CISCO.

CCNP ROUTE Exam Preparation

CCNP ROUTE 642-902

Official Certification Guide

- ✔ Master the **CCNP® Route 642-902** exam with this official study guide
- ✔ Assess your knowledge with **chapter-opening quizzes**
- ✔ Review key concepts with **Exam Preparation Tasks**
- ✔ Practice with **realistic exam questions** on the CD-ROM

ciscopress.com

Wendell Odom, CCIE® No. 1624

CCNP ROUTE 642-902

Official Certification Guide

Wendell Odom, CCIE No. 1624

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

CCNP ROUTE 642-902 Official Certification Guide

Wendell Odom

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

Printed in the United States of America

First Printing January 2010

Odom, Wendell.
CCNP Route 642-902 official certification guide / Wendell Odom.
p. cm.
ISBN 978-1-58720-253-7 (hardback w/cd)
1. Routers (Computer networks)--Examinations--Study guides. 2. Routing protocols (Computer network protocols)--Examinations--Study guides. 3. Internetworking (Telecommunication)--Examinations--Study guides. 4. Telecommunications engineers--Certification--Examinations--Study guides. I. Title.
TK5105.543.O36 2010
004.6'2--dc22
2009049908

ISBN-13: 978-1-58720-253-7

ISBN-10: 1-58720-253-0

Warning and Disclaimer

This book is designed to provide information about the Cisco ROUTE exam (642-902). 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.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States please contact: International Sales international@pearsoned.com

Feedback Information

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

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

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Executive Editor: Brett Bartow

Managing Editor: Patrick Kanouse

Development Editor: Dayna Isley

Project Editor: Mandie Frank

Book Designer: Louisa Adair

Composition: Mark Shirar

Indexer: Ken Johnson

Business Operation Manager, Cisco Press: Anand Sundaram

Manager Global Certification: Erik Ullanderson

Technical Editors: Michelle Plumb, Jerold Swan, Rick Graziani

Copy Editor: Apostrophe Editing Services

Proofreader: Barbara Hacha

Editorial Assistant: Vanessa Evans

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

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

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

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

About the Author

Wendell Odom, CCIE No. 1624, is a 28-year veteran of the networking industry. He currently works as an independent author of Cisco certification resources and occasional instructor of Cisco authorized training for Skyline ATS. He has worked as a network engineer, consultant, systems engineer, instructor, and course developer. Wendell is the author of several best-selling Cisco certification titles. He maintains lists of current titles, links to Wendell's blogs, and other certification resources at www.thecertzone.com.

About the Technical Reviewers

Michelle Plumb is a full-time Cisco certified instructor for Skillsoft. Michelle has more than 19 years experience in the field as an IT professional and telephony specialist. She maintains a high level of Cisco and Microsoft certifications. Michelle has been a technical reviewer for numerous books related to the Cisco CCNP and CCVP course material track. Michelle currently lives in Scottsdale, Arizona, with her husband and two dogs.

Jerold Swan, CCIE No. 17783, CCSP, works as a senior network engineer for the Southern Ute Indian Tribe Growth Fund in southwest Colorado. Prior to that he was a Cisco instructor for Global Knowledge. He has also worked in IT in the service provider and higher education sectors. His areas of interest include routing protocols, security, and network monitoring. He is a graduate of Stanford University. His other interests include trail running, mountain biking, and volunteer search and rescue.

Rick Graziani teaches computer science and computer networking courses at Cabrillo College in Aptos, California. Rick has worked and taught in the computer networking and information technology field for almost 30 years. Prior to teaching Rick worked in IT for various companies including Santa Cruz Operation, Tandem Computers, and Lockheed Missiles and Space Corporation. He holds an M.A. degree in computer science and systems theory from California State University Monterey Bay. Rick also does consulting work for Cisco and other companies. When Rick is not working he is most likely surfing. Rick is an avid surfer who enjoys surfing at his favorite Santa Cruz breaks.

Dedications

For Jeffrey Lanier Odom. My favorite brother. Gentle soul. Lover of stupid jokes ("baby bigger," "tankety-tankety-tank," "supplies"…) Nice guy. Good friend. Miss you, bro. 10/7/1959–6/15/2009.

Acknowledgments

As usual, Brett Bartow, executive editor, deserves thanks for allowing me to be involved with this book. Brett continually keeps an eye on the horizon for the right projects for me, essentially completing a run of books from the basics, to CCENT, CCNA, now CCNP, and CCIE. My work life wouldn't be possible without Brett keeping me pointed in the right direction. Thanks, Brett!

Jay Swan and Michelle Plumb did a nice job for us with technical edits of the book. Jay was particularly helpful with both ends of the tech edit spectrum, noticing specific and easy-to-overlook errors, while keeping an eye out for the big picture of how the text in one section impacted other sections. Michelle's diligent work helped us uncover several specific issues and make this a better book. Thanks to you both for helping make this book much better!

Rick Graziani deserves thanks with this book for several reasons. First, Rick wrote the questions on the CD with this book, a task that can be laborious–but Rick did a great job and with a positive outlook. Additionally, Rick gave us an additional set of experienced and thoughtful technical editor eyes on the BGP chapters. And while he was working on the CD questions, Rick gladly went the extra mile to point out technical edits to the other book chapters as well. Rick's great attitude toward helping with the book was very impressive.

Dayna Isley worked as the development editor for this book. Dayna and I have worked very well together for a long time, and having such a trusted editor look over every word on this new book has helped quite a bit. Dayna's attention to detail helps keep me on the authoring straight-and-narrow, this time while navigating a sometimes fluid set of processes. Dayna, thanks for sifting through this process and making me look good on paper!

Patrick Kanouse, managing editor, led us through many new production tools (WriteRAP) and processes. Additionally, Patrick happily agreed to continue several additional production tasks at my request (translated: more work for him and his team), while allowing me to manage the entire illustration process for the first time on one of my books–none of which he had to do. Patrick, thanks for your great attitude and willingness to work with me on so many extras.

The folks on Patrick's production team probably had the biggest challenge with this book compared to my other books. Mandie Frank worked as project editor, guiding the book through the various back-end processes to complete the book. Mandie got to sift though all the changing processes, help figure out when we were doing which tasks, and keep us all on track. Thanks, Mandie! San Dee Phillips retired last year so she could work even more, coming back to do the copyedit work–thanks for jumping in again, San Dee! And for Mark Shirar, Ken Johnson, and Barbara Hacha, doing the composition, indexing, and proofreading, thanks so much for handling these details–I do see the difference with

having professionals working on every step of the book creation process, and I do appreciate the results.

Thanks to Rich Bennett, good friend and part-time do-everything guy for my books and other projects. Thanks for doing all the Illustrator drawings and editing them all so many times!

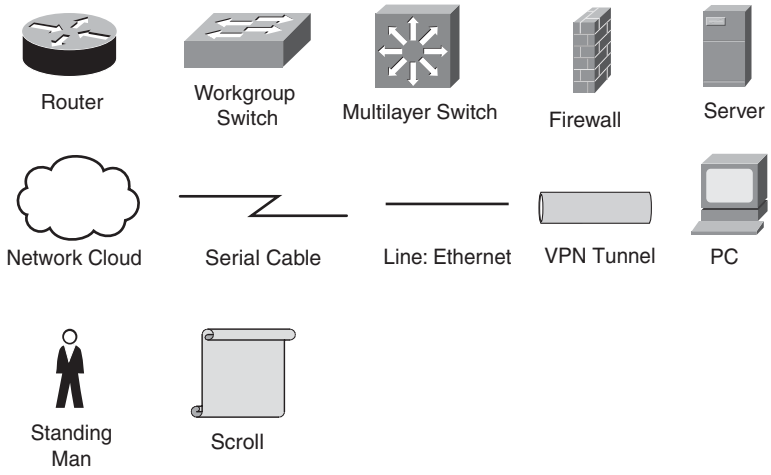
The old expression “my better half” is lived out every day here at the Odom house in the person of my wife Kris. Many thanks to Kris, who listens when I need to talk through something in the book, and lets me go hide in the basement for a few weeks to meet the latest writing deadline. Thanks, doll!

Finally and foremost, many thanks to Jesus Christ, for demonstrating your love, and for helping me and my family learn better each day how to not be a clanging symbol, but instead to show others your love.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

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

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

Foreword

CCNP ROUTE 642-902 Official Certification Guide is an excellent self-study resource for the CCNP ROUTE exam. Passing this exam is a crucial step to attaining the valued CCNP Routing and Switching certification.

Gaining certification in Cisco technology is key to the continuing educational development of today's networking professional. Through certification programs, Cisco validates the skills and expertise required to effectively manage the modern enterprise network.

Cisco Press Certification Guides and preparation materials offer exceptional—and flexible—access to the knowledge and information required to stay current in your field of expertise or to gain new skills. Whether used as a supplement to more traditional training or as a primary source of learning, these materials offer users the information and knowledge validation required to gain new understanding and proficiencies.

Developed in conjunction with the Cisco certifications and training team, Cisco Press books are the only self-study books authorized by Cisco and offer students a series of exam practice tools and resource materials to help ensure that learners fully grasp the concepts and information presented.

Additional authorized Cisco instructor-led courses, e-learning, labs, and simulations are available exclusively from Cisco Learning Solutions Partners worldwide. To learn more, visit http://www.cisco.com/go/training.

I hope that you find these materials to be an enriching and useful part of your exam preparation.

Erik Ullanderson
Manager, Global Certifications
Learning@Cisco
January 2010

Introduction

This book focuses on one major goal: to help you prepare to pass the ROUTE exam (642-902). To help you prepare, this book achieves other useful goals as well: It explains a wide range of networking topics, shows how to configure those features on Cisco routers, and explains how to determine if the feature is working. As a result, you also can use this book as a general reference for IP routing and IP routing protocols. However, the motivation for this book, and the reason it sits within the Cisco Press Certification Guide series, is that its primary goal is to help you pass the ROUTE exam.

The rest of this introduction focuses on two topics: the ROUTE exam and a description of this book.

The CCNP ROUTE Exam

Cisco announced the ROUTE (642-902) exam in January 2010. The term ROUTE does not act as an acronym; instead, the name describes the content of the exam, which focuses on IP routing. Generally, the exam includes detailed coverage of the EIGRP, OSPF, and BGP IP routing protocols, IPv6, and a few other smaller topics related to IP routing.

Cisco first announced its initial Professional level certifications in 1998 with the CCNP Routing and Switching certification. CCNP Routing and Switching certification from its inception has included the same kinds of IP routing topics found in today's ROUTE exam, but the exam names changed over the years. The exam names have tracked the names of the associated Cisco authorized courses for the same topics: Advanced Cisco Router Configuration (ACRC) in the early days, Building Scalable Cisco Internetworks (BSCI) for much of the last 10 years, and now ROUTE, because the newly revised (in 2010) Cisco authorized course also goes by the name ROUTE.

Like its ancestors, the ROUTE exam is a part of the certification requirements for several Cisco certifications, as follows:

- Cisco Certified Networking Professional (CCNP)
- Cisco Certified Internetworking Professional (CCIP)
- Cisco Certified Design Professional (CCDP)

Each of these certifications emphasizes different perspectives on some similar topics. CCNP focuses on the skills needed by a network engineer working for an Enterprise–that is, a company that deploys networking gear for its own purposes. CCIP focuses on the skills required by network engineers deploying gear at a service provider, with the service provider then offering network services to customers. Finally, CCDP focuses more on design–but good design requires solid knowledge of the technology and configuration. So, although this book frequently refers to the most popular certification of these three–CCNP–the ROUTE exam does apply to several certifications.

Contents of the ROUTE Exam

Every student who ever takes an exam wants to know what's on the exam. As with all their exams, Cisco publishes a set of exam topics. These exam topics give general guidance as to what's on the exam.

You can find the exam topics at the Cisco website. The most memorable way to navigate is to go to www.cisco.com/go/ccnp, and look for the ROUTE exam. Also, you can go to the Cisco Learning Network website (www.cisco.com/go/learnnetspace)–a less memorable URL, but a great Cisco certification site. The Cisco Learning Network site hosts exam information, learning tools, and forums in which you can communicate with others and learn more about this and other Cisco exams.

Table I-1 lists the ROUTE exam topics, with a reference to the part of the book that covers the topic.

Table I-1 ROUTE Exam Topics

Book Part	Exam Topic
Implement an EIGRP based solution, given a network design and a set of requirements	
II	Determine network resources needed for implementing EIGRP on a network
II	Create an EIGRP implementation plan
II	Create an EIGRP verification plan
II	Configure EIGRP routing
II	Verify EIGRP solution was implemented properly using **show** and **debug** commands
II	Document results of EIGRP implementation and verification
Implement a multi-area OSPF Network, given a network design and a set of requirements	
III	Determine network resources needed for implementing OSPF on a network
III	Create an OSPF implementation plan
III	Create an OSPF verification plan
III	Configure OSPF routing
III	Verify OSPF solution was implemented properly using show and debug commands
III	Document results of OSPF implementation and verification plan
Implement an eBGP based solution, given a network design and a set of requirements	
V	Determine network resources needed for implementing eBGP on a network
V	Create an eBGP implementation plan
V	Create an eBGP verification plan
V	Configure eBGP routing
V	Verify eBGP solution was implemented properly using show and debug commands
V	Document results of eBGP implementation and verification plan

Table I-1 ROUTE Exam Topics

Book Part	Exam Topic
Implement an IPv6 based solution, given a network design and a set of requirements	
VI	Determine network resources needed for implementing IPv6 on a network
VI	Create an IPv6 implementation plan
VI	Create an IPv6 verification plan
VI	Configure IPv6 routing
VI	Configure IPv6 interoperation with IPv4
VI	Verify IPv6 solution was implemented properly using show and debug commands
VI	Document results of IPv6 implementation and verification plan
Implement an IPv4 or IPv6 based redistribution solution, given a network design and a set of requirements	
IV, VI	Create a redistribution implementation plan based upon the results of the redistribution analysis.
IV, VI	Create a redistribution verification plan
IV, VI	Configure a redistribution solution
IV, VI	Verify that a redistribution was implemented
IV, VI	Document results of a redistribution implementation and verification plan
IV, VI	Identify the differences between implementing an IPv4 and IPv6 redistribution solution
Implement Layer 3 Path Control Solution	
IV	Create a Layer 3 path control implementation plan based upon the results of the redistribution analysis.
IV	Create a Layer 3 path control verification plan
IV	Configure Layer 3 path control
IV	Verify that a Layer 3 path control was implemented
IV	Document results of a Layer 3 path control implementation and verification plan
Implement basic teleworker and branch services	
VII	Describe broadband technologies
VII	Configure basic broadband connections
VII	Describe basic VPN technologies
VII	Configure GRE
VII	Describe branch access technologies

How to Take the ROUTE Exam

As of the publication of this book, Cisco exclusively uses testing vendor Pearson Vue (www.vue.com) for delivery of all Cisco career certification exams. To register, go to www.vue.com, establish a login, and register for the 642-902 ROUTE exam. You also need to choose a testing center near to your home.

Who Should Take This Exam and Read This Book?

This book has one primary audience, with several secondary audiences. First, this book is intended for anyone wanting to prepare for the ROUTE 642-902 exam. The audience includes self-study readers–people who pass the test by studying 100 percent on their own. It includes Cisco Networking Academy students taking the CCNP curriculum, who use this book to round out their preparation as they get close to the end of the Academy curriculum.

The broader question about the audience may well be why you should take the ROUTE exam. First, the exam is required for the aforementioned CCNP, CCIP, and CCDP certifications from Cisco. These certifications exist at the midpoint of the Cisco certification hierarchy. These certifications have broader and deeper technology requirements as compared to the Cisco Certified Entry Network Technician (CCENT) and Cisco Certified Network Associate (CCNA) certifications.

The real question then about audience for this book–at least the intended audience–is whether you have motivation to get one of these Professional-level Cisco certifications. CCNP in particular happens to be a popular, well-respected certification. CCIP, although less popular in numbers, focuses on topics more important to service providers, so it gives you a good way to distinguish yourself from others looking for jobs at SP companies. CCDP has been a solid certification for a long time, particularly for engineers who spend a lot of time designing networks with customers, rather than troubleshooting.

Format of the CCNP ROUTE Exam

The ROUTE exam follows the same general format as the other Cisco exams. When you get to the testing center and check in, the proctor will give you some general instructions and then take you into a quiet room with a PC. When you're at the PC, you have a few things to do before the timer starts on your exam—for instance, you can take a sample quiz, just to get accustomed to the PC and to the testing engine. Anyone who has user-level skills in getting around a PC should have no problems with the testing environment.

When you start the exam, you will be asked a series of questions. You answer the question and then move on to the next question. *The exam engine does not let you go back and change your answer.* Yes, that's true—when you move on to the next question, that's it for the earlier question.

The exam questions can be in one of the following formats:

- Multiple choice (MC)
- Testlet
- Drag-and-drop (DND)
- Simulated lab (Sim)
- Simlet

The first three types of questions are relatively common in many testing environments. The multiple choice format simply requires that you point-and-click on a circle beside the correct answer(s). Cisco traditionally tells you how many answers you need to choose, and the testing software prevents you from choosing too many answers. Testlets are questions with one general scenario, with multiple MC questions about the overall scenario. Drag-and-drop questions require you to left-click and hold, move a button or icon to another area, and release the clicker to place the object somewhere else—typically into a list. So, for some questions, to get the question correct, you might need to put a list of five things into the proper order.

The last two types both use a network simulator to ask questions. Interestingly, the two types actually allow Cisco to assess two very different skills. First, Sim questions generally describe a problem, and your task is to configure one or more routers and switches to fix the problem. The exam then grades the question based on the configuration you changed or added. Interestingly, Sim questions are the only questions that Cisco (to date) has openly confirmed that partial credit is given.

The Simlet questions may well be the most difficult style of question on the exams. Simlet questions also use a network simulator, but instead of answering the question by changing the configuration, the question includes one or more MC questions. The questions require that you use the simulator to examine the current behavior of a network, interpreting the output of any **show** commands that you can remember to answer the question. Although Sim questions require you to troubleshoot problems related to a configuration, Simlets require you to both analyze working networks and networks with problems, correlating **show** command output with your knowledge of networking theory and configuration commands.

The Cisco Learn Network (www.cisco.com/go/learnnetspace) website has tools that let you experience the environment and see how each of these question types work. The environment should be the same as when you passed CCNA (a prerequisite for CCNP, CCIP, and CCDP).

CCNP ROUTE 642-902 Official Certification Guide

This section lists a general description of the contents of this book. The description includes an overview of each chapter, and a list of book features seen throughout the book.

Book Features and Exam Preparation Methods

This book uses several key methodologies to help you discover the exam topics on which you need more review, 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. So, this book does not try to help you pass the exams only by memorization, but by truly learning and understanding the topics.

The book includes many features that provide different ways to study to be ready for the test. If you understand a topic when you read it, but do not study it any further, you probably will not be ready to pass the test with confidence. The book features included in this book give you tools that help you determine what you know, review what you know, better learn what you don't know, and be well prepared for the exam. These tools include

- **"Do I Know This Already?" Quizzes:** Each chapter begins with a quiz that helps you determine the amount of time you need to spend studying that chapter.
- **Foundation Topics:** These are the core sections of each chapter. They explain the protocols, concepts, and configuration for the topics in that chapter.
- **Exam Preparation Tasks:** The Exam Preparation Tasks section lists a series of study activities that should be done after reading the Foundation Topics section. Each chapter includes the activities that make the most sense for studying the topics in that chapter. The activities include

— **Planning Tables:** The ROUTE exam topics includes some perspectives on how an engineer plans for various tasks. The idea is that the CCNP-level engineer in particular takes the design from another engineer, plans the implementation, and plans the verification steps–handing off the actual tasks to engineers working during change-window hours. Because the engineer plans the tasks, but may not be at the keyboard when implementing a feature, that engineer must master the configuration and verification commands so that the planned commands work for the engineer making the changes off-shift. The planning tables at the end of the chapter give you the chance to take the details in the Foundation Topics core of the chapter and think about them as if you were writing the planning documents.

— **Key Topics Review:** The Key Topics icon is shown next to the most important items in the Foundation Topics section of the chapter. The Key Topics Review activity lists the Key Topics from the chapter, and page number. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic. Review these topics carefully.

— **Memory Tables:** To help you exercise your memory and memorize some lists of facts, many of the more important lists and tables from the chapter are included in a document on the CD. This document lists only partial information, allowing you to complete the table or list. CD-only Appendix D holds the incomplete tables, and Appendix E includes the completed tables from which you can check your work.

— **Definition of Key Terms:** Although the exams may be unlikely to ask a question such as "Define this term," the ROUTE exam requires that you learn and know a lot of networking terminology. This section lists the most important terms from the chapter, asking you to write a short definition and compare your answer to the glossary at the end of the book.

- **CD-based practice exam:** The companion CD contains an exam engine (from Boson software, www.boson.com), which includes 100 unique multiple-choice questions. Chapter 20 gives two suggestions on how to use these questions: either as study questions, or to simulate the ROUTE exam.

- **Companion website:** The website http://www.ciscopress.com/title/9781587202537 posts up-to-the-minute materials that further clarify complex exam topics. Check this site regularly for new and updated postings written by the author that provide further insight into the more troublesome topics on the exam.

Book Organization

This book contains 20 chapters, plus appendixes. The topics all focus in some way on IP routing and IP routing protocols, making the topics somewhat focused, but with deep coverage on those topics.

The book organizes the topics into seven major parts. Parts 1 and 7 include topics with less technical depth, and Parts 2 through 6 include the major technical topics in the book. The following list outlines the major part organization of this book:

Part I: "Perspectives on Network Planning": This part includes a single chapter:

- **Chapter 1: "Planning Tasks for the CCNP Exams":** This chapter discusses the CCNP ROUTE exam's perspectives on the planning process, including network design, implementation plans, and verification plans.

Part II: "EIGRP": This part starts with a CCNA-level EIGRP review and moves through EIGRP theory, configuration, authentication, route summarization, and more in the following chapters:

- **Chapter 2: "EIGRP Overview and Neighbor Relationships":** This chapter reviews CCNA-level EIGRP topics and then closely examines the concepts, configuration, and verification of EIGRP neighbor relationships.
- **Chapter 3: "EIGRP Topology, Routes, and Convergence":** This chapter examines the EIGRP topology database and the processes by which EIGRP processes this data to choose routes. It also examines the convergence process using feasible successors and with the Query process.
- **Chapter 4: "EIGRP Route Summarization and Filtering":** This chapter discusses the theory behind route summarization and route filtering. It also shows how to configure and verify both features for EIGRP.

Part III: "OSPF": Similar to Part II, this part starts with a CCNA-level OSPF review and moves through OSPF theory, configuration, authentication, metric tuning, default routing, route filtering, and route summarization, plus OSPF multiarea issues and different stubby area types, as follows:

- **Chapter 5: "OSPF Overview and Neighbor Relationships":** This chapter reviews CCNA-level OSPF topics and then closely examines the concepts, configuration, and verification of OSPF neighbor relationships.
- **Chapter 6: "OSPF Topology, Routes, and Convergence":** This chapter examines the OSPF topology database for routes internal to OSPF. The chapter also discusses how OSPF routers choose the best internal OSPF routes and how OSPF converges when a change occurs.

- **Chapter 7: "OSPF Route Summarization, Filtering, and Default Routing":** This chapter discusses the design, configuration, and verification of OSPF route summarization and route filtering. It also discusses default routes and how to manage the size of the OSPF database and IP routing tables by using stubby areas.
- **Chapter 8: "OSPF Miscellany":** This chapter discusses two additional OSPF topics: OSPF virtual links and OSPF issues when using NBMA networks (such as Frame Relay).

Part IV: "Path Control": The term path control refers to a wide variety of topics related to IP routing and IP routing protocols. This part examines the path control topics not specifically included in the other parts of the book:

- **Chapter 9: "Basic IGP Redistribution":** This chapter examines the concepts, configuration, and verification of IGP route redistribution. In particular, this chapter looks at the mechanics of redistribution without the use of route maps for any purpose.
- **Chapter 10: "Advanced IGP Redistribution":** This chapter essentially continues Chapter 9, in this case focusing on the more complex configuration and issues. In particular, this chapter shows how to manipulate and filter routes at the redistribution function by using route maps, and how to avoid routing loops and inefficient routes when multiple redistribution points exist.
- **Chapter 11: "Policy Routing and IP Service Level Agreement":** This chapter picks up two small path control topics that simply do not fit into any other broader chapter in this book: Policy Based Routing (PBR) and IP Service Level Agreement (IP SLA).

Part V: "BGP": This part assumes no prior knowledge of BGP. It first examines BGP design issues, to give perspective on why BGP works differently than its IGP cousins OSPF and EIGRP. This part examines basic BGP concepts, configuration, and verification, including the path control functions of incluencing both inbound and outbound BGP routes:

- **Chapter 12: "Internet Connectivity and BGP":** This chapter introduces BGP. It begins with a review of Internet connectivity from a Layer 3 perspective. It then looks at the basics of how BGP works. It also examines some Internet access design issues, discussing the cases in which BGP can be helpful and the cases in which BGP has no practical use.
- **Chapter 13: "External BGP":** This chapter examines the configuration and verification of BGP between an Enterprise and its ISP(s).
- **Chapter 14: "Internal BGP and BGP Route Filtering":** This chapter examines the cases in which routers in the same ASN need to become BGP peers, creating an Internet BGP connection. It also discusses the need for BGP filtering and the mechanics of configuring BGP filtering.

- **Chapter 15: "BGP Path Control":** This chapter discusses the concept of the BGP Best Path Algorithm to choose the best BGP routes and how to influence those choices. In particular, this chapter shows the basic configuration for BGP weight, Local Preference, AS_Path length, and Multi-Exit Discriminator (MED).

Part VI: "IPv6": This part assumes no prior knowledge of IPv6. The chapters in this part work through IPv6 addressing and IGP configuration (RIPng, EIGRP for IPv6, and OSPFv3). It also discusses route redistribution for IPv6 and IPv6/IPv4 coexistence mechanisms:

- **Chapter 16: "IP Version 6 Addressing":** This chapter begins with an overview of IP Version 6 (IPv6). It then dives into IPv6 addressing concepts, plus the related protocols, including address assignment options and neighbor discovery. The chapter shows how to configure and verify IPv6 addresses on Cisco routers.
- **Chapter 17: "IPv6 Routing Protocols and Redistribution":** This chapter introduces three IPv6 IGPs: RIP Next Generation (RIPng), EIGRP for IPv6, and OSPF Version 3 (OSPFv3). The chapter focuses on basic configuration and verification. It also discusses IPv6 redistribution in comparison with IPv4 IGP redistribution.
- **Chapter 18: "IPv4 and IPv6 Coexistence":** This chapter discusses the many options to use during the potentially long migration from a purely IPv4 network to a future purely IPv6 network.

Part VII: "Branch Office Networking": This short part includes one chapter that addresses a few small topics related to branch offices that connect to their Enterprise networks using the Internet:

- **Chapter 19: "Routing over Branch Internet Connections":** Branch office routers can be configured to use the Internet as a WAN connection path back to the rest of an Enterprise network. This chapter takes a wide look at the surprisingly large number of networking functions that must occur on a branch router in such cases. It also gives examples of configurations for IPsec and GRE tunnels, DHCP server, NAT, and DSL.

Part VIII: "Final Preparation": This short part includes one chapter as well. This chapter does not include any new technical topics:

- **Chapter 20: "Final Preparation":** This chapter suggests some study strategies for your final preparation before the ROUTE exam.

In addition to the core chapters of the book, the book has several appendixes as well. Some appendixes exist in the printed book, whereas others exist in softcopy form on the CD included with the book.

Printed Appendixes

Appendixes printed in the book include

- **Appendix A, "Answers to the "Do I Know This Already?" Quizzes":** Includes the answers to all the questions from Chapters 2 through 19.
- **Appendix B, "Conversion Tables":** Lists a decimal-to-binary conversion table, decimal values 0 through 255, along with the binary equivalents. It also lists a hex-to-decimal conversion table as well.
- **Appendix C, "CCNP ROUTE Exam Updates: Version 1.0":** Covers a variety of short topics that either clarify or expand upon topics covered earlier in the book. This appendix is updated from time to time, and posted at http://www.ciscopress.com/title/9781587202537, with the most recent version available at the time of printing included here as Appendix C. (The first page of the appendix includes instructions on how to check to see if a later version of Appendix C is available online.)

CD Appendixes

The appendixes included on the CD-ROM are

- **Appendix D, "Memory Tables":** This appendix holds the key tables and lists from each chapter with some of the content removed. You can print this appendix, and as a memory exercise, complete the tables and lists. The goal is to help you memorize facts that can be useful on the exams.
- **Appendix E, "Memory Tables Answer Key":** This appendix contains the answer key for the exercises in Appendix D.
- **Appendix F, "Completed Planning Practice Tables":** The end of Chapters 2 through 19 list planning tables that you can complete to help learn the content more deeply. If you use these tables, refer to this appendix for the suggested answers.
- **Glossary:** The glossary contains definitions for all the terms listed in the "Define Key Terms" section at the conclusion of Chapters 2 through 19.

For More Information

If you have any comments about the book, you can submit those via the www.ciscopress.com. Just go to the website, select Contact Us, and type in your message.

Cisco might make changes that affect the ROUTE exam from time to time. You should always check www.cisco.com/go/ccnp for the latest details.

This page intentionally left blank

This chapter covers the following subjects:

Perspectives on CCNP Exam Topics Related to Planning: This section outlines the goals of the CCNP certification.

How to Prepare for the Planning Topics on the Exams: This section explains what you should know generally about planning in order to be prepared for the exam.

Background Information on Implementation and Verification Plans: This short section discusses specific plans, and why there is no one specific plan used for the exam.

CHAPTER 1

Planning Tasks for the CCNP Exams

The predecessor exam to the ROUTE exam—the Building Scalable Cisco Internetworks (BSCI) exam—required mastery of the most typically used features of many routing and routing protocol technologies. The ROUTE exam requires that same mastery, but the ROUTE exam also includes many exam topics that use the words "plan" and "document." The predecessor BSCI exam did not include such wording in the exam topics, so presumably the new ROUTE exam adds this as a new requirement.

This opening chapter examines the meaning, purpose, and some perspectives on these planning and documentation tasks as they relate to preparing for and passing the ROUTE 642-902 exam.

Perspectives on CCNP Exam Topics Related to Planning

Cisco introduced the Cisco Certified Networking Professional (CCNP) certification in 1998. Since that time, Cisco has revised the exams and related courses on several occasions. Each major revision adjusted the scope of topics by expanding and adding some topics while shrinking or removing other topics. At the same time, the depth of coverage has changed over time as well, with the depth of coverage for each topic either becoming deeper or shallower.

The version of CCNP announced by Cisco in January 2010, including the 642-902 exam about which this book is written, narrows the breadth of topics included in CCNP compared to the previous version of CCNP. Cisco removed several sizable topics from CCNP—notably Quality of Service (QoS), Wireless LANs (WLANs), and many security topics. The new CCNP squarely focuses on routing and switching, but with a deeper troubleshooting requirement, with a new TSHOOT (642-832) exam. These changes also reflect that CCNP now requires only three exams instead of the four exams formerly required.

However, although the smaller number of CCNP topics may make CCNP easier, two other factors balance the CCNP so it is still a challenging, difficult, and respected certification. First, the exams appear to require a higher level of mastery for most topics. Second, that mastery is more than just technical knowledge; it requires the ability to plan the implementation and verification of a network engineering project.

Many CCNP ROUTE Exam Topics list the word "plan," collectively meaning that the CCNP candidate must approach problems in the same manner as a network engineer in a

medium- to large-sized business. The skills related to these exam topics can be built as a side-effect of doing many network engineering jobs, for instance

- The ability to analyze a network design document, extrapolate that design into the complete detailed implementation plan, including completed configurations for each router and switch.
- The ability to analyze a design document and discover the missing items—questions that must be answered before a detailed implementation plan (including configurations) can be completed.
- The ability to perform a peer review on another engineer's implementation plan, to discover weaknesses and omissions in the planned configurations, and to update the implementation plan.
- The ability to build a verification plan that lists the specific **show** commands and command options that list key pieces of information—information that directly either confirms or denies whether each planned feature has been implemented correctly.
- The ability to write a verification plan that can be understood and used by a less-experienced worker, allowing that worker to implement the change and to verify the changes worked, off-shift, when you are not on-site.
- The ability to perform a peer review on another engineer's verification plan, to discover which key design features are not verified by that plan, and to discover inaccuracies in the plan.

This chapter discusses the whole area of implementation and verification planning for the CCNP ROUTE exam, including how you should prepare for these exam topics. By considering the ideas in this chapter first, you should have the right perspectives to know how to use the tools that help you add the planning skills and perspectives needed for the exam.

CCNP Route Exam Topics That Do Not Require the CLI

Cisco lists a set of Exam Topics for each Cisco exam. These Exam Topics follow a general type of phrasing, typically starting with an action word that defines what action or skill you must do for the exam. (Unfortunately, this style seldom gives much insight into the breadth or depth of coverage of a given topic.)

For example, consider the six Exam Topics for the CCNP ROUTE exam specifically related to EIGRP:

- Determine network resources needed for implementing EIGRP on a network.
- Create an EIGRP implementation plan.
- Create an EIGRP verification plan.
- Configure EIGRP routing.
- Verify EIGRP solution was implemented properly using **show** and **debug** commands.
- Document results of EIGRP implementation and verification.

The two gray-highlighted exam topics focus on the commands available from the CLI, specifically that you need to configure EIGRP, and to use both **show** and **debug** commands to verify (confirm) whether EIGRP is working correctly. The unhighlighted topics in the list require knowledge of the commands, but the tasks do not require any hands-on activities from the CLI. Instead, when doing these tasks in real life, you would more likely be using a word processor rather than a terminal emulator.

Besides this list of the EIGRP exam topics, the entire list of CCNP ROUTE Exam Topics includes many more items that use words like "document" and "plan." Of the roughly 40 CCNP ROUTE Exam Topics, approximately half of the exam topics refer to the various types of plans and documentation. In particular, the phase "Create a _____verification plan" occurs six times in the CCNP ROUTE list of Exam Topics—one each for EIGRP, OSPF, eBGP, IPv6, redistribution, and Layer 3 path control.

Impressions on the Planning Exam Topics

After a first glance through the CCNP ROUTE Exam Topics, you might think that the new CCNP certification has been changed significantly—and you therefore need to significantly change how you prepare for CCNP. However, it turns out that by focusing on the following aspects of your study, you should be well prepared for the CCNP exams in general and the CCNP ROUTE exam in particular:

- As with any other Cisco career certification exam, understand the concepts, configuration, and verification commands (**show** and **debug** commands).
- As with any other Cisco career certification exam, master the configuration and verification tasks and commands.
- Unlike most other Cisco career certification exams, spend some time thinking about the concepts, configuration, and verification tasks as if you were writing or reviewing a network design document, a network project implementation plan, or a verification plan.

In this list, the first two tasks are simply what most people normally do when preparing for a Cisco exam. The third item represents the new type of preparation task, in which you simply think about the same concepts, commands, and features, but from a planning perspective.

At this point in this brief first chapter, you can choose whether to keep reading the topics in order, or whether to skip to the section "How to Prepare for the Planning Topics on the Exam" later in this chapter. Those of you who have a pretty good idea of the planning tasks done by most IT shops can consider skipping ahead. For those of you with little or no experience in reading network design documents, building or using network implementation and verification plans, the next section can give you some useful perspectives before you dive into studying the technologies in CCNP ROUTE.

Relating the Exam Topics to a Typical Network Engineer's Job

The need to plan, and the need to document those plans, increases as the size of the organization increases. Even if only one person at a company cares about the router and switch infrastructure, that engineer probably does not want to be writing configurations at 2 a.m. Sunday morning when the change window begins. So, even in a small IT shop, some planning occurs when the engineer creates the configurations in a text editor before the weekend change window. When the staff grows to 3 to 4 people, particularly when some of those people work different shifts, the need to document the design, implementation, and verification/operational procedures becomes more important.

For perspective, this section examines a medium- to large-sized company, along with some of the planning tasks done in the real world—the same kinds of tasks listed as part of the CCNP Exam Topics.

A Fictitious Company and Networking Staff

Imagine if you will a medium to large company, one large enough to have several network engineers on staff. For the sake of discussion, this company has roughly 50,000 employees, with 1000 smallish remote sites, four large sites with at least 2000 employees on each large campus, and maybe a smattering of other sites with 500 or so employees. Of course, the company has a few data centers, has plans for a companywide IP telephony deployment, is adding video over IP, already has the usual security needs, and has a growing teleworker community, several network connections to partner companies, and Internet connections. Oh yeah, and there's always the growing need for smart buildings to reduce energy consumption, all hooked into the IP network.

With a company of this size, the job roles for this fictitious company includes IT customer support (Help Desk, manned 24x7), an operations team that covers most hours of the day, network engineering, and a design team.

Next, consider the various roles in the network and the type of work done by the people in those roles:

- Help desk personnel may perform diagnosis of network health, taking a general problem statement from a customer down to a specific issue, for example, that a user's device is not pingable.
- Operations staff may be the second level of support for problems, both reacting to calls from the Help Desk and monitoring the network proactively. The operations staff also often implements changes on behalf of the engineering team during off-shift hours.
- The network engineering team may be the third level of support for problems but typically focuses on project work, including the detailed planning for new configurations to support new sites, new network features, and new sites in the network.
- The network designers may actually log in to the network devices far less than the operations and engineering teams, instead focusing on gathering requirements from internal and external customers, translating those requirements into a network design,

and even doing proof-of-concept testing—but leaving the details of how to deploy the design for all required sites to the network engineering team.

The number of individuals in each role varies in different organizations, of course. Maybe only a single network designer and single network engineer are required, with maybe 2 to 3 people as network operations specialists—not enough for 24x7 coverage with a specialist, but close. The Help Desk position may simply require most people to have the same fundamental networking skill set, depending on the size of the shop. On the other end of the scale, in the largest companies, the engineering staff might include several departments of network engineers.

The Design Step

Next, consider the basic workflow when a new network project happens, new sites are added, or any noticeable change occurs. The network designer begins the process by determining the requirements and creating a plan. That plan typically lists the following:

- The requirements for the project
- The sites affected
- Sample configurations
- Results from proof-of-concept testing
- Dependencies and assumptions
- Business requirements, financials, and management commitments

Many other items might be included as well.

After creating the design document, the network designer often uses a peer review process to refine and confirm the design. The designer cannot simply work in a vacuum, define the design, and then toss the design document to network engineering to be deployed. In smaller shops, a peer review may simply be 2 to 3 people standing around a dry erase board discussing the project. In larger shops, the design peer review probably requires a thorough written document be distributed before the meeting, with attendance from network engineering, operations, and the Help Desk, and with formal sign-off required.

Implementation Planning Step

The next step in the life of the project occurs when a network engineer takes the approved design document from the design team and begins planning the implementation of the project. To do this task, the network engineer must interpret the examples and general cases described in the design document and develop a very specific implementation plan that lists all significant tasks and actions by each group and on each device. The design document, for instance, may show example cases of typical branch offices—a typical one-router branch, a typical two-router larger branch, a typical district (medium-sized) site, and so on. The network engineer must then determine what must be done on every device to implement the project and document those details in an implementation plan.

For example, imagine this fictitious company plans to deploy IP telephony to the 1000 remote sites. The design document lists the following requirements:

- Switches with Power over Ethernet (PoE) at each remote office (switch models listed in the design doc).
- The convention of placing all phones at a site in one VLAN/subnet and all PCs in a second VLAN/subnet.
- VLAN trunking between the switch and the 1 or 2 routers at each remote site.
- A particular version and feature set of router IOS on the remote site routers to support Survivable Remote Site Telephony (SRST).
- More aggressive tuning of EIGRP to improve convergence time, particularly by ensuring both routes from a remote site back into the network core are in the routing table at the same time.

The design document certainly contains more details than the preceding list, but the list gives you an idea of the starting point when the network engineer first work on his implementation plan by reviewing the design document.

After a thorough review of the design, the network engineer then develops an implementation plan that includes items like the following:

- A list of all remote offices, with notations of which require a switch hardware upgrade (for PoE support) and which do not
- Total numbers of switches to be ordered, prices, and delivery schedules
- A table that lists the specific VLAN and subnet numbers used at each site for the phone VLAN/subnet and PC VLAN/subnet
- The IP address ranges from each remote site subnet that needs to be added to the DHCP servers configurations for dynamic assignment
- A list of the remote site routers that require a router hardware upgrade to support either trunking or the required IOS, including pricing and delivery schedules
- A list of remote site routers that do not require a replacement router but do require a memory upgrade to support the needed IOS
- Annotated sample configurations for typical sites, including VLAN trunking to the switch, SRST, and EIGRP convergence tuning
- A reference to the location of the switch and router configuration for every device that will be configured as part of the project

The preceding list represents the types of items that would be documented in the implementation plan for this project. The list is certainly not exhaustive but represents a smattering of what might end up in such a plan.

The implementation plan probably has many informal reviews as the network engineer works through the plan. Additionally, larger shops often have a peer review after the plan

is more fully developed, with network designers, operations, and fellow network engineers typically included in the review.

Verification Planning Step

The design step tells us "this is what we want to accomplish," whereas the implementation planning step tells us "this is exactly what we will do, and when, to accomplish this design." With that in mind, the verification plan tells whoever will actually perform the actions to implement a project how to answer this question:

> "Did the actions we took per the implementation plan work?"

The verification plan is used with the actual implementation of the changes in the network. The larger the network, the less likely that the network engineer does any of the implementation work, instead planning the implementation. More often than not, the operations staff follows the implementation plan, or more specific instructions for each individual change window, taking the appropriate actions (copying in configurations, for instance). The engineer that implements the changes then uses the verification plan to determine if the changes met the requirements.

The most important part of the verification plan, at least for being ready to pass the CCNP exams, identifies what commands confirm whether each key design point was implemented correctly. For example, the section "Implementation Planning Step" earlier in this chapter briefly describes a project to deploy IP telephony to all remote sites. For that same project, the following list describes some of the actions listed in the verification plan:

- After copy/pasting a remote site's new router configuration, use the **show ip interface brief** command, confirm an up/up state on the two subinterfaces of Fa0/0, and confirm the IP addresses match the planning chart in IP address repository.
- For each remote site router, use the **show ip route** command to confirm two routes exist for each Data Center subnet. See "Data Center Subnet Reference" in the IP address repository.
- From the WAN edge routers, use the **show ip eigrp neighbors detail** command, and find neighbors listed as stub routers. For the routers configured in tonight's change window, compare this output to the Stub Router planning table in the implementation plan, and confirm that the correct remote site routers have been configured as EIGRP stubs.

Note: The "IP address repository" mentioned in the list does not exist in this chapter; it just represents the idea that an implementation plan would include some reference to the location of all subnet/address reference information.

The important part of the verification plan lists the specific commands used, at what point in the implementation process, and what output should be seen. In practice, this plan should also include samples of output, spelling out what should be seen when correct, and what output would tell the operations staff that the change did not work correctly.

Documenting the Results of the Implementation

Continuing the story of the typical but fictitious company, when a set of changes is attempted during a change window, some documentation must be changed based on the results. If a different final configuration is used, the implementation documents must be changed.

Note: This particular step is mentioned mainly because several CCNP ROUTE Exam Topics refer to the task of documenting the results of the implementation and verification.

Summary of the Role of Network Engineer

The CCNP certification focuses on skills required to do the job of network engineer as generally described in this chapter. For perspective, then, consider the following list, which compares and contrasts some of the expectations for CCNP network engineers by interpreting the CCNP ROUTE Exam Topics:

- Does not create the design document
- Does participate in design peer reviews, finding oversights, asking further questions that impact the eventual implementation, and confirming the portions of the design that appear complete and valid
- Does not deploy the configurations off-shift
- Does plan and document the specific configurations for each device, documenting those configurations in the implementation plan so that others can add the configuration to various devices
- Does participate in peer reviews of the implementation plans written by fellow network engineers, finding omissions, caveats, and problems
- Does not verify that the changes worked as planned when implemented off-shift
- Does create the verification plan that others use to verify that the changes worked as planned when implemented off-shift
- Does perform peer reviews of other engineers' verification plans

Now that you've had a chance to think generally about the role of the network engineer, the next section brings the discussion back around to the CCNP ROUTE exam, and how you should prepare for the exam.

How to Prepare for the Planning Topics on the Exams

Can you create a networking implementation plan for each technology area on the CCNP exams? Can you create a verification plan for those same technologies? According to the CCNP Exam Topics, these skills now fall into the scope of topics on the CCNP exams. However, Cisco can't reasonably ask you a question as open-ended as "Create an entire EIGRP implementation plan based on the following design document." Such a question would take too much time relative to the typical 1 minute 15 seconds or so average question time for a typical Cisco exam.

Even though the exam may not ask you to literally create a plan, you do need the skills to perform those same tasks. As with any other exam topic, expect the exam to ask questions that zero in on a small subset of the required skill.

To prepare for the planning-oriented topics, you do NOT need to learn any more facts about the technology or commands. The CCNP Exam Topics already cover the technology, particularly the configuration, verification, and troubleshooting of the listed technologies. For the planning-related Exam Topics, however, you do need to think about those technologies with a slightly different perspective. The question is whether you could, with only pencil, paper, and a word processor—but definitely without a router or switch CLI—do the following:

Step 1. Read design goals extracted from a design document, develop a configuration that meets those goals, and discover missing information that needs to be gathered before you can complete the configuration.

Step 2. Read an extract from the design and implementation plans to determine what is wrong or missing.

Step 3. Read a configuration and design goal as stated in an implementation plan, and create the verification steps that would confirm whether the feature was working.

Step 4. Analyze a portion of a verification plan, along with the stated configuration and design goals, and determine any problems or missing elements in the verification plan.

Figure 1-1 shows the same concepts, with numbers referencing the preceding steps.

After you have a solid understanding of the configuration and verification commands for a topic, to prepare for the planning-related Exam Topics, you simply need to take the time to think about those same topics, but from the perspective of the tasks shown in Figure 1-1. To that end, this book adds some tools to the end of each chapter. These tools should help organize your thinking and remind you to take some time to ponder the topics in a chapter from a design perspective.

The end of most chapters in this book include a new section titled "Planning Preparation" as part of the broader "Foundation Summary" review section. The "Planning Preparation" guides you through various aspects of the planning process, focused on the technologies included in that chapter. All the Planning Preparation tables list some information and give you the opportunity to complete the tables as an exercise. Appendix F, "Completed Planning Practice Tables," found on the CD included with this book, lists completed versions of the tables for reference. This section uses four different types of tables that mimic some of the planning and analysis tasks, as described under the next four headings.

Note: You may want to glance at the Planning Preparation tables at the end of Chapter 2, "EIGRP Overview and Neighbor Relationships," for perspective as you read through this chapter.

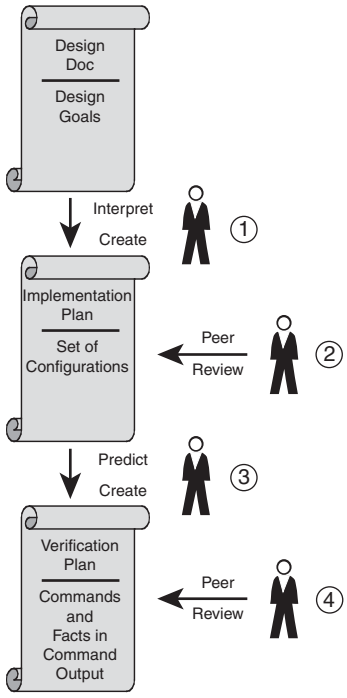


Figure 1-1 *Planning Tasks for the CCNP Candidate*

Planning Preparation: Design Review Table

When in a real design review, a network engineer must look at the requirements and decide what configuration options might be used. The Design Review table in the Planning Preparation section of most chapters lists design goals that you might see in a design document. Your job is to interpret each design goal, think about what features might be used to reach the goal, and then list those features.

This tool is designed to help you think about the broader view of a network that is typically seen in design documents.

Planning Preparation: Implementation Plan Peer Review Table

When you attend a real implementation plan peer review, you and other engineers can see the plan and immediately think of different questions. Those questions may be questions about the technology—some fact you used to know but forgot, some confusion about how a command would work given the specific design, or some question about the design goal that led to the listed configuration.

The Implementation Plan Peer Review table predicts some of the questions that might come to mind when performing a peer review. Your job with this table is to then answer the questions. Some questions may be basic, and really meant more for review, whereas others strive to force you to think about some of the subtleties of the topics in the chapter.

Create an Implementation Plan Table

The Implementation Plan table is a pure memory aid, but the task fits directly into the Exam Topics that read "Create a _____ implementation plan." This tool uses a table that simply lists the configuration topics from the chapter and asks that you write all related configuration commands from memory. Although you may not need to memorize every command and parameter for the exam, this exercise can help you mentally group commands together, which may aid in questions directed at building implementation plans.

Choose Commands for a Verification Plan Table

This final tool helps you think about verification commands by first thinking of the fact you need to discover and then choosing the command that lists the fact. Many resources—including Cisco documents, many Cisco authorized courses, and many books (mine included)—begin with the **show** commands and then explain the information displayed by the command. The reverse order of thinking is required to be prepared to create a verification plan. You must first decide what facts must be discovered to confirm a feature is working and then remember which commands supply that information.

This final planning preparation tool lists many of the key individual pieces of information that matter to the technologies included in that chapter. Your job is to list the commands that display the requested piece of information.

Background Information on Implementation and Verification Plans

The last major section of this chapter also discusses the topics of implementation and verification planning, but with a different twist. So far, this chapter has been focused on how you might prepare for CCNP exams. This section instead focuses on implementation and verification planning as an end to themselves.

No Single Plan Style

Upon reading the CCNP ROUTE Exam Topics, and seeing the references to "implementation plan" and "verification plan," you might be at least slightly curious to see examples of such plans. However, no such example implementation plan or verification plan exists in this book because no one specific type of plan matters to the exam. In fact, the Cisco authorized course for this exam, also called ROUTE, also does not offer a specific type or style of implementation or verification plan.

Several reasons exist as to why no one type of implementation or verification plan is suggested as the model for preparing for the exams, including the following:

- Every company does something different, from completely ad-hoc discussions around a white board to formal reviews and processes.
- Several standards exist for methodologies that include implementation and verification planning, but each uses different terminology, so a single suggested plan format would implicitly recommend one method or another.

- Creating a pseudo-standard example plan for the sake of CCNP was not actually necessary to assess the test-taker's planning skills.

Typical Elements in an Implementation Plan

Although no one style of plan matters, the types of items inside a plan have some interest for the exam. It is also useful to know a few terms regarding some of the formalized methods for implementation planning.

When an IT organization takes the time to require and review a written implementation plan for a project, those plans still vary in terms of depth and detail. The plan may include many details about the existing network, or it may rely on other documentation, ignoring those details in the implementation plan. The plan may list financials, particularly when hardware and software must be purchased, or it may leave those details out of the planning document, instead leaving those details for the management team to handle.

Just for perspective, Table 1-1 outlines some of the types of items you might see in a network implementation plan:

Table 1-1

The existing network:	Router and switch hardware IOS versions and feature sets RAM and flash in each device Existing configurations IP Subnet and Addressing Plan, Assignments, and Conventions
Management:	Personnel and roles, contact information Assumptions and dependencies Required management sign-offs New tools, reporting, status update process
New project details:	Design goals (reference to design doc possibly) Hardware upgrades Software upgrades Timelines to make changes Specific configurations for each device Migration issues (assuming a subset of sites are implemented in any one change window) Network diagrams, possibly for each interim step during a migration
Project completion:	Final sign-off requirements Definitions of success Submission of revised site documentation, operational procedures, and any other permanent documentation

Focus for Implementation Plans for CCNP

Although this sample includes typical elements, the three highlighted elements have some particular interest in the context of the CCNP exams. The Exam Topics do not state a general "Create a network project implementation plan," but rather focuses on a technology area. Specifically for CCNP ROUTE, the Exam Topics that mention an implementation plan specifically focus on EIGRP, OSPF, eBGP, redistribution, path control, and IPv6, respectively.

For the exam, the highlighted elements in the sample plan represent the most likely items to be within the scope of the Exam Topics. An EIGRP implementation plan in real life—say for a migration from RIP-2 to EIGRP—probably includes all the elements listed in the sample plan. However, because EIGRP is a Cisco IOS Software feature, the likely implementation plan topics relate to EIGRP configuration, verification, and issues about migrating from one design to another.

Structured Implementation Planning Methodologies

The Cisco authorized courses for the CCNP track mention a few structured methodologies that can be used to manage network projects. These methodologies include steps, terminology, and conventions that include the implementation planning, the actual implementation, and in some cases, specific items for verification planning. (In some cases, the verification plan is part of the implementation plan or change management process.) Table 1-2 lists these methodologies, and a few facts about each, for perspective.

Table 1-2 *Project Planning Methodologies*

Method	Owner	Comment
FCAPS	ISO	Fault, Configuration, Accounting, Performance, and Security: This standard focuses on network and systems management. Implementation planning falls into the change management category.
ITIL	Great Britain	Information Technology Infrastructure Library: A set of best practices for systems management that has been widely used in the IT industry. It is managed by the government of Great Britain. www.itil-official-site.com.
TMN	ITU-T	Telecommunications Management Network: Created by the ITU-T's Study Group 4, this ongoing effort defines system management practices from the ITU. Originally based on FCAPS. www.itu.int.
Cisco Lifecycle Services	Cisco	Quoting the Cisco website: "The Cisco Lifecycle Services approach defines the activities needed to help you successfully deploy and operate Cisco technologies and optimize their performance throughout the lifecycle of your network."
PPDIOO	Cisco	Prepare, Plan, Design, Implement, Operate, Optimize: The popular acronym for the steps defined by Cisco Lifestyle Services.

Typical Verification Plan Components

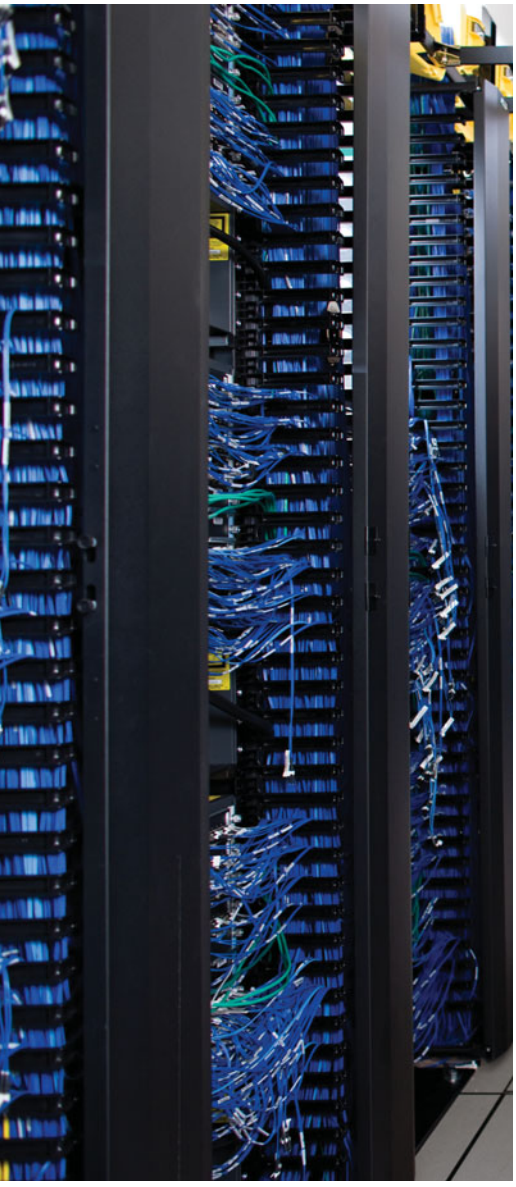
A typical verification plan is much smaller than the implementation plan. The verification plan often focuses on a specific change made on a specific day, or on a type a change made repeatedly at different sites over a longer migration period. The following list outlines the more common items in a verification plan:

- The specific **show** and **debug** commands to be used
- The specific facts in the output that must be confirmed
- Explanations of why a particular command's output confirms that a feature is or is not working
- Sample command output
- Any data that should be gathered when the output is not correct

Conclusions

The various CCNP ROUTE Exam Topics that require planning and documentation do not require more technical knowledge, but they do require you to be more comfortable with the technical topics. The best way to become more comfortable is to have a job in which you use the features but that you can become very comfortable with the technology through study as well. For those of you with little experience with the technologies included in this book, please take the time to do the exercises in the "Planning Practice" sections near the end of each chapter—we believe such practice can help you prepare for the planning perspectives expected to be on the exams.

This page intentionally left blank



This chapter covers the following subjects:

EIGRP CCNA Review: This section reviews the EIGRP concepts, configuration, and verification commands assumed as prerequisites, specifically those details included in the CCNA Exam's coverage of EIGRP.

EIGRP Neighborships: This section discusses a variety of features that impact when a router attempts to form EIGRP neighbor relationships (neighborships), what must be true for those neighborships to work, and what might prevent those neighborships.

Neighborships over WANs: This short section examines the typical usage of EIGRP neighborships over various types of WAN technologies.

CHAPTER 2

EIGRP Overview and Neighbor Relationships

Enhanced Interior Gateway Routing Protocol (EIGRP) is configured with a few relatively simple commands. In fact, for most any size network, you could go to every router, configure the **router eigrp 1** command, followed by one or more **network** *net-id* subcommands (one for each classful network to which the router is connected), and EIGRP would likely work, and work very well, with no other configuration.

In spite of that apparent simplicity, here you sit beginning the first of three chapters of EIGRP coverage in this book. Many reasons exist for the amount of EIGRP material included here. First, EIGRP includes many optional configuration features that you need to both understand and master for the CCNP ROUTE exam. Many of these features require a solid understanding of EIGRP internals as well—a topic that can be conveniently ignored if you just do the minimal configuration, but something very important to planning, implementing, and optimizing a medium/large Enterprise network.

Another reason for the depth of EIGRP coverage in this book is due to a fundamental change in the philosophy of the CCNP exams, as compared with earlier CCNP exam versions. Cisco has increased the focus on planning for the implementation and verification of new network designs. The bar has been raised, and in a way that is consistent with typical engineering jobs. Not only do you need to understand all the EIGRP features, but you also need to be able to look at a set of design requirements, and from that decide which EIGRP configuration settings could be useful—and which are not useful. You also must be able to direct others as to what verification steps would tell them if the implementation worked or not, rather than just relying on typing a **?** and looking around for that little piece of information you know exists somewhere.

Part II of this book contains three chapters. This chapter briefly reviews the basics of EIGRP, and delves into all topics related to how EIGRP routers form neighbor relationships. Chapter 3, "EIGRP Topology, Routes, and Convergence," then examines many topics related to how EIGRP chooses routes. Chapter 4, "EIGRP Route Summarization and Filtering," then moves on to examine route filtering and route summarization, which closes the discussion of specific EIGRP features.

This chapter in particular begins with the "EIGRP Basics" section, which is a review of the core prerequisite facts about EIGRP. Following the review, the chapter examines EIGRP neighbor relationships, including a variety of configuration commands that impact neighbor relationships, and the verification commands that you can use to confirm how well EIGRP neighbors work.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 2-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

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

Foundation Topics Section	Questions
EIGRP CCNA Review	1, 2
EIGRP Neighborships	3–7
Neighborships over WANs	8

1. A router has been configured with the commands **router eigrp 9** and **network 172.16.1.0 0.0.0.255**. No other EIGRP-related commands have been configured. The answers list the IP addresses that could be assigned to this router's Fa0/0 interface. Which answers list an IP address/prefix length that would cause the router to enable EIGRP on Fa0/0?

   a. 172.16.0.1/23
   b. 172.16.1.1/26
   c. 172.16.1.1/24
   d. 172.16.0.255/23
   e. None of the other answers is correct.

2. Router R1 has working interfaces S0/0, S0/1, and S0/2, with IP address/prefix combinations of 10.10.10.1/24, 10.10.11.2/24, and 10.10.12.3/22. R1's configuration includes the commands **router eigrp 9** and **network 10.0.0.0**. The **show ip eigrp interfaces** command lists S0/0 and S0/1 in the command output, but not S0/2. Which answer gives the reason for the omission? (Choose two answers.)

   a. R1 has EIGRP neighbors reachable via S0/0 and S0/1, but not via S0/2, so it is not included.
   b. S0/2 may currently be in a state other than up/up.
   c. The **network 10.0.0.0** command requires the use of mask 255.0.0.0 due to EIGRP being classful by default.
   d. S0/2 may be configured as a passive interface.

3. Routers R1 and R2 are EIGRP neighbors using their Fa0/0 interfaces, respectively. An engineer adds the **ip hello-interval eigrp 9 6** command to R1's Fa0/0 configuration. Which of the following is true regarding the results from this change?

    a. The **show ip eigrp neighbors** command on R1 lists the revised Hello timer.

    b. The **show ip eigrp interfaces** command on R1 lists the revised Hello timer.

    c. The R1-R2 neighborship fails due to Hello timer mismatch.

    d. The **show ip eigrp interfaces detail** command on R1 lists the revised Hello timer.

4. Routers R1 and R2, currently EIGRP neighbors over their Fa0/0 interfaces (respectively), both use EIGRP authentication. Tuesday at 8 p.m. the neighborship fails. Which of the following would *not* be useful when investigating whether authentication had anything to do with the failure?

    a. **debug eigrp packet**

    b. **show key chain**

    c. **show ip eigrp neighbor failure**

    d. **show clock**

5. Router R1 has been configured with the commands **router eigrp 9** and **network 172.16.2.0 0.0.0.255**, with no other current EIGRP configuration. R1's (working) Fa0/0 interface has been configured with IP address 172.16.2.2/26. R1 has found three EIGRP neighbors reachable via interface Fa0/0, including the router with IP address 172.16.2.20. When the engineer attempts to add the **neighbor 172.16.2.20 fa0/0** command in EIGRP configuration mode, which of the following occurs?

    a. Fa0/0 fails.

    b. The command is rejected.

    c. The existing three neighbors fail.

    d. The neighborship with 172.16.2.20 fails and then reestablishes.

    e. None of the other answers is correct.

6. Which of the following settings could prevent two potential EIGRP neighbors from becoming neighbors? (Choose two answers.)

    a. The interface used by one router to connect to the other router is passive in the EIGRP process.

    b. Duplicate EIGRP router IDs.

    c. Mismatched Hold Timers.

    d. IP addresses of 10.1.1.1/24 and 10.2.2.2/24, respectively.

7. An engineer has added the following configuration snippet to an implementation planning document. The configuration will be added to Router R1, whose Fa0/0 interface connects to a LAN to which Routers R2 and R3 also connect. R2 and R3 are already EIGRP neighbors with each other. Assuming the snippet shows all commands on R1 related to EIGRP authentication, which answer lists an appropriate comment to be made during the implementation plan peer review?

```
key chain fred
 key 3
 key-string whehew
interface fa0/0
 ip authentication key-chain eigrp 9 fred
```

   a. The configuration is missing one authentication-related configuration command.
   b. The configuration is missing two authentication-related configuration commands.
   c. Authentication type 9 is not supported; type 5 should be used instead.
   d. The key numbers must begin with key 1, so change the **key 3** command to **key 1**.

8. A company has a Frame Relay WAN with one central-site router and 100 branch office routers. A partial mesh of PVCs exists: one PVC between the central site and each of the 100 branch routers. Which of the following could be true about the number of EIGRP neighborships?

   a. A partial mesh totaling 100: one between the central-site router and each of the 100 branches.
   b. A full mesh – (101 * 100) / 2 = 5050–One neighborship between each pair of routers.
   c. 101–One between each router (including the central site) and its nearby PE router.
   d. None of the answers is correct.

Foundation Topics

EIGRP CCNA Review

All the CCNP exams consider CCNA materials as prerequisites, so the Cisco Press CCNP Exam Certification Guide series of books also assumes the reader is already familiar with CCNA topics. However, the CCNP exams do test on features that overlap with CCNA. Additionally, most people forget some details along the way, so this section reviews the CCNA level topics as a brief refresher.

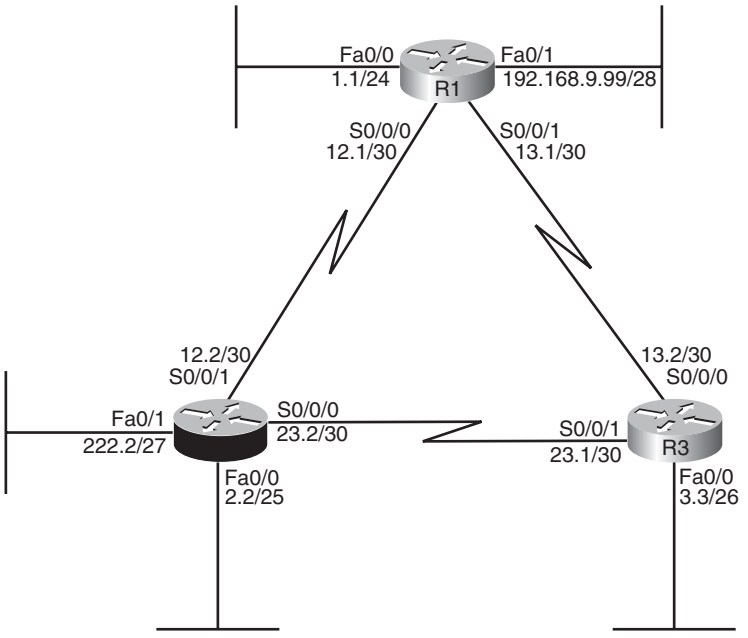
To that end, this section begins with a review of EIGRP configuration using only the **router eigrp** and **network** commands. Following that, the next section details the key fields used to verify that EIGRP is working. Finally, the last part of this introduction summarizes the basic EIGRP internals behind this initial simple example.

Configuration Review

Cisco IOS uses the **router eigrp** *asn* command, plus one or more **network** *net-id wild-card-mask* subcommands, to enable EIGRP on the router and on router interfaces. The rules for these commands are as follows:

1. Neighboring routers' **router eigrp** *asn* commands must be configured with the same ASN parameter to become neighbors.
2. IOS enables only EIGRP on interfaces matched by an EIGRP **network** command. When enabled, the router does the following:
   a. Attempts to discover EIGRP neighbors on that interface by sending multicast EIGRP Hello messages
   b. Advertises to other neighbors about the subnet connected to the interface
3. If no wildcard-mask is configured on the EIGRP **network** command, the command's single parameter should be a classful network number (in other words, a class A, B, or C network number).
4. If no wildcard-mask is configured on the EIGRP **network** command, the command enables EIGRP on all of that router's interfaces directly connected to the configured classful network.
5. If the **network** command includes a wildcard-mask, the router performs access control list (ACL) logic when comparing the net-id configured in the **network** command with each interface's IP address, using the configured wildcard-mask as an ACL wildcard mask.

Example 2-1 shows a sample configuration for each router in Figure 2-1, with several variations in the **network** commands to make the details in the preceding list more obvious.



Note: All IP addresses begin with 10.1 unless otherwise noted.

Figure 2-1 *Three Router Internetwork*

Example 2-1 *EIGRP Configuration on Routers R1, R2, and R3*

```
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 network 10.0.0.0
 network 192.168.9.0
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 network 10.1.0.0 0.0.31.255
 network 10.1.2.2 0.0.0.0
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 network 10.1.0.0 0.0.255.255
```

First, note that all three routers use the **router eigrp 1** command, so all three routers' ASN values match.

Next, consider the two **network** commands on R1. The **network 10.0.0.0** command, without a *wildcard-mask* parameter, means that R1 matches all interface in class A network 10.0.0.0—which in this case means R1's Fa0/0, S0/0/0, and S0/0/1 interfaces. The **network 192.168.9.0** command, again without a wildcard, matches interface Fa0/1.

On R2, the **network 10.1.0.0 0.0.31.255** command requires a little more thought. The router uses the 0.0.31.255 value—the wildcard (WC) mask—just like an ACL WC mask.

IOS compares the 10.1.0.0 value with each interface IP address, but only for the bit positions for which the WC mask lists a binary 0. For example, 0.0.31.255 represents 19 binary 0s, followed by 13 binary 1s, so R2 would compare the first 19 bits of 10.1.0.0 with the first 19 bits of each interface's IP address. (Note that Appendix B lists a binary/decimal conversion table.)

Two features of the mechanics of the **network** command require a little extra attention. First, IOS may convert the *address* portion of the **network** *address wc-mask* command before putting the command into the running-config. Just as IOS does for the address/WC mask combinations for the **access-list** command, IOS inverts the WC mask and then performs a Boolean AND of the address and mask. For example, if you type the **network 10.1.1.1 0.0.255.255** command, IOS inverts the WC mask (to 255.255.0.0), ANDs this value with 10.1.1.1, resulting in 10.1.0.0. As a result, IOS stores the command **network 10.1.0.0 0.0.255.255**.

The second feature is that when you know for sure the values in the **network** command, you can easily find the range of interface addresses that match the address/WC mask combination in the **network** command. The low end of the range is the address as listed in the **network** command. To find the high end of the range, just add the address and WC mask together. For example, the **network 10.1.0.0 0.0.31.255** command has a range of 10.1.0.0 through 10.1.31.255. (Note that the math suggested in this paragraph does not work when the wildcard mask does not have a single string of consecutive binary 0s followed by a single string of consecutive binary 1s.)

Finally, on R3, the **network 10.1.0.0 0.0.255.255** command tells R3 to enable EIGRP on all interfaces whose IP addresses begin with 10.1, which includes all three interfaces on R3, as shown in Figure 2-1.

Taking a step back from the details, this config has enabled EIGRP, with ASN 1, on all three routers, and on all interfaces shown in Figure 2-1—except one interface. R2's Fa0/1 interface is not matched by any **network** commands on R2, so EIGRP is not enabled on that interface. The next section reviews the commands that can be used to confirm that EIGRP is enabled, the interfaces on which it is enabled, the neighbor relationships that have been formed, and which EIGRP routes have been advertised and learned.

Verification Review

Even before starting to configure the routers, an engineer first considers all requirements. Those requirements lead to a design, which in turn leads to a chosen set of configuration commands. Then, the verification process that follows must consider the design requirements. The goal of verification is to determine that the internetwork works as designed, not just that some EIGRP routes have been learned.

For the purposes of this section, assume that the only design goal for the internetwork in Figure 2-1 is that EIGRP be used so that all routers have routes to reach all subnets shown in the figure.

To verify such a simple design, an engineer should start by confirming on which interfaces EIGRP has been enabled on each router. The next step should be to determine if the EIGRP neighbor relationships that should occur are indeed up and working. Then, the

EIGRP topology table should be examined to confirm that there is at least one entry for each subnet or network in the design. Finally, the IP routes on each router should be examined, confirming that all routes are known. To that end, Table 2-2 summarizes five key **show** commands that provide the information to answer these questions:

Note: The table mentions some information that is covered later in this chapter (passive interfaces) or in other chapters (successor/feasible successors).

Example 2-2 shows samples of each command listed in Table 2-2. Note that the output highlights various samples of items that should be verified: the interfaces on which EIGRP is enabled, the known neighbors, the subnets in the topology table, and the EIGRP routes.

Key Topic

Table 2-2 *Key EIGRP Verification Commands*

Command	Key Information
show ip eigrp interfaces	Lists the working interfaces on which EIGRP is enabled (based on the **network** commands); it omits passive interfaces.
show ip protocols	Lists the contents of the **network** configuration commands for each routing process, and a list of neighbor IP addresses.
show ip eigrp neighbors	Lists known neighbors; does not list neighbors for which some mismatched parameter is preventing a valid EIGP neighbor relationship.
show ip eigrp topology	Lists all successor and feasible successor routes known to this router. It does not list all known topology details. (See Chapter 3 for more detail on successors and feasible successors.)
show ip route	Lists the contents of the IP routing table, listing EIGRP-learned routes with a code of D on the left side of the output.

Example 2-2 *EIGRP Verification on Routers R1, R2, and R3*

```
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 1


                        Xmit Queue    Mean    Pacing Time    Multicast     Pending
Interface         Peers  Un/Reliable  SRTT    Un/Reliable    Flow Timer    Routes
Fa0/0               0        0/0          0        0/1             0            0
Se0/0/0             1        0/0         25        0/15          123            0
Se0/0/1             1        0/0         23        0/15          111            0
Fa0/1               0        0/0          0        0/1             0            0
```

```
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R2#show ip protocols
Routing Protocol is "eigrp 1"
  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 metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.1.2.2/32
    10.1.0.0/19
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.12.1              90      00:19:36
    10.1.23.1              90      00:19:36
  Distance: internal 90 external 170
```

```
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.23.2               Se0/0/1          11 00:19:53   31   200  0  6
0   10.1.13.1               Se0/0/0          10 00:19:53   32   200  0  6
```

```
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.222.2)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.13.0/30, 2 successors, FD is 2681856
         via 10.1.23.1 (2681856/2169856), Serial0/0/0
         via 10.1.12.1 (2681856/2169856), Serial0/0/1
P 10.1.12.0/30, 1 successors, FD is 2169856
         via Connected, Serial0/0/1
P 10.1.3.0/26, 1 successors, FD is 2172416
         via 10.1.23.1 (2172416/28160), Serial0/0/0
P 10.1.2.0/25, 1 successors, FD is 28160
         via Connected, FastEthernet0/0
```

```
P 10.1.1.0/24, 1 successors, FD is 2172416
        via 10.1.12.1 (2172416/28160), Serial0/0/1
P 10.1.23.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0/0
P 192.168.9.0/24, 1 successors, FD is 2172416
        via 10.1.12.1 (2172416/28160), Serial0/0/1
```

```
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

D    192.168.9.0/24 [90/2172416] via 10.1.13.1, 00:19:55, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
C       10.1.13.0/30 is directly connected, Serial0/0/0
D       10.1.12.0/30 [90/2681856] via 10.1.23.2, 00:19:55, Serial0/0/1
                     [90/2681856] via 10.1.13.1, 00:19:55, Serial0/0/0
C       10.1.3.0/26 is directly connected, FastEthernet0/0
D       10.1.2.0/25 [90/2172416] via 10.1.23.2, 00:19:55, Serial0/0/1
D       10.1.1.0/24 [90/2172416] via 10.1.13.1, 00:19:55, Serial0/0/0
C       10.1.23.0/30 is directly connected, Serial0/0/1
```

To verify the interfaces on which EIGRP is enabled, both the **show ip eigrp interfaces** command (shown on R1), and the **show ip protocols** command (shown on R2) list the information. Later in this chapter, in the "Preventing Unwanted Neighbors Using Passive Interfaces" section, the discussion around the **passive-interface** EIGRP configuration subcommand shows an example of how one command lists passive EIGRP interfaces, and the other does not. For this example, look at the list of interfaces in R2's **show ip protocols** command output: S0/0/0, S0/0/1, and FA0/0 are listed, but Fa0/1—unmatched by any of R2's **network** commands—is not.

In this design, each router should form a neighbor relationship with the other two routers, in each case over a point-to-point serial link. The **show ip eigrp neighbors** command (on R3) confirms R3's neighbors.

Finally, one design goal was for all routers to have routes for all subnets/networks. You could move on to the **show ip route** command or first look for all prefixes in the **show ip eigrp topology** command. With relatively general requirements, just looking at the IP routing table is fine. The example highlights R3's topology data and IP route for subnet 10.1.1.0/24. Of more interest might be the fact that the **show ip route** command output on

R3 lists all subnet/network numbers except one: subnet 10.1.22.0/27. This subnet exists off R2's Fa0/1 interface, which is the interface on which EIGRP has not yet been enabled.

Internals Review

To complete the review of prerequisite CCNA-level EIGRP knowledge, this section looks at a few of the internals of EIGRP. Some of the facts listed here simply need to be memorized, whereas other topics will be discussed in more detail in the next several chapters.

EIGRP follows three general steps to add routes to the IP routing table, as follows:

Step 1. **Neighbor discovery**: EIGRP routers send Hello messages to discover potential neighboring EIGRP routers and perform basic parameter checks to determine which routers should become neighbors.

Step 2. **Topology Exchange**: Neighbors exchange full topology updates when the neighbor relationship comes up, and then only partial updates as needed based on changes to the network topology.

Step 3. **Choosing Routes**: Each router analyzes their respective EIGRP topology tables, choosing the lowest-metric route to reach each subnet.

Because the majority of the rest of this chapter examines EIGRP neighborships, this review section skips any discussion of EIGRP neighbors, instead focusing on the second and third items in the preceding list.

Exchanging Topology Information

Each of these steps may cause a router to update one of three key tables used by EIGRP. First, the EIGRP neighbor table lists the neighboring routers. Second, the EIGRP topology table holds all the topology information learned from EIGRP neighbors. Finally, EIGRP chooses the best IP routes and places those into the IP routing table. (Table 2-2 earlier in this chapter lists the **show** commands that can be used to examine these tables.) EIGRP routers follow the process shown in Figure 2-2 to build the necessary information in these tables, with the end goal of populating the IP routing table.

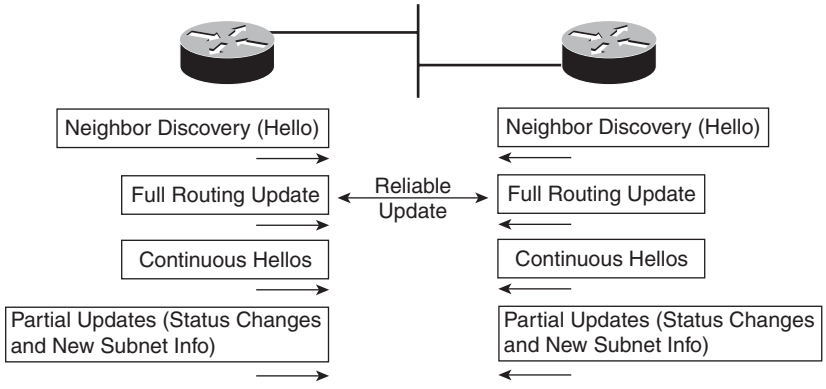


Figure 2-2 *EIGRP Discovery and Update Process*

EIGRP uses *Update messages* to send topology information to neighbors. These Update messages can be sent to multicast IP address 224.0.0.10 if the sending router needs to update multiple routers on the same subnet. Unlike OSPF, there is no concept of a designated router (DR) or backup designated router (BDR), but the use of multicast packets on LANs allows EIGRP to exchange routing information with all neighbors on the LAN efficiently.

The update messages are sent using the *Reliable Transport Protocol (RTP)*. The significance of RTP is that, like OSPF, EIGRP resends routing updates that are lost in transit. By using RTP to guarantee delivery of the EIGRP messages, EIGRP can better avoid loops.

Note: The acronym RTP also refers to a different protocol, Real-time Transport Protocol (RTP), which is used to transmit voice and video IP packets.

Neighbors use both full routing updates and partial updates as depicted in Figure 2-2. A full update means that a router sends information about all known routes, whereas a partial update includes only information about recently changed routes. Full updates occur when neighbors first come up. After that, the neighbors send only partial updates in reaction to changes to a route.

Calculating the Best Routes for the Routing Table

EIGRP topology information includes the subnet number and mask, along with the components of the EIGRP composite metric. Each router then calculates an integer metric for each route, using the individual values of the EIGRP metric components listed in the EIGRP topology database. By default, EIGRP only uses the bandwidth and delay settings when calculating the metric. Optionally, the calculation can also include interface load and interface reliability, although Cisco recommends against using either.

Note: Past documents and books often stated that EIGRP, and its predecessor IGRP, also could use MTU as a part of the metric, but MTU cannot be used and was never considered as part of the calculation. However, the MTU is listed in the EIGRP Update messages.

EIGRP calculates the metric for each possible route by inserting the values of the composite metric into a formula. If the choice is made to just use the default parameters of bandwidth and delay the formula is as follows:

$$\text{Metric} = \left(\left(\frac{10^7}{\text{least-bandwidth}} \right) + \text{cumulative-delay} \right) * 256$$

In this formula, the term *least-bandwidth* represents the lowest-bandwidth link in the route, using a unit of kilobits per second. For instance, if the slowest link in a route is a 10 Mbps Ethernet link, the first part of the formula is $10^7 / 10^4$, because 10 Mbps equals 10,000 Kbps, or 10^4 Kbps. You use 10^4 in the formula because 10 Mbps is equal to 10,000 kbps (10^4 kbps). The *cumulative-delay* value used by the formula is the sum of all the delay values for all links in the route, with a unit of "tens of microseconds." You can set both bandwidth and delay for each link, using the **bandwidth** and **delay** interface subcommands.

Table 2-3 summarizes some of the key facts about EIGRP.

This completes the CCNA-level EIGRP review. The rest of this chapter now examines EIGRP neighbor relationships.

Table 2-3 *EIGRP Feature Summary*

Feature	Description
Transport	IP, protocol type 88 (does not use UDP or TCP).
Metric	Based on constrained bandwidth and cumulative delay by default, and optionally load and reliability.
Hello interval	Interval at which a router sends EIGRP Hello messages on an interface.
Hold Timer	Timer used to determine when a neighboring router has failed, based on a router not receiving any EIGRP messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.10, with retransmissions being sent to each neighbor's unicast IP address. Can also be sent to the neighbor's unicast IP address.
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used.
Authentication	Supports MD5 authentication only.
VLSM/classless	EIGRP includes the mask with each route, also allowing it to support discontiguous networks and VLSM.
Route Tags	Allows EIGRP to tag routes as they are redistributed into EIGRP.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Manual route summarization	Allows route summarization at any point in the EIGRP network.
Automatic Summarization	EIGRP supports, and defaults to use, automatic route summarization at classful network boundaries.
Multiprotocol	Supports the advertisement of IPX and AppleTalk routes, and IP Version 6, which is discussed in Chapter 17 of this book.

EIGRP Neighborships

Like OSPF, EIGRP uses three major steps to achieve its goal of learning the best available loop-free routes:

Step 1. Establish EIGRP neighbor relationships—*neighborships*—with other routers that share a common subnet.

Step 2. Exchange EIGRP topology data with those neighbors.

Step 3. Calculate the currently best IP route for each subnet, based on the known EIGRP topology data, and add those best routes to the IP routing table.

This three-step process hinges on the first step—the successful creation of neighbor relationships between EIGRP routers. The basic EIGRP configuration described earlier in this chapter, particularly the **network** command, most directly tells EIGRP on which interfaces to dynamically discover neighbors. After EIGRP neighborships have been formed with neighboring routers that are reachable through those interfaces, the final two steps occur without any additional direct configuration.

EIGRP dynamically discovers neighbors by sending EIGRP Hello messages on each EIGRP-enabled interface. When two routers hear EIGRP Hello messages from each other, they check the EIGRP parameters listed in those messages and decide whether the two routers should or should not become neighbors.

The rest of this section focuses on topics related to EIGRP neighborship, specifically:

- Manipulating EIGRP Hello and Hold Timers
- Controlling whether routers become neighbors by using either passive interfaces or statically defined neighbors
- Authenticating EIGRP neighbors
- Examining configuration settings that can prevent EIGRP neighborships

Manipulating EIGRP Hello and Hold Timers

The word *convergence* defines the overall process by which routers notice internetwork topology changes, communicate about those changes, and change their routing tables to contain only the best currently working routes. EIGRP converges very quickly even with all default settings.

One of the slower components of the EIGRP convergence process relates to the timers EIGRP neighbors use to recognize that the neighborship has failed. If the interface over which the neighbor is reachable fails, and IOS changes the interface state to anything other than "up/up", then a router immediately knows that the neighborship should fail. However, in some cases, the interface state may stay "up/up" during times when the link may not be usable. In such cases, EIGRP convergence relies on the Hold Timer to expire, which by default on LANs means a 15-second wait. (The default EIGRP hold time with interfaces/subinterfaces with a bandwidth of T1 or slower, with encapsulation of Frame Relay, is 60 seconds.)

The basic operation of these two timers is relatively simple. EIGRP uses the Hello messages in part as a confirmation that the link between the neighbors still works. If a router does not receive a Hello from a neighbor for one entire Hold time, that router considers the neighbor to have failed. For example, with a default LAN setting of Hello of 5, and Hold of 15, the local router sends Hellos every 5 seconds. The neighbor resets its downward-counting Hold Timer to 15 upon receiving a Hello from that neighbor. Under normal operation on a LAN, with defaults, the Hold Timer for a neighbor would vary from 15, down to 10, and then be reset to 15. However, if the Hellos were no longer received for 15 seconds, the neighborship would fail, driving convergence.

To optimize convergence, an engineer could simply reduce the Hello and Hold Timers, accepting insignificant additional overhead, in return for shorter convergence times. These settings can be made per interface/subinterface, and per EIGRP process.

Note: Although expected to be outside the scope of CCNP, EIGRP can also use the Bidirectional Forwarding Detection (BFD) feature that provides a means for subsecond detection of a failure in IP connectivity between two neighboring routers.

Configuring the Hello/Hold Timers

Most design engineers would normally choose Hello/Hold Timers that match on all router interfaces on a subnet. However, these settings do not have to match. More interestingly, by setting the Hello and Hold Timers to nondefault values, you can see some oddities with the configuration of these values.

For example, consider four WAN distribution routers, as shown in Figure 2-3. These routers may each have a number of Frame Relay PVCs to remote branches, or multiple MPLS VPN connections to branches. However, to communicate with each other and with data centers at the home office, these four routers connect via a core VLAN/subnet. Note that the design shows routers, rather than Layer 3 switches, but the concept applies the same in either case.

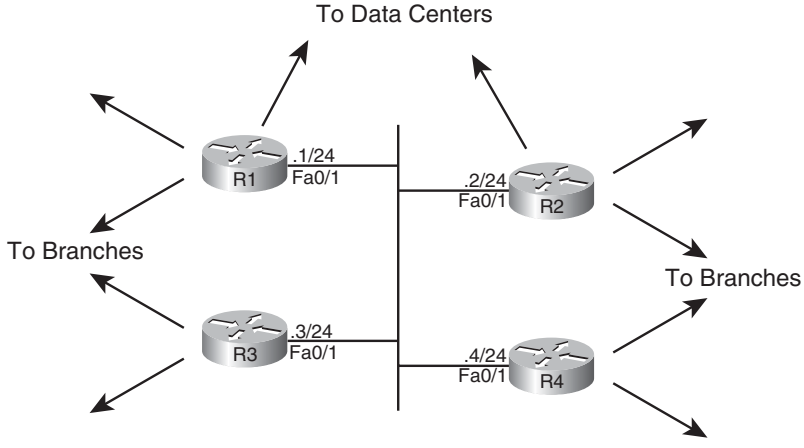


Figure 2-3 *Four WAN Distribution Routers on the Same VLAN/Subnet*

A design that hoped to speed EIGRP convergence might call for setting the Hello and Hold Timers to 2 and 6, respectively. (The Hold Timer does not have to be three times the Hello timer, but the 3:1 ratio is a reasonable guideline.) However, to make an important point about operation of the configuration commands, Example 2-3 sets only R1's Fa0/1 timers to the new values. Note that in this case, EIGRP has already been configured on all four routers, using ASN 9.

Example 2-3 *EIGRP Hello and Hold Timer Configuration—R1*

```
interface Fastethernet0/1
 ip hello-interval eigrp 9 2
 ip hold-time eigrp 9 6
```

A couple of interesting points can be made about the operation of these seemingly simple commands. First, these two settings can be made per interface/subinterface, but not per neighbor. In Figure 2-3, the Example 2-3 configuration then applies on R1 for all three neighbors reachable on interface Fa0/1.

The second interesting point about these commands is that one parameter (the Hello interval) tells R1 what to do, whereas the other (the Hold Timer) actually tells the neighboring routers what to do. As shown in Figure 2-4, **ip hello-interval eigrp 9 2** interface subcommand tells R1 to send Hellos every 2 seconds. However, the **ip hold-time eigrp 9 6** interface subcommand tells R1, again for the EIGRP process with ASN 9, to tell its neighbors to use a Hold Timer of 6 for their respective neighbor relationships with R1. In short, the EIGRP Hello message sent by R1 announces the Hold Timer that other routers should use in the neighbor relationship with R1. Figure 2-4 shows the same idea in graphical form.

Note: IOS does not prevent you from making the unfortunate configuration choice of setting the Hold timer to a value smaller than the Hello interval. In such a case, the neighborship repeatedly fails and recovers, flapping routes in and out of the routing table.

Verifying the Hello/Hold Timers

Interestingly, finding the settings for the Hello interval and Hold time requires more effort than simply using a **show** command. A router's Hello timer can be seen with the **show ip eigrp interface** *type number* **detail** command, but this command does not display a router's Hold Timer on the interface. You can of course look at a router's configuration, but the **show running-config** command may not be available to you on some question types on the ROUTE exam. However, if you have access to only user mode, you can typically make a good guess as to the settings by repeatedly using the **show ip eigrp neighbors** command. This command shows the current value of the Hold Timer for each neighbor. By repeating the command during times when everything is working, and observing the range of values, you should be able to infer each router's Hello and Hold Timer settings.

Example 2-4 shows examples of the command output on R1, R2, and R3. Note that the Hello and Hold Timer settings on R1 are all between 10—15 seconds, because the timers

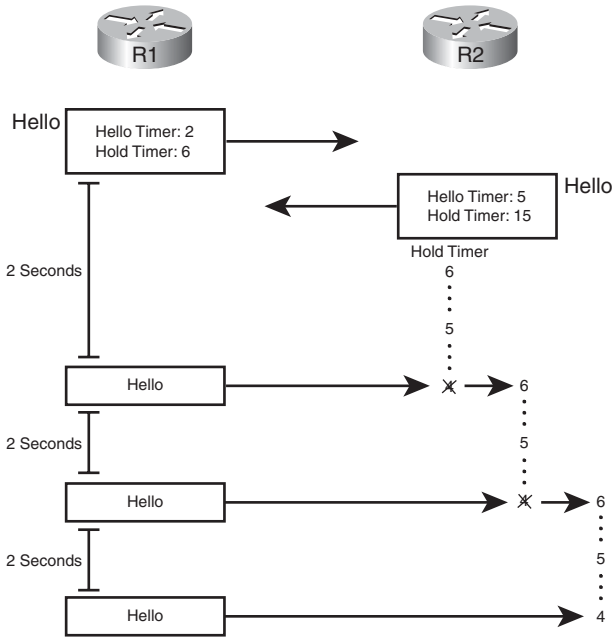


Figure 2-4 *R1 Announcing New Hello and Hold Timers*

on R2, R3, and R4 all still default to 5 and 15 seconds, respectively. R2's neighborship with R1 lists a Hold Timer of 4, which is within the expected range from 6 to 4 seconds remaining.

Example 2-4 *Demonstration that R2 and R3 Use R1's Configured Hold Timer*

```
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R1#show ip eigrp interfaces detail fa0/1
IP-EIGRP interfaces for process 9

                        Xmit Queue    Mean    Pacing Time    Multicast     Pending
Interface         Peers  Un/Reliable  SRTT    Un/Reliable    Flow Timer    Routes
Fa0/1               3        0/0        535        0/1             50             0
  Hello interval is 2 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/1  Un/reliable ucasts: 4/9
  Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 1
  Retransmissions sent: 2  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast

R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 9
```

```
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   172.16.1.4              Fa0/1             11 00:03:17 1596  5000  0  7
1   172.16.1.3              Fa0/1             11 00:05:21    1   200  0  5
0   172.16.1.2              Fa0/1             13 00:09:04    4   200  0  2
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 9
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   172.16.1.4              Fa0/1             11 00:03:36    4   200  0  6
1   172.16.1.3              Fa0/1             11 00:05:40   12   200  0  4
0   172.16.1.1              Fa0/1              4 00:09:22    1   200  0  2
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 9
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   172.16.1.4              Fa0/1             11 00:03:40    4   200  0  5
1   172.16.1.1              Fa0/1              5 00:05:44 1278  5000  0  4
0   172.16.1.2              Fa0/1            13 00:05:44 1277  5000  0  4
```

Preventing Unwanted Neighbors Using Passive Interfaces

When an EIGRP **network** configuration subcommand matches an interface, EIGRP on that router does two things:

Step 1. Attempts to find potential EIGRP neighbors by sending Hellos to the 224.0.0.10 multicast address

Step 2. Advertises about the subnet connected to that interface

In some cases, however, no legitimate EIGRP neighbors may exist off an interface. For example, consider the small internetwork of Figure 2-5, with three routers, and with only one router connected to each LAN interface. Each router needs to advertise about the subnets connected to their various FastEthernet interfaces, but at the same time, there is no benefit to multicast EIGRP Hellos on those interfaces because only one router connects to each LAN.

The network designer may reasonably choose to limit EIGRP on those interfaces that have no legitimate EIGRP neighbors. However, the subnets connected to those same interfaces also typically need to be advertised by EIGRP. For example, subnet 10.1.1.0/24, off R1's Fa0/0 interface, still needs to be advertised by EIGRP, even though R1 should never find an EIGRP neighbor on that interface.

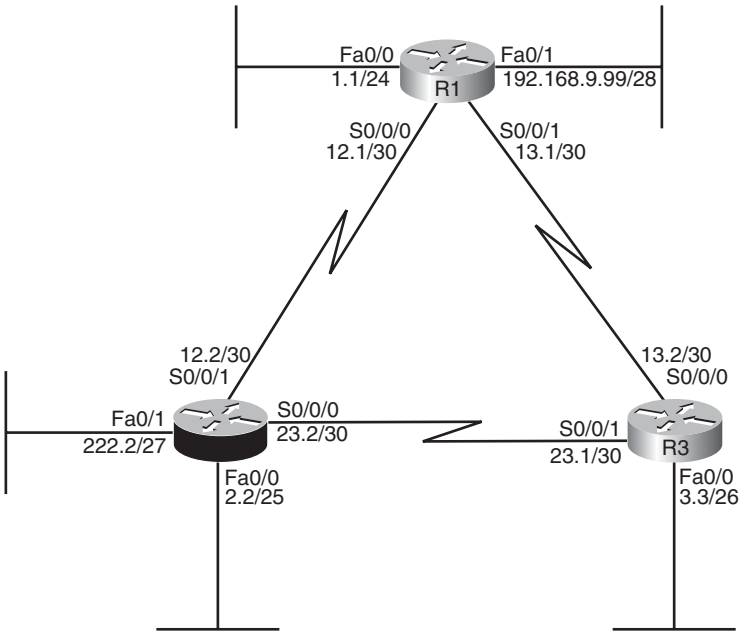


Figure 2-5 *LAN Interfaces That Benefit from the Passive Interface Feature*

Given such a requirement—to advertise about the subnet, but disallow EIGRP neighborships on the interface—an engineer has two main configuration options to add to the implementation plan:

Key Topic

Step 1. Enable EIGRP on the interface using the EIGRP **network** command, but tell the router to not send any EIGRP messages on the interface by making the interface passive (using the **passive-interface** command).

Step 2. Do not enable EIGRP on the interface, and advertise about the connected route using route redistribution (and the **redistribute connected** configuration command).

The first option relies on the passive interface feature—a feature specifically created with this design requirement in mind. When an interface is passive, EIGRP does not send any EIGRP messages on the interface—multicasts or EIGRP unicasts—and the router ignores any EIGRP messages received on the interface. However, EIGRP still advertises about the connected subnets if matched with an EIGRP **network** command. As a result, the first option in the preceding list directly meets all design requirements. It has the added advantage of being very secure in that no EIGRP neighborships are possible on the interface.

The second option—redistributing connected subnets—also works, but frankly it is the less preferred option in this case. The passive interface option clearly meets the requirement, plus the redistribution process means that EIGRP advertises the connected route as

an external EIGRP route, which could cause problems in some cases with multiple redistribution points between routing domains (as discussed in Chapter 10, "Advanced IGP Redistribution").

The configuration of passive-interface itself is somewhat straightforward. To configure the **passive-interface** option, these three routers could be configured as follows in Example 2-5.

Example 2-5 *Configuration of* **passive-interface** *Commands on R1, R2, and R3*

```
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 passive-interface fastethernet0/0
 passive-interface fastethernet0/1
 network 10.0.0.0
 network 192.168.9.0
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 passive-interface default
 no Passive-interface serial0/0/0
 no Passive-interface serial0/0/0
 network 10.0.0.0
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
router eigrp 1
 passive-interface fastethernet0/0
 network 10.0.0.0
```

R1's configuration lists two **passive-interface** commands, one per LAN interface. As a result, R1 no longer sends EIGRP messages at all on these two interfaces, including the multicast EIGRP Hellos used to discover neighbors.

R2's configuration uses a slightly different option: the **passive-interface default** command. This command essentially changes the default for an interface from not being passive to instead being passive. Then, to make an interface not passive, you have to use a **no** version of the **passive-interface** command for those interfaces.

Two commands help to verify that the passive interface design is working properly. First, the **show ip eigrp interfaces** command omits passive interfaces, listing the nonpassive interfaces matched by a **network** command. Alternatively, the **show ip protocols** command explicitly lists all passive interfaces. Example 2-6 shows samples of both commands on R2.

Example 2-6 *Verifying the Results of* **passive-interface** *on R2*

```
R2#show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface         Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0/0             1        0/0        32       0/15          159           0
Se0/0/1             1        0/0      1290       0/15         6443           0
R2#show ip protocols
Routing Protocol is "eigrp 1"
  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 metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    FastEthernet0/0
    FastEthernet0/1
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.12.1             90      00:00:39
    10.1.23.1             90      00:00:39
  Distance: internal 90 external 170
```

Controlling Neighborships Using EIGRP Authentication

EIGRP authentication causes routers to authenticate every EIGRP message. To do so, the routers should use the same preshared key (PSK), generating an MD5 digest for each EIGRP message based on that shared PSK. If a router configured for EIGRP authentication receives an EIGRP message, and the message's MD5 digest does not pass the authentication checking based on the local copy of the key, the router silently discards the message. As a result, when authentication fails, two routers cannot become EIGRP neighbors, because they ignore the EIGRP Hello messages.

From a design perspective, EIGRP authentication helps prevent denial of service (DoS) attacks, but it does not provide any privacy. The EIGRP messages can be read by the device

that physically receives the bits. Note that on LANs, the updates flow to the 224.0.0.10 multicast IP address, so any attacker could join the 224.0.0.10 multicast group and read the packets. However, authentication prevents attackers from forming neighborships with legitimate routers, preventing the advertisement of incorrect routing information.

Next, this section examines EIGRP authentication configuration generically, followed by a deeper look at the time-based authentication configuration settings, and finally showing an example of EIGRP authentication configuration.

EIGRP Authentication Configuration Checklist

The EIGRP authentication configuration process requires several commands, which are summarized as follows:

Step 1. Create an (authentication) key chain:

Create the chain and give it a name with the **key chain** *name* global command (also puts the user into key chain config mode). The name does not have to match on the neighboring routers.

Create one or more key numbers using the **key** *number* command in key chain configuration mode. The key numbers do not have to match on the neighboring routers.

Define the authentication key's value using the **key-string** *value* command in key configuration mode. The key strings must match on the neighboring routers.

(Optional) Define the lifetime (time period) for both sending and accepting each key string.

Step 2. Enable EIGRP MD5 authentication on an interface, for a particular EIGRP ASN, using the **ip authentication mode eigrp** *asn* **md5** interface subcommand.

Step 3. Refer to the correct key chain to be used on an interface using the **ip authentication key-chain eigrp** *asn name-of-chain* interface subcommand.

The configuration at Step 1 is fairly detailed, but Steps 2 and 3 are relatively simple. Essentially, IOS configures the key values separately (Step 1) and then requires an interface subcommand to refer to the key values. To support the ability to have multiple keys, and even multiple sets of keys, the configuration includes the concept of a key chain and multiple keys on each key chain.

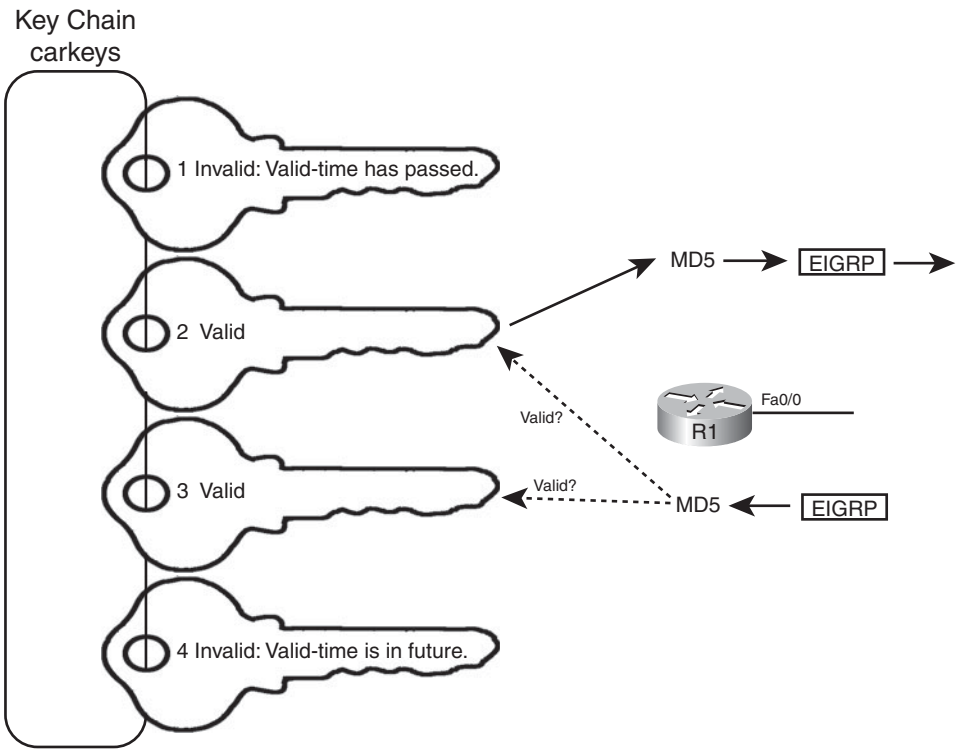
Key Chain Time-Based Logic

The key chain configuration concept, as outlined in Step 1, allows the engineer to migrate from one key value to another over time. Just like a real key chain that has multiple keys, the IOS key chain concept allows the configuration of multiple keys—each identified with a number. If no lifetime has been configured for a key, it is considered to be valid during all time frames. However, when a key has been defined with a lifetime, the key is valid only during the valid lifetime.

The existence of multiple keys in a key chain, and the existence of valid lifetimes for each key, can cause some confusion about when the keys are used. The rules can be summarized as follows:

- **Sending EIGRP messages:** Use the lowest key number among all currently valid keys.
- **Receiving EIGRP message:** Check the MD5 digest using ALL currently valid keys.

For example, consider the case shown in Figure 2-6. The figure represents the logic in a single router, Router R1, both when receiving and sending EIGRP messages on the right. The figure shows a key chain with four keys. All the keys have lifetimes configured. Key 1's lifetime has passed, making it invalid. Key 4's lifetime has yet to begin, making it invalid. However, keys 2 and 3 are both currently valid.



Key Topic

Figure 2-6 *EIGRP's Usage of Authentication Keys*

Figure 2-6 shows that the EIGRP message sent by Router R1 uses key 2, and key 2 only. Keys 1 and 4 are ignored because they are currently invalid; R1 then simply chooses the lowest-numbered key among the two valid keys. The figure also shows that R1 processes the received EIGRP message using both key 2 and key 3, because both are currently valid.

Note: Neighboring EIGRP routers that use authentication should be configured to use NTP to synchronize their time-of-day clocks. For quick tests in a lab, you can just set the time using the **clock set** exec command.

EIGRP Authentication Configuration and Verification Example

Example 2-7 shows a sample configuration, based on the network topology shown back in Figure 2-3 (the figure with four routers connected to a single LAN subnet). Key chain "carkeys" has two keys, each with different lifetimes, so that the router will use new keys

automatically over time. The example shows the configuration on a single router, but similar configuration would be required on the other routers as well.

Example 2-7 *EIGRP Authentication Configuration on R1*

```
! Chain "carkeys" will be used on R1's Fa0/1 interface. R1 will use key "fred" for
! about a month, and then start using "wilma."
key chain carkeys
 key 1
  key-string fred
  accept-lifetime 08:00:00 Feb 11 2009 08:00:00 Mar 11 2009
  send-lifetime 08:00:00 Feb 11 2009 08:00:00 Mar 11 2009
 key 2
  key-string wilma
  accept-lifetime 08:00:00 Mar 11 2009 08:00:00 Apr 11 2009
  send-lifetime 08:00:00 Mar 11 2009 08:00:00 Apr 11 2009

! Next, R1's interface subcommands are shown. First, the key chain is referenced
! using the ip authentication key-chain command, and the ip authentication mode
eigrp
! command causes the router to use an MD5 digest of the key string.
interface FastEthernet0/1
 ip address 172.16.1.1 255.255.255.0
 ip authentication mode eigrp 9 md5
 ip authentication key-chain eigrp 9 carkeys
```

The best method to confirm the authentication worked is to verify that the neighbors remain up using the **show ip eigrp neighbors** command. However, if some neighbors do not remain active, and a problem exists, two commands in particular can be helpful: **show key chain** and **debug eigrp packet**. The first of these commands lists the key chain configuration and also lists which keys are valid right now—a key consideration when troubleshooting EIGRP authentication. The **debug** command lists a message regarding why neighboring routers have failed the authentication process.

Example 2-8 shows a sample output from each of these two commands. Note that in this case, again from Figure 2-3:

- R1 and R2 have been configured to use MD5 authentication, but a typo exists in R2's key string.
- R3 and R4 have not yet been configured for MD5 authentication.

Example 2-8 *EIGRP Authentication Verification on R1*

```
R1#show key chain
Key-chain carkeys:
    key 1 — text "fred"
```

```
        accept lifetime (08:00:00 UTC Feb 11 2009) - (08:00:00 UTC Mar 11 2009)
        send lifetime (08:00:00 UTC Feb 11 2009) - (08:00:00 UTC Mar 11 2009)
    key 2 — text "wilma"
        accept lifetime (08:00:00 UTC Mar 11 2009) - (08:00:00 UTC Apr 11 2009)
[valid now]
        send lifetime (08:00:00 UTC Mar 11 2009) - (08:00:00 UTC Apr 11 2009)
[valid now]
R1#debug eigrp packet
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
SIAREPLY)
R1#
Apr  1 08:09:01.951: EIGRP: Sending HELLO on FastEthernet0/1
Apr  1 08:09:01.951:   AS 9, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
Apr  1 08:09:01.967: EIGRP: pkt key id = 2, authentication mismatch
Apr  1 08:09:01.967: EIGRP: FastEthernet0/1: ignored packet from 172.16.1.2,
opcode = 5 (invalid authentication)
Apr  1 08:09:02.287: EIGRP: FastEthernet0/1: ignored packet from 172.16.1.4,
opcode = 5 (missing authentication)
Apr  1 08:09:03.187: EIGRP: FastEthernet0/1: ignored packet from 172.16.1.3, opcode
= 5 (missing authentication)
```

In particular, note that the different highlighted phrasing in the debug output implies different problems. From 172.16.1.2 (R2), the message "invalid authentication" implies that the MD5 digest existed in the message, but was invalid. With the message "missing authentication," the meaning is that the message did not include an MD5 digest.

Also, when troubleshooting EIGRP authentication, keep the following in mind:

- Examine the configuration and the current time (**show clock**) on both routers.
- The key chain name and key number used on the two routers do not have to match.
- The key string on each of the two potential neighbors must match.
- Check which keys are currently valid using the **show key chain** command.
- Both the **ip authentication mode eigrp** *asn* **md5** interface subcommand and the **ip authentication key-chain eigrp** *asn name-of-chain* interface subcommand must be configured on the interface; if one is omitted, authentication fails.

Controlling Neighborships with Static Configuration

EIGRP supports the ability to statically define neighbors instead of dynamically discovering neighbors.

Although seldom used, you can use this feature to reduce the overhead associated with EIGRP multicast messages. Frame Relay WANs in particular may benefit from the static neighbor definitions because to support multicasts and broadcasts over Frame Relay, a router must replicate the frame and send a copy over every PVC associated with the interface or subinterface. For example, if a multicast subinterface has 10 PVCs associated

with it, but only two of the remote routers used EIGRP, without static neighbors, all 10 routers would be sent a copy of the EIGRP multicast Hello packets. With static neighbor definitions for the two routers, EIGRP messages would be sent as unicasts to each of the two neighbors, with no EIGRP messages sent to the eight non-EIGRP routers, reducing overhead.

The configuration seems simple, but it has a few subtle caveats. This section examines the straightforward configuration first and then examines the caveats.

Configuring Static EIGRP Neighbors

To define a neighbor, both routers must configure the **neighbor** *ip-address outgoing-interface* EIGRP router subcommand. The IP address is the interface IP address of the neighboring router. Also, the configured IP address must be from the subnet connected to the interface listed in the **neighbor** command; otherwise, the command is rejected. Also, note that the EIGRP configuration does not have to include a **network** command that matches the interface; EIGRP will still advertise about the subnet connected to the interface.

For example, consider Figure 2-7, which adds a new router (R5) to the internetwork of Figure 2-3. R1 and R5 have a PVC connecting them, with IP addresses and subinterface numbers shown.

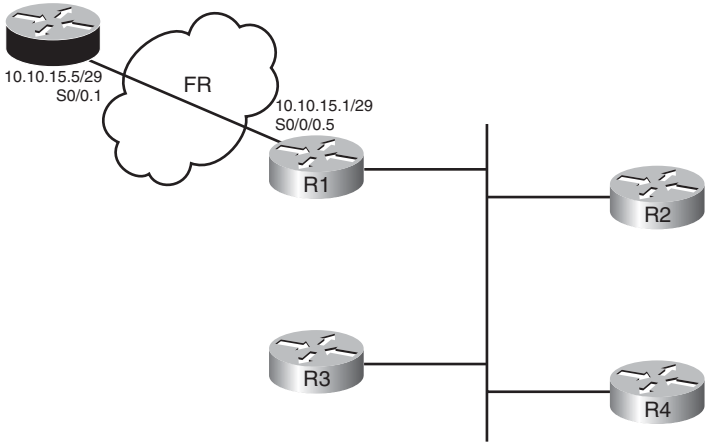


Figure 2-7 *Adding a Branch, with a Static EIGRP Neighbor*

Example 2-9 shows the configuration on both R1 and R5 to use static neighbor definitions. Of note, R1's **neighbor** command refers to R5's IP address on their common subnet (10.10.15.5), with R1's local interface (S0/0/0.5). R5 lists the reverse, with R1's 10.10.15.1 IP address, and R5's local S0/0.1 interface. Also note that neither router has a **network** command that references network 10.0.0.0, but they do advertise about subnet 10.10.15.0/29.

Example 2-9 *Static EIGRP Neighborship Between R1 and R5*

```
! New configuration on router R1
R1#show running-config
! lines omitted
router eigrp 9
 network 172.16.0.0
 no auto-summary
 Neighbor 10.10.15.5 Serial0/0/0.5
```

```
! R5's new config added to support the neighbor
R5#show running-config
! lines omitted
router eigrp 9
 no auto-summary
 neighbor 10.10.15.1 Serial0/0.1
```

```
! Back to R1
R1#show ip eigrp neighbors detail
IP-EIGRP neighbors for process 9
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.10.15.5              Se0/0/0.5         10 00:00:51   15   200  0  2
   Static neighbor
   Version 12.4/1.2, Retrans: 0, Retries: 0
2   172.16.1.2              Fa0/1             11 00:02:57    3   200  0  25
   Version 12.4/1.2, Retrans: 1, Retries: 0
1   172.16.1.3              Fa0/1             10 00:03:45    5   200  0  21
   Version 12.4/1.2, Retrans: 0, Retries: 0
0   172.16.1.4              Fa0/1             13 00:03:45    5   200  0  18
```

The **show ip eigrp neighbors** command does not identify a neighbor as static, but the **show ip eigrp neighbors detail** command does. Example 2-9 shows the more detailed output near the end, with the designation of 10.10.15.5 (R5) as a static neighbor.

Caveat When Using EIGRP Static Neighbors

IOS changes how it processes EIGRP packets on any interface referenced by an EIGRP **neighbor** command. Keeping in mind the design goal for this feature—to reduce multicasts—IOS disables all EIGRP multicast packet processing on an interface when an EIGRP **neighbor** command has been configured. For example, in Example 2-9, R1's S0/0/0.5 subinterface will not process EIGRP multicast packets any more as a result of R1's **neighbor 10.10.5.5 Serial0/0/0.5** EIGRP subcommand.

Because of the operation of the EIGRP **neighbor** command, if at least one EIGRP static neighbor is defined on an interface, no dynamic neighbors can be either discovered or continue to work if already discovered. For example, again in Figure 2-7 and Example 2-9, if R1 added a **neighbor 172.16.1.5 FastEthernet0/1** EIGRP subcommand, R1 would lose its current neighborships with Routers R2, R3, and R4.

Configuration Settings That Could Prevent Neighbor Relationships

Some of the configuration settings already mentioned in this chapter, when configured incorrectly, may prevent EIGRP neighborships. This section summarizes those settings, and introduces a few other configuration settings that can prevent neighbor relationships. The list of items that must match—and that do not have to match—can be a useful place to start troubleshooting neighbor initialization problems in real life, and to troubleshoot neighborship problems for Sim questions on the CCNP ROUTE exam.

Table 2-4 lists the neighbor requirements for both EIGRP and OSPF. (OSPF is included here just as a frame of reference for those more familiar with OSPF; this information will be repeated in Chapter 5, "OSPF Overview and Neighbor Relationships," which discusses OSPF neighborship requirements.) Following the table, the next few pages examine some of these settings for EIGRP.

Key Topic

Table 2-4 *Neighbor Requirements for EIGRP and OSPF*

Requirement	EIGRP	OSPF
The routers must be able to send/receive IP packets to one another.	Yes	Yes
Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
Must not be passive on the connected interface.	Yes	Yes
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.	Yes	No
Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	No	Yes
Must pass neighbor authentication (if configured).	Yes	Yes
Must be in same area.	N/A	Yes
IP MTU must match.	No	Yes
K-values (used in metric calculation) must match.	Yes	N/A
Router IDs must be unique.	No[1]	Yes

[1]Duplicate EIGRP RIDs do not prevent routers from becoming neighbors, but it can cause problems when adding external EIGRP routes to the routing table.

Going through Table 2-4 sequentially, the first two items (highlighted) relate to IP connectivity. Two routers must be able to send and receive IP packets with each other. Additionally, the primary IP address on the interfaces—in other words, the IP address configured *without* the secondary keyword on the **ip address** command—must be in the same subnet.

Note: It should not matter for CCNP ROUTE, but possibly for CCIE: EIGRP's rules about neighbor IP addresses being in the same subnet are less exact than OSPF. OSPF requires matching subnet numbers and masks. EIGRP just asks the question of whether the neighbor's IP address is in the range of addresses for the subnet as known to the local router. For example, two routers with addresses of 10.1.1.1/24 (range 10.1.1.1–10.1.1.254) and 10.1.1.2/30 (range 10.1.1.1–10.1.1.2) would actually allow EIGRP neighborship, because each router believes the neighbor's IP address to be in the same subnet as the local router.

The next four items in Table 2-4 (unhighlighted)—passive interfaces, matching the EIGRP ASN number, allowing mismatching Hello/Hold Timers, and authentication—have already been covered in this chapter, and do not require any further discussion.

The next two (highlighted) items in the table—matching the IP MTU and matching OSPF areas—do not prevent EIGRP neighborships. These topics, are requirements for OSPF neighborship and will be discussed in Chapter 5.

Finally, the last two items (unhighlighted) in the table (K-values and router-id) each require more than a cursory discussion for EIGRP and will be explained in the upcoming pages.

Configuring EIGRP Metric Components (K-values)

EIGRP calculates its integer metric, by default, using a formula that uses constraining bandwidth and cumulative delay. You can change the formula to use link reliability, link load, and even disable the use of bandwidth and/or delay. To change the formula, an engineer can configure five weighting constants, called *k-values*, which are represented in the metric calculation formula as constants k1, k2, k3, k4, and k5.

From a design perspective, Cisco strongly recommends against using link load and link reliability in the EIGRP metric calculation. Most shops that use EIGRP never touch the k-values at all. However, in labs, it can be useful to disable the use of bandwidth from the metric calculation, because that simplifies the metric math, and makes it easier to learn the concepts behind EIGRP.

The mechanics of setting these values (with the **metric weights** EIGRP subcommand) is covered in Chapter 3, in the "Metric Weights (K-values)" section. This command sets 5 variables (k1 through k5), each of which weights the metric calculation formula more or less heavily for various parts of the formula.

Mismatched k-value settings prevent two routers from becoming neighbors. Thankfully, determining if such a mismatch exists is easy. When a router receives an EIGRP Hello with mismatched K-values (as compared to itself), the router issues a log message stating that a k-value mismatch exists. You can also examine the values either by looking at running configuration, or look for the k-values listed in the output of the **show ip protocols** command, as shown in Example 2-10.

Example 2-10 *Mismatched K-values*

```
R2(config)#router eigrp 1
R2(config-router)#metric weights 0 1 0 1 1 0
R2(config-router)#end
Feb 23 18:48:21.599: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.12.1
(Serial0/0/1) is down: metric changed
R2#
Feb 23 18:48:24.907: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.12.1
(Serial0/0/1) is down: K-value mismatch
R2#show ip protocols
Routing Protocol is "eigrp 1"
  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 metric weight K1=1, K2=0, K3=1, K4=1, K5=0

! lines omitted for brevity
```

EIGRP Router-ID

EIGRP uses a concept of a representing each router with a router ID (RID). The EIGRP RID is a 32-bit number, represented in dotted decimal. Each router determines its RID when the EIGRP process starts, using the same general rules as does OSPF for determining the OSPF RID, as follows:

Key Topic

Step 1. Use the configured value (using the **eigrp router-id** *a.b.c.d* EIGRP subcommand).

Step 2. Use the highest IPv4 address on an up/up loopback interface.

Step 3. Use the highest IPv4 address on an up/up non-loopback interface.

Although EIGRP does require each router to have an RID, the actual value is of little practical importance. The EIGRP **show** commands seldom list the RID value, and unlike for the OSPF RID, engineers do not need to know each router's EIGRP RID to interpret the EIGRP topology database. Additionally, although it is best to make EIGRP RIDs unique, duplicate RIDs do not prevent routers from becoming neighbors.

The only time the value of EIGRP RIDs matters is when injecting external routes into EIGRP. In that case, the routers injecting the external routes must have unique RIDs to avoid confusion.

Neighborship over WANs

EIGRP configuration and neighborship rules do not differ when comparing typical LAN and typical WAN technologies. However, some design and operational differences exist, particularly regarding which routers become neighbors with which other routers. This short section closes the EIGRP neighbor discussion with a brief look at Frame Relay, MPLS VPNs, and Metro Ethernet as implemented with Virtual Private LAN Service (VPLS).

Neighborship on Frame Relay

Frame Relay provides a Layer 2 WAN service. Each router connects to the service using a physical serial link, called a Frame Relay access link. The provider then creates logical connection, called *permanent virtual circuits (PVCs)*, which is a logical path between a pair of routers connected to the Frame Relay service. Any pair of routers that connect to the ends of a Frame Relay PVC can send Frame Relay frames to each other, IP packets, and they can become EIGRP neighbors. Figure 2-8 shows a typical case, with R1 as a central-site router, and R2, R3, and R4 acting as branch routers.

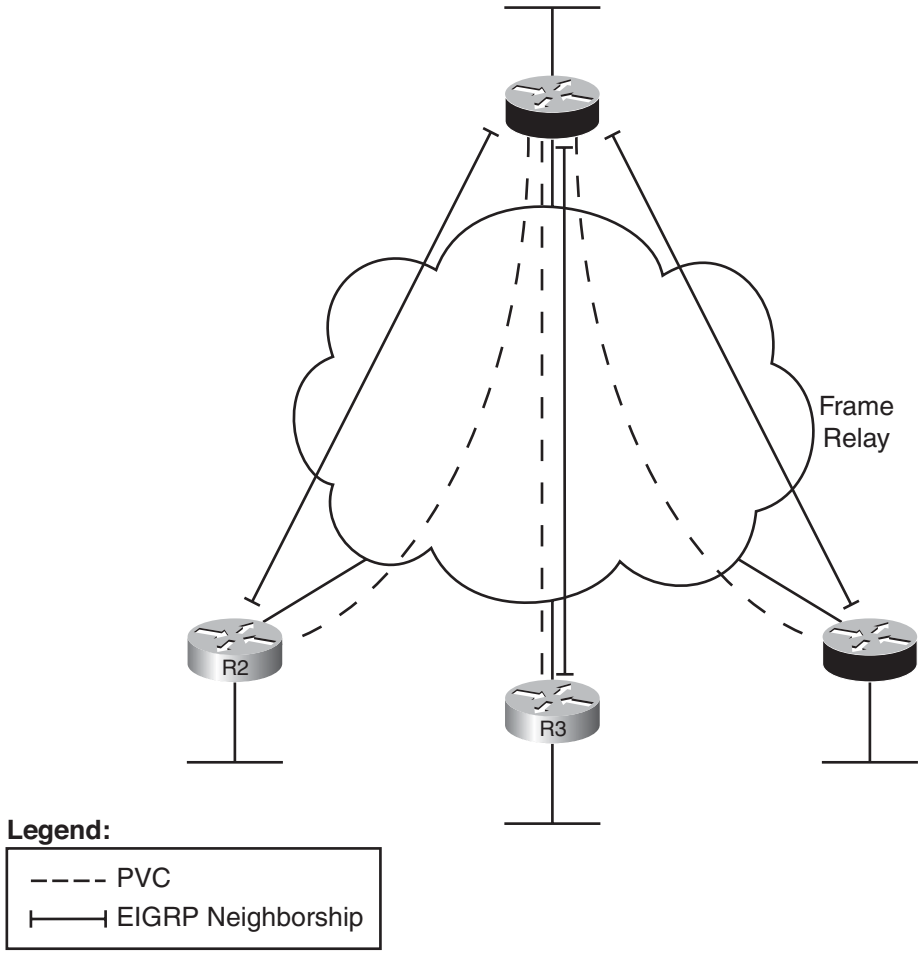


Figure 2-8 *EIGRP Neighborships over Frame Relay*

Figure 2-8 shows EIGRP neighborships, but note that all routers can learn all routes in the internetwork, even though not all routers become neighbors. The neighborships can only form when a PVC exists between the two routers.

Neighborship on MPLS VPN

Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs) create a WAN service that has some similarities but many differences when compared to Frame Relay. The customer routers connect to the service, often times with serial links, but other times with Frame Relay PVCs or with Ethernet. The service itself is a Layer 3 service, forwarding IP packets through the cloud. As a result, no pre-defined PVCs need exist between the customer routers. Additionally, the service uses routers at the edge of the service provider cloud—generically called provider edge (PE) routers—and these routers are Layer 3 aware.

That Layer 3 awareness means that the customer edge (CE) routers form an EIGRP neighborship with the PE router on the other end of their local access link, as shown in Figure 2-9. The PE routers exchange their routes, typically using Multiprotocol BGP (MP-BGP), a topic outside the scope of this book. However, all the CE routers then learn routes from each other, although each CE router has only one EIGRP neighborship for each of its connections into the MPLS VPN cloud.

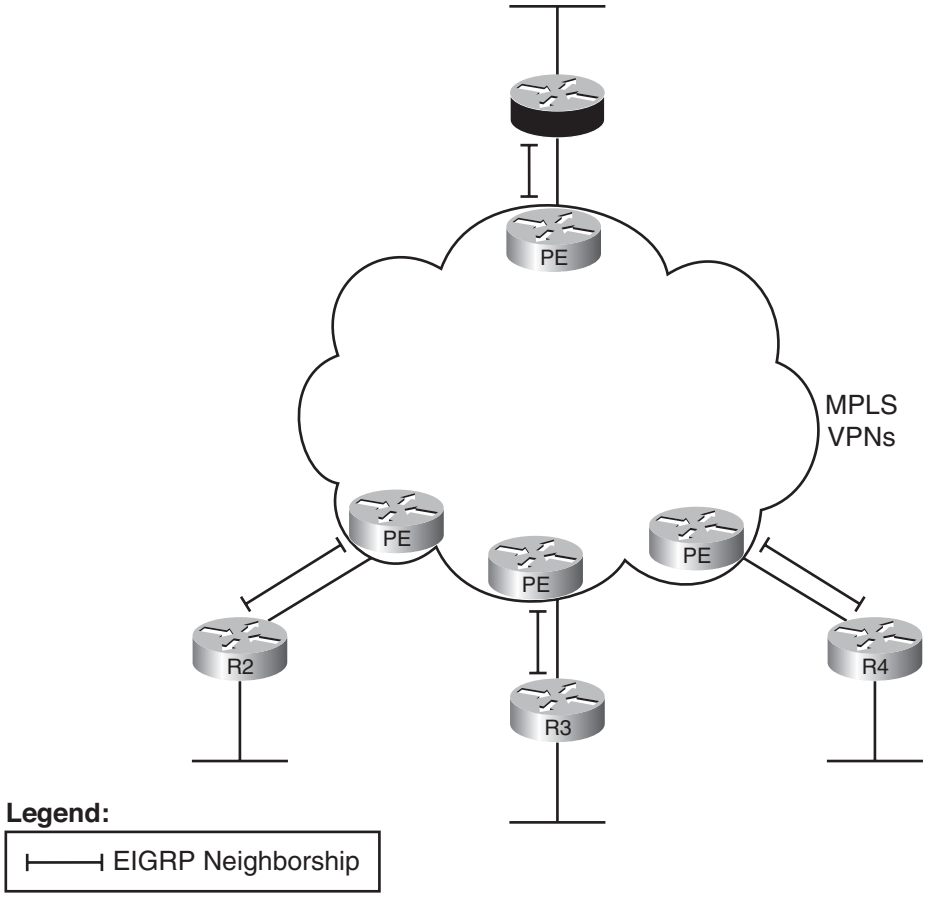


Figure 2-9 *EIGRP Neighborships over MPLS VPN*

Neighborship on Metro Ethernet

The term *Metropolitan Ethernet (MetroE)* represents a range of Layer 2 WAN services in which the CE device connects to the WAN service using some form of Ethernet. Because MetroE provides a Layer 2 Ethernet service, the service delivers an Ethernet frame sent by one customer router to one other customer router (for unicast frames), or to many other routers (for multicast or broadcast frames).

MetroE encompasses several underlying technologies to create the service. Of note for the purposes of this book are the Virtual Private Wire Service (VPWS) and the Virtual Private LAN Service (VPLS). Both technical specifications allow for connections using Ethernet links, with the service forwarding Ethernet frames. VPWS focuses on point-to-point topologies, whereas VPLS supports multipoint, approximating the concept of the entire WAN service acting like one large Ethernet switch. Because it is a Layer 2 service, MetroE does not have any Layer 3 awareness, and the customer routers (typically referenced as with the more general service provider term *customer premise equipment*, or CPE) see the MetroE service as a VLAN. Because the customer routers connect to the service as a VLAN, all the routers connected to the service can become EIGRP neighbors, as shown in Figure 2-10.

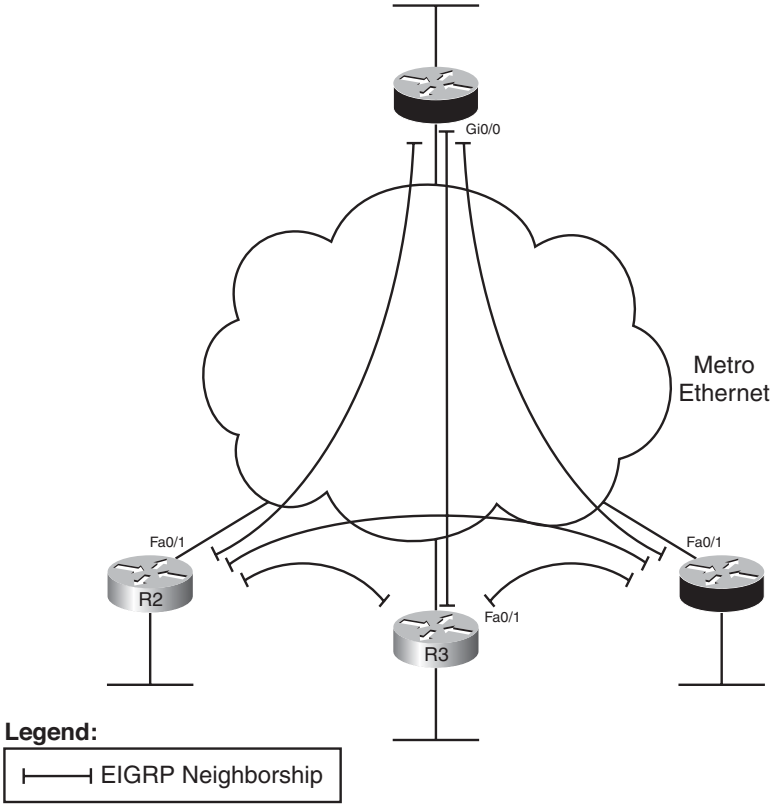


Figure 2-10 *EIGRP Neighborships over Metro Ethernet*

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to be able to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables," which you can find on the CD-ROM accompanying this book.

Design Review Table

Table 2-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 2-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Improve EIGRP convergence.	
Implement EIGRP on each router so that neighborships are formed (2).	
Limit neighborship formation on interfaces matched with an EIGRP network command (3).	

Implementation Plan Peer Review Table

Table 2-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 2-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
What happens on a router interface on which an **EIGRP network** command matches the interface? (2)	
What configuration settings prevent EIGRP neighbor discovery on an EIGRP-enabled interface? (2)	
What configuration settings prevent any neighborships on an EIGRP-enabled interface?	
What settings do potential neighbors check before becoming EIGRP neighbors? (5)	
What settings that you might think would impact EIGRP neighbor relationships actually do not prevent neighborship? (3)	
What issues typically arise when the design calls for the use of EIGRP authentication key chains with lifetime settings? (2)	

Create an Implementation Plan Table

To practice skills useful when creating your own EIGRP implementation plan, list in Table 2-7 configuration commands related to the configuration of the following features. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 2-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling EIGRP on interfaces	
Setting Hello and Hold Timers	
EIGRP authentication	
Passive interfaces	
Static EIGRP neighbors	
K-values	
EIGRP router ID	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own EIGRP verification plan, list in Table 2-8 all commands that supply the requested information. You may want to record your an-

swers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 2-8 *Verification Plan Memory Drill*

Information Needed	Command
Which routes have been added to the IP routing table by EIGRP?	
All routes in a router's routing table.	
The specific route for a single destination address or subnet.	
A list of all (both static and dynamically discovered) EIGRP neighbors.	
Notation of whether a neighbor was dynamically discovered or statically configured.	
Lists statistics regarding the numbers of EIGRP messages sent and received by a router.	
List interfaces on which EIGRP has been enabled (by virtue of the EIGRP **network** command).	
List the number of EIGRP peers known via a particular interface.	
The elapsed time since a neighborship was formed.	
The parameters of any EIGRP **network** commands.	
The configured Hello timer for an interface.	
The configured Hold Timer for an interface.	
The current actual Hold Timer for a neighbor.	
A router's EIGRP ASN.	
A list of EIGRP passive interfaces.	
A list of nonpassive EIGRP interfaces.	
The currently used EIGRP authentication key, when sending EIGRP packets.	
The currently used EIGRP authentication key, when receiving EIGRP packets.	
Lists EIGRP K-values.	
Lists traffic statistics about EIGRP.	
A router's EIGRP Router ID.	

Review All the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 2-9 lists a reference of these key topics and the page numbers on which each is found.

Table 2-9 *Key Topics for Chapter 2*

Key Topic Element	Description	Page Number
List	Configuration step review for basic EIGRP configuration	23
Table 2-2	Key EIGRP verification commands	26
Table 2-3	Summary of EIGRP features and facts	31
List	Methods of disallowing EIGRP neighborships on an interface, while still advertising the connected subnet	37
Step list	Configuration checklist for EIGRP authentication	40
Figure 2-6	Conceptual view of which keys are used by EIGRP authentication based on timeframe	41
List	EIGRP authentication troubleshooting hints	43
Table 2-4	List of items that may impact the formation of EIGRP neighborships	46
List	Rules for choosing an EIGRP Router ID	48

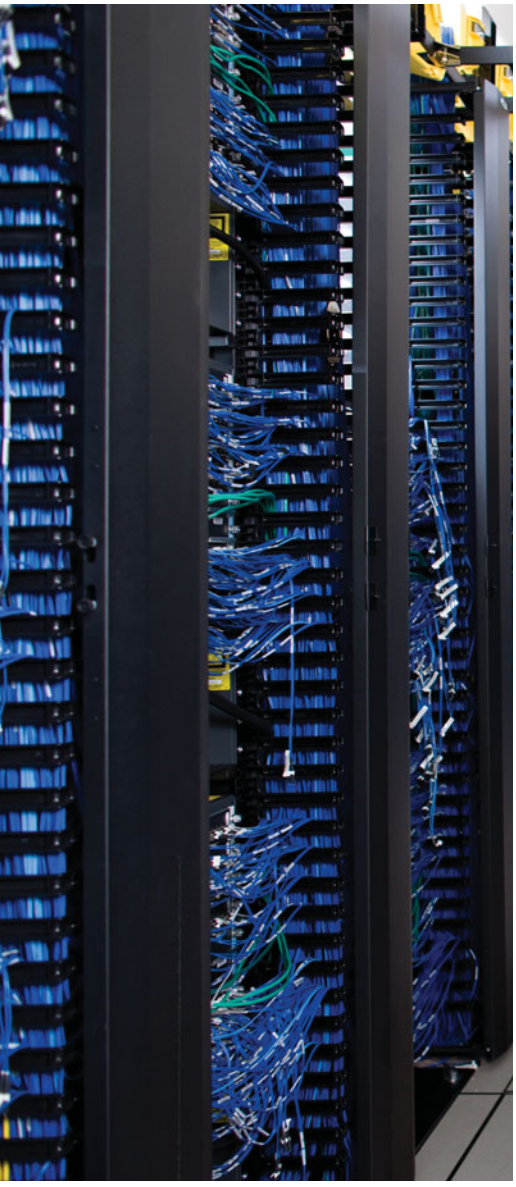
Complete the Tables and Lists from Memory

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

Define Key Terms

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

K-value, neighborship, Hello interval, Hold Timer, passive interface



This chapter covers the following subjects:

Building the EIGRP Topology Table: This section discusses how a router seeds its local EIGRP topology table, and how neighboring EIGRP routers exchange topology information.

Building the IP Routing Table: This section explains how routers use EIGRP topology data to choose the best routes to add to their local routing tables.

Optimizing EIGRP Convergence: This section examines the items that have an impact on how fast EIGRP converges for a given route.

CHAPTER 3

EIGRP Topology, Routes, and Convergence

EIGRP, like OSPF, uses three major branches of logic, each of which populates a different table. EIGRP begins by forming neighbor relationships and listing those relationships in the EIGRP neighbor table (as described in Chapter 2, "EIGRP Overview and Neighbor Relationships"). EIGRP then exchanges topology information with these same neighbors, with newly learned information being added to the router's EIGRP topology table. Finally, each router processes the EIGRP topology table to choose the currently best IP routes, adding those IP routes to the IP routing table.

This chapter moves from the first major branch (neighborships, as covered in Chapter 2) to the second and third branches: EIGRP topology and EIGRP routes. To that end, the first major section of this chapter describes the protocol used by EIGRP to exchange the topology information and details exactly what information EIGRP puts in its messages between routers. The next major section shows how EIGRP examines the topology data to then choose the currently best route for each prefix. The final section of this chapter examines how to optimize the EIGRP convergence processes so that when the topology does change, the routers in the internetwork quickly converge to the then-best routes.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these nine self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 3-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings, so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

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

Foundations Topics Section	Questions
Building the EIGRP Topology Table	1-3
Building the IP Routing Table	4–7
Optimizing EIGRP Convergence	8, 9

1. Which of the following are methods EIGRP uses to initially populate (seed) its EIGRP topology table, before learning topology data from neighbors? (Choose two.)

   a. By adding all subnets listed by the **show ip route connected** command

   b. By adding the subnets of working interfaces over which static neighbors have been defined

   c. By adding subnets redistributed on the local router from another routing source

   d. By adding all subnets listed by the **show ip route static** command

2. Which of the following are both advertised by EIGRP in the Update message and included in the formula for calculating the integer EIGRP metric? (Choose two.)

   a. Jitter

   b. Delay

   c. MTU

   d. Reliability

3. Router R1 uses S0/0 to connect via a T/1 to the Frame Relay service. Five PVCs terminate on the serial link. Three PVCs (101, 102, and 103) are configured on subinterface S0/0.1, and one each (104 and 105) are on S0/0.2 and S0/0.3. The configuration shows no configuration related to EIGRP WAN bandwidth control, and the **bandwidth** command is not configured at all. Which of the following is true about how IOS tries to limit EIGRP's use of bandwidth on S0/0?

   a. R1 limits EIGRP to around 250Kbps on DLCI 102.

   b. R1 limits EIGRP to around 250Kbps on DLCI 104.

   c. R1 limits EIGRP to around 150Kbps on every DLCI.

   d. R1 does not limit EIGRP because no WAN bandwidth control has been configured.

4. The output of **show ip eigrp topology** on Router R1 shows the following output, which is all the output related to subnet 10.11.1.0/24. How many feasible successor routes does R1 have for 10.11.1.0/24?

```
P 10.11.1.0/24, 2 successors, FD is 2172419
        via 10.1.1.2 (2172423/28167), Serial0/0/0.1
        via 10.1.1.6 (2172423/28167), Serial0/0/0.2
```

   a. 0

   b. 1

   c. 2

   d. 3

5. A network design shows that R1 has four different possible paths from itself to the Data Center subnets. Which of the following can influence which of those routes become feasible successor routes, assuming that you follow the Cisco recommended practice of not changing metric weights? (Choose two.)

   a. The configuration of EIGRP offset lists

   b. Current link loads

c. Changing interface delay settings

d. Configuration of variance

6. Router R1 is three router hops away from subnet 10.1.1.0/24. According to various **show interfaces** commands, all three links between R1 and 10.1.1.0/24 use the following settings: bandwidth: 1000, 500, 100000 and delay: 12000, 8000, 100. Which of the following answers correctly identifies a value that feeds into the EIGRP metric calculation? (Choose two correct answers.)

   a. Bandwidth of 101,500

   b. Bandwidth of about 34,000

   c. Bandwidth of 500

   d. Delay of 1200

   e. Delay of 2010

   f. Delay of 20100

7. Routers R1 and R2 are EIGRP neighbors. R1 has been configured with the **eigrp stub connected** command. Which of the following is true as a result? (Choose two correct answers.)

   a. R1 can learn EIGRP routes from R2, but R2 cannot learn EIGRP routes from R1.

   b. R1 can send IP packets to R2, but R2 cannot send IP packets to R1.

   c. R2 no longer learns EIGRP routes from R1 for routes not connected to R1.

   d. R1 no longer replies to R2's Query messages.

   e. R2 no longer sends to R1 Query messages.

8. A network design shows that R1 has four different possible paths from itself to the Data Center subnets. Which one of the following commands is most likely to show you all the possible next-hop IP addresses for these four possible routes?

   a. **show ip eigrp topology**

   b. **show ip eigrp topology all-links**

   c. **show ip route eigrp**

   d. **show ip route eigrp all-links**

   e. **show ip eigrp topology all-learned**

9. Router R1 lists 4 routes for subnet 10.1.1.0/24 in the output of the **show ip eigrp topology all-links** command. The **variance 200** command is configured, but no other related commands are configured. Which of the following rules are true regarding R1's decision of what routes to add to the IP routing table? Note that RD refers to reported distance and FD to feasible distance.

   a. Adds all routes for which the metric is <= 200 * the best metric among all routes

   b. Adds all routes because of the ridiculously high **variance** setting

   c. Adds all successor and feasible successor routes

   d. Adds all successor and feasible successor routes for which the metric is <= 200 * the best metric among all routes

Foundation Topics

Building the EIGRP Topology Table

The overall process of building the EIGRP topology table is relatively straightforward. EIGRP defines some basic topology information about each route for each unique prefix/length (subnet). This basic information includes the prefix, prefix length, metric information, and a few other details. EIGRP neighbors exchange topology information, with each router storing the learned topology information in their respective EIGRP topology table. EIGRP on a given router can then analyze the topology table, or topology database, and choose the best route for each unique prefix/length.

EIGRP uses much simpler topology data than does OSPF, which is a link state protocol that must describe the entire topology of a portion of a network with its topology database. EIGRP, essentially an advanced distance vector protocol, does not need to define nearly as much topology data, nor do EIGRP routers need to run the complex Shortest Path First (SPF) algorithm. This first major section examines the EIGRP topology database, how routers create and flood topology data, and some specific issues related to WAN links.

Seeding the EIGRP Topology Table

Before a router can send EIGRP topology information to a neighbor, that router must have some topology data in its topology table. Routers can, of course, learn about subnets and the associated topology data from neighboring routers. However to get the process started, each EIGRP router needs to adds topology data for some prefixes, so it can then advertise these routes to its EIGRP neighbors. A router's EIGRP process adds subnets to its local topology table, without learning the topology data from an EIGRP neighbor, from three sources:

- Prefixes of connected subnets for interfaces on which EIGRP has been enabled on that router using the **network** command
- Prefixes of connected subnets for interfaces referenced in an EIGRP **neighbor** command
- Prefixes learned by the redistribution of routes into EIGRP from other routing protocols or routing information sources

After a router adds such prefixes to its local EIGRP topology database, that router can then advertise the prefix information, along with other topology information associated with each prefix, to each working EIGRP neighbor. Each router adds any learned prefix information to their topology table, and then that router advertises the new information to other neighbors. Eventually, all routers in the EIGRP domain learn about all prefixes–unless some other feature, such as route summarization or route filtering, alters the flow of topology information.

The Content of EIGRP Update Message

EIGRP uses five basic protocol messages to do its work:

- Hello
- Update
- Query
- Reply
- ACK (acknowledgment)

EIGRP uses two messages as part of the topology data exchange process: Update and Ack. The Update message contains the topology information, whereas the ACK acknowledges receipt of the update packet.

The EIGRP Update message contains the following information:

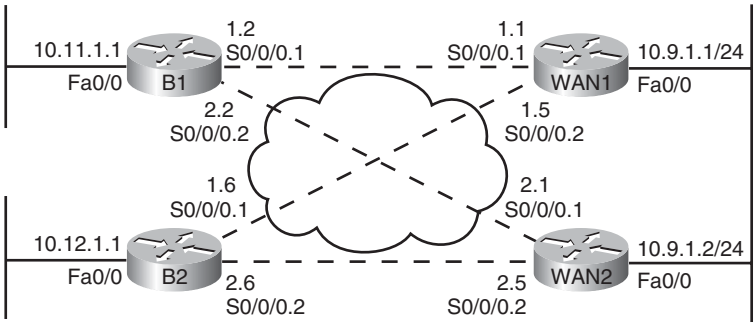
- Prefix
- Prefix length
- Metric components: bandwidth, delay, reliability, and load
- Nonmetric items: MTU and hop count

Note: Many courses and books over the years have stated that MTU is part of the EIGRP metric. In practice, the MTU has never been part of the metric calculation, although it is included in the topology data for each prefix.

To examine this whole process in more detail, see Figure 3-1 and Figure 3-2. Figure 3-1 shows a portion of an Enterprise network that will be used in several examples in this chapter. Routers B1 and B2 represent typical branch office routers, each with two Frame Relay PVCs connected back to the main site. WAN1 and WAN2 are WAN distribution routers, each of which would normally have dozens or hundreds of PVCs.

The routers in Figure 3-1 have been configured and work. For EIGRP, all routers have been configured with as many defaults as possible, with the only configuration related to EIGRP being the **router eigrp 1** and **network 10.0.0.0** commands on each router.

Next, consider what Router B1 does for its connected route for subnet 10.11.1.0/24, which is located on B1's LAN. B1 matches its Fa0/0 interface IP address (10.11.1.1) due to its **network 10.0.0.0** configuration command. So as mentioned earlier, B1 seeds its own topology table with an entry for this prefix. This topology table entry also lists the interface bandwidth of the associated interface and delay of the associated interface. Using default settings for FastEthernet interfaces, B1 uses a bandwidth of 100,000 Kbps (the same as 100 Mbps) and a delay of 10, meaning 10 tens-of-microseconds. Router B1 also includes a default setting for the load (1) and reliability (255), even though the router, using the default K-value settings, will not use these values in its metric calculations. Finally, B1 adds to the topology database the MTU of the local interface and a hop count of zero because the subnet is connected.



Note: All WAN IP addresses begin with 10.1.

Figure 3-1 *Typical WAN Distribution and Branch Office Design*

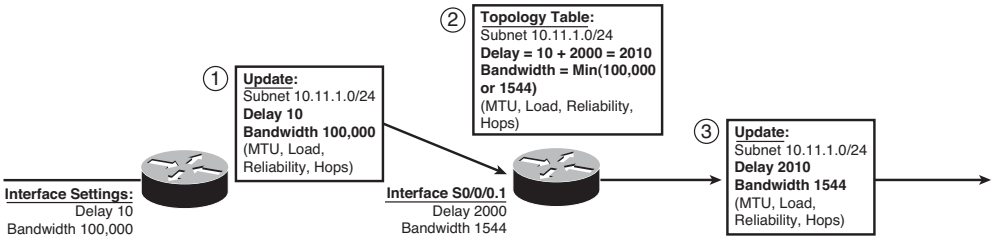


Figure 3-2 *Contents of EIGRP Update Messages*

Now that B1 has added some topology information to its EIGRP topology database, Figure 3-2 shows how B1 propagates the topology information to router WAN1 and beyond.

The steps in Figure 3-2 can be explained as follows:

Step 1. B1 advertises the prefix (10.11.1.0/24) using an EIGRP Update message. The message includes the four metric components, plus MTU and hop count–essentially the information in B1's EIGRP topology table entry for this prefix.

Step 2. WAN1 receives the Update message and adds the topology information for 10.11.1.0/24 to its own EIGRP topology table, with these changes:

WAN1 considers the interface on which it received the Update (S0/0/0.1) to be the outgoing interface of a potential route to reach 10.11.1.0/24.

WAN1 adds the delay of S0/0/0.1 (2000 tens-of-microseconds per Figure 3-2) to the delay listed in the Update message.

WAN1 compares the bandwidth of S0/0/0.1 (1544 Kbps per Figure 3-2) to the bandwidth listed in the Update message (100,000 Kbps) and chooses the lower value (1544) as the bandwidth for this route.

WAN1 also updates load (highest value), reliability (lowest value), and MTU (lowest value) based on similar comparisons, and adds 1 to the hop count.

Step 3. WAN1 then sends an Update to its neighbors, with the metric components listed in its own topology table.

This example provides a good backdrop to understand how EIGRP uses cumulative delay and minimum bandwidth in its metric calculation. Note that at Step 2, router WAN1 adds to the delay value but does not add the bandwidth. For bandwidth, WAN1 simply chooses the lowest bandwidth, comparing the bandwidth of its own interface (S0/0/0.1) with the bandwidth listed in the received EIGRP update.

Next, consider this logic on other routers—not shown in the figure—as WAN1 floods this routing information throughout the Enterprise. WAN1 then sends this topology information to another neighbor, and that router sends the topology data to another, and so on. If bandwidth of those links was 1544 or higher, the bandwidth setting used by those routers would remain the same, because each router would see that the routing update's bandwidth (1544 Kbps) was lower than the link's bandwidth. However, each router would add something to the delay.

As a final confirmation of the contents of this Update process, Example 2-1 shows the details of the EIGRP topology database for prefix 10.11.1.0/24 on both B1 and WAN1.

Example 3-1 *Topology Database Contents for 10.11.1.0/24, on B1 and WAN1*

```
! On Router B1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
B1#show ip eigrp topology 10.11.1.0/24
IP-EIGRP (AS 1): Topology entry for 10.11.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 28160
  Routing Descriptor Blocks:
  0.0.0.0 (FastEthernet0/0), from Connected, Send flag is 0x0
      Composite metric is (28160/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
```

```
! On Router WAN1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WAN1#show ip eigrp topology 10.11.1.0/24
IP-EIGRP (AS 1): Topology entry for 10.11.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  10.1.1.2 (Serial0/0/0.1), from 10.1.1.2, Send flag is 0x0
      Composite metric is (2172416/28160), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20100 microseconds
```

```
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
      Hop count is 1
```

The highlighted portions of output match the details shown in Figure 3-2, but with one twist relating to the units on the delay setting. The IOS **delay** command, which lets you set the delay, along with the data stored in the EIGRP topology database, use a unit of tens-of-microsecond. However, the **show interfaces** and **show ip eigrp topology** commands list delay in a unit of microseconds. For example, WAN1's listing of "20100 microseconds" matches the "2010 tens-on-microseconds" shown in Figure 3-2.

The EIGRP Update Process

So far, this chapter has focused on the detailed information EIGRP exchanges with a neighbor about each prefix. This section takes a broader look at the process.

When EIGRP neighbors first become neighbors, they begin exchanging topology information using Update messages using these rules:

- When a neighbor first comes up, the routers exchange full updates, meaning the routers exchange all topology information.
- After all prefixes have been exchanged with a neighbor, the updates cease with that neighbor if no changes occur in the network. There is no periodic reflooding of topology data.
- If something changes–for example, one of the metric components change, links fail, links recover, new neighbors advertise additional topology information–the routers send partial updates about only the prefixes whose status or metric components have changed.
- If neighbors fail and then recover, or new neighbor adjacencies are formed, full updates occur over these adjacencies.
- EIGRP uses Split Horizon rules on most interfaces by default, which impacts exactly which topology data EIGRP sends during both full and partial updates.

Split Horizon, the last item in the list, needs a little more explanation. Split Horizon limits the prefixes that EIGRP advertises out an interface. Specifically, if the currently best route for a prefix lists a particular outgoing interface, Split Horizon means that EIGRP will not include that prefix in the Update sent out that same interface. For example, router WAN1 uses S0/0/0.1 as its outgoing interface for subnet 10.11.1.0/24, so WAN1 would not advertise prefix 10.11.1.0/24 in its Update messages sent out S0/0/0.1.

Note: Route summarization and route filtering, as explained in Chapter 4, "EIGRP Route Summarization and Filtering," also affect which subsets of the topology table are flooded.

To send the Updates, EIGRP uses the *Reliable Transport Protocol (RTP)* to send the EIGRP updates and confirm their receipt. On point-to-point topologies such as serial links, MPLS VPN, and Frame Relay when using point-to-point subinterfaces, the EIGRP Update and ACK messages use a simple process of acknowledging each Update with an ACK. On multiaccess data links, EIGRP typically sends Update messages to multicast address 224.0.0.10 and expects a unicast EIGRP ACK message from each neighbor in reply. RTP manages that process, setting timers so that the sender of an Update waits a reasonable time, but not too long, before deciding whether all neighbors received the Update or whether one or more neighbors did not reply with an ACK.

Although EIGRP relies on the RTP process, network engineers cannot manipulate how it works.

WAN Issues for EIGRP Topology Exchange

With all default settings, after you enable EIGRP on all the interfaces in an internetwork, the topology exchange process typically does not pose any problems. However, a few scenarios exist, particularly on Frame Relay, which can cause problems. This section summarizes two issues and shows the solution.

Split Horizon Default on Frame Relay Multipoint Subinterfaces

IOS support for Frame Relay allows the configuration of IP addresses on the physical serial interface, multipoint subinterfaces, and point-to-point subinterfaces. Additionally, IP packets can be forwarded over a PVC even when the routers on the opposite ends do not have to use the same interface or subinterface type. As a result, many small intricacies exist in the operation of IP and IP routing protocols over Frame Relay, particularly related to default settings on different interface types.

Frame Relay supports several reasonable configuration options using different interfaces and subinterfaces, each meeting different design goals. For instance, if the design includes a few centralized WAN distribution routers, with PVCs connecting each branch router to each distribution router, both distribution and branch routers might use point-to-point subinterface. Such a choice makes the Layer 3 topology simple, with all links acting like point-to-point links from a Layer 3 perspective. This choice also removes issues such as Split Horizon.

In some cases, a design might include a small set of routers that have a full mesh of PVCs connecting each. In this case, multipoint subinterfaces might be used, consuming a single subnet, reducing the consumption of the IP address space. This choice also reduces the number of subinterfaces.

Both options–using point-to-point subinterfaces or using multipoint subinterfaces–have legitimate reasons for being used. However, when using the multipoint subinterface option, a particular EIGRP issue can occur when the following are true:

- Three or more routers, over Frame Relay, are configured as part of a single subnet.
- The routers use multipoint interfaces.
- Either permanently or for a time, a full mesh of PVCs between the routers does not exist.

For example, consider Router WAN1 shown earlier in Figure 3-1 and referenced again in Figure 3-3. In the earlier configurations, the WAN distribution routers and branch routers all used point-to-point subinterfaces and a single subnet per VC. To see the problem raised in this section, consider that same internetwork, but now the engineer has chosen to configure WAN1 to use a multipoint subinterface and a single subnet for WAN1, B1, and B2, as shown in Figure 3-3.

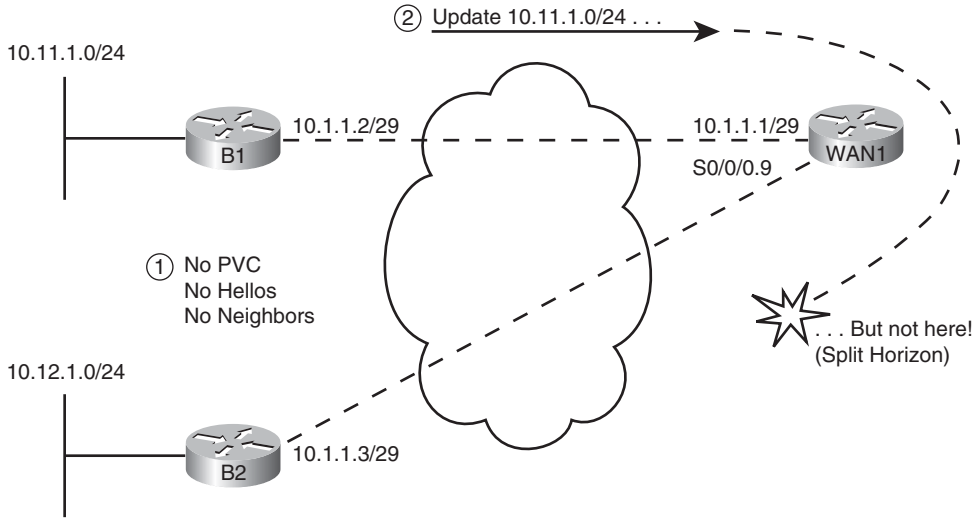


Figure 3-3 *Partial Mesh, Central Sites (WAN1) Uses Multipoint Subinterface*

The first issue to consider in this design is that B1 and B2 will not become EIGRP neighbors with each other, as noted with Step 1 in the figure. EIGRP routers must be reachable using Layer 2 frames before they can exchange EIGRP Hello messages and become EIGRP neighbors. In this case, there is no PVC between B1 and B2. B1 exchanges Hellos with WAN1, and become neighbors, as will B2 with WAN1. However, routers do not forward received EIGRP Hellos, so WAN1 will not receive a Hello from B1 and forward it to B2 or vice versa. In short, although in the same subnet (10.1.1.0/29), B1 and B2 will not become EIGRP neighbors.

The second problem occurs due to Split Horizon logic on Router WAN1, as noted with Step 2 in the figure. As shown with Step 2 in the figure, B1 could advertise its routes to WAN1, and WAN1 could advertise those routes to B2–and vice versa. However, with default settings, WAN1 will not advertise those routes due to its default setting of Split Horizon (a default interface subcommand setting of **ip split-horizon eigrp** *asn*.) As a result, WAN1 receives the Update from B1 on its S0/0/0.9 subinterface, but Split Horizon prevents WAN1 from advertising that topology data to B2 in Updates sent out interface S0/0/0.9, and vice versa.

The solution is somewhat simple–just configure the **no ip split-horizon eigrp** *asn* command on the multipoint subinterface on WAN1. The remote routers, B1 and B2 in this

case, still do not become neighbors, but that does not cause a problem by itself. With Split Horizon disabled on WAN1, B1 and B2 learn routes to the other branch's subnets. Example 3-2 lists the complete configuration and the command to disable Split Horizon:

Note: Frame Relay configuration is considered a prerequisite because it is part of the CCNA exam and courses. Example 3-2 uses **frame-relay interface-dlci** commands and relies on Inverse ARP. However, if **frame-relay map** commands were used instead, disabling Inverse ARP, the EIGRP details discussed in this example would remain unchanged.

Example 3-2 *Frame Relay Multipoint Configuration on WAN1*

```
! On Router WAN1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
interface Serial0/0/0
 no ip address
 encapsulation frame-relay

interface Serial0/0/0.9 multipoint
 ip address 10.1.1.1 255.255.255.248
 no ip split-horizon eigrp 1
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
!
router eigrp 1
 network 10.0.0.0
```

Note: The [**no**] **ip split-horizon** command controls Split Horizon behavior for RIP; the [**no**] **ip split-horizon eigrp** *asn* command controls Split Horizon behavior for EIGRP.

Displaying the EIGRP Split Horizon state of an interface is an unusually difficult piece of information to find without simply displaying the configuration. By default, IOS enables EIGRP Split Horizon. To find the setting for an interface, look for the presence or absence of the **no ip split-horizon eigrp** command on the configuration. Also, the **debug ip eigrp** command output displays messages stating when prefixes are not advertised out an interface due to split horizon.

EIGRP WAN Bandwidth Control

In a multiaccess WAN, one physical link passes traffic for multiple data link layer destinations. For example, a WAN distribution router connected to many branches using Frame Relay might literally terminate hundreds, or even thousands, of Frame Relay PVCs.

In a nonbroadcast multiaccess (NBMA) medium such as Frame Relay, when a router needs to send EIGRP updates, the Updates cannot be multicasted at Layer 2. So, the router must

send a copy of the Update to each reachable neighbor. For a WAN distribution router with many Frame Relay PVCs, the sheer amount of traffic sent over the Frame Relay access link might overload the link.

The EIGRP WAN bandwidth control allows the engineer to protect a multiaccess Frame Relay interface from being overrun with too much EIGRP message traffic. By default, a router sends EIGRP messages out an interface but only up to 50 percent of the bandwidth defined on the interface with the **bandwidth** command. The engineer can adjust this percentage using the **ip bandwidth-percent eigrp** *asn percent* interface/subinterface subcommand. Regardless of the percentage, IOS then limits the rate of sending the EIGRP messages so that the rate is not exceeded. To accomplish this, IOS queues the EIGRP messages in memory, delaying them briefly.

The command to set the bandwidth percentage is simple, but there are a few caveats to keep in mind when trying to limit the bandwidth consumed by EIGRP:

- The IOS default for bandwidth serial interfaces and subinterfaces is 1544 (Kbps).
- EIGRP limits the consumed bandwidth based on the percentage of interface/subinterface bandwidth.
- This feature keys on the bandwidth of the interface or subinterface through which the neighbor is reachable, so don't set only the physical interface bandwidth and forget the subinterfaces.
- Recommendation: Set the bandwidth of point-to-point links to the speed of the Committed Information Rate (CIR) of the single PVC on the subinterface.
- General recommendation: Set the bandwidth of multipoint subinterfaces to around the total CIR for all VCs assigned to the subinterface.
- Note that for multipoint subinterfaces, IOS WAN bandwidth control first *divides the subinterface bandwidth by the number of configured PVCs* and then determines the EIGRP percentage based on that number.

For example, consider Figure 3-4, which shows a router with one multipoint subinterface and one point-to-point subinterface. With the configuration shown in Example 3-3, WAN1 uses the following bandwidth, at most, with each neighbor:

- B1, B2, and B3: 20 Kbps (20% of 300Kbps / 3 VCs)
- B4: 30 Kbps (30% of 100 Kbps)

Example 3-3 *Configuration of WAN1, One Multipoint, One Point-to-Point*

```
! On Router WAN1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
interface Serial0/0/0.20 multipoint
 ip address 172.16.1.1 255.255.255.240
 frame-relay interface-dlci 201
 frame-relay interface-dlci 202
 frame-relay interface-dlci 203
```

```
  bandwidth 300
   ip bandwidth-percent eigrp 1 20
!
interface Serial0/0/0.21 point-to-point
 ip address 172.16.1.17 255.255.255.252
 frame-relay interface-dlci 221
  bandwidth 100
  ip bandwidth-percent eigrp 1 30
```

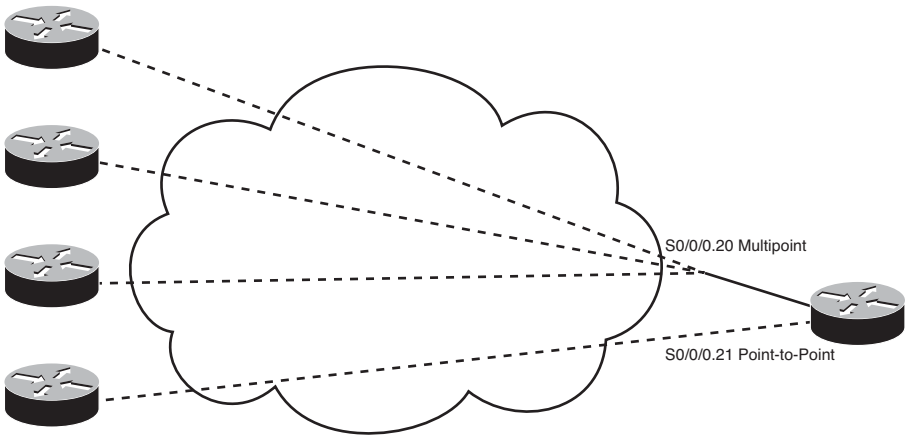


Figure 3-4 *WAN1, One Multipoint, One Point-to-Point*

Building the IP Routing Table

An EIGRP router builds IP routing table entries by processing the data in the topology table. Unlike OSPF, which uses a computationally complex SPF process, EIGRP uses a computationally simple process to determine which, if any, routes to add to the IP routing table for each unique prefix/length. This part of the chapter examines how EIGRP chooses the best route for each prefix/length and then examines several optional tools that can influence the choice of routes to add to the IP routing table.

Calculating the Metrics: Feasible Distance and Reported Distance

The EIGRP topology table entry, for a single prefix/length, lists one or more possible routes. Each possible route lists the various component metric values–bandwidth, delay, and so on. Additionally, for connected subnets, the database entry lists an outgoing interface. For routes not connected to the local router, in addition to an outgoing interface, the database entry also lists the IP address of the EIGRP neighbor that advertised the route.

EIGRP routers calculate an integer metric based on the metric components. Interesting, an EIGRP router does this calculation both from its own perspective and from the perspective of the next-hop router of the route. The two calculated values are

Key Topic

- **Feasible Distance (FD):** Integer metric for the route, from the local router's perspective, used by the local router to choose the best route for that prefix.
- **Reported Distance (RD):** Integer metric for the route, from the neighboring router's perspective (the neighbor that told the local router about the route). Used by the local router when converging to new routes.

Note: Some texts use the term *Advertised Distance (AD)* instead of Reported Distance (RD) as used in this book. Be ready for either term on the CCNP ROUTE exam. However, this book uses RD exclusively.

Routers use the FD to determine the best route, based on the lowest metric, and use the RD when falling back to an alternative route when the best route fails. (EIGRP's use of the RD is explained in the upcoming section "Successor and Feasible Successor Concepts.") Focusing on the FD, when a router has calculated the integer FD for each possible route to reach a single prefix/length, that router can then consider adding the lowest-metric route to the IP routing table.

As a reminder, the following formula shows how EIGRP calculates the metric, assuming default settings of the EIGRP metric weights (K-values). The metric calculation grows when the slowest bandwidth in the end-to-end route decreases (the slower the bandwidth, the worse the metric), and its metric grows (gets worse) when the cumulative delay grows:

Metric = 256 * ((10^7 / slowest-bandwidth) + cumulative-delay)

An example certainly helps in this case. Figure 3-5 repeats some information about the topology exchange process between Routers B1 and WAN1 (refer to Figure 3-1), essentially showing the metric components as sent by B1 to WAN1 (Step 1) and the metric components from WAN1's perspective (Step 2).

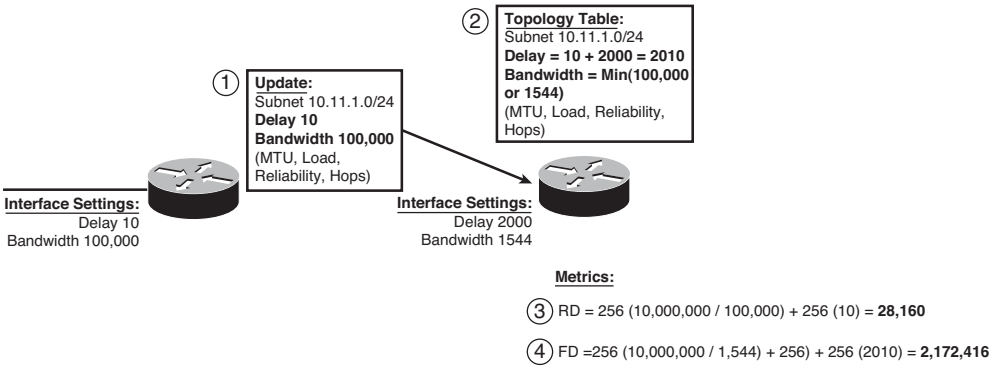


Figure 3-5 *Example Calculation of RD and FD on Router WAN1*

Steps 3 and 4 in Figure 3-5 show WAN1's calculation of the RD and FD for 10.11.1.0/24, respectively. Router WAN1 takes the metric components as received from B1, and plugs them into the formula, to calculate the RD, which is the same integer metric that Router

B1 would have calculated as its FD. Step 4 shows the same formula but with the metric components as listed at Step 2—after the adjustments made on WAN1. Step 4 shows WAN1's FD calculation, which is much larger due to the much lower constraining bandwidth plus the much larger cumulative delay.

WAN1 chooses its best route to reach 10.11.1.0/24 based on the lowest FD among all possible routes. Looking back to the much more detailed Figure 3-1, presumably a couple of other routes might have been possible, but WAN1 happens to choose the route shown in Figure 3-5 as its best route. As a result, WAN1's **show ip route** command lists the FD calculated in Figure 3-5 as the metric for this route, as shown in Example 3-4.

Example 3-4 *Router WAN1's EIGRP Topology and IP Route Information for 10.11.1.0/24*

```
! Below, note that WAN1's EIGRP topology table lists two possible next-hop
! routers: 10.1.1.2 (B1) and 10.9.1.2 (WAN2). The metric for each route,
! the first number in parenthesis, shows that the lower metric route is the one
! through 10.1.1.2 as next-hop. Also note that the metric components
! match Figure 3-5.
!
WAN1#show ip eigrp topo 10.11.1.0/24
IP-EIGRP (AS 1): Topology entry for 10.11.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  10.1.1.2 (Serial0/0/0.1), from 10.1.1.2, Send flag is 0x0
       Composite metric is (2172416/28160), Route is Internal
       Vector metric:
         Minimum bandwidth is 1544 Kbit
         Total delay is 20100 microseconds
         Reliability is 255/255
         Load is 1/255
         Minimum MTU is 1500
         Hop count is 1
  10.9.1.2 (FastEthernet0/0), from 10.9.1.2, Send flag is 0x0
       Composite metric is (2174976/2172416), Route is Internal
       Vector metric:
         Minimum bandwidth is 1544 Kbit
         Total delay is 20200 microseconds
         Reliability is 255/255
         Load is 1/255
         Minimum MTU is 1500
         Hop count is 2
!
! The next command not only lists the IP routing table entry for 10.11.1.0/24,
! it also lists the metric (FD), and components of the metric.
!
WAN1#show ip route 10.11.1.0
Routing entry for 10.11.1.0/24
```

```
  Known via "eigrp 1", distance 90, metric 2172416, type internal
  Redistributing via eigrp 1
  Last update from 10.1.1.2 on Serial0/0/0.1, 00:02:42 ago
  Routing Descriptor Blocks:
  * 10.1.1.2, from 10.1.1.2, 00:02:42 ago, via Serial0/0/0.1
      Route metric is 2172416, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
!
! Below, the route for 10.11.1.0/24 is again listed, with the metric (FD), and
! the same next-hop and outgoing interface information.
!
WAN1#show ip route eigrp
     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D       10.11.1.0/24 [90/2172416] via 10.1.1.2, 00:10:40, Serial0/0/0.1
D        10.12.1.0/24 [90/2172416] via 10.1.1.6, 00:10:40, Serial0/0/0.2
D        10.1.2.0/30 [90/2172416] via 10.9.1.2, 00:10:40, FastEthernet0/0
D       10.1.2.4/30 [90/2172416] via 10.9.1.2, 00:10:40, FastEthernet0/0
```

EIGRP Metric Tuning

EIGRP metrics can be changed using several methods: setting interface bandwidth, setting interface delay, changing the metric calculation formula by configuring k-values, and even by adding to the calculated metric using offset-lists. In practice, the most reasonable and commonly used methods are to set the interface delay and the interface bandwidth. This section examines all the methods, in part so you will know which useful tools exist, and in part to make you aware of some other design issues that then might impact the routes chosen by EIGRP.

Configuring Bandwidth and Delay

The **bandwidth** and **delay** interface subcommands set the bandwidth and delay associated with the interface. The commands themselves require little thought, other than keeping the units straight. The unit for the **bandwidth** command is Kilobits/second, and the **delay** command uses a unit of tens-of-microseconds.

If a design requires that you influence the choice of route by changing bandwidth or delay, setting the delay value is typically the better choice. IOS uses the bandwidth setting of an interface for many other reasons: calculating interface utilization, as the basis for several QoS parameters, and for SNMP statistics reporting. However, the delay setting has little influence on other IOS features besides EIGRP, so the better choice when influencing EIGRP metrics is to tune the delay.

Table 3-2 lists some of the common default values for both bandwidth and delay. As a reminder, **show** commands list the bandwidth in Kbps, which matches the **bandwidth** command, but lists the delay in microseconds, which does not match the tens-of-microseconds unit of the **delay** command.

Table 3-2 *Common Defaults for Bandwidth and Delay*

Interface Type	Bandwidth (Kbps)	Delay (Microseconds)
Serial	1544	2000
GigE	1,000,000	10
FastE	100,000	100
Ethernet	10,000	1000

Note that on LAN interfaces that can run at different speeds, the bandwidth and delay settings default based on the current actual speed of the interface.

Choosing Bandwidth Settings on WAN Subinterfaces

Frame Relay and Metro Ethernet installations often use an access link with a particular physical sending rate–clock rate if you will–but with the contracted speed, over time, being more or less than the speed of the link. For example, with Frame Relay, the provider may supply a full T1 access link, so configuring **bandwidth 1544** for such an interface is reasonable. However, the subinterfaces have one or more PVCs associated with them, and those PVCs each have Committed Information Rates (CIR) that are typically less than the access link's clock speed. However, the cumulative CIRs for all PVC often exceeds the clock rate of the physical interface. Conversely, MetroE designs use Ethernet access links of 10 Mbps, 100 Mbps, or 1 Gbps actual link speed, but often the business contract limits the amount of traffic to some number below that link speed.

Choosing a useful interface **bandwidth** setting on the subinterfaces in a Frame Relay or MetroE design requires some thought, with most of the motivations for choosing one number or another being unrelated to EIGRP. For example, imagine the network shown in Figure 3-6. Router WAN1 has a single T1 (1.544 Mbps) access link. That interface has one multipoint subinterface, with three PVCs assigned to it. It also has nine other point-to-point subinterfaces, each with a single PVC assigned.

For the sake of discussion, the design in Figure 3-6 oversubscribes the T1 access link off Router WAN1 by a 2:1 factor. Assume all 12 PVCs have a CIR of 256 Kbps, making the total bandwidth for the 12 PVCs roughly 3 Mbps. The design choice to oversubscribe the access link may be reasonable given the statistical chance of all sites sending at the same time.

Now imagine that Router WAN1 has been configured with subinterfaces as shown in the figure:

- S0/0/0.20 - multipoint, 3 PVCs
- S0/0/0.21 through S0/0/0.29 – point-to-point, 1 PVC each

Next, consider the options for setting the **bandwidth** command's value on these 10 subinterfaces. The point-to-point subinterfaces could be set to match the CIR of each PVC (256 Kbps in this example). You could choose to set the bandwidth based on the CIR of all combined PVCs on the multipoint subinterface–in this case, setting **bandwidth 768** on multipoint subinterface s0/0/0.20. However, these bandwidths would total about 3

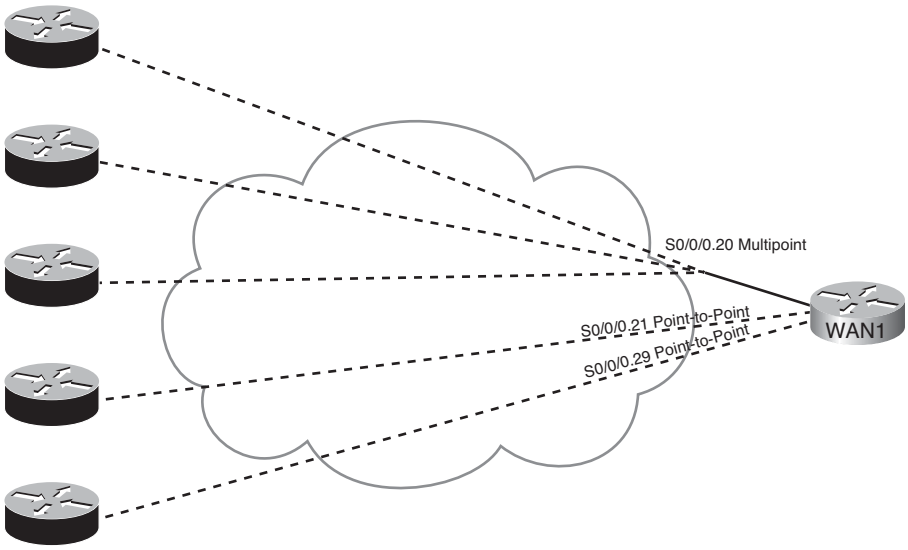


Figure 3-6 *One Multipoint and Nine Point-to-Point Subinterfaces*

Mbps–twice the actual speed of WAN1's access link. Alternatively, you could set the various bandwidths so that the total matches the 1.5 Mbps of the access link. Or you could split the difference, knowing that during times of low activity to most sites that the sites with active traffic get more than their CIR's worth of capacity anyway.

As mentioned earlier, these bandwidth settings impact much more than EIGRP. The settings impact interface statistics, both in **show** commands and in SNMP reporting. They impact QoS features to some extent as well. Given that the better option for setting EIGRP metrics is to set the interface delay, EIGRP metric tuning may not be the driving force behind the decision as to what bandwidth values to use. However, some installations may change these values over time while trying to find the right compromise numbers for features other than EIGRP. So, you need to be aware that changing those values may result in different EIGRP metrics and impact the choices of best routes.

Similar issues exist on the more modern Layer 2 WAN services like MetroE, particularly with the multipoint design of VPLS. Figure 3-7 shows a design that might exist after migrating the internetwork of Figure 3-6 to VPLS. Router WAN1 has been replaced by a Layer 3 switch, using a Gigabit interface to connect to the VPLS service. The remote sites might use the same routers as before, using a FastEthernet interface, or might be replaced with Layer 3 switch hardware as well.

Concentrating on the mechanics of what happens at the central site, WAN1 might use 802.1Q trunking. With 12 remote sites, WAN1 configures 12 VLAN interfaces, one per VLAN, with a different subnet used for the connection to each remote branch. Such a design, from a Layer 3 perspective, looks like the age-old Frame Relay design with a point-to-point link from the main site to each remote branch.

Additionally, the VPLS business contract might specify that WAN1 cannot send more than 200 Mbps of traffic into the VPLS cloud, with the excess being discarded by the

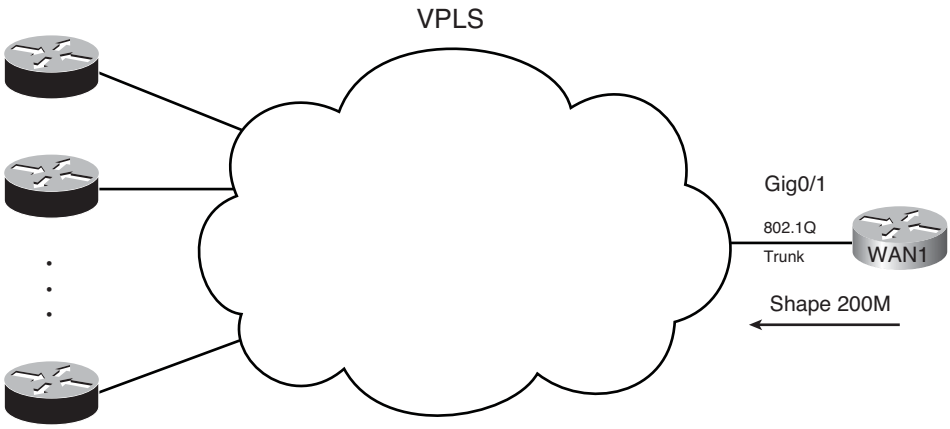


Figure 3-7 *VPLS Service–Issues in Choosing Bandwidth*

VPLS service. To prevent unnecessary discards, the engineer likely configures a feature called *shaping*, which slows down the traffic leaving the Gi0/1 interface of WAN1 (regardless of VLAN). To meet the goal of 200 Mbps, WAN1 would send only part of the time–in this case averaging a sending rate of 1/5th of the time–so that the average rate is 1/5th of 1 Gbps, or 200 Mbps.

Of note with the shaping function, the shaping feature typically limits the cumulative traffic on the interface, not per VLAN (branch). As a result, if the only traffic waiting to be sent by WAN1 happens to be destined for branch B1, WAN1 sends 200 MBps of traffic to just branch B1.

Pulling the discussion back around to EIGRP, as with Frame Relay, other design and implementation needs may drive the decision to set or change the bandwidth on the associated interfaces. In this case, Layer 3 switch WAN1 probably has 12 VLAN interfaces. Each VLAN interface can be set with a bandwidth that influences EIGRP route choices. Should this setting be 1/12th of 1 Gbps, which is the speed at which the bits are actually sent? 1/12th of 200 Mbps, the shaping rate? Or knowing that a site might get most or all of that 200 Mbps for some short period of time, should the bandwidth be set somewhere in between? As with Frame Relay, there is no set answer; for the sake of EIGRP, be aware that changes to the bandwidth settings impact the EIGRP metrics.

Metric Weights (K-values)

Engineers can change the EIGRP metric calculation by configuring the weightings (also called k-values) applied to the EIGRP metric calculation. To configure new values, use the **metric weights** *tos k1 k2 k3 k4 k5* command in EIGRP configuration mode. To configure this command, configure any integer 0–255 inclusive for the five k-values. By default, k1 = k3 = 1, and the others default to 0. The **tos** parameter has only one valid value, 0, and can be otherwise ignored.

The full EIGRP metric formula is as follows. Note that some items reduce to 0 if the corresponding k-values are also set to 0.

EIGRP requires that two routers' k-values match before those routers can become neighbors. Also note that Cisco recommends again using k-values k2, k4, and k5, because a nonzero value for these parameters causes the metric calculation to include interface load and reliability. The load and reliability change over time, which causes EIGRP to reflood topology data, and may cause routers to continually choose different routes (route flapping).

Offset Lists

EIGRP Offset Lists, the final tool for manipulating the EIGRP metrics listed in this chapter, allow an engineer to simply add a value–an offset, if you will-to the calculated integer metric for a given prefix. To do so, an engineer can create and enable an EIGRP Offset List that defines the value to add to the metric, plus some rules regarding which routes should be matched and therefore have the value added to their computed FD.

An Offset List can perform the following functions:

- Match prefixes/prefix lengths using an IP ACL, so that the offset is applied only to routes matched by the ACL with a permit clause
- Match the direction of the Update message, either sent (out) or received (in)
- Match int interface on which the Update is sent or received
- Set the integer metric added to the calculation for both the FD and RD calculations for the route

The configuration itself uses the following command in EIGRP configuration mode, in addition to any referenced IP ACLs:

```
offset-list {access-list-number | access-list-name} {in | out} offset [interface-
type interface-number]
```

For example, consider again branch office Router B1 in Figure 3-1, with its connection to both WAN1 and WAN2 over a Frame Relay network. Formerly, WAN1 calculated a metric of 2,172,416 for its route, through B1, to subnet 10.11.1.0/24. (Refer to Figure 3-5 for the math behind WAN1's calculation of its route to 10.11.1.0/24.) Router WAN1 also calculated a value of 28,160 for the RD of that same direct route. Example 3-5 shows the addition of an offset on WAN1, for received updates from Router B1.

Example 3-5 *Inbound Offset of 3 on WAN1, for Updates Received on S0/0/0.1*

```
WAN1(config)#access-list 11 permit 10.11.1.0
WAN1(config)#router eigrp 1
WAN1(config-router)#offset-list 11 in 3 Serial0/0/0.1
WAN1(config-router)#end

Mar  2 11:34:36.667: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.2
(Serial0/0/0.1) is resync: peer graceful-restart
WAN1#show ip eigrp topo 10.11.1.0/24
IP-EIGRP (AS 1): Topology entry for 10.11.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
```

```
  Routing Descriptor Blocks:
  10.1.1.2 (Serial0/0/0.1), from 10.1.1.2, Send flag is 0x0
      Composite metric is (2172419/28163), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
! output omitted for brevity
```

The configuration has two key elements: ACL 11 and the **offset-list** command. ACL 11 matches prefix 10.11.1.0, and that prefix only, with a permit clause. The **offset-list 11 in 3 s0/0/0.1** command tells Router WAN1 to examine all EIGRP Updates received on S0/0/0.1, and if prefix 10.11.1.0 is found, add 3 to the computed FD and RD for that prefix.

Note: Standard ACL 11 matches prefix 10.11.1.0 in this case, regardless of prefix length. To match the exact prefix and prefix length, use an extended ACL. When doing so, use the destination address field to match the prefix length. For example, **access-list 111 permit ip host 10.11.1.0 host 255.255.255.0** matches 10.11.1.0/24 exactly, including the prefix length.

The **show ip eigrp topology 10.11.1.0/24** command in Example 3-4 shows that the FD and RD, highlighted in parentheses, are now each three larger as compared with the earlier metrics.

Next, continuing this same example, Router B1 has now been configured to add an offset (4) in its sent updates to all routers, but for prefix 10.11.1.0/24 only.

Example 3-6 *Outbound Offset of 4 on B1, for Updates Sent to All Neighbors, 10.11.1.0/24*

```
B1(config)#access-list 12 permit 10.11.1.0
B1(config)#router eigrp 1
B1(config-router)#offset-list 12 out 4
B1(config-router)#end
B1#
```

```
! Back to router WAN1
WAN1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.9.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
```

```
P 10.11.1.0/24, 1 successors, FD is 2172419
         via 10.1.1.2 (2172423/28167), Serial0/0/0.1
! lines omitted for brevity
```

Note that the metrics, both FD and RD, are now four larger than in Example 3-4.

Optimizing EIGRP Convergence

The previous major section of this chapter focused on how EIGRP calculates metrics and how to change that metric calculation. However, that section discussed only one motivation for changing the metric: to make a router pick one route instead of another. This section, which focuses on optimizing the EIGRP convergence process, discusses another reason for choosing to manipulate the EIGRP metric calculations: faster convergence.

EIGRP converges very quickly, but EIGRP does not achieve the most optimal fast convergence times in all conditions. One design goal might be to tune EIGRP configuration settings so that EIGRP uses the faster convergence methods for as many routes as possible, and when not possible, that EIGRP converge as quickly as it can without introducing routing loops. As a result, routers might converge in some cases in a second instead of tens of seconds (from the point of a router realizing that a route has failed).

For those of you who have not thought about EIGRP convergence before now, you must first get a comfortable understanding of the concept of EIGRP Feasible Successors–the first topic in this section. Following that, the text examines the EIGRP query process. This section ends with EIGRP load balancing, which both allows spreading the load across multiple routes in addition to improving EIGRP convergence.

Fast Convergence to Feasible Successors

Earlier in this chapter, under the heading "Calculating the Metrics: Feasible Distance and Reported Distance," the text explains how a router, for each possible route, calculates two metric values. One value is the *feasible distance* (FD), which is the metric from that router's perspective. The other metic is the *reported distance* (RD), which is the integer metric from the perspective of the next-hop router.

EIGRP routers use the RD value when determining if a possible route can be considered to be a loop-free backup route called a *feasible successor*. This section explains the concepts and shows how to confirm the existence or nonexistence of such routes.

Successor and Feasible Successor Concepts

For each prefix/prefix length, when multiple possible routes exist, the router chooses the route with the smallest integer metric (smallest FD). EIGRP defines each such route as the *successor route* for that prefix, and EIGRP defines the next-hop router in such a route as the *successor*. EIGRP then creates an IP route for this prefix, with the successor as the next-hop router, and places that route into the IP routing table.

If more than one possible route exists for a given prefix/prefix length, the router examines these other (non-successor) routes and asks this question: Can any of these routes be used

immediately if the currently best route fails, without causing a routing loop? EIGRP runs a simple algorithm to identify which routes could be used without causing a routing loop, and EIGRP keeps these loop-free backup routes in its topology table. Then, if the successor route (the best route) fails, EIGRP then immediately uses the best of these alternate loop-free routes for that prefix.

EIGRP calls these alternative, immediately usable, loop-free routes *feasible* successor routes, because they can feasibly be used as a new successor route when the current successor route fails. The next-hop router of such a route is called the *feasible successor*.

Note: In general conversation, the term *successor* may refer to the route or specifically to the next-hop router. Likewise, the term *feasible successor* may refer to the route, or the next-hop router, of an alternative route.

A router determines if a route is a feasible successor based on the *feasibility condition*, defined as follows:

If a non-successor route's RD is less than the FD, the route is a feasible successor route.

Although technically correct, the preceding definition is much more understandable with an example as shown in Figure 3-8. The figure illustrates how EIGRP figures out which routes are feasible successors for Subnet 1.

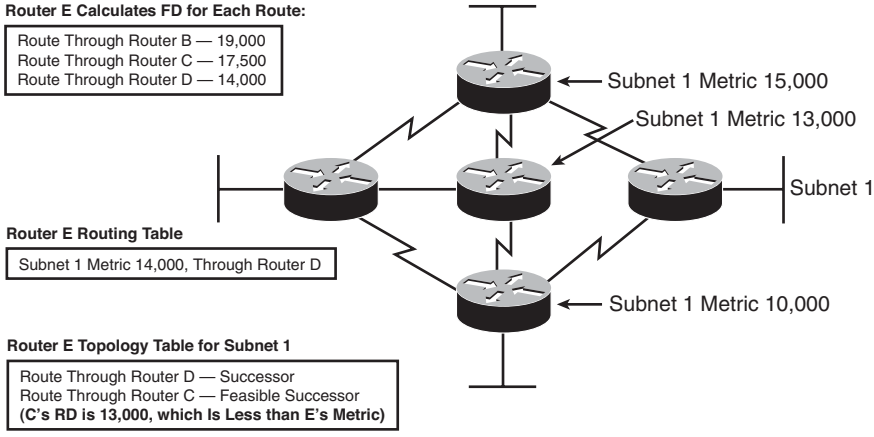


Figure 3-8 *Successors and Feasible Successors with EIGRP*

In Figure 3-8, Router E learns three routes to Subnet 1, from Routers B, C, and D. After calculating each route's metric, Router E finds that the route through Router D has the lowest metric. Router E adds this successor route for Subnet 1 to its routing table, as shown. The FD in this case for this successor route is 14,000.

EIGRP decides if a route can be a feasible successor if the reported distance for that route (the metric as calculated on that neighbor) is less than its own best computed metric (the FD). When that neighbor has a lower metric for its route to the subnet in question, that route is said to have met the *feasibility condition*.

For example, Router E computes a metric (FD) of 14,000 on its successor route (through Router D). Router C's computed metric–E's RD for this alternate router through Router C–is 13,000, which is lower than E's FD (14,000). As a result, E knows that C's best route for this subnet could not possible point toward router E, so Router E believes that its route, to Subnet 1, through Router C, would not cause a loop. As a result, Router E marks its topology table entry for the route through Router C as a feasible successor route.

Conversely, E's RD for the route through Router B, to Subnet 1, is 15,000, which is larger than Router E's FD of 14,000. So, this alternative route does not meet the feasibility condition, so Router E does not consider the route through Router B a feasible successor route.

If the route to Subnet 1 through Router D fails, Router E can immediately put the route through Router C into the routing table without fear of creating a loop. Convergence occurs almost instantly in this case. However, if both C and D fail, E would not have a feasible successor route, and would have to do additional work, as described later in the section "Converging by Going Active," before using the route through Router B.

By tuning EIGRP metrics, an engineer can create feasible successor routes in cases where none existed, improving convergence.

Verification of Feasible Successors

Determining which prefixes have both successor and feasible successor routes is somewhat simple if you keep the following in mind:

- The **show ip eigrp topology** command does not list all known EIGRP routes, but instead lists only successor and feasible successor routes.
- The **show ip eigrp topology all-links** command lists all possible routes, including those that are neither successor nor feasible successor routes.

For example, consider Figure 3-9, which again focuses on Router WAN1's route to Router B1's LAN subnet, 10.11.1.0/24. The configuration on all routers has reverted back to defaults for all settings that impact the metric: default bandwidth and delay, no offset lists, and all interfaces are up.

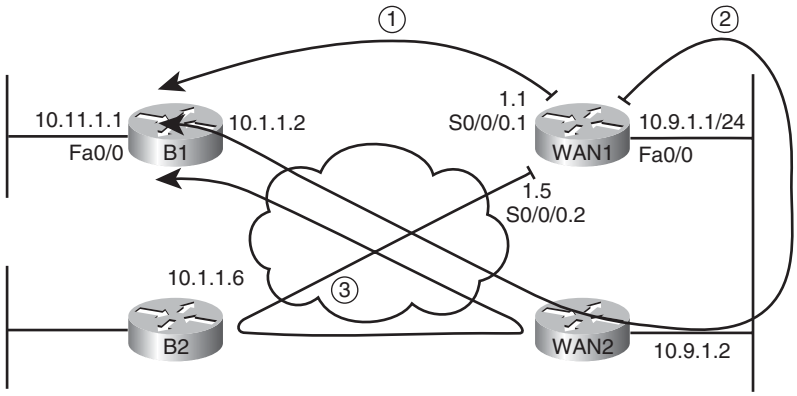
Figure 3-9 shows the three topologically possible routes to reach 10.11.1.0/24, labeled 1, 2, and 3. Route 1, direct to Router B1, is the current successor. Route 3, which goes to another branch router, back to the main site, and then to Router B1, is probably a route you would not want to use anyway. However, route 2, through WAN2, would be a reasonable back-up route.

If the PVC between WAN1 and B1 failed, WAN1 would converge to route 2 from the figure. However, with all default settings, route 2 is not an FS route, as demonstrated in Example 3-7.

Example 3-7 *Only a Successor Route on WAN1 for 10.11.1.0/24*

```
WAN1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.9.1.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.11.1.0/24, 1 successors, FD is 2172416
         via 10.1.1.2 (2172416/28160), Serial0/0/0.1
! lines omitted for brevity; no other lines of output pertain to 10.11.1.0/24.

WAN1#show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(1)/ID(10.9.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.11.1.0/24, 1 successors, FD is 2172416, serno 45
         via 10.1.1.2 (2172416/28160), Serial0/0/0.1
         via 10.9.1.2 (2174976/2172416), FastEthernet0/0
! lines omitted for brevity; no other lines of output pertain to 10.11.1.0/24.
```



Note: All WAN IP addresses begin with 10.1

Figure 3-9 *Three Possible Routes from WAN1 to 10.11.1.0/24*

A quick comparison of the two commands show that the **show ip eigrp topology** command shows only one next-hop address (10.1.1.2), whereas the **show ip eigrp topology all-links** command shows two (10.1.1.2 and 10.9.1.2). The first command lists only successor and feasible successor routes, so in this case, only one such route for 10.1.1.0/24 exists—the successor route, direct to B1 (10.1.1.2).

The output of the **show ip eigrp topology all-links** command is particularly interesting in this case. It lists two possible next-hop routers: 10.1.1.2 (B1) and 10.9.1.2 (WAN2). It does

not list the route through Router B2 (10.1.1.6) at all, because B2's current successor route for 10.11.1.0/24 is through WAN1. EIGRP Split Horizon rules tell B2 to not advertise 10.11.1.0/24 to WAN1.

Next, focus on the route labeled as option 2 in Figure 3-9, the route from WAN1, to WAN2, then to B1. Per the **show ip eigrp topology all-links** command, this route has an RD of 2,172,416–the second number in parenthesis as highlighted toward the end of Example 3-6. WAN1's successor route has an FD of that exact same value. So, this one possible alternate route for 10.11.1.0/24, through WAN2, does not meet the feasibility condition–but just barely. To be an FS route, the route's RD must be less than the FD, and in this example, the two are equal.

To meet the design requirement for quickest convergence, you could use any method to manipulate the metrics such that either WAN2's metric for 10.11.1.0 is lower, or WAN1's metric for its successor route is higher. Example 3-8 shows the results of simply adding back the offset-list on WAN1, as seen in Example 3-5, which increases WAN1's metric by 3.

Example 3-8 *Increasing WAN1's Metric for 10.11.1.0/24, Creating an FS Route*

```
WAN1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
WAN1(config)#access-list 11 permit 10.11.1.0
WAN1(config)#router eigrp 1
WAN1(config-router)#offset-list 11 in 3 s0/0/0.1
WAN1(config-router)#^Z
WAN1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.9.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.11.1.0/24, 1 successors, FD is 2172419
         via 10.1.1.2 (2172419/28163), Serial0/0/0.1
         via 10.9.1.2 (2174976/2172416), FastEthernet0/0
! lines omitted for brevity; no other lines of output pertain to 10.11.1.0/24.
```

Note that now WAN1's successor route FD is 2,172,419, which is higher than WAN2's (10.9.1.2's) RD of 2,172,416. As a result, WAN1's route through WAN2 (10.9.1.2) now meets the feasibility condition. Also, the **show ip eigrp topology** command, which lists only successor and feasible successor routes, now lists this new feasible successor route. Also note that the output still states "1 successor," so this counter indeed counts successor routes and does not include FS routes.

When EIGRP on a router notices that a successor route has been lost, if a feasible successor exists in the EIGRP topology database, EIGRP places that feasible successor route into the routing table. The elapsed time from noticing that the route failed, until the route is replaced, is typically less than 1 second. (A Cisco Live conference presentation asserts this

convergence approaches 200 milliseconds.) With well-tuned EIGRP Hold Timers and with feasible successor routes, convergence time can be held low.

Converging by Going Active

When EIGRP removes a successor route and no FS route exists, the router begins a process by which the router discovers if any loop-free alternative routes each reach that prefix. This process is called *going active* on a route. Routes for which the router has a successor route, and no failure has yet occurred, remain in a passive state. Routes for which the successor route fails, with no feasible successor routes, move to an active state, as follows:

- Change the state, as listed in the **show ip eigrp topology** command, from passive (p) to active (a).
- Send EIGRP *Query* messages to every neighbor except the neighbor in the failed route. The Query asks a neighbor whether that neighbor has a loop-free route for the listed prefix/length.
- The neighbor considers itself to have a loop-free route if that neighbor is passive for that prefix/length. If so, the neighbor 1) sends an EIGRP Reply message, telling the original router that it does indeed have a loop-free route and 2) does not forward the Query.
- If the neighbor itself is active on this route, that neighbor 1) floods EIGRP Query messages to its neighbors and 2) does not immediately send an EIGRP Reply back to the original router–instead waiting on replies to its own set of Query messages.
- When a router has received Reply messages from all neighbors to which it sent any Query messages, that router can then send a Reply message to any of its neighbors as necessary.
- When a router has received a Reply for all its Query messages, that router may safely use the best of the routes confirmed to be loop free.

Note: The EIGRP convergence process when going active on a route is sometimes also referenced by the name of the underlying algorithm, named Diffusing Update Algorithm (DUAL).

The process can and does work well in many cases, often converging to a new route in less than 10 seconds. However, in internetworks with many remote sites, with much redundancy, and with a large number of routers in a single end-to-end route, convergence when going active can be inefficient. For example, consider the internetwork in Figure 3-10. The figure shows five branch routers as an example, but the internetwork has 300 branch routers, each with a PVC connected to two WAN routers, WAN1 and WAN2. When Router WAN1 loses its route for the LAN subnet at branch B1, without an FS route, the Query process can get out of hand.

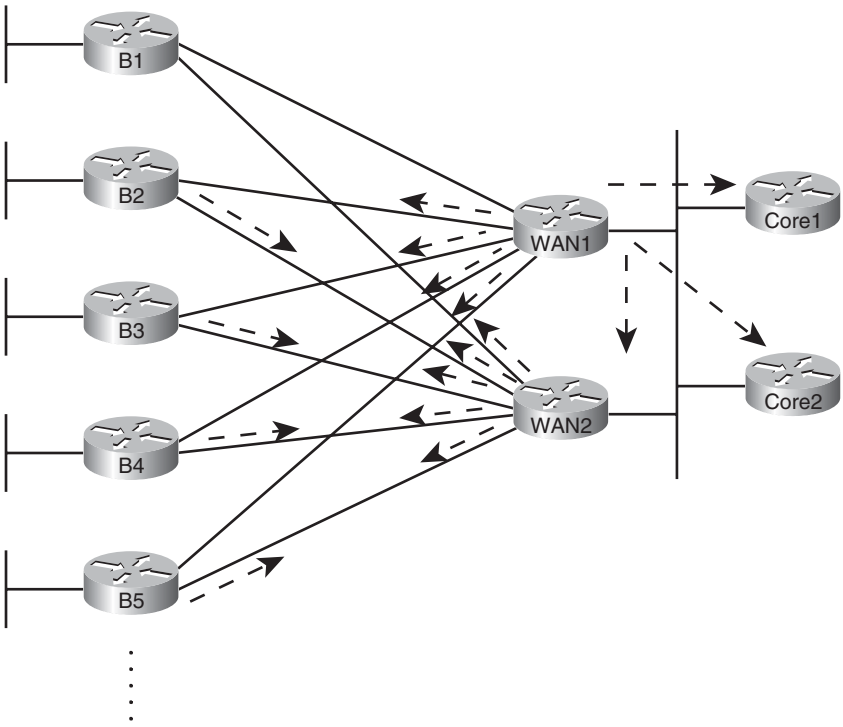


Figure 3-10 *Issues with Query Scope*

The arrowed lines show WAN1's Query messages and the reaction by several other routers to forward the Query messages. Although only 5 branch routers are shown, WAN1 would forward Query messages to 299 branch routers. WAN2 would do the same, assuming its route to B1's LAN also failed. These branch routers would then send Query messages back to the WAN routers. The network would converge, but more slowly than if an FS route existed.

Note: EIGRP sends every Query and Reply message using RTP, so every message is acknowledged using an EIGRP ACK message.

By configuring EIGRP so that a router has FS routes for most routes, the whole Query process can be avoided. However, in some cases, creating FS routes for all routes on all routers is impossible, so engineers should take action to limit the scope of queries. The next two sections discuss two tools—stub routers and route summarization—that help reduce the work performed by the DUAL algorithm and the scope of Query messages.

The Impact of Stub Routers on Query Scope

Some routers, by design, should not be responsible for forwarding traffic between different sites. For example, consider the familiar internetwork shown throughout this chapter,

most recently in Figure 3-10, and focus on the branch routers. If WAN2's LAN interface failed, and WAN1's PVC to B1 failed, then a route still exists from the core to branch B1's 10.11.1.0/24 subnet: WAN1–B2–WAN2–B1. (This is the same long route shown as route 3 in Figure 3-9.) However, this long route consumes the link bandwidth between the core and branch B2, and the traffic to/from B1 will be slower. Users at both branches will suffer, and these conditions may well be worse than just not using this long route.

Route filtering could be used to prevent WAN1 from learning such a route. However, using route filtering would require a lot of configuration on all the branch routers, with specifics for the subnets–and it would have to change over time. A better solution exists, which is to make the branch routers stub routers. EIGRP defines *stub routers* as follows:

A router that should not forward traffic between two remote EIGRP-learned subnets.

To accomplish this goal, the engineer configures the stub routers using the **eigrp stub** command. Stub routers do not advertise EIGRP-learned routes from one neighbor to other EIGRP neighbors. Additionally, and possibly more significantly, nonstub routers note which EIGRP neighbors are stub routers, and the nonstub routers do not send Query messages to the stub routers. This action greatly reduces the scope of Query messages when a route goes active, in addition to preventing the long, circuitous, and possibly harmful route.

The **eigrp stub** command has several options. When issued simply as **eigrp stub**, the router uses default parameters, which are the **connected** and **summary** options. (Note that IOS adds these two parameters onto the command as added to the running-config.) Table 3-3 lists the **eigrp stub** command options and explains some of the logic behind using them.

Table 3-3 *Parameters on the eigrp stub Command*

Option	This router is allowed to...
Connected	Advertise connected routes but only for interfaces matched with a **network** command.
Summary	Advertise auto-summarized or statically configured summary routes.
Static	Advertises static routes, assuming the **redistribute static** command is configured.
Redistributed	Advertises redistributed routes, assuming redistribution is configured.
Receive-only	Does not advertise any routes. This option cannot be used with any other option.

Note that stub routers still form neighborships, even in receive-only mode. The stub router simply performs less work and reduces the Query Scope because neighbors will not send these routes any Query messages.

For example, Example 3-9 shows the **eigrp stub connected** command on Router B2, with the results being noticable on WAN1 (**show ip eigrp neighbors detail**).

Example 3-9 *Evidence of Router B2 as an EIGRP Stub Router*

```
B2#configure terminal
B2(config)#router eigrp 1
B2(config-router)#eigrp stub connected
B2(config-router)#
Mar  2 21:21:52.361: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.9.1.14
(FastEthernet0/0.12) is down: peer info changed
! A message like the above occurs for each neighbor.
```

```
! Moving to router WAN1 next
WAN1#show ip eigrp neighbors detail
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.9.1.2                Fa0/0            11 00:00:04    7   200  0  588
   Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 8
2   10.1.1.6                Se0/0/0.2        13 00:21:23    1   200  0  408
   Version 12.4/1.2, Retrans: 2, Retries: 0, Prefixes: 2
   Stub Peer Advertising ( CONNECTED ) Routes
   Suppressing queries
0   10.9.1.6                Fa0/0.4          12 00:21:28    1   200  0  175
   Version 12.2/1.2, Retrans: 3, Retries: 0, Prefixes: 6
```

The Impact of Summary Routes on Query Scope

In addition to EIGRP stub routers, route summarization also limits EIGRP Query scope and therefore improves convergence time. The reduction in Query scope occurs due to the following rule:

Key Topic

If a router receives an EIGRP Query for a prefix/prefix length, does not have an exactly matching (both prefix and prefix length) route, but does have a summary route that includes the prefix/prefix length, that router immediately sends an EIGRP Reply and does not flood the Query to its own neighbors.

For example, consider Figure 3-11. Multilayer Switches C1 and C2 sit in the core of the network shown in various other figures in this chapter, and both C1 and C2 run EIGRP. The IP subnetting design assigns all branch office LAN subnets from the range 10.11.0.0/16 and 10.12.0.0/16. As such, Routers WAN1 and WAN2 advertise summary routes for these ranges, rather than for individual subnets. So, under normal operation, ignoring the whole Query scope issue, C1 and C2 would never have routes for individual branch subnets like 10.11.1.0/24 but would have routes for 10.11.0.0/16 and 10.12.0.0/16.

The figure highlights three steps:

Step 1. WAN1 and WAN2 advertise summary routes, so that C1, C2, and all other routers in the core have a route for 10.11.0.0/16 but not a route for 10.11.1.0/24.

Step 2. Some time in the future, WAN1 loses its route for 10.11.1.0/24, so WAN1 sends a Query for 10.11.1.0/24 to C1 and C2.

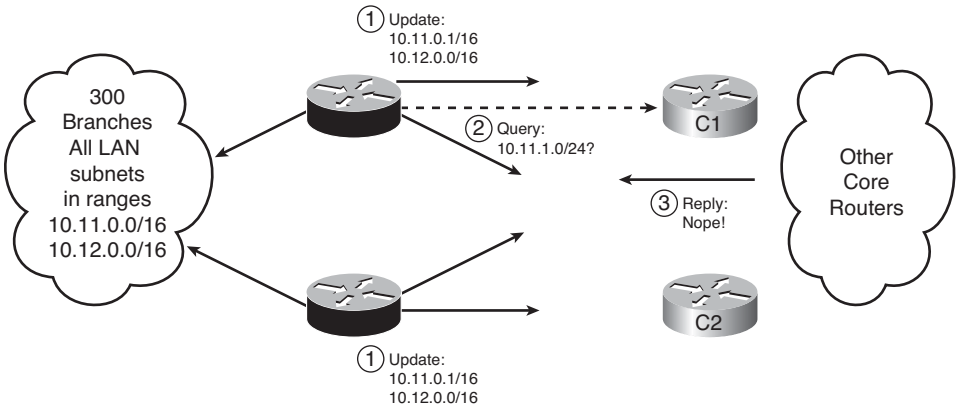


Figure 3-11 *Route Summaries Limiting Query Scope*

Step 3. C1 and C2 send an EIGRP Reply immediately afterward, because both do not have a route for that specific prefix/length (10.11.1.0/24), but both do have a summary route (10.11.0.0/16) that includes that range of addresses.

Chapter 4, "EIGRP Route Summarization and Filtering," explains the configuration of EIGRP route summarization as an end to itself.

Stuck in Active

When a router notices a route failure and moves a route from passive to active state, that router sends Query messages to its neighbors. With a sufficiently large network, particularly when routers exist several router hops away, the number of Queries may not only be large, but there also may be a string of routers that all must wait on multiple Reply messages before they can, in turn, issue a Reply. For example, in Figure 3-12, Router R1 must wait on routers R11, R12, and R13 to send a Reply. R11 must wait on routers R21, R22, and R23. R21 must wait on three other routers, and so on–meaning that R1 may have to wait quite a while before getting a response.

Although the design shown in Figure 3-12 is admittedly contrived, the point is that a router may wait awhile before getting a Reply message in response to each Query message for an Active route. A router cannot use any alternative paths for that route until all such Reply messages have been received.

To deal with this potentially long time, IOS first sets a limit on how long it should take to receive all such replies. That timer, called the *active timer*, is set to 3 minutes by default. (The timer can be configured for an entire EIGRP process using the **timers active-time** *time* EIGRP subcommand, with a units of a number of minutes.) Routes for which a router does not receive a Reply within the active timer are considered to be Stuck-in-Active (SIA) routes.

IOS has two major branches of logic when reacting to SIA routes. Earlier versions of IOS took a rather drastic action, bringing down the uncooperative neighbors that had yet to send back an EIGRP Reply for that route. For example, in Figure 3-12, if R1 received Reply messages from R11 and R12, but not R13, and the active timer expired, R1 would bring

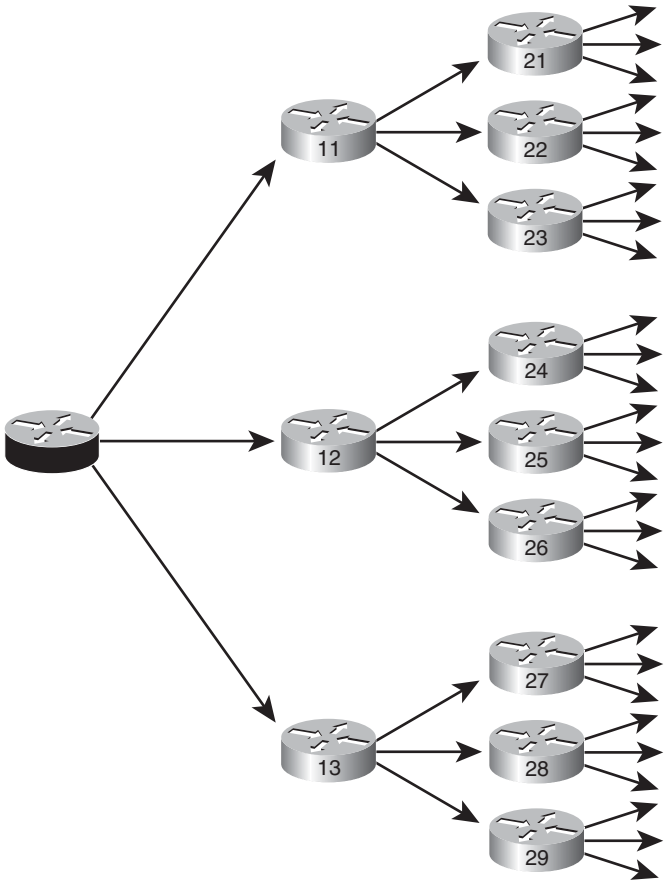


Figure 3-12 *Network Design That Causes Unreasonably Long Convergence*

down the neighborship with R13. The active route would be considered to have failed, and all routes known through the failed neighbor would also be considered to have failed–possibly generating more Query messages for other routes.

Later IOS versions (beginning in the 12.2 mainline) make an attempt to avoid failing the neighborship. At the halfway point through the Active timer–a seemingly long 90 seconds by default–a router sends an SIA-Query (Stuck-in-Active query) EIGRP message to each neighbor that has yet to send back a Reply. The purpose of the message is to either get an SIA-Reply back, meaning that the neighbor really is still waiting for replies to its own queries, or to get nothing in reply. In the first case, because the neighbor is alive and still working, there is no need to kill the neighborship. In the second case, the neighbor was not able to reply, so the action of failing the neighborship is reasonable.

Unequal Metric Route Load Sharing

Convergence to a feasible successor route should happen within a second after a router realizes the successor route has failed. Even in large well-designed networks, particularly

with features like stub routers and route summarization in use, convergence can still happen in a reasonable amount of time even when going active. The next feature, load sharing, takes convergence to another level, giving instantaneous convergence, while reaching other goals as well.

IOS allows routing protocols to place multiple routes into the routing table for an individual prefix/length. IOS then balances traffic across those routes, by default balancing traffic on a per-destination IP address basis.

Load balancing, sometimes called load sharing, provides a primary benefit of making use of the available bandwidth, rather than using some links as simply backup links. For example, with the two-PVC designs repeatedly shown in this chapter (Figures 3-1, 3-9, and 3-10), without load sharing, a branch router would send traffic over one PVC, but not both. With load sharing, some traffic would flow over each PVC.

A useful secondary benefit–faster convergence–occurs when using load balancing. By placing multiple routes into the routing table for a single prefix, convergence happens essentially instantly. For example, if a branch router has two routes for each data center subnet—one using each PVC that connects the branch to the core—and one of the routes fails, the other route is already in the routing table. In this case, the router does not need to look for FS routes nor go active on the route. The router uses the usual EIGRP convergence tools only when all such routes are removed from the routing table.

The load balancing configuration requires two commands, one of which already defaults to a reasonable setting. First, you need to define the number of allowed routes for each prefix/prefix length using the **maximum-paths** *number* EIGRP subcommand. The default setting of 4 is often big enough, because most internetworks do not have enough redundancy to have more than four possible routes.

Note: The maximum number of paths varies based on IOS version and router platform. However, for the much older IOS versions, the maximum was 6 routes, with later versions typically supporting 16 or more.

The second part of the load balancing configuration overcomes a challenge introduced by EIGRP's metric calculation. The EIGRP integer metric calculation often results in 8-to-10-digit integer metrics, so the metrics of competing routes are seldom the exact same value. Calculating the exact same metric for different routes for the same prefix is statistically unlikely.

IOS includes the concept of EIGRP *variance* to overcome this problem. Variance lets you tell IOS that the EIGRP metrics can be close in value and still be considered worthy of being added to the routing table—and you can define how close.

The **variance** *multiplier* EIGRP router subcommand defines an integer between 1 and 128. The router then multiplies the variance times the successor route's FD–the metric of the best route to reach that subnet. Any FS routes whose metric is less than the product of the variance times the FD are considered to be equal routes and may be placed into the routing table, up to the number of routes defined by the **maximum-paths** command.

For example, consider the example as shown in Figure 3-13 and Table 3-4. In this example, to keep the focus on the concepts, the metrics are small easy-to-compare numbers, rather than the usual large EIGRP metrics. The example focuses on R4's three possible routes to reach Subnet 1. The figure shows the RD of each route next to Routers R1, R2, and R3, respectively.

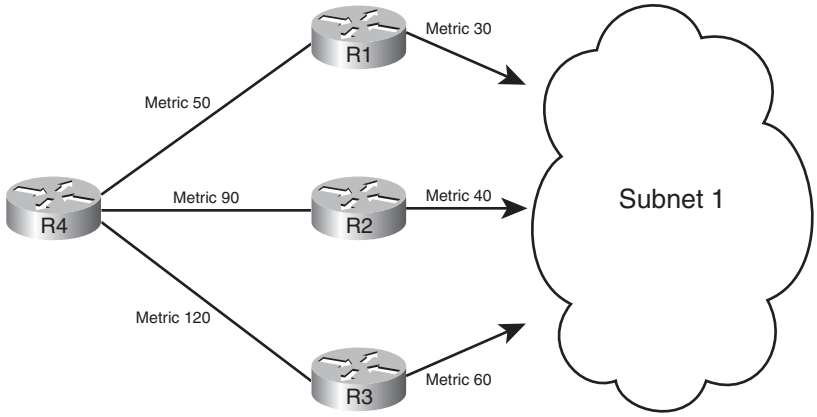


Figure 3-13 *Example of the Use of Variance*

Table 3-4 *Example of Routes Chosen as Equal Due to Variance*

Next-hop	Metric	RD	Added to Routing Table at Variance 1?	Added to Routing Table at Variance 2?	Added to Routing Table at Variance 3?
R1	50	30	Yes	Yes	Yes
R2	90	40	No	Yes	Yes
R3	120	60	No	No	No

Before considering the variance, note that in this case the route through R1 is the successor route because it has the lowest metric. This also means that the FD is 50. The route through R2 is an FS route because its RD of 40 is less than the FD of 50. The route through R3 is not an FS route, because R3's RD of 60 is more than the FD of 50.

At a default variance setting of 1, the metrics must be exactly equal to be considered equal, so only the successor route is added to the routing table (the route through R1). With variance 2, the FD (50) is multiplied by the variance (2) for a product of 100. The route through R2, with FD 90, is less than 100, so R4 will add the route through R2 to the routing table as well. The router can then load balance traffic across these two routes. Table 3-4 summarizes these cases, plus one other, which is described after the table.

In the third case, with variance 3, the product of the FD (50) times 3 results equals 150. All three routes' calculated metrics (their FD values) are less than 150. However, the route

through R3 is not an FS route, so it cannot be added to the routing table for fear of causing a routing loop. So, R4 adds only the routes through R1 and R2 to its IP routing table. (Note that the variance and maximum-paths settings can be verified by using the **show ip protocols** command.)

The following list summarizes the key points about variance:

- The variance is multiplied by the current FD (the metric of the best route to reach the subnet).
- Any FS routes whose calculated metric is less than or equal to the product of variance times FD are added to the IP routing table, assuming the **maximum-paths** setting allows more routes.
- Routes that are neither successor nor feasible successor can never be added to the IP routing table, regardless of the variance setting.

When the routes have been added to the routing table, the router supports a couple of methods for how to load balance traffic across the routes. The router can balance the traffic proportionally with the metrics, meaning that lower metric routes send more packets. Also, the router can send all traffic over the lowest-metric route, with the other routes just being in the routing table for faster convergence in case the best route fails.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 3-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 3-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Limit consumption of IP subnets in Frame Relay WAN design.	
In a relatively slow Frame Relay WAN, protect against consuming too much bandwidth with overhead EIGRP traffic.	
Plan to change bandwidth from 1X CIR to 2X CIR on all Frame Relay subinterfaces.	
Plan to set bandwidth to values other than actual interface speeds to manipulate EIGRP metrics.	
A goal of ensuring all remote routers' secondary EIGRP routes does not require Queries for convergence.	
What tools can we use to meet the design goal of fast convergence? (four items)	

Implementation Plan Peer Review Table

Table 3-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 3-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A Frame Relay multipoint interface, with 20 PVCs attached, has a configuration for 10% of the bandwidth to be used for EIGRP. How much is allocated per PVC?	
A configuration lists the **no ip split-horizon** command–when would that matter?	
The plan calls for setting all EIGRP K-values to 1. What negative effect could this have on routes in the IP routing table?	
The configuration uses offset lists. Will that impact the calculation of FD and/or RD?	
The plan lists a sample configuration migrating an interface from **delay 20** to **delay 200**. How much will the metric go up?	
The plan shows the use of the **variance 4** command. What must be configured to add other routes to a routing table? (two items)	

Create an Implementation Plan Table

To practice skills useful when creating your own EIGRP implementation plan, list in Table 3-7 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 3-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling EIGRP on interfaces	
Enabling or disabling Split Horizon for EIGRP	
Setting the Bandwidth consumed by EIGRP on an interface	
Setting an interface's logical bandwidth	
Setting an interface's logical delay	
K-values	
Configuring an EIGRP offset list that matches a prefix	
Configuring an EIGRP offset list that matches a prefix and prefix length	
Configuring unequal cost load balancing	
Configure an EIGRP stub router	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own EIGRP verification plan, list in Table 3-8 all commands that supply the requested information. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Key Topic

Table 3-8 *Verification Plan Memory Drill*

Information Needed	Command
The composite metric values for all EIGRP prefixes.	
Display EIGRP Split Horizon settings.	
Calculate the maximum bandwidth EIGRP will consume on a physical or point-to-point subinterface.	
Calculate the maximum bandwidth EIGRP will consume per PVC on a multipoint Frame Relay subinterface.	
Display the increase in RD after implementing an EIGRP offset list.	
Display interface bandwidth and delay settings.	
Lists EIGRP K-values.	
Find the number of successor and feasible successor routes.	
Find all routes, including non-successors.	
Determine if the local router is a stub router.	
Determine if a neighboring router is a stub router.	
Find the current setting of variance and maximum-paths.	
Display messages each time EIGRP suppresses a prefix advertisement due to Split Horizon.	

Key Topic

Review all the Key Topics

Review the most important topics from the chapter, noted with the key topics icon in the outer margin of the page. Table 3-9 lists a reference of these key topics and the page numbers on which each is found.

Table 3-9 *Key Topics for Chapter 3*

Key Topic Element	Description	Page Number
List	Three sources for seeding a local router's EIGRP topology table	60
List	EIGRP message types (5)	61
List	Rules for EIGRP topology exchange	64
Definitions	Feasible Distance, Reported Distance	70
Definition	Feasibility Condition	79
Figure 3-8	Conceptual view of successors and feasible successors	79
List	Two commands to find all EIGRP routes versus all successor/feasible successor routes	80
List	EIGRP process of finding routes when going active	83
Definition	EIGRP stub router	85
Table 3-3	List of options on the **eigrp stub** command	85
Definition	Rule by which summary routes reduce Query scope	86
List	Rules for unequal-cost multipath	91

Complete the Tables and Lists from Memory

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

Define Key Terms

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

feasibility condition, feasible distance, feasible successor, full update, partial update, reported distance, advertised distance, successor, Split Horizon, bandwidth, delay, K-value, offset list, going active, DUAL, Query scope, EIGRP stub router, unequal-cost load balancing, variance

This chapter covers the following subjects:

Route Filtering: This section examines how to filter prefixes from being sent in EIGRP Updates or filter them from being processed when received in an EIGRP Update.

Route Summarization: This section discusses the concepts and configuration of EIGRP route summarization.

Default Routes: This section examines the benefits of using default routes, and the mechanics of two methods for configuring default routes with EIGRP.

CHAPTER 4

EIGRP Route Summarization and Filtering

Engineers can choose to implement routing protocols such that all routers know routes to all prefixes in an internetwork. By allowing all routers to share all known prefixes, every router in an Enterprise has the potential ability to forward packets to all subnets in the Enterprise, assuming the internetwork is working correctly.

However, a reasonable network design may not need nor want the unfettered flow of routing information in an Enterprise. The design should consider the flow of data in the network, in particular, the fact that most data applications do not send packets directly between end user hosts. Most often, the packets go from one host to a server, and then the server communicates directly with the other end user host. So the design may require or desire that some routes are filtered so that routers in one part of the internetwork cannot forward packets to another part.

Additionally, the design may call for a general goal to reduce the size of the typical router's IP routing table. Although routers consume only a small amount of CPU to forward each packet, the amount of CPU required does increase slightly based on the number of routes. Additionally, a smaller routing table may be simpler and require less time when troubleshooting problems.

This chapter explains three categories of tools that you can use to limit the number of routes in the routing table: route filtering, route summarization, and default routes.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these nine self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 4-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

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

Foundations Topics Section	Questions
Route Filtering	1–4
Route Summarization	5–7
Default Routes	8, 9

1. Router R1 has been configured for EIGRP. The configuration also includes an ACL with one line–**access-list 1 permit 10.100.32.0 0.0.15.255**–and the EIGRP configuration includes the **distribute-list 1 in** command. Which of the following routes could not be displayed in the output of the **show ip eigrp topology** command as a result?

   a. 10.10.32.0/19
   b. 10.10.44.0/22
   c. 10.10.40.96/27
   d. 10.10.48.0/23
   e. 10.10.60.0/30

2. The command output that follows was gathered from router R1. If correctly referenced by an EIGRP distribution list that filters outbound Updates, which of the following statements is true about the filtering of various prefixes by this Prefix list?

```
R1#sh ip prefix-list
ip prefix-list question: 4 entries
   seq 5 deny 10.1.2.0/24 ge 25 le 27
   seq 15 deny 10.2.0.0/16 ge 30 le 30
   seq 20 permit 0.0.0.0/0
```

   a. Prefix 10.1.2.0/24 will be filtered due to clause 5.
   b. Prefix 10.1.2.224/26 will be filtered due to clause 5.
   c. Prefix 10.2.2.4/30 will be filtered due to clause 15.
   d. Prefix 10.0.0.0/8 will be permitted.
   e. Prefix 0.0.0.0/0 will be permitted.

3. R1 has correctly configured EIGRP to filter routes using a route map named *question*. The configuration that follows shows the entire route map and related configuration. Which of the following is true regarding the filtering action on prefix 10.10.10.0/24 in this case?

```
route-map question deny 10
 match ip address 1
route-map question permit 20
 match ip address prefix-list fred
!
access-list 1 deny 10.10.10.0 0.0.0.255
ip prefix-list fred permit 10.10.10.0/23 le 25
```

   a. It will be filtered due to the deny action in route map clause 10.
   b. It will be allowed because of the double negative (two deny references) in clause 10.
   c. It will be permitted due to matching clause 20's reference to prefix-list fred.
   d. It will be filtered due to matching the implied deny all route map clause at the end of the route map.

4. An engineer has typed four different single-line prefix lists in a word processor. The four answers show the four different single-line prefix lists. The engineer then does a copy/paste of the configuration into a router. Which of the lists could match a subnet whose prefix length is 27?

   a. **ip prefix-list fred permit 10.0.0.0/24 ge 16 le 28**
   b. **ip prefix-list barney permit 10.0.0.0/24 le 28**
   c. **ip prefix-list wilma permit 10.0.0.0/24 ge 25**
   d. **ip prefix-list betty permit 10.0.0.0/24 ge 28**

5. An engineer plans to configure summary routes with the **ip summary-address eigrp** *asn prefix mask* command. Which of the following, when added to such a command, would create a summary that includes all four of the following subnets: 10.1.100.0/25, 10.1.101.96/27, 10.1.101.224/28, and 10.1.100.128.25?

   a. 10.1.0.0 255.255.192.0
   b. 10.1.64.0 255.255.192.0
   c. 10.1.100.0 255.255.255.0
   d. 10.1.98.0 255.255.252.0

6. R1 has 5 working interfaces, with EIGRP neighbors existing off each interface. R1 has routes for subnets 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24, with EIGRP integer metrics of roughly 1 million, 2 million, and 3 million, respectively. An engineer then adds the **ip summary-address eigrp 1 10.1.0.0 255.255.0.0** command to interface Fa0/0. Which of the following is true?

   a. R1 loses and then reestablishes neighborships with all neighbors.
   b. R1 no longer advertises 10.1.1.0/24 to neighbors connected to Fa0/0.
   c. 1 advertises a 10.1.0.0/16 route out Fa0/0, with metric of around 3 million (largest metric of component subnets).
   d. R1 advertises a 10.1.0.0/16 route out Fa0/0, with metric of around 2 million (median metric of component subnets).

7. In a lab, R1 connects to R2, which connects to R3. R1 and R2 each have several working interfaces, all assigned addresses in class A network 10.0.0.0. Router R3 has some working interfaces in class A network 10.0.0.0, and others in class B network 172.16.0.0. The engineer experiments with the **auto-summary** command on R2 and R3, enabling and disabling the command in various combinations. Which of the following combinations will result in R1 seeing a route for 172.16.0.0/16, instead of the individual subnets of class B network 172.16.0.0?

   a. **auto-summary** on R2 and **no auto-summary** on R3
   b. **auto-summary** on R2 and **auto-summary** on R3
   c. **no auto-summary** on R2 and **no auto-summary** on R3
   d. **no auto-summary** on R2 and **auto-summary** on R3

8. Router R1 exists in an Enterprise that uses EIGRP as its routing protocol. The **show ip route** command output on router R1 lists the following phrase: "Gateway of last resort is 1.1.1.1 to network 2.0.0.0". Which of the following is most likely to have caused this output to occur on R1?

    a. R1 has been configured with an **ip default-network 2.0.0.0** command.

    b. R1 has been configured with an **ip route 0.0.0.0 0.0.0.0 1.1.1.1** command.

    c. R1 has been configured with an **ip route 2.0.0.0 255.0.0.0 1.1.1.1** command.

    d. Another router has been configured with an **ip default-network 2.0.0.0** command.

    e. Another router has been configured with an **ip route 2.0.0.0 255.0.0.0 1.1.1.1** command.

9. Enterprise Router R1 connects an Enterprise to the Internet. R1 needs to create and advertise a default route into the Enterprise using EIGRP. The engineer creating the implementation plan has chosen to base this default route on the **ip route** command, rather than using **ip default-network**. Which of the following is not a useful step with this style of default route configuration?

    a. Create the default route on R1 using the **ip route 0.0.0.0 0.0.0.0** *outgoing-interface* command.

    b. Redistribute the statically configured default route.

    c. Disable auto-summary.

    d. Configure the **network 0.0.0.0** command.

    e. Ensure R1 has no manually configured summary routes using the **ip summary-address eigrp** command.

Foundation Topics

Route Filtering

Does a router in a branch office need to be able to forward packets to hosts in another branch office? Does a router in the sales division need to be able to forward packets to hosts in the manufacturing division? These questions are just a sampling of design questions for which route filtering can be part of the solution.

Route filtering allows the engineer to filter which routes are advertised in an EIGRP update. If routers in a branch do not need to learn routes about subnets in other branches, routers can filter that routing information. This filtering reduces the size of routing tables, saving memory, possibly improving routing performance, and makes the internetwork more secure by limiting the flow of packets.

EIGRP enables route filtering using the **distribute-list** router subcommand. The concept is relatively straightforward: The distribute list refers to either an access control list (ACL), prefix list, or route map. These three tools classify whether a route should be permitted to be sent/received in an EIGRP Update or be denied (filtered). The **distribute-list** command also specifies the direction–outbound updates or inbound updates–and optionally, the specific interface on which to filter updates.

For example, Figure 4-1 shows an expanded version of the internetwork used frequently in the previous two chapters. The figure adds several links between the WAN routers and some core Layer 3 switches. It also notes the address ranges for all data centers (10.16.0.0/16) and the range of addresses used for subnets in the manufacturing division (10.17.32.0/19).

The design engineer could make many choices about what routes to filter, for example:

- Filter routes to WAN subnets so that the core and manufacturing do not learn those routes, because these subnets should not be the destination of any user traffic.
- Filter manufacturing routes from being advertised to the branches, because the branches are in the sales division.
- Filter routes for the subnets sitting between the Layer 3 switches in the core, preventing them from being advertised to either manufacturing or the sales branches, because no users in these divisions should be sending packets to these subnets.

The examples in this chapter focus on the first of these design options.

Filtering the subnets that exist between Layer 3 devices, as is suggested in the second and third items in the list, have both pros and cons. For example, the second design goal filters the WAN subnets, because no end users need to be able to send packets to those subnets. This meets the goal of having smaller routing tables. However, operations personnel may have a larger challenge when monitoring and troubleshooting, because when a **ping** or **traceroute** fails, they also need to figure out whether the command failed by design due to the purposefully filtered routes, or whether a problem has occurred.

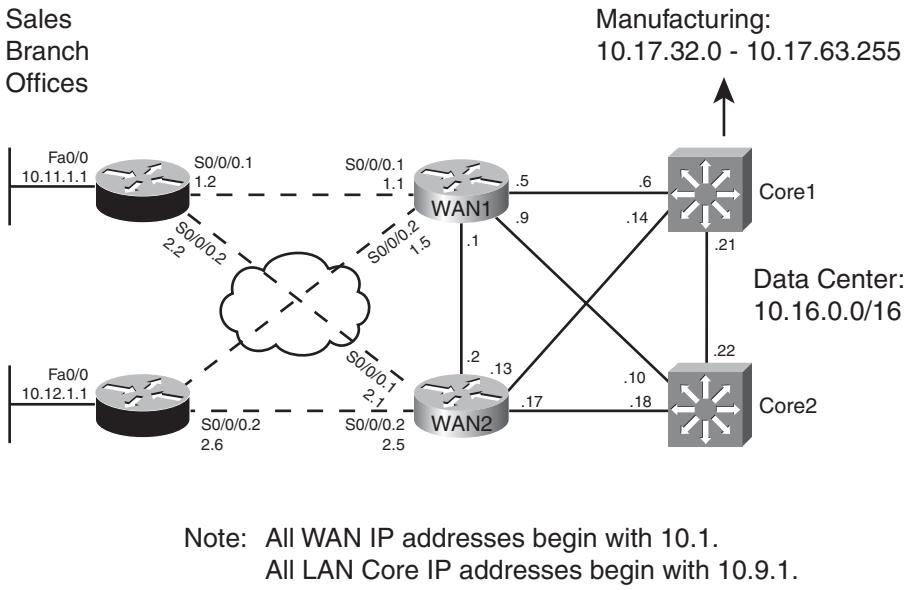


Figure 4-1 *Expanded Design with a Range of Addresses in Manufacturing*

This section next examines how to filter EIGRP routes using ACLs, prefix lists, and then route maps. All three of these tools will be used throughout this book, so this chapter lays the foundation for understanding these tools, in addition to showing how to use these tools when filtering EIGRP routes.

Filtering by Referencing ACLs

To filter EIGRP routes by matching them using ACLs, the ACL must match a route with a **permit** clause to then allow the route to be advertised, and match the route with a **deny** clause to filter the route. Before getting into how an ACL matches a route, first it is important to review what can be examined based on the configuration of an IP ACL.

EIGRP distribute lists support the use of standard IP ACLs. The syntax of both numbered and named standard ACLs allows a configuration of one dotted decimal number and its corresponding wildcard (WC) mask. When used for packet filtering, this number is compared to the source IP address of the packet. When referenced by the **distribute-list** command for the purpose of EIGRP route filtering, EIGRP compares the standard ACL source-address field to the subnet number (prefix) of each EIGRP route.

The best way to learn the specifics is to consider several examples. Figure 4-2 shows the specific size subnets being advertised from the manufacturing division into the core. The design calls for the WAN routers to filter these routes from being advertised toward the Sales division's branch offices.

Figure 4-2 shows the **distribute-list 2 out s0/0/0.1** command on Router WAN1 as one sample of the syntax. A command like this would need to be included in WAN1's configuration for each interface connected to a branch. ACL number 2 would then be configured

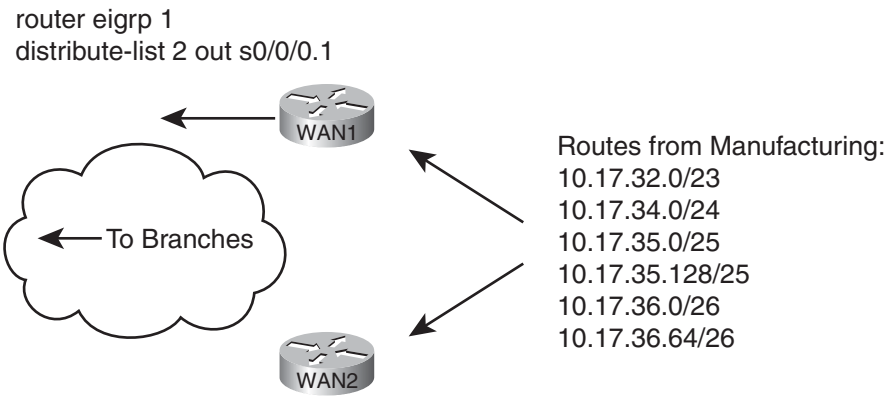


Figure 4-2 *Specific Manufacturing Routes to Be Filtered*

to match the manufacturing routes with a deny clause, and all other routes with a permit clause, filtering the routes.

If WAN1 has hundreds of serial subinterfaces for its WAN connections, then following the sample in the previous paragraph, WAN1 would have hundreds of **distribute-list 2 out serial** *number* commands, one per WAN interface/subinterface. Alternatively, the engineer could configure a single **distribute-list 2 out** command on Router WAN1, not specifying an interface. In this case, Router WAN1 would not advertise these routes to any neighbors, greatly reducing WAN1's configuration.

Given that this section represents the first use of ACLs in this book, at your option, take the time to try the following exercise as a review of ACLs. Consider the following **access-list** commands. Imagine that each command in this list is the first of two commands in a single access-list. The second and only other command is a command that permits all other routes—for example, **access-list 2 permit any**. Then, ask yourself—if used by a **distribute-list** on WAN1 to filter the manufacturing routes (as seen in Figure 4-2), and you want that ACL to filter only manufacturing routes, which of these 2-line ACLs meet the requirements?

```
access-list 3 deny 10.17.32.0
access-list 4 deny 10.17.32.0 0.0.0.255
access-list 5 deny 10.17.32.0 0.0.3.255
access-list 6 deny 10.16.0.0 0.1.255.255
```

To keep from giving away the answers, Table 4-2, which supplies the answers and explanation, is located on the next page.

Table 4-2 *Analysis of the Sample ACLs Used with the distribute-list Command*

ACL	Routes Filtered	Explanation
3	10.17.32.0/23	The ACL matches exactly prefix 10.17.32.0, so it matches a single manufacturing route.
4	10.17.32.0/23	The ACL matches all prefixes that begin with 10.17.32 because of the WC mask, again matching a single route.
5	10.17.32.0/23 10.17.34.0/24 10.17.35.0/25 10.17.35.128/25	The ACL matches all prefixes in the range 10.17.32.0–10.17.35.255, which includes four manufacturing routes.
6	All manufacturing and data center routes	The ACL matches all prefixes in the range 10.16.0.0–10.17.255.255, which includes the data center routes.

Note: To find the range of numbers matched by an ACL's address and wildcard mask values, use the address field as the low end of the range, and simply add the address and wildcard mask to find the high end of the range.

Example 4-1 shows the configuration on router WAN1 to filter the manufacturing routes, with distribute lists enabled on its two WAN subinterfaces. The ACL matches (with a deny action) all manufacturing routes, and matches all other routes with a permit clause.

Example 4-1 *WAN1's distribute-list to Filter Manufacturing Routes*

```
! On Router B1, before the filtering is applied:
B1#show ip route ¦ include 10.17
D        10.17.35.0/25 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
D        10.17.34.0/24 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
D        10.17.32.0/23 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
D        10.17.36.0/26 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
D        10.17.36.64/26 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
D        10.17.35.128/25 [90/2300416] via 10.1.1.1, 00:00:18, Serial0/0/0.1
```

```
! On Router WAN1:

WAN1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
WAN1(config)#access-list 2 deny 10.17.32.0 0.0.31.255
WAN1(config)#access-list 2 permit any
WAN1(config)#router eigrp 1
WAN1(config-router)#distribute-list 2 out
WAN1(config-router)#^Z
WAN1#
```

```
! On Router B1, after the filtering is applied
B1#show ip route ¦ include 10.17
B1#
```

Note: The same configuration added to Router WAN1 was also added to Router WAN2; however, the commands were not repeated in Example 4-1.

The ACL in this case, ACL 2, matches all subnets with a value between 10.17.32.0–10.17.63.255, inclusive, based on the IP address value of 10.17.32.0 and WC mask of 0.0.31.255. By matching these routes with a **deny** clause, the ACL, used as a distribute list, filters the routes. The **access-list 2 permit any** command matches all other routes, allowing them to be advertised.

Table 4-2 lists the answers to the exercise listed a few pages back.

Filtering by Referencing IP Prefix Lists

The IOS IP **prefix-list** feature gives the network engineer another tool for matching routes when performing route filtering. IP prefix lists can examine both the prefix and the prefix length, and a range of prefixes or a range of prefix lengths. The command then sets either a deny or permit action for each matched prefix/length. To use the prefix-list, the configuration simply refers to the **prefix-list** with the same **distribute-list** command seen earlier.

Using IP prefix lists for route filtering has several advantages. First, IP prefix lists allow matching of the prefix length, whereas the ACLs used by the EIGRP **distribute-list** command cannot. (Some other route filtering configurations can match both the prefix and prefix length using extended ACLs.) Many people find IP prefix lists more intuitive for configuring route filtering. Finally, the internal processing of the IP prefix lists uses an internal tree structure that results in faster matching of routes as compared with ACLs.

This section begins by examining IP prefix lists as an end to itself, followed by an example of filtering EIGRP routes using a prefix list.

IP Prefix List Concepts

IP prefix lists provide mechanisms to match two components of an IP route:

- The route prefix (the subnet number)
- The prefix length (the subnet mask)

Each single IP prefix list has similar characteristics to a single ACL, with subtle similarities to both numbered and named ACLs. The IP prefix list consists of one or more global configuration commands (like numbered ACLs), with commands using the same name being in the same list (like named ACLs). As with named ACLs, each **ip prefix-list** command has a sequence number to allow later deletion of individual commands and insertion of commands into a particular sequence position. Each command has a **permit** or **deny** action—but because it is used only for matching routes, and not for packet filtering, the **permit** or **deny** keyword just implies whether a route is matched (**permit**) or not (**deny**).

The generic command syntax is as follows:

```
ip prefix-list list-name [seq seq-value] {deny | permit  prefix/prefix-
length}[ge ge-value] [le le-value]
```

The following statements summarize the logic:

Key Topic

Step 1. The route's prefix must be within the range of addresses implied by the **prefix-list** command's *prefix/prefix-length* parameters.

Step 2. The *route's prefix length* must match the *range of prefixes* implied by the **prefix-list** command's prefix-*length*, **ge**, and **le** parameters.

The matching of the prefix works much like the ACL matching logic. The configured prefix/prefix-length implies a range of IP addresses. For example, an **ip prefix-list barney deny 10.0.0.0/8...** implies any number whose first 8 bits (per the /8) match 10.0.0.0–in other words, all IPv4 addresses that begin with 10. Any route whose prefix is in this range–for example, 10.0.0.0, 10.1.1.0, and 10.5.255.128—would be considered to match this part of the logic.

However, IP prefix lists always examine the prefix length as well. To perform the logic of matching a route's prefix length, IOS considers the following parts of the **ip prefix-list** command:

- The required *prefix-length* parameter
- The optional *ge-value*, which stand for *greater-than-or-equal-to*
- The optional *le-value, which stand for less-than-or-equal-to*

For a given **ip prefix-list** command, one of four configuration combinations affect the logic of matching prefix lengths, as listed in Table 4-3. The text following the table provides a more detailed explanation as compared with the summarized information in the table.

Key Topic

Table 4-3 *LE and GE Parameters on IP Prefix List, and the Implied Range of Prefix Lengths*

Prefix List Parameter	Range of Prefix Length
Neither	*conf-length must = route-length*
Both **ge** and **le**	*ge-value <= route-length <= le-value*
Only **le**	*conf-length <= route-length <= le-value*
Only **ge**	*ge-value <= route-length <= 32*

The first case in the table occurs when neither **ge** nor **le** is configured. In that case, an exact match of prefix-length must occur between the configured prefix length and a route's prefix length. For example, the command **ip prefix-list fred deny 10.0.0.0/8** matches route 10.0.0.0/8, but not 10.0.0.0/20.

The second case in the table occurs when both **ge** and **le** are configured. In that case, the route's prefix length must be between the configured **ge** and **le** values, inclusive. For in-

stance, **ip prefix-list fred deny 10.0.0.0/8 ge 20 le 22** matches route 10.0.0.0/20, but not 10.0.0.0/8, because the prefix length must either be 20, 21, or 22.

The cases in which either **ge** or **le** is configured, but not both, requires a little more thought. A visual representation can help, as shown in Figure 4-3.

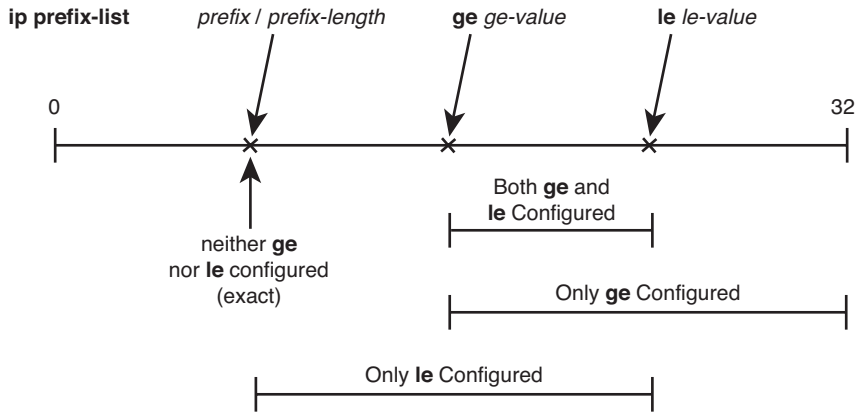


Figure 4-3 *Representation of Prefix Length Ranges for ip prefix-list Command*

In short, with only **ge** configured, the command matches prefix-length ranges from the *ge-value* up to 32 (the longest IPv4 prefix length), inclusive. With only **le** configured, the command matches prefix-length ranges between the *prefix-length* parameter and the *le-value*, inclusive.

Note: IOS requires that the configured prefix-length, ge-value, and le-value meet the following requirement: prefix-length <= ge-value <= le-value. Otherwise, IOS rejects the **ip prefix-list** command.

Samples of Prefix List Matching

Several examples can really help nail down prefix list logic. The following routes will be examined by a variety of prefix lists, with the routes numbered for easier reference:

Step 1. 10.0.0.0/8

Step 2. 10.128.0.0/9

Step 3. 10.1.1.0/24

Step 4. 10.1.2.0/24

Step 5. 10.128.10.4/30

Step 6. 10.128.10.8/30

Next, Table 4-4 shows the results of seven different one-line prefix lists applied to these six example routes. The table lists the matching parameters in the **prefix-list** commands,

omitting the first part of the commands. The table explains which of the six routes would match the listed prefix list, and why.

Table 4-4 *Example Prefix Lists Applied to the List of Routes*

prefix-list Command Parameter	Routes Matched from previous list of prefixes	Result
10.0.0.0/8	1	Without **ge** or **le** configured, both the prefix (10.0.0.0) and length (8) must be an exact match.
10.128.0.0/9	2	Without **ge** or **le** configured, the prefix (10.128.0.0) and length (9) must be an exact match.
10.0.0.0/8 ge 9	2–6	The 10.0.0.0/8 means "all routes whose first octet is 10." The prefix length must be between 9 and 32, inclusive.
10.0.0.0/8 ge 24 le 24	3, 4	The 10.0.0.0/8 means "all routes whose first octet is 10," and the prefix range is 24 to 24—meaning only routes with prefix length 24.
10.0.0.0/8 le 28	1–4	The prefix length needs to be between 8 and 28, inclusive.
0.0.0.0/0	None	0.0.0.0/0 means "match all prefixes." However, because no le nor ge parameter is configured, the /0 also means that the prefix length must be 0. So, it would match all routes' prefixes, but none of their prefix lengths. Only a default route would match this prefix list.
0.0.0.0/0 le 32	All	The range implied by 0.0.0.0/0 is all IPv4 addresses. The **le 32** combined with prefix length 0 implies any prefix length between 0 and 32, inclusive. This is the syntax for "match all" prefix list logic.

Note: Pay particular attention to the match all logic of the final entry in the table.

Using IP Prefix Lists to Filter EIGRP Routes

After you master the logic behind IP prefix lists, using them with the **distribute-list** command requires minimal extra effort. For example, to refer to a prefix list name Fred, you could configure the **distribute-list prefix Fred...** command, instead of **distribute-list 2...** to refer to ACL 2. (Note that the prefix list names are case-sensitive.)

For example, using the internetwork of Figure 4-1 and Figure 4-2 again, consider the following revised design requirements for route filtering:

- Of the routes from manufacturing, filter only those routes that begin with 10.17.35 and 10.17.36.
- Of the routes for subnets on the WAN link, filter routes to prevent the core routers and branch routers from learning routes whose prefix length is /30.

Although the first of the preceding two requirements mainly exists to demonstrate the **ip prefix-list** command, the second goal may be more useful for real networks. Often, routes with a /30 prefix length are routes used between two routers, either on WAN links or over LANs between Layer 3-enabled devices. Users should not need to send packets to addresses in these subnets. So, the only need to have routes to these subnets is for network management (ping tests, for instance).

Example 4-2 shows the configuration on WAN1; the equivalent configuration has been added on WAN2 as well.

Example 4-2 *Filtering All Routes with a /30 Prefix Length*

```
! On Router WAN1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WAN1#show running-config
! lines omitted for brevity
router eigrp 1
 network 10.0.0.0
 distribute-list prefix fred out
auto-summary
!
ip prefix-list fred seq 5 deny 10.17.35.0/24 ge 25 le 25
ip prefix-list fred seq 10 deny 10.17.36.0/24 ge 26 le 26
ip prefix-list fred seq 15 deny 0.0.0.0/0 ge 30 le 30
ip prefix-list fred seq 20 permit 0.0.0.0/0 le 32
```

```
! On Router B1:
B1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
C       10.11.1.0/24 is directly connected, FastEthernet0/0
D       10.12.1.0/24 [90/2684416] via 10.1.2.1, 00:06:15, Serial0/0/0.2
                     [90/2684416] via 10.1.1.1, 00:06:15, Serial0/0/0.1
C       10.1.2.0/30 is directly connected, Serial0/0/0.2
C       10.1.1.0/30 is directly connected, Serial0/0/0.1
D       10.16.1.0/24 [90/2172672] via 10.1.2.1, 00:00:32, Serial0/0/0.2
                     [90/2172672] via 10.1.1.1, 00:00:32, Serial0/0/0.1
D       10.17.34.0/24 [90/2300416] via 10.1.2.1, 00:06:15, Serial0/0/0.2
                      [90/2300416] via 10.1.1.1, 00:06:15, Serial0/0/0.1
D       10.17.32.0/23 [90/2300416] via 10.1.2.1, 00:06:15, Serial0/0/0.2
                      [90/2300416] via 10.1.1.1, 00:06:15, Serial0/0/0.1

B1#show ip route 10.17.32.0 255.255.248.0 longer-prefixes
! The legend is normally displayed; omitted here for brevity

     10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
D       10.17.34.0/24 [90/2300416] via 10.1.2.1, 00:04:12, Serial0/0/0.2
                      [90/2300416] via 10.1.1.1, 00:04:12, Serial0/0/0.1
D       10.17.32.0/23 [90/2300416] via 10.1.2.1, 00:04:12, Serial0/0/0.2
                     [90/2300416] via 10.1.1.1, 00:04:12, Serial0/0/0.1
```

The configuration on WAN1 includes a four-line prefix list. The first line (sequence number 5) matches 10.17.35.0/25 and 10.17.35.128/25, in part because it asks for a range of prefix lengths from 25 to 25–meaning an exact length of 25. Similarly, the second statement (sequence number 10) matches routes 10.17.36.0/26 and 10.17.36.64/26. The third statement (sequence number 15) uses wildcard logic (**0.0.0.0/0**) to match all prefixes, but only those with prefix length 30 (**ge 30 le 30**). The last command matches all prefixes, with prefix lengths from 0 to 32 (all prefix lengths).

The resulting IP routing table on branch Router B1 shows only a small number of routes. B1 has a route to the other example branch's subnet (10.12.1.0), and another in the range of addresses for the data centers (10.16.1.0/24). It has the two routes leaked from manufacturing. Note that the only two /30 routes known on B1 are two connected routes, so the **distribute-list** is filtering all the /30 routes.

Filtering by Using Route Maps

Route maps, the third EIGRP route filtering tool that can be referenced with the **distribute-list** command, provides programming logic similar to the If/Then/Else logic seen in programming languages. A single route map has one or more **route-map** commands in it, and routers process **route-map** commands in sequential order based on sequence numbers. Each **route-map** command has underlying matching parameters, configured with the aptly named **match** command. (To match all packets, the **route-map** clause simply omits the **match** command.)

Route maps can be used for many functions besides being used to filter routes for a single routing protocol like EIGRP. Route maps can be used to filter routes during the route redistribution process, and to set BGP Path Attributes (PAs) for the purpose of influencing the choice of the best routes in an internetwork.

When used for filtering EIGRP routes, route maps do provide a few additional features beyond what can be configured using ACLs and prefix lists. However, route maps can be tricky to understand and sometimes counterintuitive. This section begins with an examination of the concepts behind IOS route maps, followed by some examples of their use for filtering EIGRP routes.

Route Map Concepts

Route maps have many similarities when compared to ACLs and prefix lists. A single route map has several **route-map** commands, with the commands in the same route map all having the same text name. When referenced by the **distribute-list** command, IOS processes the commands in the route map sequentially, based on the sequence numbers in the commands. Like ACLs and prefix lists, IOS adds the sequence numbers automatically if omitted when configuring the **route-map** commands. And once a particular route has been matched and determined to be either filtered (deny) or allowed to pass (permit), even if more **route-map** commands exist later in the list, IOS stops processing the route-map for that route.

Each **route-map** command includes the name of the route map, an action (**permit** or **deny**), and possibly a sequence number (optional). After typing this command, the CLI user is in **route-map** configuration mode for that route-map clause. Any **match** commands configured in that mode apply to that single **route-map** command. For instance, Example 4-3 shows the configuration of a sample route map on router WAN1.

Example 4-3 *Pseudocode for Route Map Used as EIGRP Route Filter*

```
route-map sample-rm deny 8
 match (1st set of criteria)
route-map sample-rm permit 17
 match (2nd set of criteria)
route-map sample-rm deny 30
 match (3rd set of criteria)
route-map sample-rm permit 35
!
router eigrp 1
 distribute-list route-map sample-rm out
```

Example 4-3 shows pseudocode, ignoring the specifics of what is matched with the **match** commands. Focus on the actions in the **route-map** command (permit or deny), and the overall logic, as listed here:

- **Seq #8:** The action is **deny**, so discard or filter all routes matched by the **match** command (1st set of criteria).

- **Seq #17:** The action is **permit**, so allow through all routes matched by the **match** command (2nd set of criteria).
- **Seq #30:** The action is **deny**, so discard or filter all routes matched by the **match** command (3rd set of criteria).
- **Seq #35:** The action is **permit**. The absence of a **match** command means "match all," so allow through all remaining routes.

The **match** command can reference an ACL or prefix list, but doing so does introduce the possibility of confusion. The confusing part is that the decision to filter a route or allow the route through is based on the **deny** or **permit** in the **route-map** command, and *not the deny or permit in the ACL or prefix list*. When referencing an ACL or prefix list from a route map, the ACL or prefix list simply matches all routes permitted by the ACL or prefix list. Routes that are denied by the ACL or prefix list simply do not match that **match** command's logic, making IOS then consider the next **route-map** command.

The following list summarizes the key points about route map logic when used for redistribution:

Key Topic

- **route-map** commands with the **permit** option either cause a route to be allowed through (if matched by the **match** command) or remain in the list of routes to be examined by the next **route-map** clause.
- **route-map** commands with the **deny** option either filter the route (if matched by the **match** command) or remain in the list of routes to be examined by the next **route-map** clause.
- If a clause's **match** commands refer to an ACL or prefix list, and the ACL or prefix list matches a route with the **deny** action, the route is not necessarily filtered. Instead, it just means that route does not match that particular **match** command and can then be considered by the next **route-map** clause.
- The **route-map** command includes an implied deny all clause at the end; to configure a permit all, use the **route-map** command, with a **permit** action, but without a **match** command.

Route maps have several more options on the **match command** as compared to what can be examined by ACLs and IP prefix lists. However, for the purposes of EIGRP route filtering, the items that may be matched do not provide significant help in filtering routes. However, when redistributing routes from other routing protocols, as is covered in Chapter 9, "Basic IGP Redistribution," and Chapter 10, "Advanced IGP Redistribution," some of the **match** command's other options can be very helpful.

Using Route Maps to Filter EIGRP Routes

The mechanics of the configuration works much like the other two filtering features. The **distribute-list** command refers to the feature that matches the packets, in this case a **route-map** command option. The **distribute-list** command again lists a direction (in or out), and optionally an interface.

Example 4-4 shows the configuration results in an excerpt from the **show running-config** command, along with the output of the **show route-map** command. The configuration im-

plements the same logic as used in Example 4-2 earlier in this chapter, under the heading "Using IP Prefix Lists to Filter EIGRP Routes." The design criteria are the same as with that earlier example:

- Of the routes from manufacturing, filter only those routes that begin with 10.17.35 and 10.17.36.
- Filter WAN routers from advertising any /30 routes in the Layer 3 core.

Example 4-4 *Filtering All Routes with a /30 Prefix Length, Plus Some Routes from Manufacturing*

```
! On Router WAN1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WAN2#show running-config
! lines omitted for brevity
router eigrp 1
 network 10.0.0.0
 distribute-list route-map filter-man-slash30 out
 auto-summary
!
ip prefix-list manufacturing seq 5 permit 10.17.35.0/24 ge 25 le 25
ip prefix-list manufacturing seq 10 permit 10.17.36.0/24 ge 26 le 26
!
ip prefix-list slash30 seq 5 permit 0.0.0.0/0 ge 30 le 30
!
route-map filter-man-slash30 deny 8
 match ip address prefix-list manufacturing
!
route-map filter-man-slash30 deny 15
 match ip address prefix-list slash30
!
route-map filter-man-slash30 permit 23

! Notice - no match commands, so the above clause matches all remaining routes
!
! lines omitted for brevity
WAN2#show route-map
route-map filter-man-slash30, deny, sequence 8
  Match clauses:
    ip address prefix-lists: manufacturing
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map filter-man-slash30, deny, sequence 15
  Match clauses:
    ip address prefix-lists: slash30
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map filter-man-slash30, permit, sequence 23
```

```
Match clauses:
Set clauses:
Policy routing matches: 0 packets, 0 bytes
```

In particular, note that the first two **route-map** commands list a **deny** action, meaning that all routes matched in these two clauses will be filtered. The IP prefix lists referenced in the **match** commands, called **manufacturing** and **slash30**, respectively, each match (permit) the routes listed in one of the two design goals. Note that the logic of both prefix lists could have easily been configured into a single prefix list, reducing the length of the **route-map** command as well. Finally, note that the last **route-map** command has a **permit** action, with no **match** command, meaning that the default action is to allow the route to be advertised.

Also, it can be useful to take a moment and review Example 4-2 as a point of comparison for the use of the IP prefix lists in each case. In the **route-map** of Example 4-4, the prefix list needs to match the routes with a permit clause so that the route-map **deny** action causes the routes to be filtered. Earlier, Example 4-2 shows the same basic logic in the prefix list, but with an action of deny. The reasoning is that when the **distribute-list prefix-list...** command refers directly to an IP prefix list, IOS then filters routes denied by the prefix list.

Route Summarization

As mentioned in the introduction to this chapter, keeping routing tables small helps conserve memory and may improve the time required by a router to forward packets. Route filtering allows an engineer to reduce the size of the routing table, but with the side effect of limiting the destinations reachable by each router. That effect may or may not be acceptable, given the other design goals of a particular internetwork, and given the need to operate the network.

Route summarization allows an engineer to keep the routing tables more manageable, without limiting reachability. Instead of advertising routes for every subnet, a router advertises a single route that represents the same range of IP addresses as more than one subnet. Each router can forward packets to the same set of destinations, but the routing table is smaller. For example, instead of advertising routes 10.11.0.0/24, 10.11.1.0/24, 10.11.2.0/24, and so on–all subnets up through 10.11.255.0/24–a router could advertise a single route for 10.11.0.0/16, which includes the exact same range of addresses.

This section begins by examining some design issues related to route summarization. Then the text moves on to explain how to explicitly configure EIGRP summary routes, finishing with a discussion of automatically created summaries based on the **auto-summary** command and feature.

Route Summarization Design

Route summarization works best when the subnet planning process considers route summarization. To accommodate summarization, the engineer assigning subnets can assign

larger address blocks to one part of the topology. The engineers working with that part of the internetwork can break the address blocks into individual subnets as needed. At the edge of that part of the network, the engineers can configure route summaries to be advertised to the other parts of the internetwork. In short, when possible, plan the route summaries before deploying the new parts of an internetwork, and then assign addresses to different parts of the internetwork within their assigned address blocks.

For example, consider Figure 4-4, which shows a variation on the same internetwork shown earlier in this chapter, with the address blocks planned before deployment.

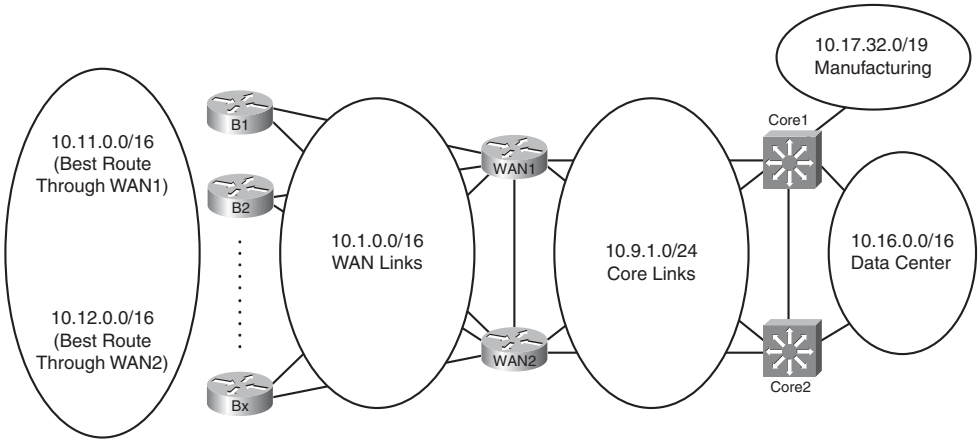


Figure 4-4 *Address Blocks Planned for Example Enterprise Internetwork*

Figure 4-4 shows the address blocks planned for various parts of the internetwork, as follows:

- Assign branch subnets come from two consecutive ranges–10.11.0.0/16 and 10.12.0.0/16.
- Assign WAN router-to-router subnets from the range 10.1.0.0/16.
- Assign core LAN router-to-router subnets from the range 10.9.0.0/16.
- Assign Data Center subnets from the range 10.16.0.0/16.
- Give the manufacturing division, which has a separate IT staff, address block 10.17.32.0/19.

Inside each of the circles in Figure 4-4, the engineering staff can assign subnets as the need arises. As long as addresses are not taken from one range and used in another part of the internetwork, the routers at the boundary between the regions (circles) in Figure 4-4 can configure EIGRP route summarization to both create one large summary route and prevent the advertisement of the smaller individual routes.

Calculating Summary Routes

Note that the examples in this chapter generally use simpler examples of summary routes, using prefix lengths like /24 and /16 most often. However, for the exam, you need to be comfortable interpreting prefix/prefix length pairs, and subnet/mask pairs, whether they represent an actual subnet or a summary route.

The math to analyze a subnet/mask pair, or prefix/length pair, is identical to the math included as part of the CCNA certification. As such, this book does not attempt to explain those same concepts, other than this brief review of one useful shortcut when working with potential summary routes.

If you can trust that the subnet/mask or prefix/length is a valid subnet or summary, then the following method can tell you the range of numbers represented. For example, consider 10.11.0.0/16. Written in subnet/mask form, it is 10.11.0.0/255.255.0.0. Then, invert the mask by subtracting the mask from 255.255.255.255, yielding 0.0.255.255 in this case. Add this inverted mask to the subnet number (10.11.0.0 in this case), and you have the high end of the range (10.11.255.255). So, summary 10.11.0.0/16 represents all numbers from 10.11.0.0–10.11.255.255.

When using less obvious masks, the process works the same. For example, consider 10.10.16.0/20. Converting to mask format, you have 10.10.16.0/255.255.240.0. Inverting the mask gives you 0.0.15.255. Adding the inverted mask to the subnet number gives you 10.10.31.255, and a range of 10.10.16.0–10.10.31.255.

Before closing this short section about calculating summary routes, note that the the process of adding the inveretd subnet mask assumes that the prefix/length or subnet/mask is a valid subnet number or valid summary route. If it is not, then you can still do the math, but neither the low end nor high end of the range is valid. For example, 10.10.16.0/19, similar to the previous example, is not actually a subnet number. 10.10.16.0 would be an IP address in subnet 10.10.0.0/19, with range of addresses 10.10.0.0–10.10.31.255.

Choosing Where to Summarize Routes

EIGRP supports route summarization at any router, unlike OSPF, which requires that summarization be performed only at area border routers (ABR) or autonomous system border routers (ASBR). EIGRP's flexibility helps when designing the internetwork, but it also poses some questions as to where to summarize EIGRP routes.

In some cases, the options are relatively obvious. For example, consider the 10.17.32.0/19 address block in manufacturing in Figure 4-4. The manufacturing division's router could summarize all its routes as a single 10.17.32.0/19 route when advertising to Core1. Alternately, Core1 could summarize all those same routes, advertising a summary for 10.17.32.0/19. In either case, packets from the rest of the internetwork shown in Figure 4-4 will flow toward Core1 and then to the Manufacturing division.

Next, consider the 10.16.0.0/16 address block in the Data Center. Because all these subnets reside to the right of Layer 3 switches Core1 and Core2, these two devices could summarize 10.16.0.0/16. However, these routes could also be summarized on WAN1/WAN2 for advertisement to the branches on the left. Summarizing on Core1/Core2 helps reduce the

size of the routing tables on WAN1 and WAN2. However, the sheer number of subnets in a Data Center is typically small compared to the number of small remote sites, so the savings of routing table space may be small.

One advantage of summarizing 10.16.0.0/16 on WAN1/WAN2 instead of Core1/Core2 in this case is to avoid routing inefficiencies in the core of the internetwork. The later section "Suboptimal Forwarding with Summarization" discussed the concept with a different example.

Influencing the Choice of Best Route for Summary Routes

Often, engineers plan route summarization for the same address block on multiple routers. Such a design takes advantage of redundancy and can be used to perform basic load balancing of traffic across the various paths through the internetwork. Figure 4-5 shows one such example, with Routers WAN1 and WAN2 summarizing routes for the two address blocks located on the branch office LANs: 10.11.0.0/16 and 10.12.0.0/16.

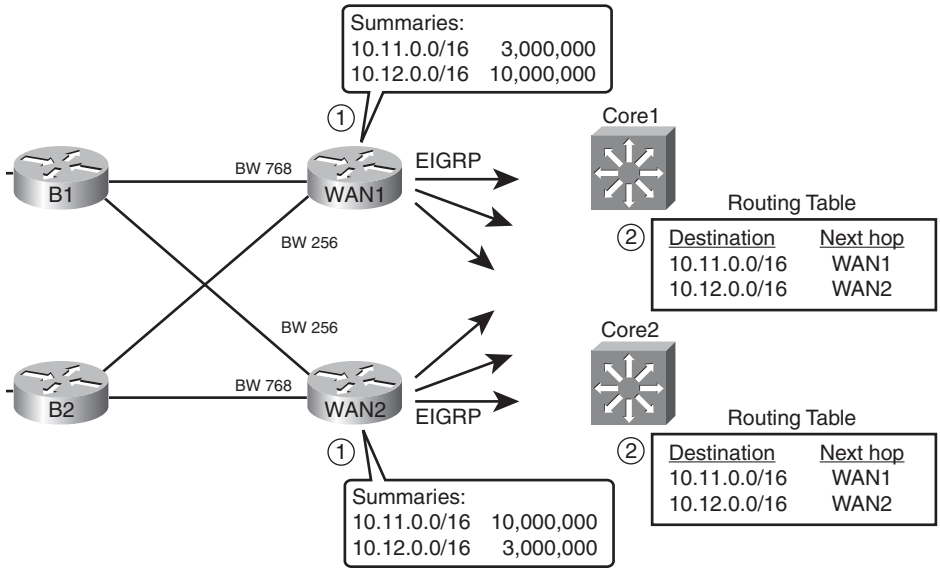


Figure 4-5 *Choosing Locations for Route Summarization*

The figure shows the advertisements of the summary routes. WAN1 and WAN2 both advertise the same summarizes: 10.11.0.0/16 for some branches and 10.12.0.0/16 for the others. Note that by advertising the WAN routes, instead of filtering, the operations staff might have an easier time monitoring and troubleshooting the internetwork, while still meeting the design goal of reducing the size of the routing table. (Also, note that Router WAN1 summarizes Manufacturing's routes of 10.17.32.0/19.)

In some cases, the network designer has no preference for which of the two or more routers should be used to reach hosts within the summary route range. For example, for most Data

Center designs, as shown earlier in Figure 4-4, the routes from the left of the figure toward the Data Center, through Core1 and Core2, would typically be considered equal.

However, in some cases, as in the design shown in Figure 4-5, the network designer wants to improve the metric of one of the summary routes for a single address block to make that route the preferred route. Using 10.11.0.0/16 as an example, consider this more detailed description of the design:

- Use two PVCs to each branch–one faster PVC with 768 Kbps CIR and one slower PVC (either 128 Kbps or 256 Kbps CIR).
- Roughly half the branches should have a faster PVC connecting to Router WAN1, and the other half of the branches should have a faster PVC connecting to Router WAN2.
- Assign user subnets from the range 10.11.0.0/16 for branches that use WAN1 as the primary WAN access point, and from 10.12.0.0/16 for the branches that use WAN2 as primary.
- Routing should be influenced such that packets flow in both directions over the faster WAN link, assuming that link is working.

This design requires that both directions of packets flow over the faster PVC to each branch. Focusing in the outbound (core-toward-branch) direction for now, by following the design, and setting the interface bandwidth settings to match the PVC speeds, the outbound routes will send packets over the faster PVCs. The main reason for the route choices is the following fact about summary routes with IOS:

> Set the summary route's metric components based on the lowest metric route upon which the summary route is based.

By setting the interface bandwidth settings to match the design, the two WAN routers should summarize and advertise routes for 10.11.0.0/16 and 10.12.0.0/16, advertising these routes toward the core–but with different metrics.

WAN1 advertises its 10.11.0.0/16 route with a lower metric than WAN2's summary for 10.12.0.0/16 because all of WAN1's routes for subnets that begin 10.11 are reachable over links set to use 768 Kbps of bandwidth. All WAN1's links to branches whose subnets begin 10.12 are reachable over links of speed 128 KBps or 256 Kbps, so WAN1's metric is higher that WAN2's metric for the 10.12.0.0/16 summary. WAN2 follows the same logic but with the lower metric route for 10.12.0.0/16.

As a result of the advertisements on WAN1 and WAN2, the core routers both have routing table entries that drive traffic meant for the faster-through-WAN1 branches to WAN1, and traffic for the faster-through-WAN2 branches to WAN2.

Suboptimal Forwarding with Summarization

The final concept to consider when summarizing routes is that the packets may take a longer path than if summarization is not used. The idea works a little like this story. Say you were travelling to Europe from the USA. You knew nothing of European geography, other than that you wanted to go to Paris. So, you look around and find hundreds of flights to Europe and just pick the cheapest one. When you get to Europe, you worry

about how to get the rest of the way to Paris–be it a Taxi ride from the Paris airport, or whether it takes a day of train travel. Although you do eventually get to Paris, if you had chosen to know more about European geography before you left, you could have saved yourself some travel time in Europe.

Similarly, routers that learn a summary route do not know about the details of the subnets inside the summary. Instead, like the person who just picked the cheapest flight to Europe, the routers pick the lowest metric summary route for a prefix. That router forwards packets based on the summary route. Later, when these packets arrive at routers that do know all the subnets inside the summary, those routers can then use the best route–be it a short route or long route.

For example, Figure 4-6 shows the less efficient routing of packets to host 10.11.1.1, a host off Router B1, assuming that the route summarization shown in Figure 4-5 still exists. When R1's 768 Kbps CIR PVC to Router B1 fails, WAN1 does not change its route advertisement for its 10.11.0.0/16 summary route. When EIGRP advertises a summary route, the advertising router considers the summary route to be up and working unless all subordinate routes fail. Unless all of WAN1's specific routes in the 10.11.0.0/16 range failed, R1 would not notify routers on the right about any problem. So, when the example shown in Figure 4-6 begins, the 10.11.0.0/16 summary advertised by WAN1, as seen earlier in Figure 4-5, is still working, and both Core1 and Core2 use WAN1 as their next-hop router for their routes to 10.11.0.0/16.

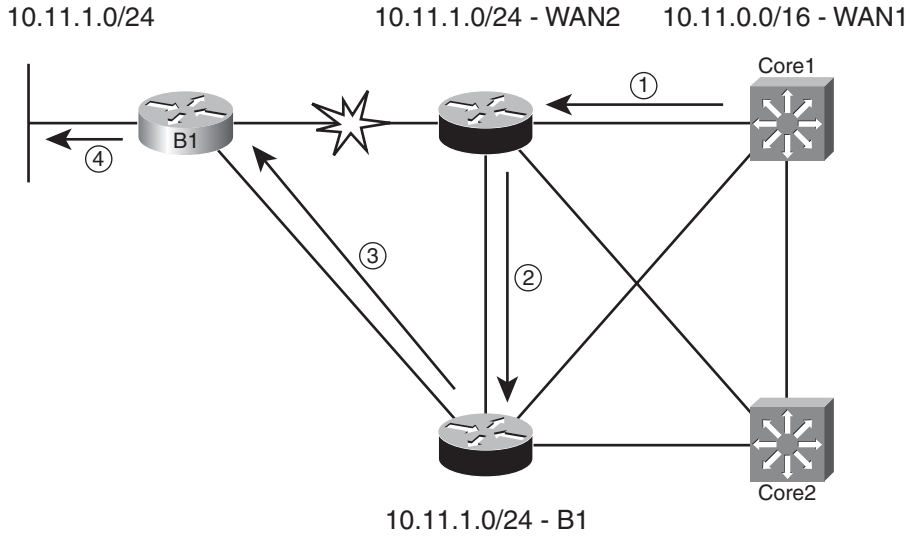


Figure 4-6 *Suboptimal Forwarding Path when Primary PVC Fails*

Following the steps in the figure:

Step 1. Core 1 sends a packet to 10.11.1.1, using its route for 10.16.0.0/16, to WAN1.

Step 2. WAN1, which has routes for all the subnets that begin 10.11, has a route for 10.11.1.0/24 with WAN2 as the next hop (because WAN1's link to B1 has failed).

Step 3. WAN2 has a route for 10.11.1.0/24, with B1 as the next hop, so WAN2 forwards the packet.

Step 4. B1 forwards the packet to host 10.11.1.1.

Route Summarization Benefits and Trade-Offs

The previous section showed details of a classic trade-off with route summarization: the benefits of the summary route versus the possibility of inefficient routing. For easier study, the benefits and trade-offs for route summarization are listed here:

Benefits:

- Smaller routing tables, while all destinations still reachable.
- Reduces Query scope: EIGRP Query stops at a router that has a summary route that includes the subnet listed in the Query but not the specific route listed in the Query.
- EIGRP supports summarization at any location in the internetwork.
- The summary has the metric of the best of the subnets being summarized.

Trade-offs:

- Can cause suboptimal routing.
- Packets destined for inaccessible destinations will flow to the summarizing router before being discarded.

Configuring EIGRP Route Summarization

The more difficult part of EIGRP route summarization relates to the planning, design, and analysis of trade-offs as covered in the preceding section. After you have made those design choices, configuring route summarization requires the addition of a few instances of the following interface subcommand:

```
ip summary-address eigrp asn prefix subnet-mask
```

When configured on an interface, the router changes its logic for the EIGRP Update messages sent out the interface, as follows:

- The router brings down, and then back up, all EIGRP neighbors reachable on that interface, effectively causing neighbors to forget previous topology information, and listen to new information (when the neighborships recover).
- When the neighborships recover, the router advertises the summary route, per the **ip summary-address** command, assuming the router has at least one route whose address range is inside the range of the summary route.
- The router does *not* advertise the subordinate routes. (The term *subordinate route refers to the routes whose address ranges are inside the range of addresses are defined by the summary route.)*

- The router adds a route to its own routing table, for the summary prefix/prefix-length, with an outgoing interface of null0.

In Figure 4-7, WAN1 and WAN2 summarize the routes for the Data Center in the range 10.16.0.0/16, instead of sending individual routes for this range to the branch offices. Example 4-5 shows the results of summarization on both routers.

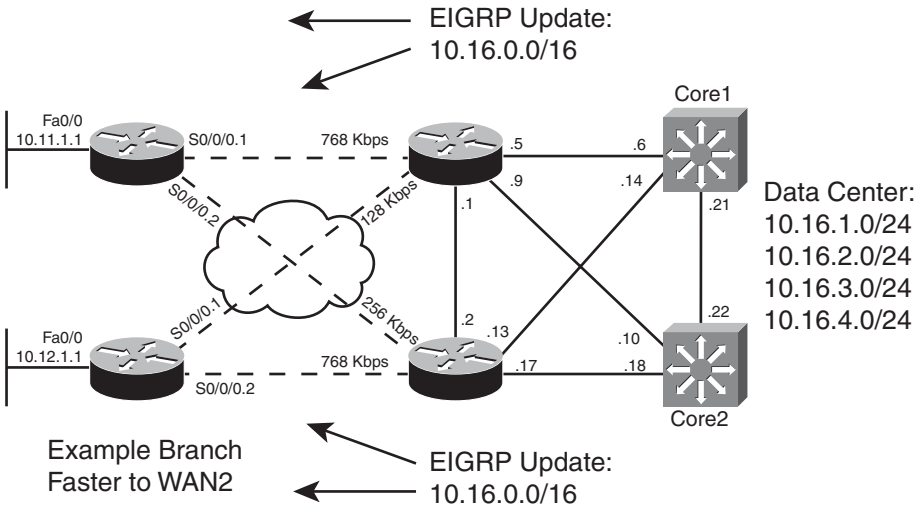


Figure 4-7 *Summary for 10.16.0.0/16 on WAN1, WAN2*

Example 4-5 *Summarizing Routes for Data Center (10.16.0.0/16) on WAN1/WAN2*

```
! On Router WAN2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WAN2#show running-config
! lines omitted for brevity
!
interface Serial0/0/0.1 point-to-point
 bandwidth 256
 ip address 10.1.2.1 255.255.255.252
 ip summary-address eigrp 1 10.16.0.0 255.255.0.0 5
 frame-relay interface-dlci 103
!
interface Serial0/0/0.2 point-to-point
 bandwidth 768
 ip address 10.1.2.5 255.255.255.252
 ip summary-address eigrp 1 10.16.0.0 255.255.0.0 5
 frame-relay interface-dlci 104
!
WAN2#show ip eigrp topology 10.16.0.0/16
IP-EIGRP (AS 1): Topology entry for 10.16.0.0/16
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 28416
```

```
  Routing Descriptor Blocks:
  0.0.0.0 (Null0), from 0.0.0.0, Send flag is 0x0
       Composite metric is (28416/0), Route is Internal
       Vector metric:
         Minimum bandwidth is 100000 Kbit
         Total delay is 110 microseconds
         Reliability is 255/255
         Load is 1/255
         Minimum MTU is 1500
         Hop count is 1
  10.1.2.2 (Serial0/0/0.1), from 10.1.2.2, Send flag is 0x0
       Composite metric is (11026688/3847936), Route is Internal
       Vector metric:
         Minimum bandwidth is 256 Kbit
         Total delay is 40110 microseconds
         Reliability is 255/255
         Load is 1/255
         Minimum MTU is 1500
         Hop count is 3

! Note that the following command lists only routes in the range
! of the summary - 10.16.0.0 - 10.16.255.255.
WAN2#show ip route 10.16.0.0 255.255.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 23 subnets, 6 masks
D       10.16.2.0/24 [90/156160] via 10.9.1.14, 00:19:06, FastEthernet0/0.12
D       10.16.3.0/24 [90/156160] via 10.9.1.14, 00:19:06, FastEthernet0/0.12
D       10.16.0.0/16 is a summary, 00:14:07, Null0
D       10.16.1.0/24 [90/28416] via 10.9.1.18, 00:19:06, FastEthernet0/1.16
                     [90/28416] via 10.9.1.14, 00:19:06, FastEthernet0/0.12
D       10.16.4.0/24 [90/156160] via 10.9.1.14, 00:19:06, FastEthernet0/0.12
WAN2#show ip route 10.16.0.0 255.255.0.0
Routing entry for 10.16.0.0/16
  Known via "eigrp 1", distance 5, metric 28416, type internal
  Redistributing via eigrp 1
```

```
Routing Descriptor Blocks:
* directly connected, via Null0
    Route metric is 28416, traffic share count is 1
    Total delay is 110 microseconds, minimum bandwidth is 100000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
   Loading 1/255, Hops 0
```

Example 4-5 shows the results only on Router WAN2, but WAN1 will be identically configured with the **ip summary-address** command. With only two branch office routers actually implemented in my lab, WAN2 needs only two **ip summary-address** commands: one for the subinterface connected to Router B1, and another for the subinterface connected to B2. With a full implementation, this same command would be needed on each subinterface connected to a branch router.

The example also shows how a router like WAN2 uses a summary route to null0. This route–10.16.0.0/16 with an outgoing interface of null0–causes the router (WAN2) to discard packets matched by this route. However, as you can see from the end of Example 4-5, WAN2 also has routes for all the known specific subnets. Pulling all these thoughts together, when the summarizing router receives a packet within the summary route's range

- If the packet matches a more specific route than the summary route, the packet is forwarded based on that route.
- When the packet does not match a more specific route, it matches the summary route and is discarded.

To ensure that the router adds this local summary route, the router uses the administrative distance (AD) setting of 5. The user may have typed the **ip summary-address eigrp 1 10.16.0.0 255.255.0.0** command, without the 5 at the end. Even so, IOS will add this default AD value as seen in Example 4-5. With an AD of 5, WAN2 will ignore any EIGRP-advertised summary routes for 10.16.0.0/16–for example, the summary created by neighbor WAN1—because EIGRP's default AD for internal routes is 90. In fact, the output of WAN2's **show ip eigrp topology 10.16.0.0/16** command lists two known routes for 10.16.0.0/16: one to null0 and the other to branch router WAN1 (outgoing interface S0/0/0.1). WAN1 uses the lower-AD route to null0, which prevents a routing loop. (Note that this summary route with outgoing interface null0 is often called a *discard route.)*

Next, consider the results on the branch routers. The following might be reasonable design requirements that should be verified on the branch routers:

- Each branch router's route for 10.16.0.0/16 should use the primary (faster) PVC (see Figure 4-7).
- Each branch router should be able to converge quickly to the other 10.16.0.0/16 summary route without using EIGRP Queries (in other words, there should be an FS route).

Example 4-6 confirms that both requirements are met.

Example 4-6 *Results of the 10.16.0.0/16 Summary on Routers B1, B2*

```
! Router B1 first !!!!!!!!!!!!!!!!!!!
B1#show ip route 10.16.0.0 255.255.0.0 longer-prefixes
! lines omitted for brevity

     10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
D       10.16.0.0/16 [90/3847936] via 10.1.1.1, 00:16:53, Serial0/0/0.1

B1#show ip eigrp topology
! lines omitted for brevity
P 10.16.0.0/16, 1 successors, FD is 3847936
         via 10.1.1.1 (3847936/28416), Serial0/0/0.1
         via 10.1.2.1 (10514688/28416), Serial0/0/0.2
```

```
! Router B2 Next !!!!!!!!!!!!!!!!!!!
B2#show ip route 10.16.0.0 255.255.0.0 longer-prefixes
! lines omitted for brevity

     10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
D       10.16.0.0/16 [90/3847936] via 10.1.2.5, 00:16:44, Serial0/0/0.2
```

First, on Router B1, the router has an IP route for 10.16.0.0/16, with outgoing interface S0/0/0.1. Per Figure 4-7, this subinterface indeed connects to the primary PVC. Per the **show ip eigrp topology** command, two possible routes for 10.16.0.0/16 are listed; this command only lists successor and feasible successor routes. Also, note that the FS route's RD (28,416) is less than the successor route's FD (3,847,936), which means the secondary route indeed meets the feasibility condition.

The reverse is true on Router B2. B2's best route for 10.16.0.0/16 uses its S0/0/0.2, which connects to B2's primary (faster) PVC through WAN2. Although not shown, it also lists its backup route over the slower PVC as a feasible successor.

The route summarization feature discussed in this section is sometimes referred to as *manual* route summarization to contrast it with the term *auto* summarization. EIGRP auto summarization is explained next.

Auto-summary

Automatic summarization, also called auto-summary, causes a router to automatically advertise a summary route under certain conditions, without the use of the **ip summary-address** command. When using auto-summary, if a router has interfaces in more than one Class A, B, or C network, then that router will advertise a single summary route for an entire Class A, B, or C network into the other classful network, rather than advertise routes for the individual subnets. The following is a more formal definition:

> When a router has multiple working interfaces, and those interfaces use IP addresses in different classful networks, the router advertises a summary route for each classful network on interfaces attached to a different classful network.

The auto-summary feature first existed as a required feature of *classful routing protocols*. By definition, classful routing protocols (RIPv1 and IGRP) do not advertise subnet mask information. The omission of the subnet mask in routing updates causes several design problems–in particular, these protocols cannot support variable length subnet masks (VLSM), route summarization, or discontiguous network designs.

The newer IGPs–EIGRP, OSPF, and RIP-2–are classless routing protocols because they advertise the subnet mask and support VLSM. However, with auto-summary enabled, EIGRP acts like classful routing protocols in one specific way: they do not support discontiguous networks. To support discontiguous networks with EIGRP, simply disable auto-summary. The rest of this section further defines the terms and the problem, and shows the solution of disabling auto-summary.

To better understand discontiguous networks, consider this analogy. U.S. residents can appreciate the concept of a discontiguous network based on the common term *contiguous 48*, referring to the 48 U.S. states other than Alaska and Hawaii. To drive to Alaska from the contiguous 48 U.S. states, for example, you must drive through another country (Canada, for the geographically impaired), so Alaska is not contiguous with the 48 states. In other words, it is discontiguous.

More formally:

- **Contiguous network:** A single classful network in which packets sent between every pair of subnets will pass only through subnets of that same classful network, without having to pass through subnets of any other classful network.
- **Discontiguous network:** A single classful network in which packets sent between at least one pair of subnets must pass through subnets of a different classful network.

Figure 4-8 shows a classic example of a discontiguous network 10.0.0.0. Subnets of class A network 10.0.0.0 exist on the left and the right, with subnets of class B network 172.16.0.0 in the middle of the internetwork. Following the figure, the problem created by the auto-summary feature is described.

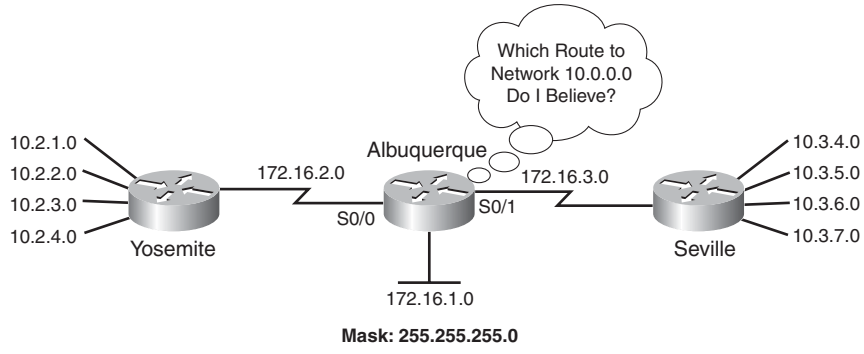


Figure 4-8 *Discontiguous Network 10.0.0.0*

The problem is that when EIGRP auto-summarizes routes at the boundary between classful networks, then routers in other classful networks cannot route packets to all the desti-

nations. For example, because both Yosemite and Seville use auto-summary, they both advertise a route for 10.0.0.0/8 to Albuquerque. Albuquerque may choose one of the two as the better route–for example, it may choose the route to the left, through Yosemite. However, in that case, then Albuquerque cannot forward packets to the network 10.0.0.0 hosts on the right. Even if Albuquerque decided to add both routes to its routing table, the load sharing typically occurs per destination IP address, not per subnet. So, some packets might be delivered to the correct host, and others not.

For EIGRP, two solutions exist. First, you could design the network to not use a discontiguous network. Alternatively, you can just disable auto-summary using the **no auto-summary** subcommand inside EIGRP configuration mode. This command affects the behavior of the router on which it is configured only and tells that router to not advertise a summary route for the entire classful network. Instead, that router advertises all the subnets, as if the auto-summary feature did not exist.

Note: The **auto-summary** and **no auto-summary** commands have no effect on routers that connect to a single classful network.

For classful routing protocols, the only solution is to not use discontiguous classful networks.

Note: Some confusion exists related to EIGRP's default for auto-summary. Some IOS documentation claims that EIGRP defaults to use **no auto-summary** at later IOS releases, including 12.4T, but experiments show the opposite. You can confirm the actual setting by looking at the output of the **show ip protocols** command.

Default Routes

A router's default route matches the destination of all packets that are not matched by any other route in the IP routing table. In fact, a default route can be thought of as the ultimate summary route–a route for the prefix that includes all IPv4 addresses, as represented by prefix/length 0.0.0.0/0.

This section first examines the most common use of default routes inside an Enterprise: to draw Internet traffic toward the Internet-connected routers without having to put routes for all Internet destinations into the Enterprise routers' routing tables. Following that, this section examines two methods for EIGRP to advertise the default route.

Default Routing to the Internet Router

Consider an Enterprise network and its connection to the Internet, as shown in Figure 4-9. For now, the design shows a single Internet-facing router (I1). As is often the case, the entire Enterprise in this figure uses private IP addresses. In this case, all Enterprise subnets are part of private class A network 10.0.0.0.

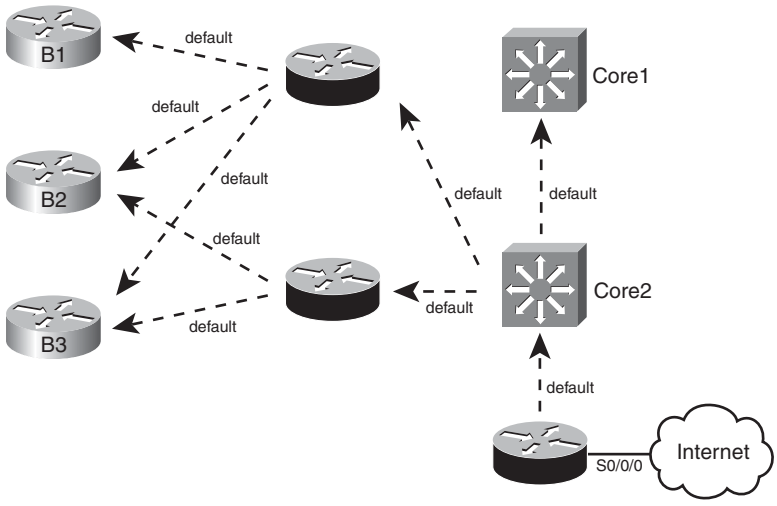


Figure 4-9 *Pulling Packets to the Internet Router (I1)*

From a design perspective, the entire Enterprise can use a default route to forward packets to the Internet. To accomplish this design, the Internet-facing router advertises a default route. All routers flood this default prefix throughout the EIGRP domain, building their own default routes.

When converged, all routers have a default route, plus the usual Enterprise routes. Packets destined for addresses inside the Enterprise use the same old routes, ignoring the default route. Packets destined outside the Enterpriseuse each router's respective default route because no other routes match the destination. Eventually, these packets arrive at Router I1. When I1 receives these packets, it can forward toward the Internet, either based on a default route or on routes learned using BGP.

Figure 4-9 shows a case with just one Internet-facing router, but with multiple, the same concepts can be used. The multiple Internet-facing routers can each advertise a default route, and each Enterprise router will think one of the available defaults is best–causing the packets to arrive at the nearest Internet access point.

Default Routing Configuration with EIGRP

This section examines the two main options for EIGRP to advertise default routes: to define a static default route and advertise it with EIGRP and to flag an existing route to be used also as a default route.

Advertising Static Default Routes with EIGRP

To cause the advertisement of the default routes shown in Figure 4-9, Router I1 can follow these steps:

Step 1. Create a static route default route using the **ip route 0.0.0.0 0.0.0.0 S0/0/0** command.

Key Topic

Step 2. Inject this route into the EIGRP topology database, either using the **network 0.0.0.0** command or by redistributing the static route.

First, examine the command listed for Step 1: **ip route 0.0.0.0 0.0.0.0 S0/0/0**. The prefix and mask together represent all IPv4 addresses. The reasoning is that if a mask of 255.255.0.0 means "the last two octets can be any value," and 255.0.0.0 means "the last three octets can be any value," then a subnet mask of 0.0.0.0 means that all four octets can be any value. The outgoing interface, S0/0/0 in this case, tells I1 to send packets for otherwise unknown destinations over the link to the Internet, as intended.

After Step 1, Router I1 has a route in its routing table, but EIGRP does not yet advertise the route. I1 could be configured to perform route redistribution for this static route. (Refer to Chapter 9 for more information on route redistribution.) The other option is to use the **network 0.0.0.0** EIGRP subcommand. Oddly enough, this command does not actually match interface IP addresses of interfaces, but is a special case in which IOS thinks "if my routing table has a default route in it, put a default route (0.0.0.0/0) into the EIGRP table." (If the route leaves the routing table, then the router will notify neighbors that the route has failed.)

Configuring a Default Network

The second option for creating a default route is to flag a route for a classful network–for a prefix that will be advertised into the EIGRP domain–as a route that can be used as a default route. Then each router can use the forwarding details in that route–the outgoing interface and next-hop router–as its default route.

Configuring this feature requires a couple of steps. The concepts require the most thought, with the configuration commands that follow being relatively simple:

Key Topic

Step 1. On the router to which all traffic should be directed, identify a classful network that can be advertised into the EIGRP domain, and ensure that network is being advertised into EIGRP (typically using the EIGRP **network** command).

Step 2. Configure that network as a default network using the global command **ip default-network** *network-number*.

Step 1 requires a class A, B, or C network, known in the routing table of the router that will generate the default route (Router I1 in Figure 4-9). Most often, that route is either created off a loopback interface for the purpose of making this process work, or an existing route on the Internet side of the router is used.

Figure 4-10 shows two examples. First, class C network 198.133.219.0/24 exists off I1's S0/0/0 interface, so I1 has a connected route for this class C network in its routing table. Alternatively, the engineer could configure a loopback interface, such as loopback 9, so that I1 would have a connected route for 192.31.7.0/24. In both cases, the routes would need to be advertised into EIGRP, by matching the address using the **network** command.

If the configuration stopped at Step 1, then the Enterprise routers simply know yet another route. By adding the **ip default-network** command to refer to one of these networks, EIGRP then flags this route as a candidate default route. As a result, each EIGRP router treats their route for this particular network also as if it were a default route.

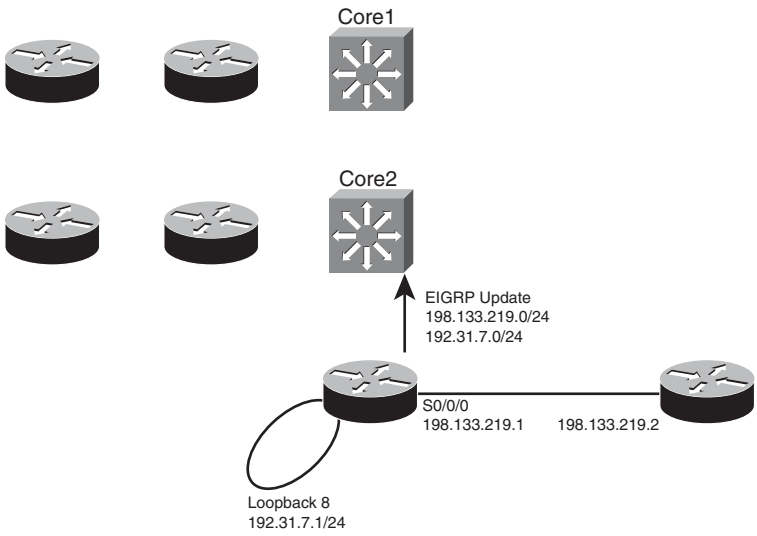


Figure 4-10 *Example Default Networks*

Example 4-7 shows an example of the configuration on Router I1, along with some of the **show** commands on Router I1.

Example 4-7 *Configuring a Default Network on Router I1*

```
I1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
I1(config)#interface loopback 8
I1(config-if)#ip address 192.31.7.1 255.255.255.0
I1(config-if)#router eigrp 1
I1(config-router)#network 192.31.7.0
I1(config-router)#exit
I1(config)#ip default-network 192.31.7.0
I1(config-router)#^Z

I1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
```

```
! lines omitted for brevity
C*   192.31.7.0/24 is directly connected, Loopback8

I1#show ip eigrp topology 192.31.7.0/24
IP-EIGRP (AS 1): Topology entry for 192.31.7.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
  0.0.0.0 (Loopback8), from Connected, Send flag is 0x0
      Composite metric is (128256/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 5000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1514
        Hop count is 0
      Exterior flag is set
```

The configuration has several results, as seen in the example:

- A connected route for 192.31.7.0/24, a class C network
- The advertisement of that network into EIGRP due to the **network 192.31.7.0** command
- The setting of the exterior flag on the route

Because of the **ip default-network 192.31.7.0** command, the routing table lists the route as a candidate default route, as denoted by an asterisk.

Interestingly, the router with the **ip default-network** command configured–I1 in this case–does not use that route as a default route, as indicated by the highlighted phrase "Gateway of last resort not set." (*Gateway of last resort* refers to the next-hop router of a router's current default route.) Although I1 flags the route as a candidate default route, I1 itself does not use that route as its default, because I1 is actually the original advertiser of the default.

Moving on to another Enterprise router, in this case B1, you can see in Example 4-8 that not only does the remote router learn the candidate default route, but that the B1 uses this same information as B1's default route.

Example 4-8 *Gateway of Last Resort on Router B1*

```
B1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 192.31.7.0


     10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
Lines omitted for brevity
D*   192.31.7.0/24 [90/2297856] via 10.1.1.1, 00:05:10, Serial0/0/0.1
```

In this case, B1 has indeed learned an EIGRP route for 192.31.7.0/24, a route flagged as exterior. Because this happens to be the only candidate default route learned by B1 at this point, it is the best default route. So, B1 sets its gateway of last resort to 10.1.1.1–the next-hop IP address of B1's route to 192.31.7.0/24. If B1 knew of multiple candidate default routes, it would have chosen the best route based on administrative distance and then metric, and used that route as the basis for the gateway of last resort.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 4-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 4-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Prevent the edge routers in sites for one division of the company from knowing routes for subnets in another division.	
The design shows the use of class B networks 172.16.0.0, 172.17.0.0, and 172.18.0.0 throughout the Enterprise, with a goal to reduce routing table sizes when possible. (3 options)	
R1 and R2 will advertise the same summary route; ensure that R1 is the preferred EIGRP path for that summary.	
Always ensure that the shortest path is taken with each route.	

Implementation Plan Peer Review Table

Table 4-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 4-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
The plan calls for filtering 10.10.10.0/26 and 10.10.12.0/26, but not 10.10.11.0/24. What tools can be used?	
A plan lists a configuration of an EIGRP distribution list, referring to route-map *one*. The route-maps clauses use only the deny option. However, it refers to prefix lists that use some deny and permit actions. Can any routes pass through the filter?	
A plan lists a configuration of an EIGRP distribution list, with a route-map with two clauses, each with a permit action. Both clauses refer to a different prefix list, each of which has some permit and deny actions. Characterize which routes will be filtered, and which will not, based on matching deny and permit clauses in each prefix list.	
The plan shows extensive use of class C private networks inside a large Enterprise. What effect might EIGRP auto-summary have?	
The plan shows a sample configuration of the **ip summary-address eigrp 1 10.10.0.0 255.255.252.0** command on Router R1. What routes should I see on R1? What will their administrative distance be?	

Create an Implementation Plan Table

To practice skills useful when creating your own EIGRP implementation plan, list in Table 4-7 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 4-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Command/Notes
Filtering EIGRP routes using numbered ACLs	
Filtering EIGRP routes using prefix lists	
Enabling filtering EIGRP routes using route-maps	
Commands to create a route-map clause, and match based on a standard numbered ACL, a standard named ACL, and a prefix-list	
Configuring a summary route	
Enable/disable auto-summary	
Configure a default route using ip default-network	
Configure a default route using static routes	

Choose Commands for a Verification Plan Table

To practice skills that are useful when creating your own EIGRP verification plan, list in Table 4-8 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 4-8 *Verification Plan Memory Drill*

Information Needed	Command
Display access lists.	
Display prefix lists.	
Display route maps.	
Verify whether a filter worked by displaying all known routes in a range of addresses.	
Display a summary IP route.	
On summarizing router, display EIGRP topology info on a summary route.	
On summarizing router, display IP routes for a summary route and it subordinate routes.	
On summarizing router, display the administrative distance of the null route.	
Display the current auto-summary setting.	
Determine if a prefix in the EIGRP topology table has been flagged as a candidate default route.	
Determine if an IP route has been flagged as a candidate default route.	
Display a router's preferred default route.	

Review all the Key Topics

Review the most important topics from this chapter, noted with the key topics icon in the outer margin of the page. Table 4-9 lists a reference of these key topics and the page numbers on which each is found.

Table 4-9 *Key Topics for Chapter 4*

Key Topic Element	Description	Page Number
List	Summary of matching logic for prefix lists	106
Table 4-3	Summary of comparisons of prefix length for IP prefix lists	106
List	Summary of matching logic for route maps	112
List	Benefits and negatives regarding the use of route summarization	120
List	A summary of what occurs when configuring an EIGRP summary route	120
Definition	auto-summary	124
List	Steps to advertise static default routes	127
List	Steps to configure a default network	128

Complete the Tables and Lists from Memory

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

Define Key Terms

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

Prefix list, route map, distribute list, address block, subordinate route, auto summary, default network, static default route, gateway of last resort

This chapter covers the following subjects:

OSPF Review: This section reviews the OSPF concepts, configuration, and verification commands assumed as prerequisites, specifically those details included in the CCNA Exam's coverage of OSPF.

OSPF Neighbors and Adjacencies on LANs: This section discusses a variety of features that impact when a router attempts to form OSPF neighbor relationships (neighborships), what must be true for those neighborships to work, and what might prevent those neighborships.

OSPF Neighbors and Adjacencies on WANs: This short section examines the typical usage of OSPF neighborships over various types of WAN technologies.

CHAPTER 5

OSPF Overview and Neighbor Relationships

Open Shortest Path First (OSPF) requires only a few relatively simple commands when using it in a small- to medium-sized internetwork. However, behind those commands resides a fairly complex routing protocol, with internals that can intimidate those new to OSPF. When compared to the less-complex EIGRP, OSPF requires more thought when planning and a few more configuration commands than does EIGRP. Additionally, the underlying complexity of OSPF makes operating and verifying an OSPF internetwork more challenging.

Part III of this book contains four chapters. This chapter briefly reviews the basics of OSPF and delves into all topics related to how OSPF routers form neighbor relationships. Chapter 6, "OSPF Topology, Routes, and Convergence," then examines how OSPF exchanges topology data, as stored in its Link State Database (LSDB), for internal OSPF routes. Chapter 6 also discusses how OSPF then chooses the best internal OSPF routes. Chapter 7, "OSPF Route Summarization, Filtering, and Default Routing," moves examine several tools that optimize the operation of OSPF, including route filtering, route summarization, and special OSPF area types. Finally, Chapter 8, "OSPF Virtual Links and Frame Relay Operations," discusses a few miscellaneous topics.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 5-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz."

Table 5-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Question
OSPF Review	1–3
OSPF Neighbors and Adjacencies on LANs	4–7
OSPF Neighbors and Adjacencies on WANs	8

1. A router has been configured with the commands **router ospf 9**, **network 172.16.1.0 0.0.0.255 area 8**, and **network 172.16.0.0 0.0.255.255 area 9**, in that order. No other OSPF-related commands have been configured. The answers list the IP addresses that could be assigned to this router's Fa0/0 interface. Which answers list an IP address/prefix length that would cause the router to put Fa0/0 into area 9? (Choose two.)

   a. 172.16.0.1/23
   b. 172.16.1.1/26
   c. 172.16.1.1/24
   d. 172.16.0.255/23
   e. None of the other answers is correct.

2. Which of the following is true about an OSPF area border router (ABR)?

   a. The ABR must have multiple interfaces connected to the backbone area.
   b. An ABR is a router with two interfaces, each connected to a different nonbackbone area.
   c. The only requirement to be considered an ABR is at least one interface connected to the backbone area.
   d. An ABR must have at least one interface in the backbone area plus at least one other interface in a nonbackbone area.

3. Which of the following can either directly or indirectly identify all the interfaces for which both 1) OSPF has been enabled and 2) OSPF is not passive? (Choose two.)

   a. **show ip ospf database**
   b. **show ip ospf interface brief**
   c. **show ip protocols**
   d. **show ip route ospf**
   e. **show ip ospf neighbors**

4. Router R1 directly connects to subnet 10.1.1.0/24 with its Fa0/0 interface. R1 can ping four other working OSPF routers in that subnet. R1 is neither the designated router (DR) nor backup DR (BDR). OSPF is working correctly on all five routers. Which of the following is true on R1? (Choose two.)

   a. The **show ip ospf neighbors** command lists two neighbors off Fa0/0.
   b. The **show ip ospf neighbors** command lists four neighbors off Fa0/0.
   c. The **show ip ospf neighbors** command lists two neighbors off Fa0/0 in the FULL state.
   d. The **show ip ospf neighbors** command lists two neighbors off Fa0/0 in the DISCO state.

5. Routers R1 and R2 are OSPF neighbors using their Fa0/0 interfaces, respectively, using default settings for all timers. An engineer adds the **ip ospf hello-interval 6** command to R1's Fa0/0 configuration. Which of the following is true regarding the results from this change? (Choose 2)

   a. The **show ip ospf neighbor** command on R1 lists the revised Hello timer.
   b. The **show ip ospf interface brief** command on R1 lists the revised Hello timer.
   c. The R1-R2 neighborship fails due to Hello timer mismatch.
   d. The **show ip ospf interface** command on R1 lists the revised Hello timer.

6. Routers R1 and R2, OSPF neighbors in area 0 over their Fa0/0 interfaces (respectively), currently both successfully use OSPF MD5 authentication. The OSPF configuration includes the **area 0 authentication** command under the **router ospf 1** command. Which of the following commands must have been configured on R1's Fa0/0 interface? (Choose two.)

   a. **ip ospf authentication null**
   b. **ip ospf authentication message-digest**
   c. **ip ospf authentication-key whatever-it-is**
   d. **ip ospf message-digest-key 1 md5 whatever-it-is**
   e. **ip ospf md5 1 key whatever-it-is**

7. Which of the following settings do not prevent two potential OSPF neighbors from becoming neighbors?

   a. The interface used to connect to that neighbor being passive in the OSPF process
   b. Duplicate OSPF router IDs
   c. Mismatched dead timers
   d. IP addresses of 10.1.1.1/24 and 10.2.2.2/24
   e. Mismatched OSPF process IDs

8. A company has a Frame Relay WAN with one central-site router and 100 branch office routers. A partial mesh of PVCs exists: one PVC between the central site and each of the 100 branch routers. All routers use point-to-point subinterfaces and one subnet per PVC. Which of the following is true about OSPF in this design?

   a. The central site router has 100 fully adjacent neighborships with the 100 branches.
   b. The central site router has neighborships with all branch routers, but fully adjacent neighbors with only two branches.
   c. The central site router has a neighborship with the Frame Relay switch.
   d. None of the other answers is correct.

Foundation Topics

OSPF Review

All the CCNP exams consider CCNA materials as prerequisites, so the Cisco Press CCNP Exam Certification Guide series of books also assumes the reader is already familiar with CCNA topics. However, the CCNP exams do include features that overlap with CCNA. Additionally, most people forget some details about CCNA topics along the way. This section is intended as a quick reminder of the basics from your earlier CCNA studies related to OSPF, with the addition of a few related details you may not have seen suring your CCNA study.

Note that this section does not cover every detail of CCNA-level OSPF topics–the main goal is a quick refamiliarization. Following this review, throughout this and the next three chapters, the rest of the CCNA-level OSPF features, plus many new OSPF-features, will be detailed.

To that end, this section begins with a review of OSPF terminology and link state theory, followed by a configuration and verification sample.

OSPF Link State Concepts

OPSF uses link state (LS) logic, which can be broken into three major branches. The first step, neighbor discovery, has the same overall goal as EIGRP's neighbor discovery process: to find the neighboring routers, and exchange enough information so that the two routers know whether they should exchange topology data. (Like EIGRP, OSPF keeps a list of neighbors in its neighbor table.)

The second step, topology database exchange, requires each OSPF router to cooperate by sending messages so that all routers learn topology information–information that is the equivalent of the kinds of information a human would draw and write in a diagram of the internetwork. Each router stores this topology information in its topology database, sometimes called its link state database (LSDB). The information communicated by OSPF routers and held in their LSDBs includes:

- The existence of, and an identifier for, each router (router ID)
- Each router interface, IP address, mask, and subnet
- The list of routers reachable by each router on each interface

The third major step, route computation, means that each router independently analyzes the topology data to choose the best routes from their perspective. In particular, LS algorithms such as OSPF use a Shortest Path First (SPF) algorithm to analyze the data, choose the shortest (best) route for each reachable subnet, and add the correct next-hop/outgoing interface information for those routes to the IP routing table.

OSPF requires more planning than does EIGRP, particularly with regard to the necessity for a hierarchical design using OSPF areas. Each router interface exists in a single area, with some special routers, called area border routers (ABR), being the boundary between

areas. Inside an area, routers exchange detailed topology information. However, the detailed topology information does not flow between areas. Instead, the ABRs advertise briefer information between areas, including information about subnets/masks, but the information advertised into one area does not include details about the topology of the other area. For perspective on the OSPF design issues, consider Figure 5-1, which shows a typical hierarchical design.

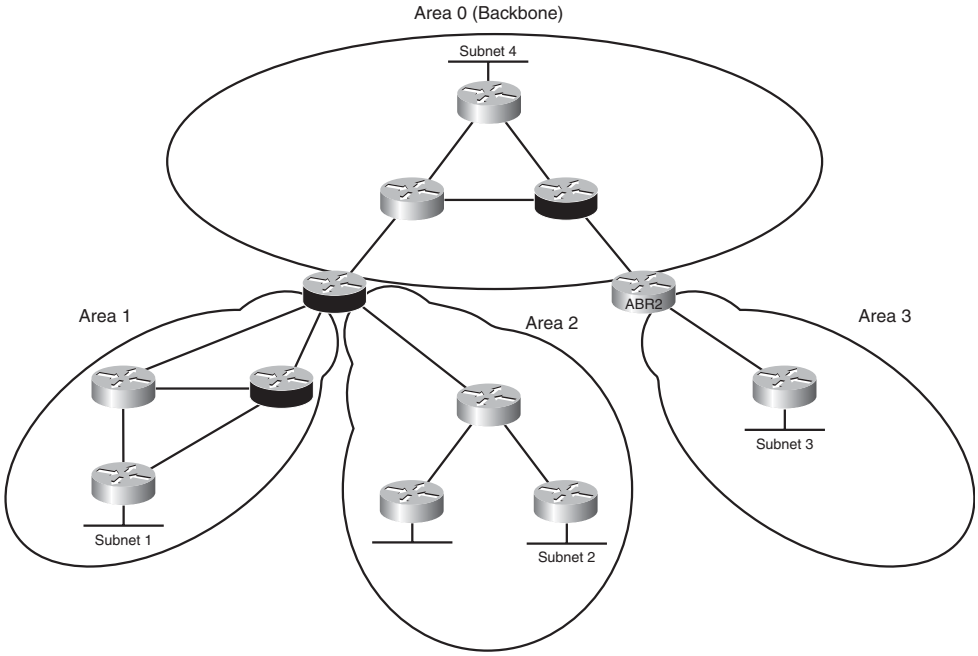


Figure 5-1 *Typical Hierarchical OSPF Design*

One area, called the backbone area, must connect to all other areas. Packets that need to pass between two nonbackbone areas must pass through (at least) one backbone router. The ABRs must keep a copy of the LSDB for each area to which they attach; for example, ABR1 has LSDBs for area 0, area 1, and area 2. However, the ABRs do not forward all the topology details between areas; instead, they simply advertise the subnets (prefix/length) between the areas.

Because of the sparse information advertised into one area about another area, topologically, routers inside one area know only about the subnets in another area. They do not know about the details of the topology in the other area; instead, from a topology perspective, it appears as if the subnets from another area connect to the ABR. Figure 5-2 shows the concept with the two routers in area 3 from Figure 5-1.

Figure 5-2 essentially shows the contents of area 3's LSDB in graphical form. Two routers exist, with a link between them, and one LAN subnet (Subnet 3) internal to the area. However, the other three sample subnets shown in Figure 5-1 (Subnets 1, 2, and 3) appear

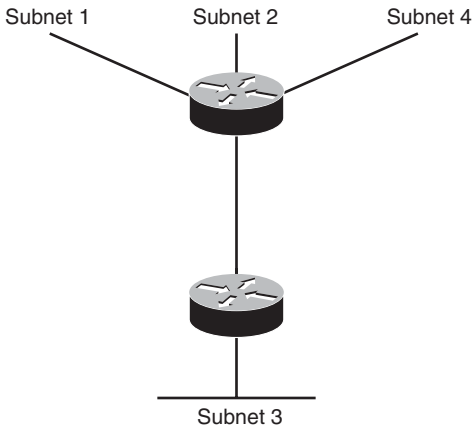


Figure 5-2 *Area 3 LSDB Concept*

connected to ABR2. (Other subnets exist outside area 3 as well; the figure just shows a few as examples.) The routers inside area 3 can calculate and add routes to their routing tables, but without needing all the topology shown in Figure 5-1. By using areas as a design as in Figure 5-1, network engineers can group routers and interfaces into areas, which results of smaller topology databases on those routers, as shown in Figure 5-2. As a result, each router reduces the processing time, memory consumption, and effort to calculate the best routes.

OSPF does have a fairly large number of terms. Table 5-2 lists some of the more common OSPF terms as an early reference as you read through the chapter.

Key Topic

Table 5-2 *Commonly Used OSPF Terms*

Term	Definition
Link state database	The data structure held by an OSPF router for the purpose of storing topology data.
Shortest Path First (SPF)	The name of the algorithm OSPF uses to analyze the LSDB. The analysis determines the best (lowest cost) route for each prefix/length.
Link State Update (LSU)	The name of the OSPF packet that holds the detailed topology information, specifically LSAs

Table 5-2 *Commonly Used OSPF Terms*

Term	Definition
Link State Advertisement (LSA)	The name of a class of OSPF data structures that hold topology information. LSAs are held in memory in the LSDB and communicated over the network in LSU messages.
Area	A contiguous grouping of routers and router interfaces. Routers in an area strive to learn all topology information about the area, but they do not learn topology information about areas to which they do not connect.
Area Border Router (ABR)	A router that has interfaces connected to at least two different OSPF areas, including the backbone area. ABRs hold topology data for each area, and calculate routes for each area, and advertise about those routes between areas.
Backbone router	Any router that has at least one interface connected to the backbone area.
Internal routers	A router that has interfaces connected to only one area, making the router completely internal to that one area.
Designated Router (DR)	On multiaccess data links like LANs, an OSPF router elected by the routers on that data link to perform special functions. These functions include the generation of LSAs representing the subnet, and playing a key role in the database exchange process.
Backup Designated Router (BDR)	A router on a multiaccess data link that monitors the DR and becomes prepared to take over for the DR, should the DR fail.

OSPF Configuration Review

Other than the configuration of the OSPF areas, the configuration of OSPF basics looks similar to a simple EIGRP configuration. IOS uses the **router ospf** *process-id* command, plus one or more **network** *net-id wildcard-mask* **area** *area-id* subcommands, to enable OSPF on the router and on router interfaces. The rules for these commands are as follows:

Key Topic

Step 1. Neighboring routers' **router ospf** *process-id* commands do not have to be configured with the same *process-id* parameter to become neighbors.

Step 2. IOS only enables OSPF on interfaces matched by an OSPF **network** command. When enabled, the router does the following:

Attempts to discover OSPF neighbors on that interface by sending multicast OSPF Hello messages

Includes the connected subnet in future topology database exchanges

Step 3. To match an interface with the **network** command, IOS compares the *net-id* configured in the **network** command with each interface's IP address, while using the configured wildcard-mask as an ACL wildcard mask.

Step 4. Regardless of the order in which the **network** commands are added to the configuration, IOS puts these commands into the configuration file with the most specific (most binary 0s) wildcard mask first. IOS lists the **network** commands in this sorted order in the configuration.

Step 5. The first **network** command that matches an interface, per the order shown in the output of the **show running-config** command, determines the OSPF area number associated with the interface.

Example 5-1 shows a sample configuration for each router in Figure 5-3.

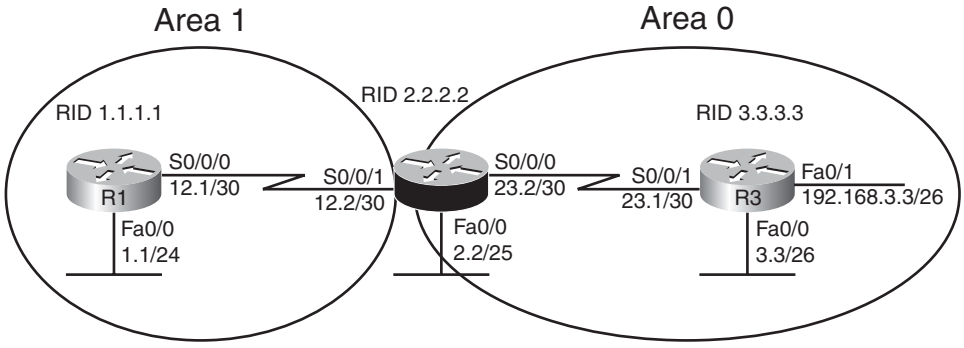


Figure 5-3 *Three Router Internetwork with Two OSPF Areas*

Example 5-1 *OSPF Configuration on Routers R1, R2, and R3*

```
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
interface loopback 1
 ip address 1.1.1.1 255.255.255.255
router ospf 1
 network 10.0.0.0 0.255.255.255 area 1
```

```
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
interface loopback 1
 ip address 2.2.2.2 255.255.255.255

router ospf 2
 network 10.1.12.2 0.0.0.0 area 1
 network 10.1.0.0 0.0.255.255 area 0
```

```
! On Router R3: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
interface loopback 1
 ip address 3.3.3.3 255.255.255.255

router ospf 3
 network 10.1.0.0 0.0.255.255 area 0
 network 192.168.3.3 0.0.0.0 area 0
```

First, note that all three routers use a different process ID on their respective **router ospf** *process-id* commands; these mismatches do not prevent neighborship.

Next, consider the requirement that R1's S0/0/0 and R2's S0/0/1 must be in the same area. Typically, all routers on the same subnet need to be in the same area; the routers themselves are the boundary between areas. In this case, R1's **network 10.0.0.0 0.255.255.255 area 1** command matches all interfaces whose addresses begin with 10 in the first octet and assigns those interfaces (Fa0/0 and S0/0/0) to area 1. Similarly, R2's **network 10.1.12.2 0.0.0.0 area 1** command matches only one IP address–R2's S0/0/1 IP address–and places it in area 1. Looking further at R2's OSPF configuration, note that both **network** commands actually match the 10.1.12.2 S0/0/1 IP address: one with area 0, and one with area 1. However, R2 orders these two **network** commands with the most-specific wildcard mask first, placing the command with wildcard mask 0.0.0.0 first, and the one with wildcard 0.0.255.255 second. Then, R2 compares the commands to the interface IP addresses in order, so R2 places S0/0/1 into area 1. (Note that in real internetworks, choosing wildcard masks such that it is clear which **network** command should match each interface is the better choice.)

On R3, the **network 10.1.0.0 0.0.255.255 area 0** command matches interfaces Fa0/0 and S0/0/0, adding them to area 0. R3 then needs an additional **network** command to enable OSPF on R3's Fa0/1 interface with all three interfaces in area 0.

Finally, note that the addition of the loopback interfaces causes each router to choose an obvious OSPF router ID (RID). OSPF uses the same logic as does EIGRP to choose a

router ID on each router, at the time the OSPF process is initialized, as follows, in the listed order of precedence:

Key Topic

Step 1. Use the router ID defined in the **router-id** *x.x.x.x* OSPF router subcommand.

Step 2. Use the highest IP address of any up/up loopback interface.

Step 3. Use the highest IP address of any up/up non-loopback interface.

Note that for the second and third choices, the interface does not need to have OSPF enabled.

OSPF Verification Review

The verification process, whether it uses a formal verification plan, must have some knowledge of the intended design and function of the network. The design and implementation documents dictate what the network should do, and the verification plan should confirm whether the network is meeting those goals.

For the purposes of this OSPF review section, assume that the only design goal for the internetwork in Figure 5-3 is that OSPF be used so that all routers have routes to reach all subnets shown in the figure, within the constraints of the area design.

To verify such a simple design, an engineer should start by confirming on which interfaces OSPF has been enabled on each router. The next step should be to determine if the OSPF neighbor relationships that should occur are indeed up and working. Then, the OSPF topology table should be examined to confirm that non-ABRs have only topology information for their respective areas. Finally, the IP routes on each router should be examined, confirming that all routes are known. To that end, Table 5-3 summarizes five key **show** commands that provide the information to answer these questions:

Key Topic

Table 5-3 *Most Commonly Used OSPF show Commands*

Command	Key Information
show ip ospf interface brief	Lists the interfaces on which OSPF is enabled (based on the **network** commands); it omits passive interfaces.
show ip protocols	Lists the contents of the **network** configuration commands for each routing process, and a list of enabled but passive interfaces.
show ip ospf neighbors	Lists known neighbors, including neighbor state; does not list neighbors for which some mismatched parameter is preventing a valid OSPF neighbor relationship.
show ip ospf database	Lists all LSAs for all connected areas. (See Chapter 6 for more detail on the LSA types seen in the database.)
show ip route	Lists the contents of the IP routing table, listing OSPF-learned routes with a code of O on the left side of the output.

Example 5-2 shows samples of each command listed in Table 5-3. Note that the output highlights various samples of items that should be verified, including the interfaces on which OSPF is enabled, the known neighbors, the neighbors' state, the LSAs in the topology table, and the OSPF routes.

Example 5-2 *OSPF Verification on Routers R1, R2, and R3*

```
! On Router R2: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Note that S0/0/1 is shown as in area 1, while the other 3 interfaces are all
in
! Area 0.


R2#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Se0/0/0      2     0               10.1.23.2/30       64    P2P   1/1
Fa0/0        2     0               10.1.2.2/25        1     DR    0/0
Se0/0/1      2     1               10.1.12.2/30       64    P2P   1/1

! Next, note that R2 lists two "Routing Information Sources", 1.1.1.1 (R1) and
! 3.3.3.3 (R3). These routers, listed by RID, should mirror those listed
! in the output of the show ip ospf neighbors command that follows.

R2#show ip protocols
Routing Protocol is "ospf 2"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.2.2.2
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.12.2 0.0.0.0 area 1
    10.1.0.0 0.0.255.255 area 0

 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    3.3.3.3              110      00:01:08
    1.1.1.1              110      00:01:08
  Distance: (default is 110)

! Note that the Full state means that the database exchange process is
```

```
! fully completed between these two neighbors.
R2#show ip ospf neighbors

Neighbor ID     Pri   State           Dead Time   Address         Interface
3.3.3.3           0   FULL/  -        00:00:34    10.1.23.1       Serial0/0/0
1.1.1.1           0   FULL/  -        00:00:34    10.1.12.1       Serial0/0/1
! On Router R1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Note that R1's LSDB includes a "Router Link State" for RID 1.1.1.1 (R1)
! and R2 (2.2.2.2), but not 3.3.3.3 (R3), because R3 is not attached  to area
1.

R1#show ip ospf database

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         210         0x80000004 0x001533 3
2.2.2.2         2.2.2.2         195         0x80000002 0x0085DB 2

                Summary Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.2.0        2.2.2.2         190         0x80000001 0x00B5F0
10.1.3.0        2.2.2.2         190         0x80000001 0x00AE76
10.1.23.0       2.2.2.2         190         0x80000001 0x0031A4
192.168.3.0     2.2.2.2         191         0x80000001 0x008B3B

! Below, note that R1 has routes for all remote subnets, including R3's
! LAN subnets, even though R1 does not list R3 in its LSDB.


R1#show ip route ospf
     10.0.0.0/8 is variably subnetted, 5 subnets, 4 masks
O IA    10.1.3.0/26 [110/129] via 10.1.12.2, 00:04:13, Serial0/0/0
O IA    10.1.2.0/25 [110/65] via 10.1.12.2, 00:04:13, Serial0/0/0
O IA    10.1.23.0/30 [110/128] via 10.1.12.2, 00:04:13, Serial0/0/0
     192.168.3.0/26 is subnetted, 1 subnets
O IA    192.168.3.0 [110/129] via 10.1.12.2, 00:04:13, Serial0/0/0
```

OSPF Feature Summary

Table 5-4 summarizes some of the key facts about OSPF. The table includes some review items from the CCNA level OSPF topics, plus some topics that will be developed in chapters 5 through 8. The items that are not CCNA topics are included just for convenience when reviewing for final preparation before taking the exam.

Table 5-4 *OSPF Feature Summary*

Feature	Description
Transport	IP, protocol type 89 (does not use UDP or TCP).
Metric	Based on cumulative cost of all outgoing interfaces in a route. The interface cost defaults to a function of interface bandwidth but can be set explicitly.
Hello interval	Interval at which a router sends OSPF Hello messages on an interface.
Dead interval	Timer used to determine when a neighboring router has failed, based on a router not receiving any OSPF messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.5 (All SPF Routers) and 225.0.0.6 (All Designated Routers).
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used.
Authentication	Supports MD5 and clear-text authentication.
VLSM/classless	OSPF includes the mask with each route, also allowing it to support discontiguous networks and VLSM.
Route Tags	Allows OSPF to tag routes as they are redistributed into OSPF.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Manual route summarization	Allows route summarization at ABR routers only.

This concludes the review of OSPF topics. The rest of this chapter focuses on OSPF topics related to the formation of OSPF neighbor relationships.

OSPF Neighbors and Adjacencies on LANs

With EIGRP, neighborship is relatively simple. If two EIGRP routers discover each other (using Hellos) and meet several requirements (like being in the same subnet), the two routers become neighbors. After becoming neighbors, the two EIGRP routers exchange topology information.

Comparing OSPF and EIGRP, OSPF neighborship is more complex. First, with EIGRP, two routers either become neighbors or they do not. With OSPF, even after all the neighbor parameter checks pass, two classes of neighborships exist: neighbors and fully adjacent neighbors. The OSPF neighbor discovery process has many pitfalls when the internetwork uses Frame Relay, with a class of issues that simply do not exist with EIGRP. Finally, OSPF uses an underlying Finite State Machine (FSM) with eight neighbor states used to describe the current state of each OSPF neighbor, adding another layer of complexity compared to EIGRP.

This section breaks down the OSPF neighbor relationship, the logic, and the OSPF configuration settings–anything that impacts OSPF neighborship on LAN interfaces. In particular, this section examines the following questions:

- On what interfaces will this router attempt to discover neighbors by sending multicast OSPF Hello messages?
- When a potential neighbor is discovered, do they meet all requirements to become neighbors at all?

This section examines these topics, in sequence.

Enabling OSPF Neighbor Discovery on LANs

OSPF sends multicast OSPF Hello messages on LAN interfaces, attempting to discover OSPF neighbors, when two requirements are met:

- OSPF has been enabled on the interface, either through the **network** router subcommand or the **ip ospf area** interface subcommand.
- The interface has not been made passive by the **passive-interface** router subcommand.

When both requirements are met, OSPF sends Hellos to the 224.0.0.5 multicast address, an address reserved for all OSPF-speaking routers. The Hello itself contains several parameters that must be checked, including the OSPF RID of the router sending the Hello, and the OSPF area that router has assigned to that LAN subnet.

Of the three configuration commands that might impact whether a router attempts to discover potential neighbors on an interface, one is commonly understood (**network**) and was already covered in this chapter's "OSPF Configuration Review" section. The second configuration command that impacts whether potential neighbors discover each other, **passive-interface**, works just like it does with EIGRP. In short, when a router configures an interface as passive to OSPF, OSPF quits sending OSPF Hellos, so the router will not discover neighbors. The router will still advertise about the interface's connected subnet if OSPF is enabled on the interface, but all other OSPF processing on the interface is stopped.

The third configuration command that impacts whether a router discovers potential neighbors using Hellos is the **ip ospf** *process-id* **area** *area-id* interface subcommand. This command acts as a replacement for the OSPF **network** command. Simply put, this command enables OSPF directly on the interface and assigns the area number.

To demonstrate the **ip ospf area** and **passive-interface** commands, Example 5-3 shows a revised configuration on Router R3 as seen originally back in Example 5-1. In this new example configuration, R3 has made two interfaces passive, because no other OSPF routers exist on its LAN subnets–making any attempt to discover OSPF neighbors have no benefit. Additionally, R3 has migrated its configuration away from the older **network** commands, instead using the **ip ospf area** interface subcommand.

Example 5-3 *Configuring passive-interface and ip ospf area*

```
Interface loopback 1
 Ip address 3.3.3.3 255.255.255.255

router ospf 3
 passive-interface FastEthernet0/0
 passive-interface FastEthernet0/1

interface FastEthernet0/0
 ip ospf 3 area 0
interface FastEthernet0/1
 ip ospf 3 area 0
interface Serial0/0/1
 ip ospf 3 area 0

R3#show ip protocols
Routing Protocol is "ospf 3"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 3.3.3.3
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
  Routing on Interfaces Configured Explicitly (Area 0):
    Serial0/0/1
    FastEthernet0/1
    FastEthernet0/0
 Reference bandwidth unit is 100 mbps
  Passive Interface(s):
    FastEthernet0/0
    FastEthernet0/1
  Routing Information Sources:
    Gateway          Distance      Last Update
  Distance: (default is 110)
```

Note that in the second half of Example 5-3 the **show ip protocols** command now lists the interfaces as matched with the **ip ospf area** commands, and it lists the passive interfaces. You can take the list of explicitly configured interfaces, remove the passive interfaces, and know which interfaces on which R3 will attempt to discover OSPF neighbors. Also, take a

moment to compare this output with the same command's output in Example 5-2, with the earlier example listing the parameters of the configured **network** commands.

Settings That Must Match for OSPF Neighborship

After an OSPF router has discovered a potential neighbor by receiving a Hello from the other router, the local router considers the router that sent the Hello as a potential neighbor. The local router must examine the contents of the received Hello, plus a few other factors, compare those settings to its own, check for agreement, and only then may that other router be considered an OSPF neighbor.

For reference, the following list details the items seen in OSPF Hello messages. Note that some fields might not be present in a Hello, depending on the conditions in the network.

- OSPF Router ID
- Stub area flag
- Plus the following interface-specific settings:
  - Hello interval
  - Dead Interval
  - Subnet mask
  - List of neighbors reachable on the interface
  - Area ID
  - Router priority
  - Designated Router (DR) IP address
  - Backup DR (BDR) IP address
  - Authentication digest

Table 5-5 summarizes the items that two routers will compare when deciding whether they can become OSPF neighbors. For study purposes, the table also lists some items that one might think prevent OSPF neighborship but do not, with comparisons to EIGRP.

Key Topic

Table 5-5 *Neighbor Requirements for EIGRP and OSPF*

Requirement	OSPF	EIGRP
Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
Must not be passive on the connected interface.	Yes	Yes
Must be in same area.	Yes	N/A
Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	Yes	No
Router IDs must be unique.	Yes	No
IP MTU must match.	Yes[1]	No

Must pass neighbor authentication (if configured).	Yes	Yes
K-values (used in metric calculation) must match.	N/A	Yes
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.	No	Yes

[1]May allow the other router to be listed in the **show ip ospf neighbor** command, but the MTU mismatch will prevent proper operation of the topology exchange.

Note: Table 5-5 repeats most of the information listed in Chapter 2, Table 2-4, but in an order that focuses on OSPF issues.

The first few items in Table 5-5 require only a minor amount of discussion. First, OSPF checks the IP address (found as the source address of the Hello message) and mask (listed in the Hello message) of the potential neighbor, calculates the subnet number, and compares the subnet number and mask to its own interface IP address. Both the subnet number and mask must match. Additionally, the OSPF Hello messages include the area number on the subnet, as defined by that router. The receiving router compares the received Hello with its own configuration and rejects the potential neighbor if the area numbers do not match.

The next several headings inside this section examine the other three settings that can prevent OSPF neighborship: Hello and Dead intervals, OSPF Router ID, IP MTU, and authentication.

Optimizing Convergence Using Hello and Dead Timers

Using the same concept as EIGRP, but with different terminology, OSPF uses two timers to monitor the reachability of neighbors. With OSPF, the Hello interval defines how often the router sends a Hello on the interface. The Dead interval defines how long a router should wait, without hearing any Hello messages from that neighbor, before deciding that the neighbor failed. For example, with a default LAN interface setting of Hello of 10, and Dead of 40, the local router sends Hello messages every 10 seconds. The neighbor resets its downward-counting Hold timer to 40 upon receiving a Hello from that neighbor. Under normal operation on a LAN, with defaults, the Dead timer for a neighbor would vary from 40, down to 30, and then be reset to 40 upon receipt of the next Hello. However, if the Hello messages were no longer received for 40 seconds, the neighborship would fail, driving convergence.

To tune for faster convergence, you can configure OSPF to set a lower Hello and Dead timer. It speeds convergence in some cases; note that if the interface fails, OSPF will immediately realize that all neighbors reached through that interface have also failed and not wait on the Dead timer to count down to zero. For example, consider the internetwork in Figure 5-4. This internetwork has four routers connected to the same VLAN, with the interfaces, IP addresses, masks, and OSPF areas as shown.

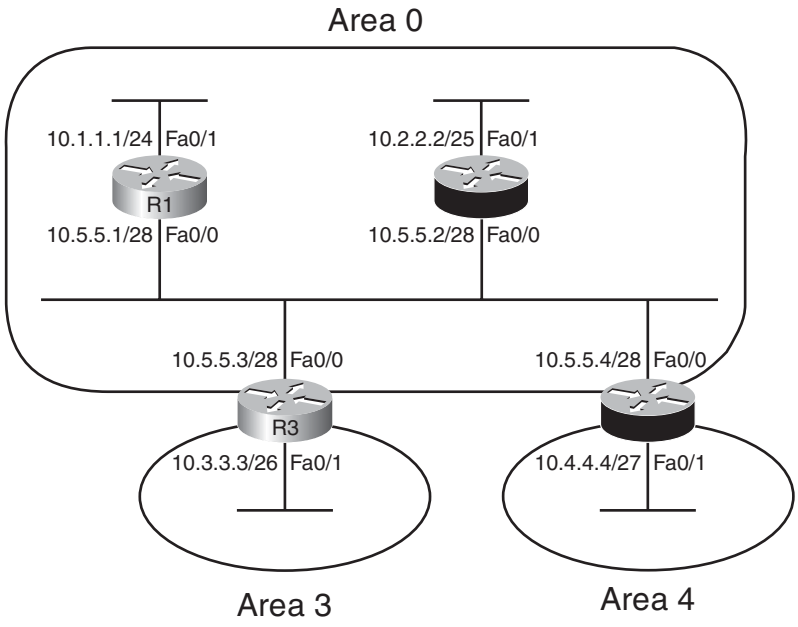


Figure 5-4 *Four OSPFs Routers on the Same Subnet, with Two OSPF Areas*

Example 5-4 verifies some of the facts about the routers in Figure 5-4, showing the changes to the Hello interval and the resulting failed neighborships. Each router has been assigned an obvious RID: 1.1.1.1 for R1, 2.2.2.2 for R2, and so on.

Example 5-4 *The Effect of Configuring a Different OSPF Hello Interval*

```
R4#show ip ospf neighbors


Neighbor ID      Pri   State           Dead Time   Address         Interface
1.1.1.1            1   2WAY/DROTHER    00:00:35    10.5.5.1
FastEthernet0/0
2.2.2.2            1   FULL/BDR        00:00:39    10.5.5.2
FastEthernet0/0
3.3.3.3            1   FULL/DR         00:00:38    10.5.5.3
FastEthernet0/0
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#interface fastethernet0/0
R4(config-if)#ip ospf hello-interval 9
R4(config-if)#^Z
*Apr 28 00:06:20.271: %SYS-5-CONFIG_I: Configured from console by console
R4#show ip ospf interface fa0/0
FastEthernet0/0 is up, line protocol is up
```

```
  Internet Address 10.5.5.4/28, Area 0
  Process ID 4, Router ID 4.4.4.4, Network Type BROADCAST, Cost: 1
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 10.5.5.3
  Backup Designated router (ID) 2.2.2.2, Interface address 10.5.5.2
  Timer intervals configured, Hello 9, Dead 36, Wait 36, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 3
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 2.2.2.2  (Backup Designated Router)
    Adjacent with neighbor 3.3.3.3  (Designated Router)
  Suppress hello for 0 neighbor(s)
R4#
*Apr 28 00:06:51.559: %OSPF-5-ADJCHG: Process 4, Nbr 1.1.1.1 on FastEthernet0/0
from 2WAY to DOWN, Neighbor Down: Dead timer expired
*Apr 28 00:06:57.183: %OSPF-5-ADJCHG: Process 4, Nbr 3.3.3.3 on FastEthernet0/0
from FULL to DOWN, Neighbor Down: Dead timer expired
*Apr 28 00:06:58.495: %OSPF-5-ADJCHG: Process 4, Nbr 2.2.2.2 on FastEthernet0/0
from FULL to DOWN, Neighbor Down: Dead timer expired
```

This example demonstrates several interesting facts. First, note that upon configuring the **ip ospf hello-interval 9** command under Fa0/0, the **show ip ospf interface fa0/0** command shows that not only did the Hello interval change, but the Dead timer was set to 4X the Hello interval, or 36. To directly set the Dead timer on the interface, use the **ip ospf dead-interval** *value* interface subcommand. Then, at the end of the example, note that all three of R4's neighbor relationships failed, because those routers now have mismatched Hello and Dead timers. However, the neighbor relationships failed only after the dead timers expired, as noted in the messages, and as confirmed by the timestamps on the messages.

Example 5-4 also shows the two normal, stable, and working neighbor states. Look to the heading "state" in the output of the **show ip ospf neighbors** command at the top of the example. The first word (before the /) lists the state or status of each neighbor. FULL refers to a fully adjacent neighbor, meaning the OSPF topology has been fully exchanged with that neighbor. The other state listed there, 2WAY, is a normal, stable, working state for neighbors with which topology data was not exchanged directly. As described in Chapter 6, in some cases OSPF routers exchange their topology information to one specific router on a LAN, called the designated router (DR), but they do not exchange their database directly with other routers. In the preceding example, taken from R4, R4 lists its relationship with R1 as 2WAY, which happens to be the status for a working neighbor that does not become fully adjacent.

Chapter 6's section "Exchange with a Designated Router" discusses the database exchange process when using a DR.

Note: OSPF has two methods to tune the Hello and Dead intervals to subsecond values. Like EIGRP, OSPF supports Bidirectional Forwarding Detection (BFD). Additionally, OSPF supports command **ip ospf dead-interval minimal hello-multiplier** *multiplier*, which sets the dead interval to one second, and the Hello interval to a fraction of a second based on the multiplier. For example, the command **ip ospf dead-interval minimal hello-multiplier 4** sets the dead interval to one second, with Hellos occurring four times (the multiple) per second, for an effective Hello interval of ¼ seconds.

Using a Unique OSPF Router-ID

As mentioned earlier in the "OSPF CCNA Review" section, each OSPF router assigns itself a router ID, based on the same rules as EIGRP. In OSPF's case, that means a router first looks for the OSPF **router-id** *rid-value* OSPF subcommand; next, to the highest IP address of any up/up loopback interface; and finally, to the highest IP address of any up/up non-loopback interface.

An OSPF RID mismatch makes for unpredictable results because OSPF routers base their view of the topology on the topology database, and the database identifies routers based on their RIDs. By design, all OSPF RIDs in a domain should be unique; to avoid such issues, OSPF prevents neighborships between routers with duplicate RIDs.

The next example shows what happens when two routers discover each other as potential neighbors, but notice a duplicate RID. Using the same network as in Figure 5-4, each router has been assigned an obvious RID: 1.1.1.1 for R1, 2.2.2.2 for R2, and so on. Unfortunately, R4 has been mistakenly configured with RID 1.1.1.1, duplicating R1's RID. R4 is powered on after all three other routers have established neighbor relationships. Example 5-5 shows some of the results.

Example 5-5 *OSPF RID Mismatch – R1 and R4, R4 Connects after R1*

```
! On R1... the following output occurs AFTER R4 powers on. R1, RID 1.1.1.1,
! does not form a neighbor relationship with R4.

R1#show ip ospf neighbors

Neighbor ID      Pri   State           Dead Time   Address         Interface
2.2.2.2            1   FULL/BDR        00:00:35    10.5.5.2
FastEthernet0/0
3.3.3.3            1   FULL/DR         00:00:33    10.5.5.3
FastEthernet0/0
! On R3
```

```
! R3 does form a neighbor relationship, but does not learn routes from the
! R4. Note that R3 does not have a route for R4's 10.4.4.0/27 subnet.
```

```
R3#show ip ospf neighbors

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/DROTHER    00:00:38    10.5.5.1
FastEthernet0/0
1.1.1.1           1   FULL/DROTHER    00:00:37    10.5.5.4
FastEthernet0/0
2.2.2.2           1   FULL/BDR        00:00:35    10.5.5.2
FastEthernet0/0
R3#show ip route ospf
     10.0.0.0/8 is variably subnetted, 4 subnets, 4 masks
O       10.2.2.0/25 [110/2] via 10.5.5.2, 00:06:56, FastEthernet0/0
O       10.1.1.0/24 [110/2] via 10.5.5.1, 00:01:34, FastEthernet0/0
```

As you can see from the output on R1, whose RID is duplicated with R4, the routers with duplicate RIDs do not form a neighbor relationship. Additionally, other routers, such as R3 as shown in the example, do form a neighbor relationship with the two routers, but the duplication confuses the topology flooding process. Because R3 formed its neighborship with R1 before R4, R3 does learn a route for R1's 10.1.1.0/24 subnet, but does not for R4's 10.4.4.0/27 subnet. However, with the same configuration, but a different sequence and timing of neighbors coming up, R3 might learn about 10.4.4.0/27 instead of 10.1.1.0/24.

Note: Note that the OSPF process will not start without an RID.

Using the Same IP MTU

The maximum transmission unit (MTU) of an interface tells IOS the largest IP packet that can be forwarded out the interface. This setting protects the packet from being discarded on data links whose Layer 2 features will not pass a frame over a certain size. For example, routers typically default to an IP MTU of 1500 bytes to accommodate Ethernet's rules about frames not exceeding 1526 bytes.

From a data plane perspective, when a router needs to forward a packet larger than the outgoing interface's MTU, the router either fragments the packet or discards it. If the IP header's don't fragment (DF) bit is set, the router discards the packet. If the DF bit is not set, the router can perform Layer 3 fragmentation on the packet, creating two (or more) IP packets with mostly identical IP headers, spreading the data that follows the original IP packet header out among the fragments. The fragments can then be forwarded, with the reassembly process being performed by the receiving host.

From a design perspective, the MTU used by all devices attached to the same data link ought to be the same value. However, routers have no dynamic mechanism to prevent the misconfiguration of MTU on neighboring routers.

When an MTU mismatch occurs between two OSPF neighbors, one router will attempt to become neighbors with the other router whose MTU differs. The other router will be listed in the list of neighbors (**show ip ospf neighbor**). However, the two routers will not

exchange topology information, and the two routers will not calculate routes that use this neighbor as the next-hop router.

The IP MTU can be set on an interface using the **ip mtu** *value* interface subcommand and for all Layer 3 protocols with the **mtu** *value* interface subcommand. Example 5-6 shows an example, with R4 again configured so that it has problems.

Example 5-6 *Setting IP MTU and Failing the OSPF Database Exchange Process*

```
R4#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#int fastethernet0/0
R4(config-if)#ip mtu 1498
R4(config-if)#^Z
R4#
R4#show ip interface fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.5.5.4/28
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1498 bytes
! lines omitted for brevity
R4#show ip ospf neighbors

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   EXSTART/DROTHER 00:00:39    10.5.5.1
FastEthernet0/0
2.2.2.2           1   EXSTART/DROTHER 00:00:37    10.5.5.2
FastEthernet0/0
3.3.3.3           1   EXSTART/BDR     00:00:39    10.5.5.3
FastEthernet0/0

*Apr 28 12:36:00.231: %OSPF-5-ADJCHG: Process 4, Nbr 2.2.2.2 on FastEthernet0/0
from EXSTART to DOWN, Neighbor Down: Too many retransmissions
R4#show ip ospf neighbors

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   INIT/DROTHER    00:00:39    10.5.5.1
FastEthernet0/0
2.2.2.2           1   DOWN/DROTHER      -         10.5.5.2
FastEthernet0/0
3.3.3.3           1   INIT/DROTHER    00:00:39    10.5.5.3        FastEthernet0/0
```

Note that you could argue that the mismatched MTU does not prevent routers from becoming neighbors, but it does prevent them from successfully exchanging topology data. When the mismatch occurs, a pair of routers tries to become neighbors, and they list each other in the output of **show ip ospf neighbors**, as seen in Example 5-6. However, the neighbor state (listed before the /, under heading "State") moves from EXSTART (which means the database exchange process is starting), but it fails as implied by the highlighted

message in the example. Then, the state changes to DOWN, and later one router tries again, moving to INIT (initializing) state. So, the neighbor is listed in the output of **show ip ospf neighbors** command, but never succeeds at exchanging the topology data.

Chapter 6 discusses the database exchange process, making reference to neighbor states. Table 6-5, in Chapter 6s section "OSPF Message and Neighbor State Reference," summarizes the neighbor state values and their meaning.

OSPF Authentication

OSPF authentication causes routers to authenticate every OSPF message. To do so, the routers use the same preshared key value, generating an MD5 digest for each OSPF message and sending that digest as part of each OSPF message. If a router configured for OSPF authentication receives an OSPF message, and the received message's MD5 digest does not pass the authentication checking based on the local key value, the router silently discards the message. As a result, when authentication fails, two routers cannot become OSPF neighbors, because they ignore the inauthentic OSPF Hello messages.

OSPF authentication uses one of three types: type 0 (no authentication), type 1 (clear text), and type 2 (MD5), with MD5 being the only reasonable option in production networks. To configure to have no authentication, do nothing, because IOS defaults to not attempt OSPF authentication. However, for either of the other two OSPF authentication options, you need to follow two configuration steps, as follows:

Step 1. Authentication must be enabled, plus the authentication type must be selected, through one of two means:

Enabling per interface using the ip ospf authentication [message-digest] interface subcommand

Enabling on all interfaces in an area by changing the area-wide authentication setting using the area area-no authentication [message-digest] subcommand under router ospf

Step 2. The authentication keys must be configured per interface.

Table 5-6 lists the three OSPF authentication types, along with the *interface* commands to both enable authentication and to define the authentication keys. Note that the three authentication types can be seen in the messages generated by the **debug ip ospf adjacency** command.

Table 5-6 *OSPF Authentication Types*

Type	Meaning	Enabling Interface Subcommand	Authentication Key Configuration Interface Subcommand
0	None	**ip ospf authentication null**	—
1	Clear text	**ip ospf authentication**	**ip ospf authentication-key** *key-value*
2	MD5	**ip ospf authentication message-digest**	**ip ospf message-digest-key** *key-number* **md5** *key-value*

Note: The maximum length of the key is 16 characters.

Although IOS defaults to use type 0 authentication (no authentication), you can override this default using the **area authentication** command in OSPF configuration mode. For example, on a router with 12 interfaces, you plan to use type 2 (MD5) authentication on all interfaces. Rather than add the **ip ospf authentication message-digest** command to all 12 interfaces, you could override the default that applies to all interface on that router, in a given area. Table 5-7 lists the commands to set the default values.

Key Topic

Table 5-7 *Effect of the area authentication Command on OSPF Interface Authentication Settings*

area authentication Command	Interfaces in That Area Default to Use...
default; no configuration required	Type 0
area *num* **authentication**	Type 1
area *num* **authentication message-digest**	Type 2

Note that if the area-wide default and the interface subcommand have both been configured, the interface setting takes precedence. For example, if the interface has been configured with the **ip ospf authentication** subcommand (type 1, clear text), and the **area authentication** command sets MD5 authentication, that particular interface uses clear text authentication.

Example 5-7 shows the authentication configuration on Router R3 from Figure 5-4. In this case, all four routers use MD5 authentication on their common subnet, with key number 1, and key value *really-a-secret*. R3 also configures simple password authentication on its Fa0/1 interface, just to demonstrate the various commands. Also for the purposes of demonstrating the commands, R3 overrides the default authentication, setting the default area 0 authentication to MD5.

Example 5-7 *OSPF Authentication Using Only Interface Subcommands*

```
! First, overriding the default in area 0 so that it uses MD5 authentication,
! meaning Fa0/0, in area 0, will use MD5.

R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 3
R3(config-router)# area 0 authentication message-digest
R3(config-router)#interface fastethernet0/0
R3(config-if)# ip ospf message-digest-key 1 md5 really-a-secret
!
```

```
! Next, two interface subcommands to enable simple password authentication
! on Fa0/1, which is in area 3.

R3(config-if)#interface fastethernet0/1
R3(config-if)#ip ospf authentication
R3(config-if)#ip ospf authentication-key secret
R3(config-if)# ctl-z
R3#
R3#show ip ospf interface fastethernet0/0
! Lines omitted for brevity - the last few lines list authentication information
  Message digest authentication enabled
    Youngest key id is 1


R3#show ip ospf interface fastethernet0/1
! Lines omitted for brevity - the last few lines list authentication information
  Simple password authentication enabled
```

Note that to verify whether OSPF authentication is enabled, and the authentication type, use the **show ip ospf interface** command, and look to the very end of the output, as shown at the end of Example 5-7. Note that these messages identify the configuration setting and do nothing to confirm whether authentication passed. To confirm whether it worked, look for the neighbor in the output of **show ip ospf neighbor**: A neighbor with whom authentication fails will not be listed. The reason is that the Hello will be rejected due to authentication, and the Hello contains several parameters that must be checked before a router will choose to become an OSPF neighbor with that router. You can also use the **debug ip ospf adj** command, whose output explicitly states that a mismatch exists with the authentication key.

Unlike EIGRP authentication, OSPF authentication does not allow the configuration of a key chain with time-based authentication keys. However, multiple keys can be configured on an interface, each with a different key number. To migrate to a new key, you would first configure a new key value on all routers in a subnet and then delete the configuration of the old keys.

To avoid having network failures during this cutover, OSPF actually sends and accepts messages that use all the currently configured authentication keys on the interface. For example, if the four routers in Figure 5-4 had been configured with key numbers 1 and 2, every OSPF message would be sent twice–once with each key.

This concludes the discussion of OSPF neighborships on LANs. The final brief section examines some issues with OSPF neighborship on WANs.

OSPF Neighbors and Adjacencies on WANs

To form OSPF neighbor relationships on WAN connections, OSPF still must meet the same requirements as on LANs. The area number must match with each neighbor; the IP subnet number and mask of each router must match; authentication must pass; and so on. In short, the items in Table 5-5 earlier in this chapter must be true.

However, the operation of OSPF on WAN links of various types requires some additional thought, particularly when developing an implementation and verification plan. In particular, depending on the WAN technology and configuration, the following additional questions may matter for proper OSPF operation over WAN connections:

- Will the routers discover each other using multicast OSPF Hello messages, or do the neighbors require predefinition?
- Will the routers try to elect a DR, and if so, which router should be allowed to be the DR?
- With which other routers should each router become an OSPF neighbor?

The first two of these items depend in part on the setting of the OSPF *network type*, and the third question depends on the WAN service. This section first examines the concept of OSPF network types and then examines the use of OSPF over common WAN technologies included in the CCNP ROUTE exam.

OSPF Network Types

The OSPF network type–a per-interface setting–directs OSPF in regard to three important facts:

- Whether the router can expect to discover neighbors using multicast Hello messages
- Whether only two or more than two OSPF routers can exist in the subnet attached to the interface
- Whether the router should attempt to elect an OSPF DR on that interface

For instance, LAN interfaces require a DR because of the default OSPF network type of *broadcast*. OSPF defines this interface network type to use multicast Hellos to dynamically discover neighbors, allows more than two routers to be in the same subnet, and to attempt to elect a DR. Conversely, point-to-point links and point-to-point WAN subinterfaces default to use a network type of *point-to-point*, meaning that only two OSPF routers can exist in the subnet, neighbors can be dynamically discovered through Hellos, and that the routers do not elect a DR.

Note: The discussion of the motivation for using OSPF DRs occurs in Chapter 6 section "Background of Designated Routers." For now, note that DRs become useful but are not always required, when a subnet supports more than two OSPF routers (for example, on a LAN subnet).

In production networks, the network type is often ignored, because there is no motivation to change this setting–you pick a combination that works, and most everyone ignores it. For the sake of CCNP ROUTE, you need to be aware of the setting and know a few of the caveats found in some Frame Relay configurations. To keep the discussion focused on the core topics, Chapters 6 and 7 avoid the Frame Relay configurations that require more consideration of the OSPF network type setting. Chapter 8's section "OSPF over Multipoint Frame Relay" discusses the OSPF network types in more depth.

Table 5-8 summarizes the OSPF network types and their meanings. Note that this per-interface or per-subinterface setting is configured with the **ip ospf network** *type* interface subcommand; the first column in the table lists the exact keyword according to this command. Note that the gray highlighted rows will be discussed in Chapter 8.

Table 5-8 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Default Hello Interval	Dynamic Discovery of Neighbors?	More Than Two Routers Allowed in the Subnet?
Broadcast	Yes	10	Yes	Yes
Point-to-point[1]	No	10	Yes	No
Loopback	No	—	—	No
Nonbroadcast[2] (NBMA)	Yes	30	No	Yes
Point-to-multipoint	No	30	Yes	Yes
Point-to-multipoint nonbroadcast	No	30	No	Yes

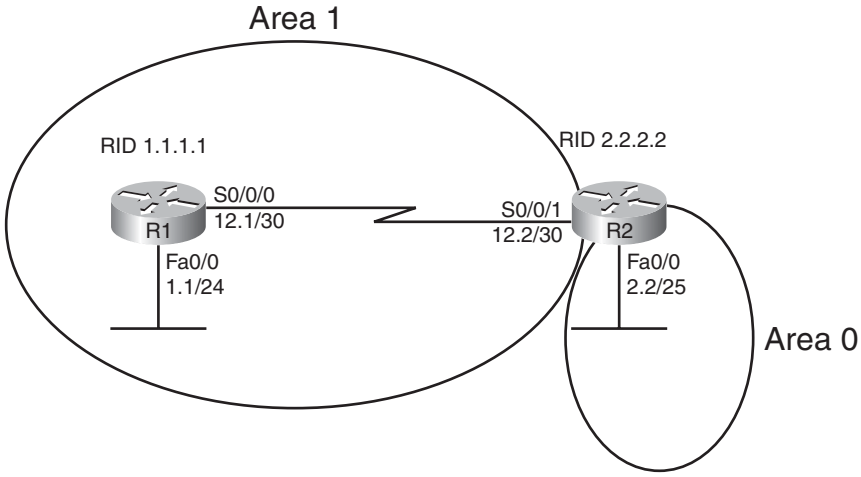
[1]Default on Frame Relay point-to-point subinterfaces.
[2]Default on Frame Relay physical and multipoint subinterfaces.

OSPF Neighborship over Point-to-Point Links

Point-to-point serial links can be a bit boring. You configure IP addresses on either end, configure the **clock rate** if using a back-to-back serial cable in a lab, and **no shutdown** the interfaces. When enabling OSPF on the interfaces, no extra effort is required compared to LANs–just enable OSPF on the interface, and rely on the default OSPF network type of *point-to-point*.

However, serial links can provide a convenient and uncluttered place to experiment with OSPF network types. As such, Figure 5-5 shows a small network with two routers, with

Example 5-8 that follows showing several examples of the OSPF network type. (This small network matches a portion of the network shown in Figure 5-1 earlier in this chapter.)



Note: All IP addresses begin with 10.1 unless otherwise noted.

Figure 5-5 *Simple Two Router Internetwork*

Example 5-8 demonstrates OSPF network types with all defaults on the HDLC link between R1 and R2.

Example 5-8 *OSPF Network Types, Default, on an HDLC Link*

```
R1#show run int s0/0/0
Building configuration...

Current configuration : 102 bytes
!
interface Serial0/0/0
 ip address 10.1.12.1 255.255.255.252
 no fair-queue
 clock rate 1536000
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
!
end

R1#show ip ospf interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Internet Address 10.1.12.1/30, Area 1
```

```
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 64
! lines omitted for brevity

R1#show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address         Interface
2.2.2.2            0   FULL/  -        00:00:31    10.1.12.2       Serial0/0/0
```

Example 5-6 begins listing R1's configuration on the serial link, mainly to make the point that the OSPF network type has not been explicitly configured. The **show ip ospf interface** command then lists the network type (point-to-point). Based on Table 5-7, this type should dynamically discover neighbors, and it does, with neighbor 2.2.2.2 (R2) being listed at the end of the example. In particular, note that under the state heading in the **show ip ospf neighbor** command output, after the /, only a dash is listed. This notation means that no attempt was made to elect a DR. If the network type had implied a DR should be elected, then some text would be listed after the /, for example, "/DR" meaning that the neighbor was the DR. (Refer back to the end of Example 5-4 for an example of the output of **show ip ospf neighbor** in which a DR has been elected.)

Example 5-9 shows an alternative where both routers change their OSPF network type on the serial link to nonbroadcast. This change is nonsensical in real designs and is only done for the purposes of showing the results: that the neighbors are not discovered dynamically, but once defined, a DR is elected.

Note: R2 has been preconfigured to match the configuration on R1 in Example 5-9, namely, the OSPF network type has been changed (**ip ospf network non-broadcast**), and R2 has been configured with a **neighbor 10.1.12.1** OSPF router subcommand.

Example 5-9 *Configuring OSPF Network Type Nonbroadcast on an HDLC Link*

```
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface s0/0/0
R1(config-if)#ip ospf network ?
  broadcast             Specify OSPF broadcast multi-access network
  non-broadcast         Specify OSPF NBMA network
  point-to-multipoint   Specify OSPF point-to-multipoint network
  point-to-point        Specify OSPF point-to-point network
R1(config-if)#ip ospf network non-broadcast
R1(config-if)#^Z

R1#show ip ospf neighbor

R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
```

```
R1(config-router)#neighbor 10.1.12.2
R1(config-router)#^Z
R1#

*Apr 28 20:10:15.755: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial0/0/0 from
LOADING to FULL, Loading Done
R1#show ip ospf neighbor

Neighbor ID      Pri   State           Dead Time   Address         Interface
2.2.2.2            1   FULL/DR         00:01:58    10.1.12.2       Serial0/0/0
```

The example begins with R2 already configured, so the neighbor relationship has already failed. When the OSPF network type changes on R1's S0/0/0, the routers do not dynamically discover each other, based on the network type (nonbroadcast). However, by completing the configuration in the example by adding R1's **neighbor 10.1.12.2** command, the neighbor relationship is formed. Also, note that the final **show ip ospf neighbor** command lists a state of FULL, then a /, and then DR, meaning that a DR was indeed elected, as required by this OSPF network type.

Neighborship over Frame Relay Point-to-Point Subinterfaces

Frame Relay design allows several options for IP addressing and subnetting. One option treats each pair of routers on the ends of each PVC as a point-to-point topology, with one subnet assigned to each pair of routers. Another option treats more than two routers as a group, whether connected with a full mesh or partial mesh of PVCs, with a single subnet assigned to that group.

Many Frame Relay designs use the first option, treating each pair of routers on the ends of a PVC as a single subnet, as shown in Figure 5-6. In such cases, it makes sense to treat each PVC as a separate point-to-point connection, assigning a single subnet (at Layer 3) to each Layer 2 PVC.

With this design, if all the routers use point-to-point subinterfaces as shown in R1's configuration in the figure, you can ignore the OSPF network (interface) type, and OSPF works fine. IOS point-to-point subinterfaces unsurprisingly default to use OSPF network type point-to-point. The two routers discover each other using multicast OSPF Hellos, they do not bother to elect a DR, and everything works well.

Chapter 8 discusses alternative Frame Relay configurations.

Neighborship on MPLS VPN

Multiprotocol Label Switching (MPLS) virtual private networks (VPN) create a WAN service that has some similarities but many differences when compared to Frame Relay. The customer routers connect to the service, often times with serial links, but other times with Frame Relay PVCs or with Ethernet. The service itself is a Layer 3 service, forwarding IP packets through the cloud. As a result, no predefined PVCs need exist between the customer routers. Additionally, the service uses routers at the edge of the service provider cloud–generically called provider edge (PE) routers–and these routers are Layer 3-aware.

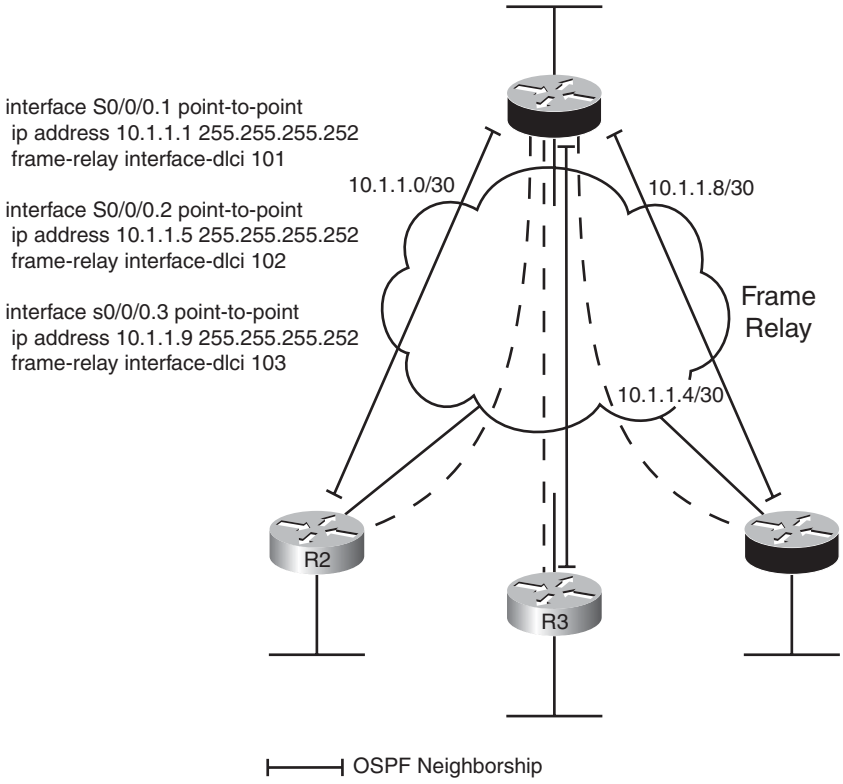


Figure 5-6 *Partial Meshed Frame Relay Network*

That Layer 3 awareness means that the customer edge (CE) routers form an OSPF neighborship with the PE router on the other end of their local access link, as shown in Figure 5-7. The PE routers exchange their routes, typically using Multiprotocol BGP (MP-BGP), a topic outside the scope of this book. So, unlike the design seen previously in Figure 5-6, the central site router will not have an OSPF neighborship with each branch office router but will have a neighborship with the MPLS VPN provider's PE router.

MPLS VPN does impact the data seen in the LSDB in the Enterprise routers and requires some different thinking in regard to area design. However, these details remain outside the scope of this book.

Neighborship on Metro Ethernet

In the like-named section "Neighborship on Metro Ethernet" in Chapter 2, that chapter explained the basic terminology with Metro Ethernet, including of VPWS, a point-to-point service, and VPLS, a multipoint service. In both cases, however, if the customer connects to the service using a router, the configuration typically uses VLAN trunking with subinterfaces off the FastEthernet or Gigabit Ethernet interface. If connecting with a Layer 3 switch, the configuration again often uses VLAN trunking, with the Layer 3 configuration being made on various VLAN interfaces inside the switch configuration.

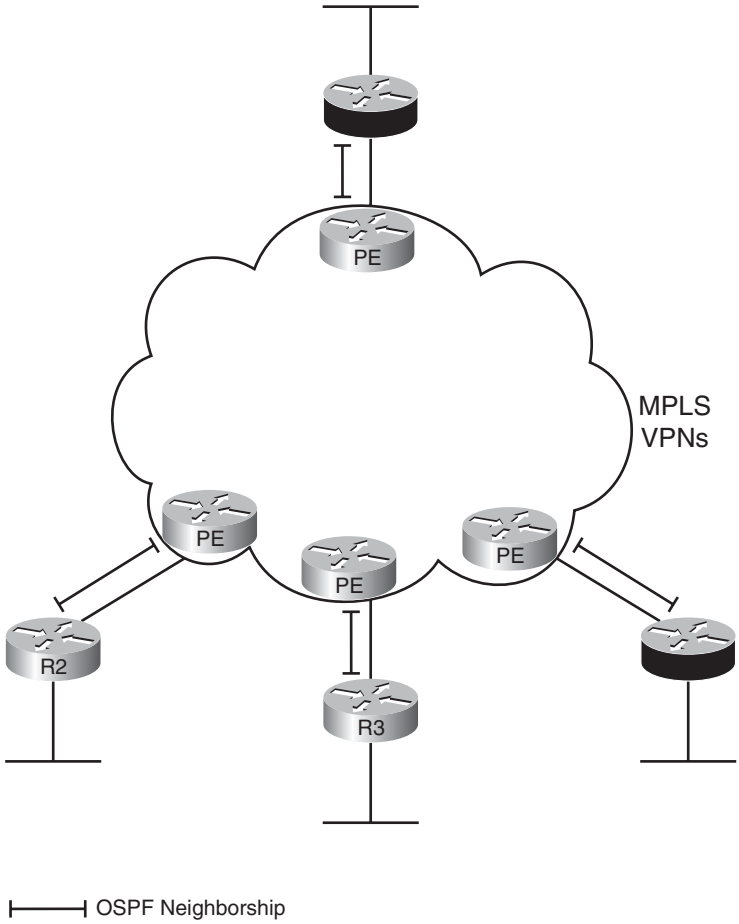


Figure 5-7 *OSPF Neighborships over MPLS VPN*

Because MetroE services provide Layer 2 connectivity, customer routers do not form OSPF neighborships with routers inside the service provider's network. Instead, the OSPF neighborships forms between customer routers, essentially as if the service were a large WAN. Figure 5-8 shows the basic idea, with four routers connected to the service.

Figure 5-8 shows four routers with any-to-any connectivity, typical of a VPWS service. However, from an OSPF design perspective, each pair of routers could communicate over a different VLAN, using a different Layer 3 subnet. Each Layer 3 subnet could be in a different area. Although (like MPLS VPN area design) beyond the scope of this book, these options give the network designer many options on how to define areas.

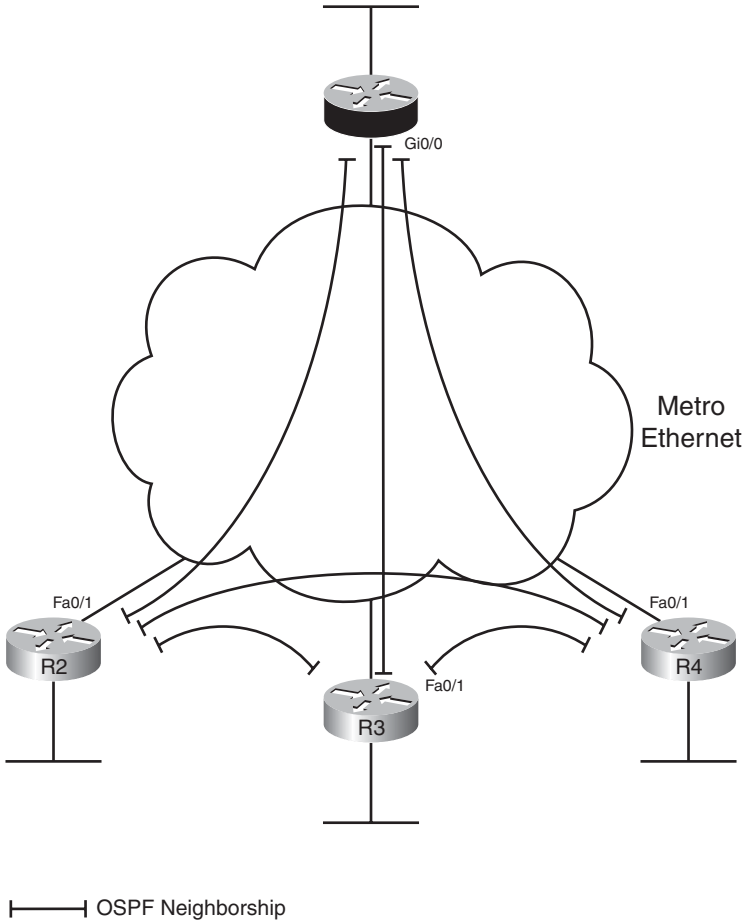


Figure 5-8 *OSPF Neighborships over Metro Ethernet*

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Table."

Design Review Table

Table 5-9 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 5-9 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Improve OSPF convergence.	
Implement OSPF on each router so that neighborships are formed (2).	
Limit neighborship formation on OSPF-enabled interfaces (2).	
The design shows branch routers with WAN interfaces in area 0 and LAN interfaces in different areas for each branch. What LSDB information do you expect to see in the branch routers?	

Implementation Plan Peer Review Table

Table 5-10 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 5-10 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
What happens on a router interface on which an OSPF **network** command matches the interface? (2)	
What configuration settings prevent OSPF neighbor discovery on an OSPF-enabled interface?	
What settings do potential neighbors check before becoming OSPF neighbors? (7)	
What settings that you might think would impact OSPF neighbor relationships actually do not prevent neighborship?	
A design shows one main site and 100 branches, with OSPF, and MPLS VPNs. How many OSPF neighborships over the WAN do you expect to see on the central site router?	
A design shows one main site and 100 branches, with one Frame Relay PVC between the main site and each branch. How many OSPF neighborships over the WAN do you expect to see on the central site router?	
A design shows six routers connected to the same VLAN and subnet. How many OSPF fully adjacent neighborships over this subnet do you expect each router to have?	
A design shows one main site and 100 branches, each connected with a VPWS service. The configuration shows that the central site router uses a separate VLAN subinterface to connect to each branch, but the branch routers do not have a VLAN connecting to other branches. How many OSPF fully adjacent neighborships over the WAN do you expect to see on the central site router?	

Create an Implementation Plan Table

To practice skills useful when creating your own OSPF implementation plan, list in Table 5-11 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 5-11 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling OSPF on interfaces–traditional method	
Enabling OSPF on interfaces–using interface subcommands	
Setting Hello and Dead intervals	

Table 5-11 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
MD5 authentication, with no **router** subcommands	
MD5 authentication, with **router** subcommands	
Passive interfaces, with router subcommands	
OSPF router ID	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own OSPF verification plan, list in Table 5-12 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 5-12 *Verification Plan Memory Drill*

Information Needed	Command
Which routes have been added to the IP routing table by OSPF	
All routes in a router's routing table	
The specific route for a single destination address or subnet	
A list of all (both static and dynamically discovered) OSPF neighbors	
List interfaces on which OSPF has been enabled	
List the number of OSPF neighbors and fully adjacent neighbors known via a particular interface	
The elapsed time since a neighborship was formed	
The configured Hello timer for an interface	
The configured Dead Interval timer for an interface	
The current actual dead timer for a neighbor	
A router's RID	
A list of OSPF passive interfaces	
The currently used OSPF authentication type	
The smallest-numbered OSPF authentication key number	
Lists traffic statistics about OSPF	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review All the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 5-13 lists a reference of these key topics and the page numbers on which each is found.

Table 5-13 *Key Topics for Chapter 5*

Key Topic Element	Description	Page Number
Table 5-2	Common OSPF terminology	142
List	Base OSPF configuration steps	144
List	Rules for choosing the OSPF Router ID	146
Table 5-3	Five most commonly used OSPF **show** commands	146
Table 5-4	Summary table of OSPF features	149
List	Requirements before OSPF will attempt to dynamically discover neighbors	150
Table 5-5	Requirements for neighbor formation–OSPF and EIGRP	152
List	Rules for enabling OSPF authentication on an interface	159
Table 5-6	OSPF Authentication command summary	159
Table 5-7	Commands to set the default OSPF authentication type	160
Table 5-8	OSPF network types	163

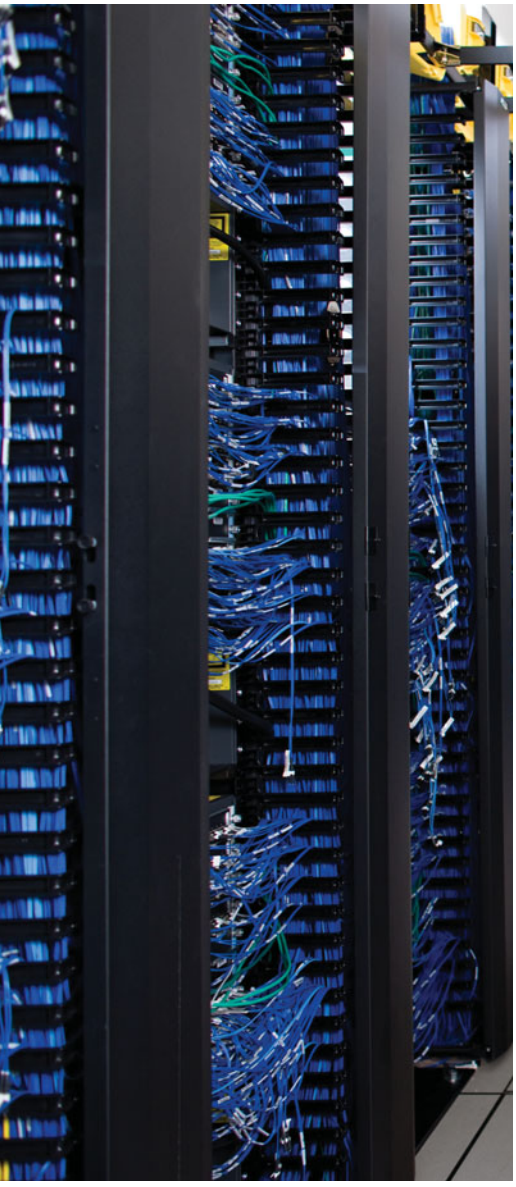
Complete the Tables and Lists from Memory

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

Define Key Terms

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

Area, Area Border Router (ABR), Backbone router, Router ID, Hello Interval, Dead Interval, Fully adjacent, OSPF network type



This chapter covers the following subjects:

LSAs and the OSPF Link State Database: This section examines LSA Types 1, 2, and 3, and how they allow OSPF routers to model the topology and choose the best routes for each known subnet.

The Database Exchange Process: This section details how neighboring routers use OSPF messages to exchange their LSAs.

Choosing the Best Internal OSPF Routes: This section examines how OSPF routers calculate the cost for each possible route to each subnet.

CHAPTER 6

OSPF Topology, Routes, and Convergence

OSPF and EIGRP both use three major branches of logic, each of which populates a different table: the neighbor table, the topology table, and the IP routing table. This chapter examines topics related to the OSPF topology table–the contents, and the processes by which routers exchange this information–and how OSPF routers choose the best routes in the topology table to be added to the IP routing table.

In particular, this chapter begins by looking at the building blocks of OSPF topology, namely the OSPF link state advertisement (LSA). Following that, the chapter examines the process by which OSPF routers exchange LSAs with each other. Finally, the last major section of the chapter discusses how OSPF chooses the best route among many when running the Shortest Path First (SPF) algorithm.

Note that this chapter focuses on OSPF Version 2, the long-available version of OSPF that supports IPv4 routes. Chapter 17, "IPv6 Routing Protocols and Redistribution," discusses OSPF Version 3, which applies to IPv6.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these nine self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 6-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 6-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
LSAs and the OSPF Link State Database	1–3
The Database Exchange Process	4, 5
Choosing the Best OSPF Routes	6–9

1. A network design shows area 1 with three internal routers, area 0 with four internal routers, and area 2 with five internal routers. Additionally, one ABR (ABR1) connects areas 0 and 1, plus a different ABR (ABR2) connects areas 0 and 2. How many Type 1 LSAs would be listed in ABR2's LSDB?

    a. 6
    b. 7
    c. 15
    d. 12
    e. None of the other answers is correct.

2. A network planning diagram shows a large internetwork with many routers. The configurations show that OSPF has been enabled on all interfaces, IP addresses correctly configured, and OSPF working. For which of the following cases would you expect a router to create and flood a Type 2 LSA?

    a. When OSPF is enabled on a LAN interface, and the router is the only router connected to the subnet.
    b. When OSPF is enabled on a point-to-point serial link, and that router has both the higher router ID and higher interface IP address on the link.
    c. When OSPF is enabled on a Frame Relay point-to-point subinterface, has the lower RID and lower subinterface IP address, and otherwise uses default OSPF configuration on the interface.
    d. When OSPF is enabled on a working LAN interface on a router, and the router has been elected BDR.
    e. None of the other answers is correct.

3. A verification plan shows a network diagram with branch office Routers B1 through B100, plus two ABRs, ABR1, and ABR2, all in area 100. The branches connect to the ABRs using Frame Relay point-to-point subinterfaces. The verification plan lists the output of the **show ip ospf database summary 10.100.0.0** command on a router B1, one of the branches. Which of the following is true regarding the output that could be listed for this command?

    a. The output lists nothing unless 10.100.0.0 has been configured as a summary route using the **area range** command.
    b. If 10.100.0.0 is a subnet in area 0, the output lists one Type 3 LSA, specifically the LSA with the lower metric when comparing ABR1's and ABR2's LSA for 10.100.0.0.
    c. If 10.100.0.0 is a subnet in area 0, the output lists two Type 3 LSAs, one each created by ABR1 and ABR2.
    d. None, because the Type 3 LSAs would exist only in the ABR's LSDBs.

4. Which of the following OSPF messages contains entire complete LSAs used during the database exchange process?

   a. LSR
   b. LSAck
   c. LSU
   d. DD
   e. Hello

5. Routers R1, R2, R3, and R4 connect to the same 10.10.10.0/24 LAN-based subnet. OSPF is fully working in the subnet. Later, R5, whose OSPF priority is higher than the other four routers, joins the subnet. Which of the following are true about the OSPF database exchange process over this subnet at this point? (Choose two.)

   a. R5 will send its DD, LSR, and LSU packets to the 224.0.0.5 all-DR-routers multicast address.
   b. R5 will send its DD, LSR, and LSU packets to the 224.0.0.6 all-DR-routers multicast address.
   c. The DR will inform R5 about LSAs by sending its DD, LSR, and LSU packets to the 224.0.0.6 all-SPF-routers multicast address.
   d. The DR will inform R5 about LSAs by sending its DD, LSR, and LSU packets to the 224.0.0.5 all-SPF-routers multicast address.

6. R1 is internal to area 1, and R2 is internal to area 2. Subnet 10.1.1.0/24 exists in area 2 as a connected subnet off R2. ABR ABR1 connects area 1 to backbone area 0, and ABR2 connects area 0 to area 2. Which of the following LSAs must R1 use when calculating R1's best route for 10.1.1.0/24?

   a. R2's Type 1 LSA
   b. Subnet 10.1.1.0/24's Type 2 LSA
   c. ABR1's Type 1 LSA in area 0
   d. Subnet 10.1.1.0/24's Type 3 LSA in Area 0
   e. Subnet 10.1.1.0/24's Type 3 LSA in Area 1

7. Which of the following LSA types describes topology information that, when changed, requires a router in the same area to perform an SPF calculation? (Choose two.)

   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
   f. 7

8. The following output was taken from Router R3. A scan of R3's configuration shows that no **bandwidth** commands have been configured in this router. Which of the following answers lists configuration settings could be a part of a configuration that results in the following output? (Choose two.)

```
R3#show ip ospf interface brief
Interface     PID    Area            IP Address/Mask     Cost  State Nbrs F/C
Se0/0/0.2     3      34              10.10.23.3/29       647   P2P   1/1
Se0/0/0.1     3      34              10.10.13.3/29       1000  P2P   1/1
Fa0/0          3      34              10.10.34.3/24       20    BDR   1/1
```

   a. An **auto-cost reference-bandwidth 1000** command in router ospf mode
   b. An **auto-cost reference-bandwidth 2000** command in router ospf mode
   c. An **ip ospf cost 1000 interface S0/0/0.1** command in router ospf mode
   d. An **auto-cost reference-bandwidth 64700** command in router ospf mode

9. Which of the following LSA types describe information related to topology or subnets useful for calculating routes for subnets inside the OSPF domain? (Choose three.)

   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
   f. 7

Foundation Topics

LSAs and the OSPF Link State Database

Every router that connects to a given OSPF area should learn the exact same topology data. Each router stores the data, composed of individual link state advertisements (LSA), in their own copy of the link state database (LSDB). Then, the router applies the Shortest Path First (SPF) algorithm to the LSDB to determine the best (lowest cost) route for each reachable subnet (prefix/length).

When a router uses SPF to analyze the LSDB, the SPF process has some similarities to how humans put a jigsaw puzzle together–but without a picture of what the puzzle looks like. Humans faced with such a challenge might first look for the obvious puzzle pieces, such as the corner and edge pieces, because they are easily recognized. You might then group puzzle pieces together if they have the same color or look for straight lines that might span multiple puzzle pieces. And of course, you would be looking at the shapes of the puzzle pieces to see which ones fit together.

Similarly, a router's SPF process must examine the individual LSAs and see how they fit together, based on their characteristics. To better appreciate the SPF process, the first section of this chapter examines the three LSA types OSPF uses to describe an Enterprise OSPF topology inside the OSPF domain. By understanding the types of LSAs, you can get a better understanding of what a router might look for to take the LSAs–the pieces of a network topology puzzle if you will–and build the equivalent of a network diagram.

For reference, Table 6-2 lists the various OSPF LSA types. Note that Chapter 9 explains three other LSA types, all used when redistributing routes into the OSPF domain.

Table 6-2 *OSPF LSA Types*

LSA Type	Common Name	Description
1	Router	Each router creates its own Type 1 LSA to represent itself for each area to which it connects. The LSDB for one area contains one Type 1 LSA per router per area, listing the RID and all interface IP addresses on that router that are in that area. Represents stub networks as well.
2	Network	One per transit network. Created by the DR on the subnet, and represents the subnet and the router interfaces connected to the subnet.
3	Net Summary	Created by ABRs to represent subnets listed in one area's type 1 and 2 LSAs when being advertised into another area. Defines the links (subnets) in the origin area, and cost, but no topology data.
4	ASBR Summary	Like a type 3 LSA, except it advertises a host route used to reach an ASBR.

Table 6-2 *OSPF LSA Types*

LSA Type	Common Name	Description
5	AS External	Created by ASBRs for external routes injected into OSPF.
6	Group Membership	Defined for MOSPF; not supported by Cisco IOS.
7	NSSA External	Created by ASBRs inside an NSSA area, instead of a type 5 LSA.
8	External Attributes	Not implemented in Cisco routers.
9–11	Opaque	Used as generic LSAs to allow for easy future extension of OSPF; for example, type 10 has been adapted for MPLS traffic engineering.

LSA Type 1: Router LSA

An LSA type 1, called a *router LSA*, identifies an OSPF router based on its OSPF router ID (RID). Each router creates a Type 1 LSA for itself and floods the LSA throughout the same area. To flood the LSA, the originating router sends the Type 1 LSA to its neighbors inside the same area, who in turn send it to their other neighbors inside the same area, until all routers in the area have a copy of the LSA.

Besides the RID of the router, this LSA also lists information about the attached links. In particular, the Type 1 LSA lists:

- For each interface on which no DR has been elected, it lists the router's interface subnet number/mask and interface OSPF cost. (OSPF refers to these subnets as *stub networks*.)
- For each interface on which a DR has been elected, it lists the IP address of the DR and a notation that the link attaches to a *transit network* (meaning a type 2 LSA exists for that network).
- For each interface with no DR, but for which a neighbor is reachable, it lists the neighbor's RID.

As with all OSPF LSAs, OSPF identifies a Type 1 LSA using a 32-bit *link state identifier* (LSID). When creating its own Type 1 LSA, each router uses its own OSPF RID value as the LSID.

Internal routers each create a single Type 1 LSA for themselves, but ABRs create multiple Type 1 LSAs for themselves: one per area. The Type 1 LSA in one area will list only interfaces in that area and only neighbors in that area. However, the router still has only a single RID, so all its Type 1 LSAs for a single router list the same RID. The ABR then floods each of its Type 1 LSAs into the appropriate area.

To provide a better backdrop for the upcoming LSA discussions, Figure 6-1 shows a sample internetwork, which will be used in most of the examples in this chapter.

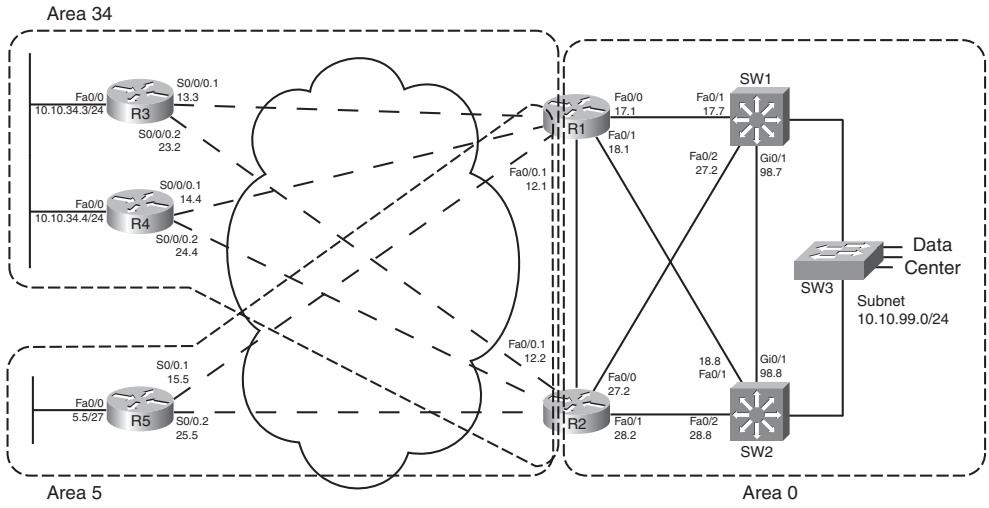


Figure 6-1 *Sample OSPF Multi-Area Design*

All routers that participate in an area, be they internal routers or ABRs, create and flood a Type 1 LSA inside the area. For example, in Figure 6-1, area 5 has one internal router (R5, RID 5.5.5.5), and two ABRs: R1 with RID 1.1.1.1 and R2 with RID 2.2.2.2. Each of these three routers create and flood their own Type 1 LSA inside area 5 so that all three routers know the same three Type 1 LSAs.

Next, to further understand the details inside a Type 1 LSA, first consider the OSPF configuration of R5 as an example. R5 has three IP-enabled interfaces: Fa0/0, S0/0/0.1, and S0/0.2. R5 uses point-to-point subinterfaces, so R5 should form neighbor relationships with both R1 and R2 with no extra configuration beyond enabling OSPF, in area 5, on all three interfaces. Example 6-1 shows this baseline configuration on R5.

Example 6-1 *R5 Configuration–IP Addresses and OSPF*

```
interface Fastethernet0/0
 ip address 10.10.5.5 255.255.255.224
 ip ospf 5 area 5
!
interface s0/0.1 point-to-point
 ip addr 10.10.15.5 255.255.255.248
 frame-relay interface-dlci 101
 ip ospf 5 area 5
!
interface s0/0.2 point-to-point
```

```
 ip addr 10.10.25.5 255.255.255.248
 frame-relay interface-dlci 102
 ip ospf 5 area 5
!
router ospf 5
 router-id 5.5.5.5
!
R5#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
se0/0.2      5     5                10.10.25.5/29      64    P2P   1/1
se0/0.1      5     5                10.10.15.5/29      64    P2P   1/1
fa0/0        5     5                10.10.5.5/27       1     DR    0/0
R5#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2         0     FULL/  -        00:00:30    10.10.25.2      Serial0/0.2
1.1.1.1         0     FULL/  -        00:00:38    10.10.15.1      Serial0/0.1
```

R5's OSPF configuration enables OSPF, for process ID 5, placing three interfaces in area 5. As a result, R5's type 1 LSA will list at least these three interfaces as links, plus it will refer to the two working neighbors. Example 6-2 displays the contents of R5's area 5 LSDB, including the detailed information in R5's Type 1 LSA, including the following:

- The LSID of R5's Type 1 LSA (5.5.5.5)
- Three links that connect to a stub network, each listing the subnet/mask
- Two links that state a connection to another router, one listing R1 (RID 1.1.1.1) and one listing R2 (RID 2.2.2.2)

Example 6-2 *R5 Configuration–IP Addresses and OSPF*

```
R5#show ip ospf database


            OSPF Router with ID (5.5.5.5) (Process ID 5)

                Router Link States (Area 5)


Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         835         0x80000002 0x006BDA 2
2.2.2.2         2.2.2.2         788         0x80000002 0x0082A6 2
5.5.5.5         5.5.5.5         787         0x80000004 0x0063C3 5

                Summary Net Link States (Area 5)
```

```
Link ID         ADV Router      Age         Seq#       Checksum
10.10.12.0      1.1.1.1         835         0x80000001 0x00F522
10.10.12.0      2.2.2.2         787         0x80000001 0x00D73C
! lines omitted for brevity

R5#show ip ospf database router 5.5.5.5

            OSPF Router with ID (5.5.5.5) (Process ID 5)

                Router Link States (Area 5)

  LS age: 796
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 5.5.5.5
  Advertising Router: 5.5.5.5
  LS Seq Number: 80000004
  Checksum: 0x63C3
  Length: 84
  Number of Links: 5

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 2.2.2.2
     (Link Data) Router Interface address: 10.10.25.5
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.10.25.0
     (Link Data) Network Mask: 255.255.255.248
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 1.1.1.1
     (Link Data) Router Interface address: 10.10.15.5
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.10.15.0
     (Link Data) Network Mask: 255.255.255.248
      Number of TOS metrics: 0
       TOS 0 Metrics: 64
```

```
Link connected to: a Stub Network
 (Link ID) Network/subnet number: 10.10.5.0
 (Link Data) Network Mask: 255.255.255.224
  Number of TOS metrics: 0
  TOS 0 Metrics: 1
```

The first command, **show ip ospf database**, displays a summary of the LSAs known to R5. The output mainly consists of a single line per LSA, listed by LSA ID. The three highlighted lines of this command, in Example 6-2, highlight the RID of the three router (Type 1) LSAs, namely 1.1.1.1 (R1), 2.2.2.2 (R2), and 5.5.5.5 (R5).

The output of the **show ip ospf database router 5.5.5.5** command displays the detailed information in R5's router LSA. Looking at the highlighted portions, you see three stub networks–three interfaces on which no DR has been elected–and the associated subnet numbers. The LSA also lists the neighbor IDs of two neighbors (1.1.1.1 and 2.2.2.2) and the interfaces on which these neighbors can be reached.

Armed with the same kind of information in R1's and R2's Type 1 LSAs, a router has enough information to determine which routers connect, over which stub links, and then use the interface IP address configuration to figure out the interfaces that connect to the other routers. Figure 6-2 shows a diagram of area 5 that could be built just based on the detailed information held in the router LSAs for R1, R2, and R5.

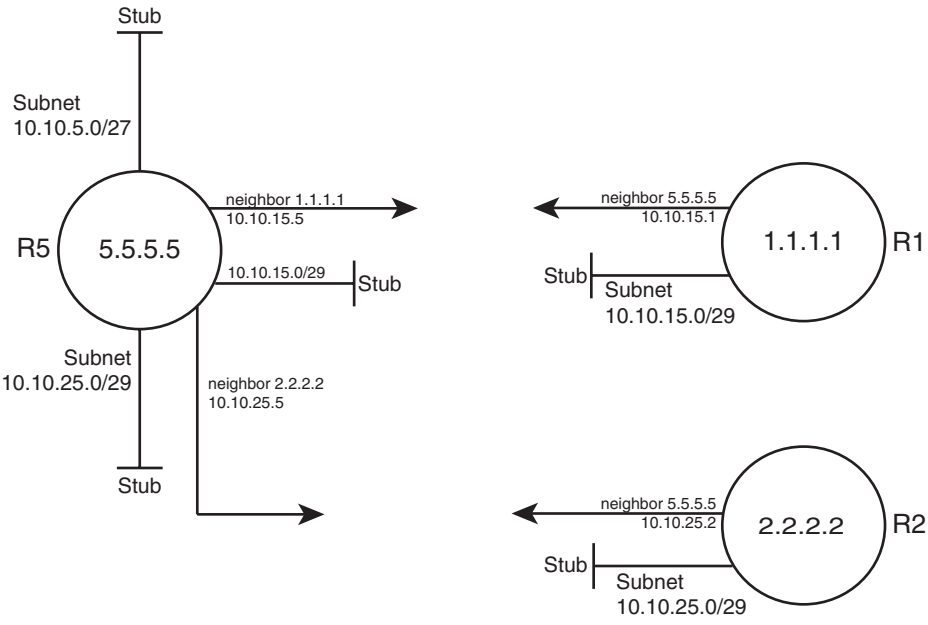


Figure 6-2 *Three Type 1 LSAs in Area 5*

Note that Figure 6-2 displays only information that could be learned from the Type 1 router LSAs inside area 5. Each Type 1 router LSA lists information about a router but only the details related to a specific area. As a result, Figure 6-2 shows R1's interface in area 5 but none of the interfaces in area 34 nor in area 0. To complete the explanation surrounding Figure 6-2, Example 6-3 lists R1's Type 1 router LSA for area 5.

Example 6-3 *R1's Type 1 LSA in Area 5*

```
R5#show ip ospf database router 1.1.1.1



            OSPF Router with ID (5.5.5.5) (Process ID 5)

                Router Link States (Area 5)

  Routing Bit Set on this LSA
  LS age: 1306
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 1.1.1.1
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000002
  Checksum: 0x6BDA
  Length: 48
  Area Border Router
  Number of Links: 2

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 5.5.5.5
     (Link Data) Router Interface address: 10.10.15.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.10.15.0
     (Link Data) Network Mask: 255.255.255.248
      Number of TOS metrics: 0
       TOS 0 Metrics: 64
```

Note: Because the OSPF uses the RID for many purposes inside different LSAs–for instance, as the LSID of a type 1 LSA–Cisco recommends setting the RID to a stable, predictable value. To do this, use the OSPF **router-id** *value* OSPF subcommand, or define a loopback interface with an IP address, as discussed in Chapter 5's section "Using a Unique OSPF Router ID."

LSA Type 2: Network LSA

SPF requires that the LSDB model the topology with nodes (routers) and connections between nodes (links). In particular, each link must be between a pair of nodes. When a multiaccess data link exists—for instance, a LAN—OSPF must somehow model that LAN so that the topology represents nodes and links between only a pair of nodes. To do so, OSPF uses the concept of a Type 2 Network LSA.

OSPF routers actually choose whether to use a Type 2 LSA for a multiaccess network based on whether a designated router (DR) has or has not been elected on an interface. So, before discussing the details of the Type 2 network LSA, a few more facts about the concept of a DR need to be discussed.

Background on Designated Routers

As discussed in Chapter 5's section "OSPF Network Types," the OSPF network type assigned to a router interface tells that router whether to attempt to elect a DR on that interface. Then, when a router has heard a Hello from at least one other router, the routers elect a DR and BDR.

OSPF uses a DR in a particular subnet for two main purposes:

- To create and flood a Type 2 network LSA for that subnet
- To aid in the detailed process of database exchange over that subnet

Routers elect a DR, and a backup DR (BDR), based on information in the OSPF Hello. The Hello message lists each router's RID and a priority value. When no DR exists at the time, routers use the following election rules when neither a DR nor BDR yet exists:

- Choose the router with the highest priority (default 1, max 255, set with **ip ospf priority** *value* interface subcommand).
- If tied on priority, choose the router with highest RID.
- Choose a BDR, based on next-best priority, or if a tie, next-best (highest) RID.

Although the preceding describes the election when no DR currently exists, the rules differ a bit when a DR and BDR already exist. After a DR and BDR are elected, no election is held until either the DR or BDR fails. If the DR fails, the BDR becomes the DR-—regardless of whether a higher priority router has joined the subnet—and a new election is held to choose a new BDR. If the BDR fails, a new election is held for BDR, and the DR remains unchanged.

On LANs, the choice of DR matters little from a design perspective, but does matter from an operational perspective. Throughout this chapter, note the cases in which output of **show** commands identify the DR and its role. Now, back to the topic of Type 2 LSAs.

Note: On Frame Relay WAN links, the choice of DR may impact whether OSPF functions at all. This topic is covered in Chapter 8, "OSPF Virtual Links and Frame Relay Operations."

Type 2 Network LSA Concepts

OSPF uses the concept of a Type 2 LSA to model a multiaccess network–a network with more than two routers connected to the same subnet–while still conforming to the "a link connects only two nodes" rule for the topology. For example, consider the network in Figure 6-3 (also shown as Figure 5-4 in the previous chapter). As seen in Chapter 5, all four routers form neighbor relationships inside area 0, with the DR and BDR becoming fully adjacent with the other routers.

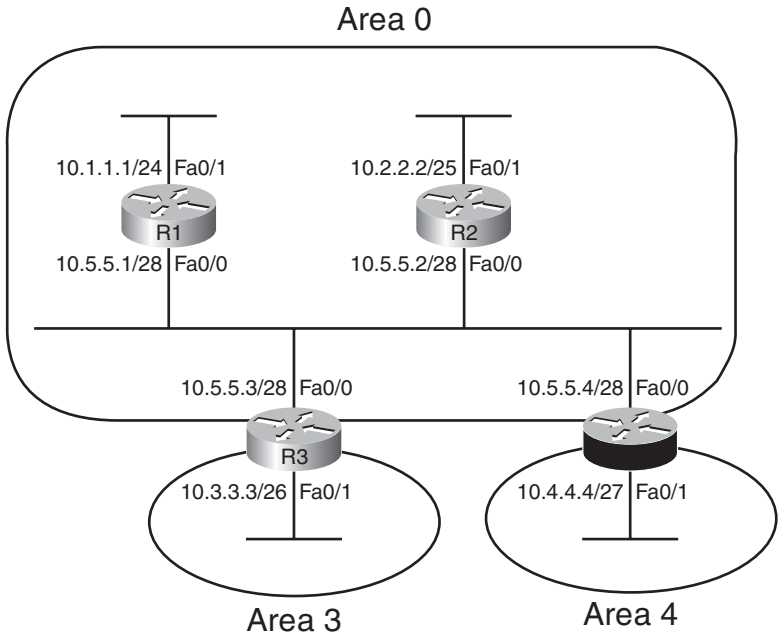


Figure 6-3 *Small Network, Four Routers, on a LAN*

OSPF cannot represent the idea of four routers connected via a single subnet by using a link connected to all four routers. Instead, OSPF defines the Type 2 network LSA, used as a *pseudonode*. Each router's Type 1 router LSA lists a connection this pseudonode, often called a *transit network*, which is then modeled by a Type 2 network LSA. The Type 2 network LSA itself then lists references back to each Type 1 router LSA connected to it–four in this example, as shown in Figure 6-4.

The elected DR in a subnet creates the Type 2 LSA for that subnet. The DR identifies the LSA by assigning an LSID of the DR's interface IP address in that subnet. The type 2 LSA also lists the DR's RID as the router advertising the LSA.

Type 2 LSA show Commands

To see these concepts in the form of OSPF **show** commands, next consider area 34 back in Figure 6-1. This design shows that R3 and R4 connect to the same LAN, which means that

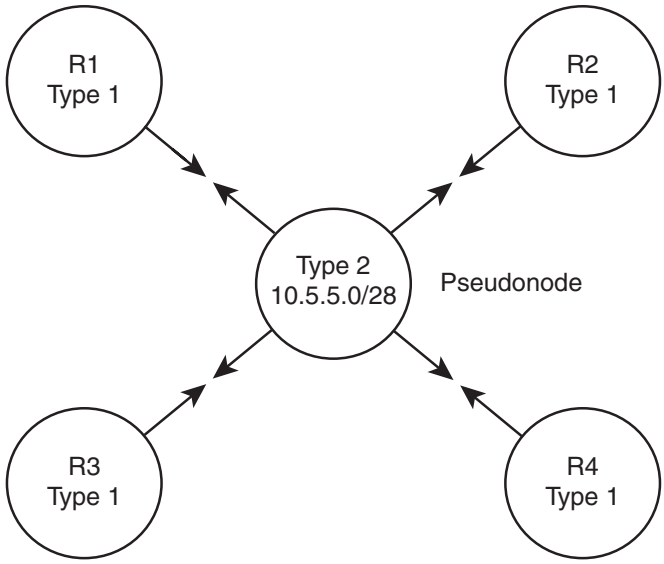


Figure 6-4 *OSPF Topology when Using a Type 2 Network LSA*

a DR will be elected. (OSPF elects a DR on LANs when at least two routers pass the neighbor requirements and can become neighbors.) If both R3 and R4 default to use priority 1, then R4 wins the election, due to its 4.4.4.4 RID (versus R3's 3.3.3.3). So, R4 creates the Type 2 LSA for that subnet and floods the LSA. Figure 6-5 depicts the area 34 topology, and Example 6-4 shows the related LSDB entries.

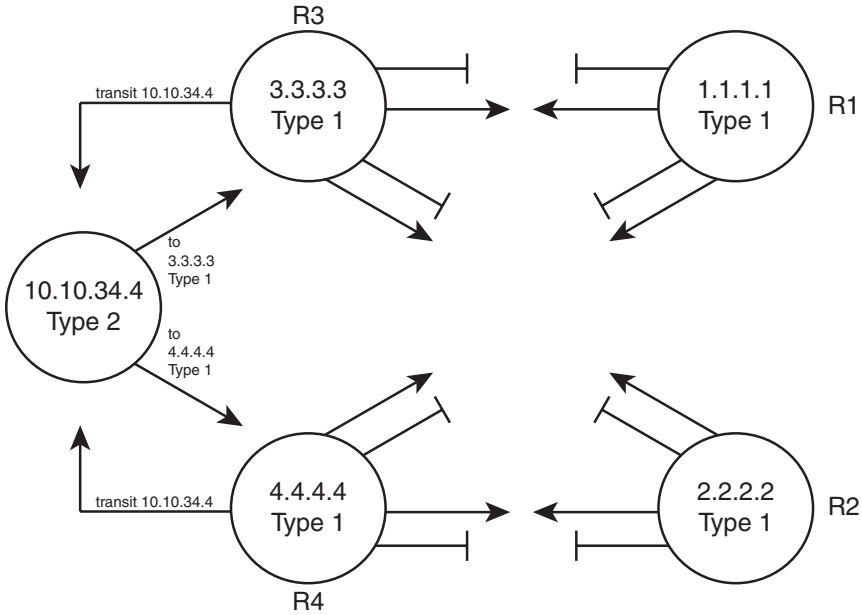


Figure 6-5 *Area 34 Topology with Four Type 1 LSAs and One Type 2 LSA*

Example 6-4 *Area 34 LSAs for R4, Network 10.10.34.0/24*

```
R5#show ip ospf database

            OSPF Router with ID (3.3.3.3) (Process ID 3)

                Router Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1061        0x80000002 0x00EA7A 4
2.2.2.2         2.2.2.2         1067        0x80000001 0x0061D2 4
3.3.3.3         3.3.3.3         1066        0x80000003 0x00E2E8 5
4.4.4.4         4.4.4.4         1067        0x80000003 0x007D3F 5

                Net Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.34.4      4.4.4.4         1104        0x80000001 0x00AB28

                Summary Net Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.5.0       1.1.1.1         1023        0x80000001 0x000BF2
10.10.5.0       2.2.2.2         1022        0x80000001 0x00EC0D
! lines omitted for brevity

R3#show ip ospf database router 4.4.4.4

            OSPF Router with ID (3.3.3.3) (Process ID 3)

                Router Link States (Area 34)

  LS age: 1078
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 4.4.4.4
  Advertising Router: 4.4.4.4
  LS Seq Number: 80000003
  Checksum: 0x7D3F
  Length: 84
```

```
  Number of Links: 5

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 2.2.2.2
     (Link Data) Router Interface address: 10.10.24.4
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.10.24.0
     (Link Data) Network Mask: 255.255.255.248
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 1.1.1.1
     (Link Data) Router Interface address: 10.10.14.4
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.10.14.0
     (Link Data) Network Mask: 255.255.255.248
      Number of TOS metrics: 0
       TOS 0 Metrics: 64

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.10.34.4
     (Link Data) Router Interface address: 10.10.34.4
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

R3#show ip ospf database network 10.10.34.4

            OSPF Router with ID (3.3.3.3) (Process ID 3)

                Net Link States (Area 34)

  Routing Bit Set on this LSA
  LS age: 1161
  Options: (No TOS-capability, DC)
  LS Type: Network Links
  Link State ID: 10.10.34.4 (address of Designated Router)
  Advertising Router: 4.4.4.4
```

```
LS Seq Number: 80000001
Checksum: 0xAB28
Length: 32
Network Mask: /24
      Attached Router: 4.4.4.4
      Attached Router: 3.3.3.3
```

The **show ip ospf database** command lists a single line for each LSA. Note that the (highlighted) heading for network LSAs lists one entry, with LSID 10.10.34.4, which is R4's Fa0/0 IP address. The LSID for Type 2 Network LSAs is the interface IP address of the DR that creates the LSA.

The **show ip ospf database router 4.4.4.4** command shows the new style of entry for the reference to a *Transit Network*, which again refers to a connection to a Type 2 LSA. The output lists a LSID of 10.10.34.4, which again is the LSID of the Type 2 LSA.

Finally, the **show ip ospf database network 10.10.34.4** command shows the details of the Type 2 LSA, based on its LSID of 10.10.34.4. Near the bottom, the output lists the attached routers, based on RID. The SPF process can then use the cross-referenced information, as shown in Figure 6-5, to determine which routers connect to this transit network (pseudonode). The SPF process has information in both the Type 1 LSAs that refer to the transit network link to a Type 2 LSA, and the Type 2 LSA has a list of RIDs of Type 1 LSAs that connect to the Type 2 LSA, making the process of modeling the network possible.

OSPF can model all the topology inside a single area using Type 1 and 2 LSAs. When a router uses its SPF process to build a model of the topology, it can then calculate the best (lowest cost) route for each subnet in the area. The next topic completes the LSA picture for internal OSPF routes by looking at Type 3 LSAs, which are used to model interarea routes.

LSA Type 3: Summary LSA

OSPF areas exist in part so that engineers can reduce the consumption of memory and compute resources in routers. Instead of having all routers, regardless of area, know all Type 1 and Type 2 LSAs inside an OSPF domain, ABRs do not forward Type 1 and Type 2 LSAs from one area into another area, and vice versa. This convention results in smaller per-area LSDBs, saving memory and reducing complexity for each run of the SPF algorithm, which saves CPU and improves convergence time.

However, even though ABRs do not flood Type 1 and Type 2 LSAs into other areas, routers still need to learn about subnets in other areas. OSPF advertises these interarea routes using the Type 3 summary LSA. ABRs generate a Type 3 LSA for each subnet in one area, and advertises each Type 3 LSA into the other areas.

For example, if subnet A exists in area 3, then the routers in area 3 learn of that subnet as part of Type 1 and Type 2 LSAs. However, an ABR connected to area 3 will not forward

the Type 1 and Type 2 LSAs into other areas, instead creating a Type 3 LSA for each subnet (including subnet A). The routers inside the other areas can then calculate a route for the subnets (like subnet A) that exist inside another area.

Type 3 summary LSAs do not contain all the detailed topology information, so in comparison to Types 1 and 2, these LSAs summarize the information–hence the name *summary LSA*. Conceptually, a Type 3 LSA appears to be another subnet connected to the ABR that created and advertised the Type 3 LSA. The routers inside that area can calculate their best route to reach the ABR, which gives the router a good loop-free route to reach the subnet listed in a Type 3 LSA.

An example can certainly help in this case. First, consider the comparison shown in the top and bottom of Figure 6-6. The top depicts the topology shown back in Figure 6-1 if that design had used a single area. In that case, every router would have a copy of each Type 1 LSA (shown as a router name in the figure), and each Type 2 (abbreviated as T2 in the figure). The bottom of Figure 6-6 shows the area 5 topology, when holding to the three area design shown in Figure 6-1.

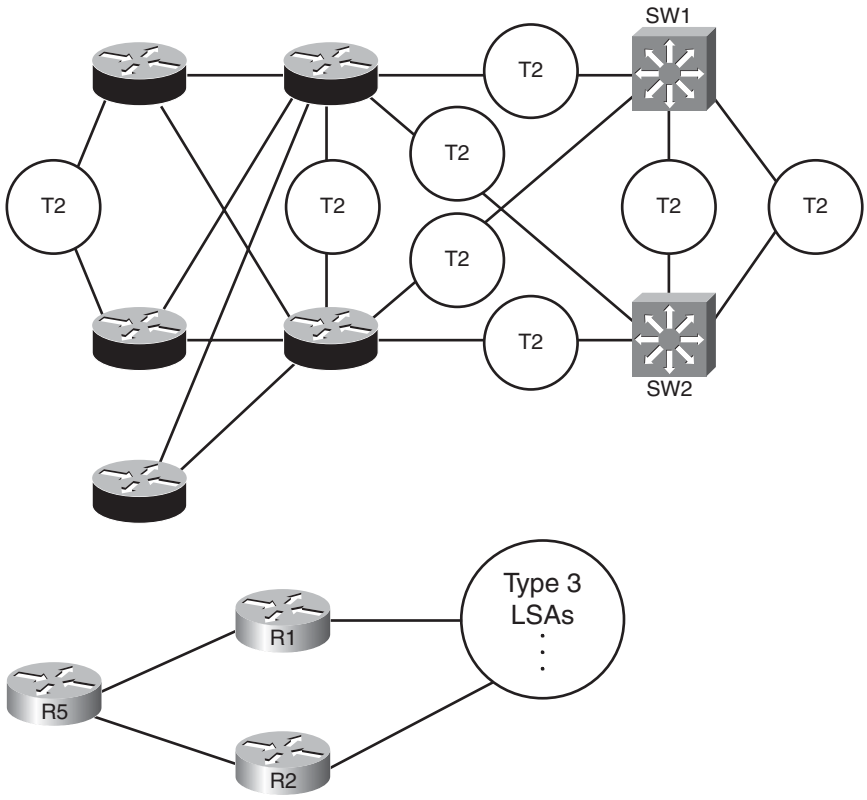


Figure 6-6 *Comparing a Single Area LSDB to a Three Area LSDB*

The ABR creates and floods each Type 3 LSA into the next area. The ABR assigns an LSID of the subnet number being advertised. It also adds its own RID to the LSA as well, so that routers know which ABR advertised the route. It also includes the subnet mask. The correlation between the advertising router's RID and the LSID (subnet number) allows the OSPF processes to create the part of the topology as shown with Type 3 LSAs at the bottom of Figure 6-6.

Example 6-5 focuses on the Type 3 LSAs in Area 34 of the network shown in Figure 6-1. Ten subnets exist outside area 34. As ABRs, both R1 and R2 create and flood a Type 3 LSA for each of these 10 subnets, resulting in 20 Type 3 LSAs listed in the output of the **show ip ospf database** command inside area 34. Then, the example focuses specifically on the Type 3 LSA for subnet 10.10.99.0/24.

Example 6-5 *Type 3 LSAs in Area 34*

```
R3#show ip ospf database



            OSPF Router with ID (3.3.3.3) (Process ID 3)

                Router Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         943         0x80000003 0x00E87B 4
2.2.2.2         2.2.2.2         991         0x80000002 0x005FD3 4
3.3.3.3         3.3.3.3         966         0x80000004 0x00E0E9 5
4.4.4.4         4.4.4.4         977         0x80000004 0x007B40 5

                Net Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.34.4      4.4.4.4         977         0x80000002 0x00A929

                Summary Net Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum
10.10.5.0       1.1.1.1         943         0x80000002 0x0009F3
10.10.5.0       2.2.2.2         991         0x80000002 0x00EA0E
10.10.12.0      1.1.1.1         943         0x80000002 0x00F323
10.10.12.0      2.2.2.2         991         0x80000002 0x00D53D
10.10.15.0      1.1.1.1         943         0x80000002 0x0021BA
10.10.15.0      2.2.2.2         993         0x80000003 0x008313
10.10.17.0      1.1.1.1         946         0x80000002 0x00BC55
10.10.17.0      2.2.2.2         993         0x80000002 0x00A864
10.10.18.0      1.1.1.1         946         0x80000002 0x00B15F
10.10.18.0      2.2.2.2         994         0x80000002 0x009D6E
```

```
10.10.25.0      1.1.1.1         946         0x80000002 0x00355C
10.10.25.0      2.2.2.2         993         0x80000002 0x009439
10.10.27.0      1.1.1.1         946         0x80000002 0x0058AE
10.10.27.0      2.2.2.2         993         0x80000002 0x0030D3
10.10.28.0      1.1.1.1         947         0x80000002 0x004DB8
10.10.28.0      2.2.2.2         993         0x80000002 0x0025DD
10.10.98.0      1.1.1.1         946         0x80000002 0x004877
10.10.98.0      2.2.2.2         993         0x80000002 0x002A91
10.10.99.0      1.1.1.1         946         0x80000002 0x003D81
10.10.99.0      2.2.2.2         993         0x80000002 0x001F9B

R3#show ip ospf database summary 10.10.99.0

            OSPF Router with ID (3.3.3.3) (Process ID 3)

                Summary Net Link States (Area 34)

  Routing Bit Set on this LSA
  LS age: 1062
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.10.99.0 (summary Network Number)
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000002
  Checksum: 0x3D81
  Length: 28
  Network Mask: /24
        TOS: 0  Metric: 2

  Routing Bit Set on this LSA
  LS age: 1109
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.10.99.0 (summary Network Number)
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000002
  Checksum: 0x1F9B
  Length: 28
  Network Mask: /24
        TOS: 0  Metric: 2
```

Note: The Type 3 Summary LSA is not used for the purpose of route summarization. OSPF does support route summarization, and Type 3 LSAs may indeed advertise such a summary, but the Type 3 LSA does not inherently represent a summary route. The term

summary reflects the idea that the information is sparse compared to the detail inside Type 1 and Type 2 LSAs.

The upcoming section "Calculating the Cost of Inter-area Routes" discusses how a router determines the available routes to reach subnets listed in a Type 3 LSA and how a router chooses which route is best.

Limiting the Number of LSAs

By default, Cisco IOS does not limit the number of LSAs a router can learn. However, it may be useful to protect a router from learning too many LSAs to protect router memory. Also, with a large number of LSAs, the router may be unable to process the LSDB with SPF well enough to converge in a reasonable amount of time.

The maximum number of LSAs learned from other routers can be limited by a router using the **max-lsa** *number* OSPF subcommand. When configured, if the router learns more than the configured number of LSAs from other routers (ignoring those created by the router itself), the router reacts. The first reaction is to issue log messages. The router ignores the event for a time period, after which the router repeats the warning message. This ignore-and-wait strategy can proceed through several iterations, ending when the router closes all neighborships, discards its LSDB, and then starts adding neighbors again. (The ignore time, and the number of times to ignore the event, can be configured with the **max-lsa** command.)

Summary of Internal LSA Types

OSPF uses Type 1, 2, and 3 LSAs to calculate the best routes for all routes inside the OSPF routing domain. Later, Chapter 9 explains Types 4, 5, and 7, which OSPF uses to calculate routes for external routes–routes redistributed into OSPF.

Table 6-3 summarizes some of the key points regarding OSPF Type 1, 2, and 3 LSAs. In particular for the ROUTE exam, the ability to sift through the output of various **show ip ospf database** commands can be important. Knowing what the OSPF LSID represents can help you interpret the output, and knowing the keywords used with the **show ip ospf database** *lsa-type lsid* commands can also be very useful. Table 6-3 summarizes these details.

Table 6-3 *Facts about LSA Types 1, 2, and 3*

LSA Type (Number)	LSA Type (Name)	This Type Represents	Display Using show ip ospf database *keyword*...	LSID Is Equal To	Created By
1	Router	A router	**router**	RID of router	Each router creates its own

Table 6-3 *Facts about LSA Types 1, 2, and 3*

LSA Type (Number)	LSA Type (Name)	This Type Represents	Display Using show ip ospf database *keyword*...	LSID Is Equal To	Created By
2	Network	A subnet in which a DR exists	**network**	DR's IP address in the subnet	The DR in that subnet
3	Summary	Subnet in another area	**summary**	Subnet number	An ABR

The Database Exchange Process

Every router in an area, when OSPF stabilizes after topology changes occur, should have an identical LSDB for that area. Internal routers (routers inside a single area) have only that area's LSAs, but an ABR's LSDB will contain LSAs for each area to which it connects. The ABR does, however, know which LSAs exist in each area.

OSPF routers flood both the LSAs they create, and the LSAs they learn from their neighbors, until all routers in the area have a copy of each of the most recent LSAs for that area. To manage and control this process, OSPF defines several messages, processes, and neighbor states that indicate the progress when flooding LSAs to each neighbor. This section begins by listing reference information for the OSPF messages and neighbor states. Next, the text describes the flooding process between two neighbors when a DR does not exist, followed by a description of the similar process used when a DR does exist. This section ends with a few items related to how the routers avoid looping the LSA advertisements and how they periodically reflood the information.

OSPF Message and Neighbor State Reference

For reference, Table 6-4 lists the OSPF message types that will be mentioned in the next few pages. Additionally, Table 6-5 lists the various neighbor states as well. Although useful for study, when you first learning this topic, feel free to skip these tables for now.

Key Topic

Table 6-4 *OSPF Message Types and Functions*

Message Name/number	Description
Hello (1)	Used to discover neighbors, supply information used to confirm two routers should be allowed to become neighbors, to bring a neighbor relationship to a 2-way state, and to monitor a neighbor's responsiveness in case it fails
Database Description (DD or DBD) (2)	Used to exchange brief versions of each LSA, typically on initial topology exchange, so that a router knows a list of that neighbor's known LSAs

Table 6-4 *OSPF Message Types and Functions*

Message Name/number	Description
Link-State Request (LSR) (3)	A packet that lists the LSIDs of LSAs the sender of the LSR would like the receiver of the LSR to supply during database exchange
Link-State Update (LSU) (4)	A packet that contains fully detailed LSAs, typically sent in response to an LSR message
Link-State Acknowledgment (LSAck) (5)	Sent to confirm receipt of an LSU message

Table 6-5 *OSPF Neighbor State Reference*

State	Meaning
Down	No Hellos have been received from this neighbor for more than the dead interval.
Attempt	Used when the neighbor is defined with the **neighbor** command, after sending a Hello, but before receiving a Hello from that neighbor.
Init	A Hello has been received from the neighbor, but it did not have the local router's RID in it or lists parameters that do not pass the neighbor verification checks. This is a permanent state when Hello parameters do not match.
2Way	A Hello has been received from the neighbor, it has the router's RID in it, and all neighbor verification checks passed.
ExStart	Currently negotiating the DD sequence numbers and master/slave logic used for DD packets.
Exchange	Finished negotiating the DD process particulars, and currently exchanging DD packets.
Loading	All DD packets are exchanged, and the routers are currently sending LSR, LSU, and LSAck packets to exchange full LSAs.
Full	Neighbors are fully adjacent, meaning they believe that their LSDBs for that area are identical. Routing table (re)calculations can begin.

Exchange Without a Designated Router

As discussed in Chapter 5, the OSPF interface network type tells that router whether to attempt to elect a DR on that interface. The most common case for which routers do not elect a DR occur on point-to-point topologies, such as true point-to-point serial links and point-to-point subinterfaces. This section examines the database exchange process on such interfaces, in preparation for the slightly more complex process when using a DR on an OSPF broadcast network type, like a LAN.

Every OSPF neighborship begins by exchanging Hellos until the neighbors (hopefully) reach the 2-Way state. During these early stages, the routers discover each other by sending multicast Hellos and then check each other's parameters to make sure all items match

(as listed in Chapter 5's Table 5-5). Figure 6-7 shows the details, with the various neighbor states listed on the outside of the figure and the messages listed in the middle.

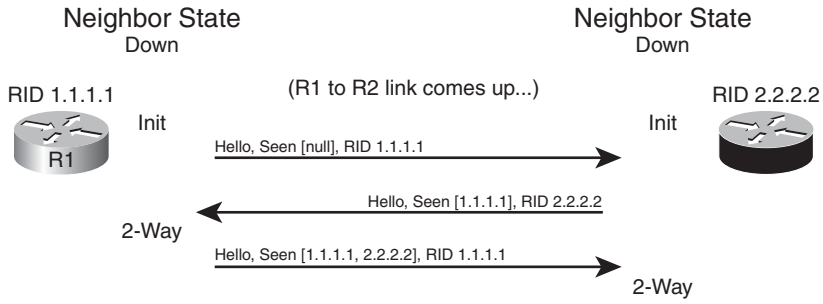


Figure 6-7 *Neighbor Initialization–Early Stages*

Figure 6-7 shows an example that begins with a failed neighborship, so the neighborship is in a down state. When a router tries to reestablish the neighborship, each router sends a multicast Hello and moves to an INIT state. After a router has both received a Hello and verified that all the required parameters agree, the router lists the other router's RID in the Hello as being seen, as shown in the bottom two Hello messages in the figure. When a router receives a Hello that lists its own RID as having been seen by the other router, the router can transition to 2-Way state.

When a router has reached the 2-Way state with a neighbor, as shown at the end of Figure 6-7, the router then decides whether it should exchange its LSDB entries. When no DR exists, the answer is always "yes." Each router next follows this general process:

Step 1. Discover the LSAs known to the neighbor but unknown to me.

Step 2. Discover the LSAs known by both routers, but the neighbor's LSA is more up to date.

Step 3. Ask the neighbor for a copy of all the LSAs identified in the first two steps.

Figure 6-8 details the messages and neighbor states used to exchange the LSAs between two neighbors.

Figure 6-8 shows many details. As with Figure 6-7, Figure 6-8 shows neighbor states on the outer edges of the flows (refer to Table 6-5 for reference). Routers display these neighbor states (in the **show ip ospf neighbor** command variants), so a particular state may be useful in determining how far two neighbors have gotten in the database exchange process. The more important neighbor states will be mentioned throughout the chapter.

The inner portions of Figure 6-8 represent the OSPF message flows, with Table 6-2 earlier in the chapter listing the messages for reference. The next several pages examine the process shown in Figure 6-8 in more detail.

Discovering a Description of the Neighbor's LSDB

After a router has decided to move forward from 2-Way state and exchange its LSDB with a neighbor, the routers use the sequence shown in Figure 6-8. The next step in that process

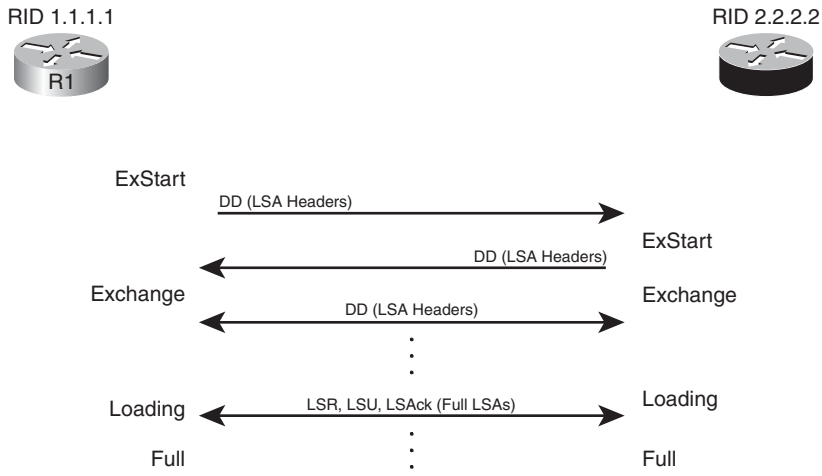


Figure 6-8 *Overview of the Database Exchange Process Between Two Neighbors*

requires both routers to tell each other the LSIDs of all their known LSAs in that area. The primary goal is for each neighbor to realize which LSAs it does not know, so it can then ask for those full LSAs to be sent. To learn the list of LSAs known by the neighbor, the neighboring routers follow these steps:

Step 1. Multicast database description packets (abbreviated as both DD and DBD, depending on the reference) to 224.0.0.5, which is the all SPF routers multicast address.

Step 2. When sending the first DD message, transition to the *ExStart* state until one router, the one with the higher RID, becomes master in a master/slave relationship.

Step 3. After electing a master, transition the neighbor to the *Exchange* state.

Step 4. Continue multicasting DD messages to each other until both routers have the same shared view of the LSIDs known collectively by both routers, in that area.

Note that the DD messages themselves do not list the entire LSAs, but rather LSA headers. These headers include the LSIDs of the LSAs and the LSA sequence number. The LS sequence number for an LSA begins at value 0x80000001 (hex) when initially created; the router creating the LSA increments the sequence number, and refloods the LSA, whenever the LSA changes. For example, if an interface moves from up to down state, that router changes its Type 1 LSA to list that interface state as down, increments the LSA sequence number, and refloods the LSA.

The master router for each exchange controls the flow of DD messages, with the slave responding to the master's DD messages. The master keeps sending DD messages until it lists all its known LSIDs in that area. The slave responds by placing LSA headers in its DD messages. Some of those LSA headers simply repeat what the slave heard from the master, for the purpose of acknowledging to the master that the slave learned that LSA header

from the master. Additionally, the slave also includes the LSA headers for any LSAs that the master did not list.

This exchange of DD messages ends with each router knowing a list of LSAs that it does not have in its LSDB, but that the other router does have those LSAs. Additionally, each router also ends this process with a list of LSAs that the local router already knows, but for which the other router has a more recent copy (based on the sequence number).

Exchanging the LSAs

When the two neighbors realize that they have a shared view of the list of LSIDs, they transition to the Loading state and start exchanging the full LSAs–but only those that they do not yet know about or those that have changed.

For example, when the two routers in Figure 6-8 first become neighbors, neither router will have a copy of the Type 1 LSA for the other router. So, R1 will request that R2 send its LSA with LSID 2.2.2.2; R2 will send its Type 1 LSA; and R1 will acknowledge receipt. The mechanics work like this:

Step 1. Transition the neighbor state to Loading.

Step 2. For any missing LSAs, send a *Link State Request* (LSR) message, listing the LSID of the requested LSA.

Step 3. Respond to any LSR messages with an *Link State Update* (LSU), listing one or more LSAs in each message.

Step 4. Acknowledge receipt by either sending a *Link State Acknowledgment* (LSAck) message (called explicit acknowledgment), or by sending the same LSA that was received back to the other router in an LSU message (implicit acknowledgment).

Step 5. When all LSAs have been sent, received, and acknowledged, transition the neighborship to the FULL state (fully adjacent).

Note: Because this section examines the case without a DR, all these messages flow as multicasts to 224.0.0.5, the all SPF routers multicast address, unless the neighbors have been defined with an OSPF **neighbor** command.

By the end of this process, both routers should have an identical LSDB for the area to which the link has been assigned. At that point, the two routers can run the SPF algorithm to choose the currently best routes for each subnet.

Exchange with a Designated Router

Database exchange with a DR differs slightly than database exchange when no DR exists. The majority of the process is similar, with the same messages, meanings, and neighbor states. The big difference is the overriding choice of with whom each router chooses to perform database exchange.

Non-DR routers do not exchange their databases directly with all neighbors on a subnet. Instead, they exchange their database with the DR. Then, the DR exchanges any new/changed LSAs with the rest of the OSPF routers in the subnet.

The concept actually follows along with the idea of a Type 2 LSA as seen earlier in Figure 6-4. Figure 6-9 represents four Type 1 LSAs, for four real routers on the same LAN, plus a single Type 2 LSA that represents the multiaccess subnet. The DR created the Type 2 LSA as part of its role in life.

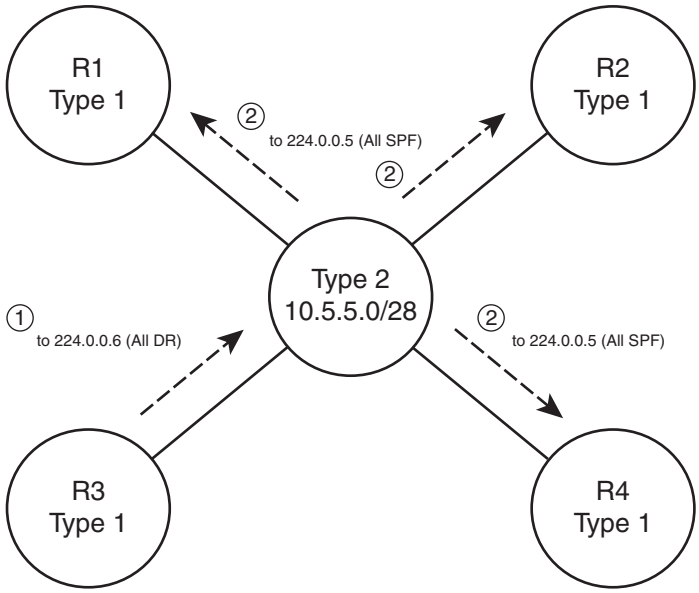


Figure 6-9 *Conceptual View–Exchanging the Database with a Pseudonode*

Figure 6-9 shows two conceptual steps for database exchange. The non-DR router (R3) first exchanges its database with the pseudonode, and then the type 2 pseudonode exchanges its database with the other routers. However, the pseudonode is a concept, not a router; to make the process depicted in Figure 6-9 work, the DR takes on the role of the Type 2 pseudonode. The messages differ slightly as well, as follows:

- The non-DR performs database exchange with the same messages, as shown in Figure 6-9, but sends these messages to the 224.0.0.6 All-DR-routers multicast address.
- The DR performs database exchange with the same messages but sends the messages to the 224.0.0.5 all-SPF-routers multicast address.

Key Topic

Consider these two conventions one at a time. First, the messages sent to 224.0.0.6 are processed by the DR and the BDR only. The DR actively participates, replying to the messages, with the BDR acting as a silent bystander. In effect, this allows the non-DR router to exchange their database directly with the DR and BDR, but with none of the other routers in the subnet.

Next, consider the multicast messages from the DR to the 224.0.0.5 all-SPF-router multicast address. All OSPF routers process these messages, so the rest of the routers—the DROthers to use the IOS term—also learn the newly exchanged LSAs. This process completes the second step shown in the conceptual Figure 6-9, where the DR, acting like the pseudonode, floods the LSAs to the other OSPF routers in the subnet.

The process occurs in the background and can be generally ignored. However, for operating an OSPF network, an important distinction must be made. With a DR in existence, a DROther router performs the database exchange process (as seen in Figure 6-9) with the DR/BDR only and not any other DROther routers in the subnet. For example, in Figure 6-9, R1 acts as DR, R2 acts as BDR, and R3/R4 act as DROther routers. Because the underlying process does not make R3 and R4 perform database exchange with each other, the routers do not reach the FULL neighbor state, remaining in 2-Way state.

Example 6-6 shows the resulting output for the LAN shown in Figure 6-9, with four routers. The output, taken from DROther R3, shows a 2-Way state with R4, the other DROther. It also shows on interface Fa0/0 that its own priority is 1. This output also shows a neighbor count (all neighbors) of 3 and an adjacent neighbor count (all fully adjacent neighbors) of 2, again because the neighborship between DROthers R3 and R4 is not a full adjacency.

Example 6-6 *Demonstrating OSPF FULL and 2-Way Adjacencies*

```
R3#show ip ospf interface fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.16.1.3/24, Area 0
  Process ID 75, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DROTHER, Priority 2
  Designated Router (ID) 1.1.1.1, Interface address 172.16.1.1
  Backup Designated router (ID) 2.2.2.2, Interface address 172.16.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 1.1.1.1  (Designated Router)
    Adjacent with neighbor 2.2.2.2  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

```
R3#show ip ospf neighbor fa0/0

Neighbor ID      Pri   State           Dead Time   Address         Interface
1.1.1.1            4   FULL/DR         00:00:37    172.16.1.1
FastEthernet0/0
2.2.2.2            3   FULL/BDR        00:00:37    172.16.1.2
FastEthernet0/0
44.44.44.44        1   2WAY/DROTHER    00:00:36    172.16.1.4      FastEthernet0/0
```

Flooding Throughout the Area

So far in this section, the database exchange process has focused on exchanging the database between neighbors. However, LSAs need to be flooded throughout an area. To do so, when a router learns new LSAs from one neighbor, that router then knows that its other neighbors in that same area may not know of that LSA. Similarly, when an LSA changes, for example, when an interface changes state, a router may learn the same old LSA but with a new sequence number, and again need to flood the changed LSA to other neighbors in that area.

Figure 6-10 shows a basic example of the process. In this case, R2, R3, and R4 have established neighbor relationships, with four LSAs in their LSDB in this area. R1 is again the new router added to the internetwork.

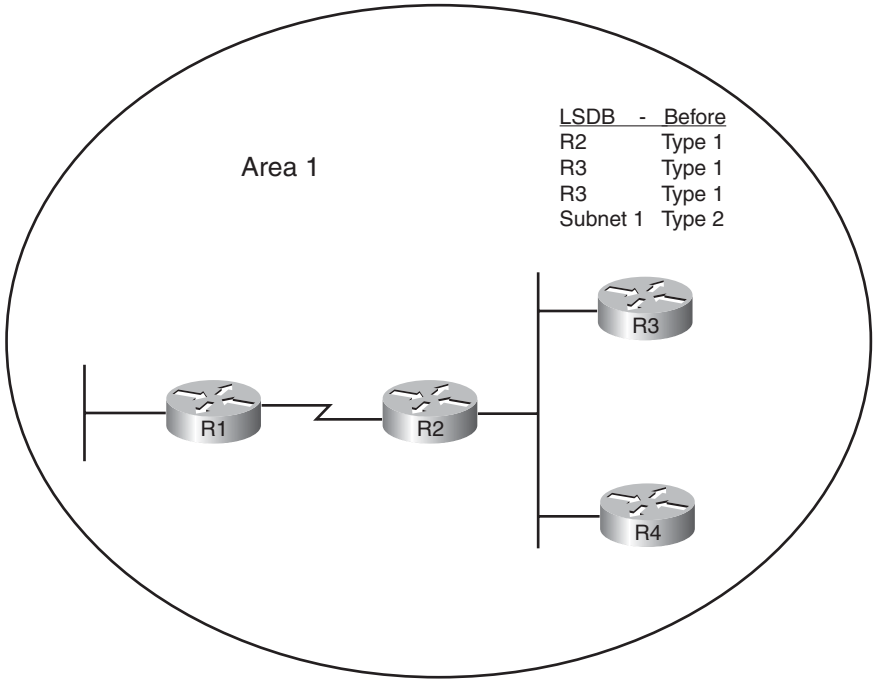


Figure 6-10 *Flooding Throughout an Area*

First, consider what happens as the new R1-R2 neighborship comes up and goes through database exchange. When R1 loads, and the link comes up, R1 and R2 reach a full state and have a shared view of the Area 1 LSDB. R2 has learned all R1's new LSAs (should only be R1's Type 1 router LSA), and R1 has learned all the area 1 LSAs known to R2, including the Type 1 LSAs for R3 and R4.

Next, think about the LSDBs of R3 and R4 at this point. The database exchange between R1-R2 did not inform R3 nor R4 about any of the new LSAs known by R1. So, R2, when it learns of R1's Type 1 LSA, sends DD packets to the DR on the R2/R3/R4 LAN. LSR/LSU packets follow, resulting in R3 and R4 learning about the new LSA for R1. If more routers existed in area 1, the flooding process would continue throughout the entire area, until all routers know of the best (highest sequence number) copy of each LSA.

The flooding process prevents the looping of LSAs as a side-effect of the database exchange process. Neighbors use DD messages to learn the LSA headers known by the neighbor, and then only request the LSAs known by the neighbor but not known by the local router. By requesting only unknown LSAs or new versions of old LSAs, routers prevent the LSAs advertisements from looping.

Periodic Flooding

Although OSPF does not send routing updates on a periodic interval, as do distance vector protocols, OSPF does reflood each LSA every 30 minutes based on each LSA's age variable. The router that creates the LSA sets this age to 0 (seconds). Each router then increments the age of its copy of each LSA over time. If 30 minutes pass with no changes to an LSA–meaning no other reason existed in that 30 minutes to cause a reflooding of the LSA–the owning router increments the sequence number, resets the timer to 0, and refloods the LSA.

Because the owning router increments the sequence number and resets the LSAge every 1800 seconds (30 minutes), the output of various **show ip ospf database** commands should also show an age of less than 1800 seconds. For example, referring back to Example 6-5, the Type 1 LSA for R1 (RID 1.1.1.1) shows an age of 943 seconds and a sequence number of 0x80000003. Over time the sequence number should increment once per every 30 minutes, with the LSAge cycle upward toward 1800 and then back to 0 when the LSA is reflooded.

Note also that when a router realizes it needs to flush an LSA from the LSDB for an area, it actually sets the age of the LSA to the MaxAge setting (3600) and refloods the LSA. All the other routers receive the LSA, see the age is already at the maximum, causing those routers to also remove the LSA from their LSDBs.

Choosing the Best OSPF Routes

All this effort to define LSA types, create areas, and fully flood the LSAs has one goal in mind: to allow all routers in that area to calculate the best, loop-free routes for all known subnets. Although the database exchange process may seem laborious, the process by which SPF calculates the best routes requires a little less thought, at least to the level re-

quired for the CCNP ROUTE exam. In fact, the choice of the best route for a given subnet, and calculated by a particular router, can be summarized as follows:

- Analyze the LSDB to find all possible routes to reach the subnet.
- For each possible route, add the OSPF interface cost for all outgoing interfaces in that route.
- Pick the route with the lowest total cost.

Key Topic

For humans, if you build a network diagram, note the OSPF cost for each interface (as shown with **show ip ospf interface**), you can easily add up the costs for each router's possible routes to each subnet and tell which route OSPF will choose. The routers must use a more complex SPF algorithm to derive a mathematical model of the topology based on the LSAs. This section examines both the simpler human view of metric calculation and folds in some of the basics of what SPF must do on a router to calculate the best routes. It also goes through the options for tuning the metric calculation to influence the choice of routes.

OSPF Metric Calculation for Internal OSPF Routes

The process of calculating the cost from a router to each subnet may be intuitive to most people. However, spending a few minutes considering the details is worthwhile, in part to link the concepts with the LSAs, and to be better prepared for questions on the ROUTE exam. This section breaks the discussion into three sections: intra-area routes, interarea routes, a short discussion about cases when both intra-area and interarea routes exist for the same subnet, and an explanation of SPF calculations.

Calculating the Cost of Intra-Area Routes

When a router analyzes the LSDB to calculate the best route to each subnet, it does the following:

Step 1. Finds all subnets inside the area, based on the stub interfaces listed in the Type 1 LSAs and based on any Type 2 network LSAs

Step 2. Runs SPF to find all possible paths through the area's topology, from itself to each subnet

Step 3. Calculates the OSPF interface costs for all outgoing interfaces in each route, picking the lowest total cost route for each subnet as the best route

Key Topic

For example, Figure 6-11 shows the routers and links inside area 34, as a subset of the internetwork also shown in Figure 6-1. Figure 6-11 shows the interface numbers and OSPF costs.

Following the basic three-step process, at Step 1, R1 can determine that subnet 10.10.34.0/24 exists in area 34 because of the Type 2 LSA created by the DR in that subnet. For Step 2, R1 can then run SPF and determine four possible routes, two of which are clearly more reasonable to humans: R1-R3 and R1-R4. (The two other possible routes, R1-R3-R2-R4 and R1-R4-R2-R3, are possible and would be considered by OSPF but would clearly be higher cost.) For Step 3, R1 does the simple math of adding the costs of the outgoing interfaces in each route, as follows:

- R1-R3: Add R1's S0/0/0.3 cost (647) and R3's Fa0/0 cost (10), total 657
- R1-R4: Add R1's S0/0/0.4 cost (647) and R4's Fa0/0 cost (10), total 657

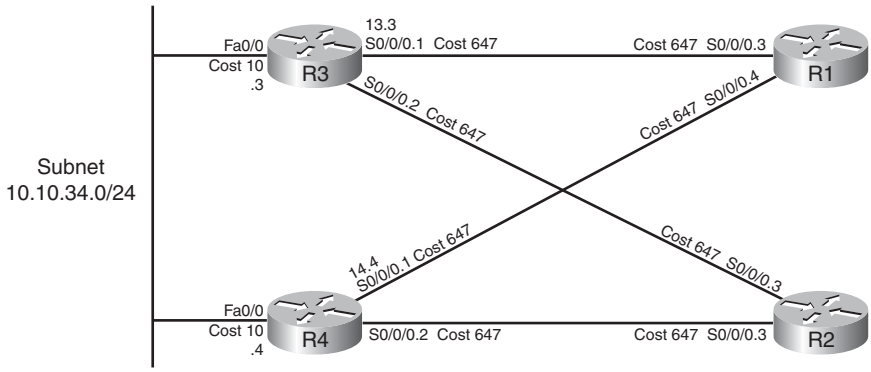


Figure 6-11 *Area 34 Portion of Figure 6-1*

The metrics tie, so with a default setting of **maximum-paths 4**, R1 adds both routes to its routing table. In particular, the routes list the metric of 657, and the next-hop IP address on the other end of the respectively links: 10.10.13.3 (R3's S0/0/0.1) and 10.10.14.4 (R4's S0/0/0.1).

Note that OSPF supports equal-cost load balancing, but it does not support unequal-cost load balancing. The **maximum-paths** OSPF subcommand can be set as low as 1, with the maximum being dependent on router platform and IOS version. Modern IOS versions typically support 16 or 32 concurrent routes to one destination (maximum).

Calculating the Cost of Interarea Routes

From a human perspective, the cost for interarea routes can be calculated just like for intra-area routes if we have the full network diagram, subnet numbers, and OSPF interface costs. To do so, just find all possible routes from a router to the destination subnet, add up the costs of the outgoing interfaces, and choose the router with the lowest total cost.

However, OSPF routers cannot do the equivalent for interarea routes, because routers internal to one area do not have topological data–LSA Types 1 and 2–for other areas. Instead, ABRs create and flood Type 3 summary LSAs into an area, listing the subnet number and mask, but not listing details about routers and links in the other areas. For example, Figure 6-12 shows both Areas 34 and 0 from Figure 6-1, including interface costs. Then consider how OSPF determines the lowest-cost route from router R3 for subnet 10.10.99.0/24, the Data Center subnet on the right.

R3 has a large number of possible routes to reach subnet 10.10.99.0/24. For example, just to get from R3 to R1, there are several possibilities: R3-R1, R3-R4-R1, and R3-R2-R1. From R1 the rest of the way to subnet 10.10.99.0/24, many more possibilities exist. The SPF algorithm has to calculate all possible routes inside an area to the ABR, so with more redundancy, SPF's run time goes up. And SPF has to consider all the options, whereas we humans can rule out some routes quickly because they appear to be somewhat ridiculous.

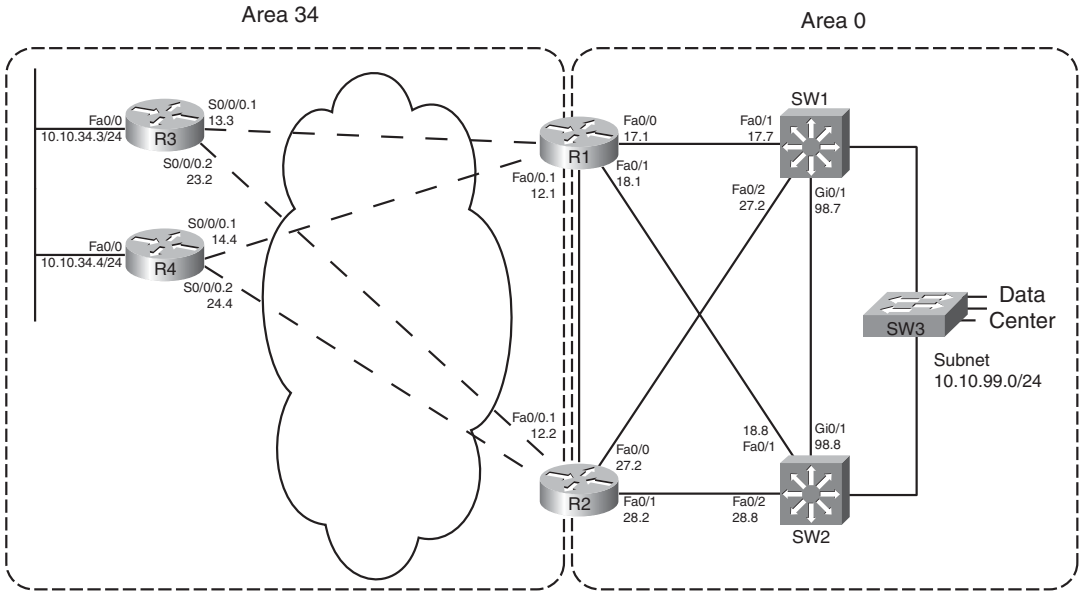


Figure 6-12 *Area 34 and Area 0 Portion of Figure 6-1*

Because of the area design, with R1 and R2 acting as ABRs, R3 does not process all the topology shown in Figure 6-12. Instead, R3 relies on the Type 3 Summary LSAs created by the ABRs, which have the following information:

- The subnet number/mask represented by the LSA
- The cost of the ABR's lowest-cost route to reach the subnet
- The RID of the ABR

Example 6-7 begins to examine the information R3 will use to calculate its best route for subnet 10.10.99.0/24, on the right side of Figure 6-12. To see these details, Example 6-7 lists several commands taken from R1. It lists R1's best route (actually two that tie) for subnet 10.10.99.0/24, with cost 11. It also lists the Type 3 LSA R1 generated by R1 for 10.10.99.0/24, again listing cost 11, and listing the Type 3 LSA created by ABR R2 and flooded into area 34.

Example 6-7 *Route and Type 3 LSA on R1 for 10.10.99.0/24*

```
R1#show ip route ospf
      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
O        10.10.5.0/27 [110/648] via 10.10.15.5, 00:04:19, Serial0/0/0.5
O        10.10.23.0/29 [110/711] via 10.10.13.3, 00:04:19, Serial0/0/0.3
```

```
O        10.10.24.0/29 [110/711] via 10.10.14.4, 00:04:19, Serial0/0/0.4
O        10.10.25.0/29 [110/711] via 10.10.15.5, 00:04:19, Serial0/0/0.5
O        10.10.27.0/24 [110/11] via 10.10.17.7, 00:04:19, FastEthernet0/0
                       [110/11] via 10.10.12.2, 00:04:19, FastEthernet0/0.1
O        10.10.28.0/24 [110/11] via 10.10.18.8, 00:04:19, FastEthernet0/1
                       [110/11] via 10.10.12.2, 00:04:19, FastEthernet0/0.1
O        10.10.34.0/24 [110/648] via 10.10.14.4, 00:04:19, Serial0/0/0.4
                       [110/648] via 10.10.13.3, 00:04:19, Serial0/0/0.3
O        10.10.98.0/24 [110/11] via 10.10.18.8, 00:04:19, FastEthernet0/1
                       [110/11] via 10.10.17.7, 00:04:19, FastEthernet0/0
O        10.10.99.0/24 [110/11] via 10.10.18.8, 00:04:19, FastEthernet0/1
                       [110/11] via 10.10.17.7, 00:04:19, FastEthernet0/0

R1#show ip ospf database summary 10.10.99.0

             OSPF Router with ID (1.1.1.1) (Process ID 1)

! omitting output for area 5...
                 Summary Net Link States (Area 34)

  LS age: 216
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.10.99.0 (summary Network Number)
  Advertising Router: 1.1.1.1

  LS Seq Number: 80000003
  Checksum: 0x951F
  Length: 28
  Network Mask: /24
        TOS: 0  Metric: 11


  LS age: 87
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.10.99.0 (summary Network Number)
  Advertising Router: 2.2.2.2

  LS Seq Number: 80000002
  Checksum: 0x7938
  Length: 28
  Network Mask: /24
       TOS: 0  Metric: 11
```

Note: The examples use default bandwidth settings, but with all routers configured with the **auto-cost reference-bandwidth 1000** command. This command is explained in the upcoming section "Reference Bandwidth."

For routers in one area to calculate the cost of an interarea route, the process is simple when you realize that the Type 3 LSA lists the ABR's best cost to reach that interarea subnet. To calculate the cost:

Step 1. Calculate the intra-area cost from that router to the ABR listed in the type 3 LSA.

Step 2. Add the cost value listed in the Type 3 LSA. (This cost represents the cost from the ABR to the destination subnet.)

Key Topic

A router applies these two steps for each possible route to reach the ABR. Following the example of router R3 and subnet 10.10.99.0/24, Figure 6-13 shows the components of the calculation.

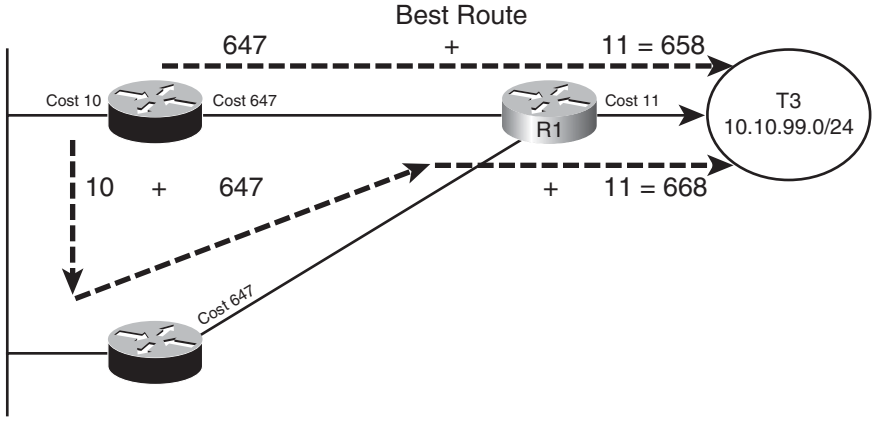


Figure 6-13 *R3's Calculation of Cost for 10.10.99.0/24*

Figure 6-13 shows the calculation of both routes, with intra-area cost to reach R1 either 647 or 657 in this case. For both routes, the cost listed in the Type 3 LSA sourced by R1, cost 11, is added.

When more than one ABR exists, as is the case as shown in Figure 6-12, each ABR should have created a Type 3 LSA for the subnet. In fact, the output in Example 6-7 showed the Type 3 LSA for 10.10.99.0/24 created by both R1 and another created by R2. For instance, in the internetwork used throughout this chapter, ABRs R1 and R2 would create a Type 3 LSA for 10.10.99.0/24. So, in this particular example, R3 would also have to calculate the best route to reach 10.10.99.0/24 through ABR R2. Then, R3 would choose the best route among all routes for 10.10.99.0/24.

Each router repeats this process for all known routes to reach the ABR, considering the Type 3 LSAs from each ABR. In this case, R3 ties on metrics for one route through R1 and one through R2, so R3 adds both routes to its routing table, as shown in Example 6-8.

Example 6-8 *Route and Type 3 LSA on R1 for 10.10.99.0/24*

```
R3#show ip route 10.10.99.0 255.255.255.0
Routing entry for 10.10.99.0/24
  Known via "ospf 3", distance 110, metric 658, type inter area
  Last update from 10.10.13.1 on Serial0/0/0.1, 00:08:06 ago
  Routing Descriptor Blocks:
  * 10.10.23.2, from 2.2.2.2, 00:08:06 ago, via Serial0/0/0.2
      Route metric is 658, traffic share count is 1
    10.10.13.1, from 1.1.1.1, 00:08:06 ago, via Serial0/0/0.1
      Route metric is 658, traffic share count is 1

R3#show ip route ospf
     10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
O IA    10.10.5.0/27 [110/1304] via 10.10.23.2, 00:07:57, Serial0/0/0.2
                     [110/1304] via 10.10.13.1, 00:07:57, Serial0/0/0.1
O IA    10.10.12.0/24 [110/657] via 10.10.23.2, 00:08:17, Serial0/0/0.2
                      [110/657] via 10.10.13.1, 00:08:17, Serial0/0/0.1
! lines omitted for brevity
O IA    10.10.99.0/24 [110/658] via 10.10.23.2, 00:08:17, Serial0/0/0.2
                      [110/658] via 10.10.13.1, 00:08:17, Serial0/0/0.1
```

Besides the information that matches the expected outgoing interfaces per the figures, the output also flags these routes as interarea routes. The first command lists "type inter area" explicitly, and the **show ip route ospf** command lists the same information with the code "O IA," meaning OSPF, interarea. Simply put, interarea routes are routes for which the subnet is known from a Type 3 summary LSA.

Special Rules Concerning Intra-area and Interarea Routes on ABRs

OSPF has a couple of rules concerning intra-area and interarea routes that take precedence over the simple comparison of the cost calculated for the various routes. The issue exists when more than one ABR connects to the same two areas. Many designs use two routers between the backbone and each nonbackbone area for redundancy, so this design occurs in many OSPF networks.

The issue relates to the fact that with two or more ABRs, the ABRsthemselves, when calculating their own routing tables, can calculate both an intra-area route and interarea route for subnets in the backbone area. For example, consider the perspective of Router R1 from the last several examples, as depicted in Figure 6-14.

Conceptually, R1 could calculate both the intra-area route and interarea route to 10.10.99.0/24. However, the OSPF cost settings could be set so that the lower cost route

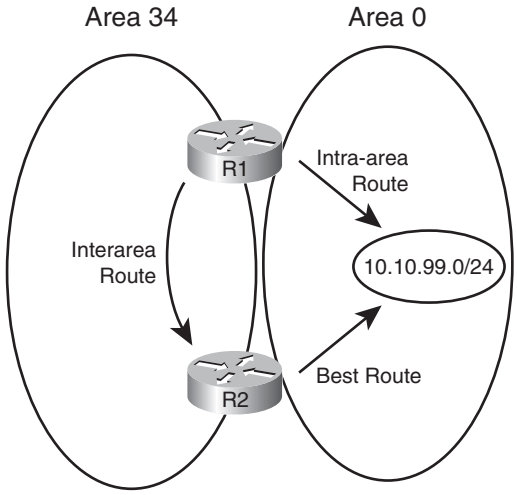


Figure 6-14 *R1's Choice: Intra-Area or Interarea Route to 10.10.99.0/24*

for R1 actually goes through area 34, to ABR R2, and then on through Area 0 to 10.10.99.0/24. However, two OSPF rules prevent such a choice by R1:

Step 1. When choosing the best route, an intra-area route is always better than a competing interarea route, regardless of metric.

Step 2. If an ABR learns a Type 3 LSA inside a nonbackbone area, the ABR ignores that LSA when calculating its own routes.

Because of the first rule, R1 would never choose the interarea route if the intra-area route were available. The second rule goes further, stating that R1 could never choose the interarea route at all–R1 simply ignores that LSA for the purposes of choosing its own best IP routes.

Metric and SPF Calculations

Before moving on to discuss how to influence route choices by changing the OSPF interface costs, first take a moment to consider the CPU-intensive SPF work done by a router. SPF does the work to piece together topology information to find all possible routes to a destination. As a result, SPF must execute when the intra-area topology changes, because changes in topology impact the choice of best route. However, changes to Type 3 LSAs do not drive a recalculation of the SPF algorithm, because the Type 3 LSAs do not actually describe the topology.

To take the analysis a little deeper, remember that an internal router, when finding the best interarea route for a subnet, uses the intra-area topology to calculate the cost to reach the ABR. When each route is identified, the internal router adds the intra-area cost to the ABR, plus the corresponding Type 3 LSA's cost. A change to the Type 3 LSA–it fails, comes back up, or the metric changes–does impact the choice of best route, so the changed Type 3 LSA must be flooded. However, no matter the change, the change does not affect the topology between a router and the ABR—and SPF focuses on processing that topology data. So, only changes to Type 1 and 2 LSAs require an SPF calculation.

You can see the number of SPF runs, and the elapsed time since the last SPF run, using several variations on the **show ip ospf** command. Each time a Type 3 LSA changes and is flooded, SPF does not run, and the counter does not increment. However, each time a Type 1 or 2 LSA changes, SPF runs, and the counter increments. Example 6-9 highlights the counter that shows the number of SPF runs on that router, in that area, and the time since the last run. Note that ABRs list a group of messages per area, showing the number of runs per area.

Example 6-9 *Example with New Route Choices but No SPF Run*

```
R3#show ip ospf  ¦ begin Area 34
    Area 34
         Number of interfaces in this area is 3
         Area has no authentication
         SPF algorithm last executed 00:41:02.812 ago
         SPF algorithm executed 15 times
         Area ranges are
         Number of LSA 25. Checksum Sum 0x0BAC6B
         Number of opaque link LSA 0. Checksum Sum 0x000000
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
```

Metric Tuning

Engineers have a couple of commands available that allow them to tune the values of the OSPF interface cost, thereby influencing the choice of best OSPF route. This section discusses the three methods: changing the reference bandwidth, setting the interface bandwidth, and setting the OSPF cost directly.

Changing the Reference Bandwidth

OSPF calculates the default OSPF cost for an interface based on the following formula:

$$\frac{\text{Reference-brandwidth}}{\text{interface-bandwidth}}$$

The reference-bandwidth, which you can set using the **auto-cost reference-bandwidth** *bandwidth* router subcommand, sets the numerator of the formula for that one router, with a unit of Mbps. This setting may be different on different routers, but Cisco recommends using the same setting on all routers in an OSPF routing domain.

For example, serial interfaces default to a bandwidth setting of 1544, meaning 1544 Kbps. The reference bandwidth defaults to 100, meaning 100 Mbps. After converting the reference bandwidth units to Kbps to match the bandwidth, the cost, calculated per the defaults, for serial links would be

$$\frac{100{,}000}{1544} = 64$$

Note: OSPF always rounds down when the calculation results in a fraction.

The primary motivation for changing the reference bandwidth is to accommodate good defaults for higher-speed links. With a default of 100 Mbps, the cost of FastEthernet interfaces calculates to cost 1. However, the minimum OSPF cost is 1, so Gigabit Ethernet and 10 Gigabit interfaces also then default to OSPF cost 1. By setting the OSPF reference bandwidth so that there is some difference in cost between the higher speed links, OSPF can then choose routes that use those higher speed interfaces.

Note: Although Cisco recommends that all routers use the same reference bandwidth, the setting is local to each router.

Note that in the examples earlier in this chapter, the bandwidth settings used default settings, but the **auto-cost reference-bandwidth 1000** command was used on each router to allow different costs for FastEthernet and Gigabit interfaces.

Setting Bandwidth

You can indirectly set the OSPF cost by configuring the **bandwidth** *speed* interface subcommand. In such cases, the formula shown in the previous section is used, just with the configured bandwidth value.

While on the topic of the interface bandwidth subcommand, a couple of seemingly trivial facts may matter to your choice of how to tune the OSPF cost. First, on serial links, the bandwidth defaults to 1544. On subinterfaces of those serial interfaces, the same bandwidth default is used.

On Ethernet interfaces, if not configured with the **bandwidth** command, the interface bandwidth matches the actual speed. For example, on an interface that supports autonegotiation for 10/100, the bandwidth is either 100,000 (kbps, or 100 Mbps) or 10,000 (Kbps, or 10 Mbps) depending on whether the link currently runs at 100 or 10 Mbps, respectively.

Configuring Cost Directly

The most controllable method to configure OSPF costs, but the most laborious, is to configure the interface cost directly. To do so, use the **ip ospf cost** *value* interface subcommand, substituting your chosen value as the last parameter.

Verifying OSPF Cost Settings

Several commands can be used to display the OSPF cost settings of various interfaces. Example 6-10 shows several, along with the configuration of all three methods for changing the OSPF cost. In this example, the following has been configured:

- The reference bandwidth is set to 1000.
- Interface S0/0/0.1 has its bandwidth set to 1000 Kbps.
- Interface Fa0/0 has its cost set directly to 17.

Example 6-10 *R3 with OSPF Cost Values Set*

```
router ospf 3
 auto-cost reference-bandwidth 1000
interface S0/0/0.1
 bandwidth 1000
interface fa0/0
 ip ospf cost 17


R3#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Se0/0/0.2    3     34              10.10.23.3/29      647   P2P   1/1
Se0/0/0.1    3     34              10.10.13.3/29      1000  P2P   1/1
Fa0/0        3     34              10.10.34.3/24      17    BDR   1/1

R3#show ip ospf interface fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.10.34.3/24, Area 34
  Process ID 3, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 17
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 10.10.34.4
  Backup Designated router (ID) 3.3.3.3, Interface address 10.10.34.3
! lines omitted for brevity
```

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 6-6 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 6-6 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design sets specific limits to the number of Type 1 and 2 LSAs in each area. Describe how to predict the number of each type of LSA.	
How could you tune OSPF metrics to favor 10 Gbps links over 1 Gbps and 1 Gig over 100 Mbps (2)?	
The design shows one physical path from ABR1 to core subnet 1 inside area 0, and one longer area 1 path to the same subnet. What can be done to ensure both paths can be used?	

Implementation Plan Peer Review Table

Table 6-7 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 6-7 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
What conditions must be true for a router to create/flood a Type 2 LSA? (2)	
The plan shows Frame Relay with all point-to-point subinterfaces. By default, will a DR/BDR be elected?	
The plan shows a reference bandwidth change planned for all routers with high speed links, but not all other routers. What is the impact? (2)	
The plan shows many different WAN links speeds but with the interface bandwidths not matching the actual speed. tAll OSPF cost changes are made explicitly with the **ip ospf cost** interface subcommand. Do the incorrect bandwidths cause any OSPF problems?	

Create an Implementation Plan Table

To practice skills useful when creating your own OSPF implementation plan, list in Table 6-8 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 6-8 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Tune metrics by changing the formula for calculating OSPF cost based on interface bandwidth.	
Tune metrics by changing interface bandwidth.	
Change metrics by setting cost directly.	
Set the number of equal-cost OSPF routes allowed in a router's routing table.	
Influence the choice of DR on a LAN. (2)	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own OSPF verification plan, list in Table 6-9 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 6-9 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display a summary of the OSPF database.	
Display all Type 1 router LSAs known to a router.	
Display the details of a particular Type 1 router LSA.	
Display all Type 2 network LSAs known to a router.	
Display the details of a particular Type 2 router LSA.	
Display all Type 3 summary LSAs known to a router.	
Display the details of a particular Type 3 router LSA.	
Display a list of OSPF-enabled interfaces on a router.	
Determine on which interfaces a router has formed at least one OSPF neighborship.	
Determine the number of fully adjacent neighbors on an interface.	
Determine which transit networks connect to a Type 1 LSA.	
Determine the router that created and flooded a Type 3 LSA.	
Determine the router that created and flooded a Type 2 LSA.	
Determine the router that created and flooded a Type 1 LSA.	
Display the IP address of the current DR and BDR on a LAN.	
Display the OSPF interface cost (metric).	
Display all OSPF-learned routes.	
Display statistics about the number of SPF algorithm runs.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review All the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 6-10 lists a reference of these key topics and the page numbers on which each is found.

Table 6-10 *Key Topics for Chapter 6*

Key Topic Element	Description	Page Number
Table 6-2	OSPF LSA types	179
List	Two main functions of a DR	186
Table 6-3	Facts about LSA Types 1, 2, and 3	195
Table 6-4	OSPF Message Types	196
Table 6-5	OSPF neighbor states	197
List	Key differences between database exchange with and without a DR	201
List	Three steps a router considers when choosing the best OSPF IP routes	205
List	Three steps to calculate OSPF costs for intra-area routes	205
List	Two steps for calculating OSPF costs for interarea routes	209

Complete the Tables and Lists from Memory

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

Define Key Terms

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

Link State Identifier (LSID), Designated Router (DR), Backup Designated Router (BDR), Internal Router, Area Border Router (ABR), All SPF Routers multicast, All DR multicast, Link State Advertisement, Database Description (DD) packet, Link State Request (LSR) packet, Link State Acknowledgement (LSA) packet, Link State Update (LSU) packet, Router LSA, Network LSA, Summary LSA, Type 1 LSA, Type 2 LSA, Type 3 LSA, Reference bandwidth, SPF calculation

This page intentionally left blank

This chapter covers the following subjects:

Route Filtering: This section introduces three separate methods of route filtering with OSPF and discusses the commands to configure two of these methods.

Route Summarization: This section examines how OSPF can summarize routes at ABRs and at ASBRs.

Default Routes and Stub Areas: This section examines the two main reasons an Enterprise might use default routes and then shows OSPF's solution to each need: flooding a domain-wide default route and using OSPF stub areas.

CHAPTER 7

OSPF Route Summarization, Filtering, and Default Routing

This chapter discusses several features that optimize OSPF operations: route filtering, route summarization, default routing, plus OSPF stub areas.

Of these topics, the chapter focuses most on route filtering and route summarization, both of which have the same reasoning and motivation as the same features in EIGRP, as discussed throughout Chapter 4, "EIGRP Route Summarization and Filtering." Route filtering can be used to purposefully prevent hosts in one part of an internetwork from sending packets to another part. It can also reduce the size of the topology table and IP routing table, reducing both OSPF memory and CPU consumption, plus make the packet forwarding process run slightly better. Route summarization can also reduce routing protocol and packet forwarding overhead, but with a potential negative effect of creating less-efficient paths through the internetwork.

Additionally, this chapter briefly covers default routing, again the same motivations as the equivalent EIGRP feature, as discussed in Chapter 4. Finally, an OSPF-unique feature, OSPF stub routers, can also be used to limit the amount of topology data in an area, again reducing overhead.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 7-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 7-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Route Filtering	1–3
Route Summarization	4, 5
Default Routing and Stub Areas	6–8

1. Router B1, an internal router in area 1, displays the following output. The only two ABRs connected to area 1 are performing Type 3 LSA filtering. Which of the following answers is true based on the information in the output from B1?

```
R1# show ip route 10.1.0.0 255.255.0.0 longer-prefixes
! Legend lines omitted for brevity

     10.0.0.0/8 is variably subnetted, 17 subnets, 3 masks
O       10.1.2.0/24 [110/658] via 10.10.13.1, 00:00:32, Serial0/0/0.1
O IA    10.1.1.0/24 [110/658] via 10.10.23.2, 00:41:39, Serial0/0/0.2
O IA    10.1.3.0/24 [110/658] via 10.10.23.2, 00:41:39, Serial0/0/0.2
```

   a. A Type 3 LSA for 10.2.2.0/24 was filtered by both ABRs.
   b. A Type 3 LSA for 10.1.2.0/24 was not filtered by both ABRs.
   c. A Type 3 LSA for 10.1.3.0/24 was not filtered by at least one ABR.
   d. A Type 3 LSA for 10.1.1.0/24 filtered by both ABRs.

2. The following command output was gathered from Router R1, an ABR between areas 0 (backbone) and area 1. In this internetwork, area 0 contains all the subnets of class A network 10.0.0.0. R1's OSPF process has a **distribute-list prefix question in** command configured. Assuming the subnets listed in the answers actually exist in area 0, which of the following occurs on router R1?

```
R1#sh ip prefix-list
ip prefix-list question: 4 entries
   seq 5 deny 10.1.2.0/24 ge 25 le 27
   seq 15 deny 10.2.0.0/16 ge 30 le 30
   seq 20 permit 0.0.0.0/0 le 32
```

   a. R1 will not create/flood a type 3 LSA for subnet 10.1.2.0/26 into area 1.
   b. R1 will not create/flood a Type 3 LSA for subnet 10.1.2.0/24 into area 1.
   c. R1 will not have an OSPF route for subnet 10.1.2.0/26 in its IP routing table.
   d. R1 will not have an OSPF route for subnet 10.1.2.0/24 in its IP routing table.

3. Use the same scenario as the previous question, with one change. Instead of the **distribute-list prefix question in** command configured on R1, R1's OSPF process has an **area 1 filter-list prefix question in** command configured. Again assuming that the subnets listed in the answers actually exist in area 0, which of the following occurs on router R1?

```
R1#sh ip prefix-list
ip prefix-list question: 4 entries
   seq 5 deny 10.1.2.0/24 ge 25 le 27
   seq 15 deny 10.2.0.0/16 ge 30 le 30
   seq 20 permit 0.0.0.0/0 le 32
```

   a. R1 will not create/flood a type 3 LSA for subnet 10.1.2.0/26 into area 1.
   b. R1 will not create/flood a Type 3 LSA for subnet 10.1.2.0/24 into area 1.
   c. R1 will not have an OSPF route for subnet 10.1.2.0/26 in its IP routing table.
   d. R1 will not have an OSPF route for subnet 10.1.2.0/24 in its IP routing table.

4. R1, an ABR between backbone area 0 and area 1, has intra-area routes in area 0 for 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24. These routes have metrics of 21, 22, and 23, respectively. An engineer then adds the **area 0 range 10.1.0.0 255.255.0.0** command under the OSPF process of R1. Which of the following is true? (Choose two.)

   a. R1 loses and then re-establishes neighborships with all neighbors.

   b. R1 no longer advertises 10.1.1.0/24 to neighbors into area 1.

   c. R1 advertises a 10.1.0.0/16 route into area 1 with a metric of 23 (largest metric).

   d. R1 advertises a 10.1.0.0/16 route into area 1 with metric of 21 (lowest metric).

5. The following output exists on Router R1, a router internal to area 1. What can you determine as true from the output of the **show ip ospf database summary** command?

```
Routing Bit Set on this LSA
LS age: 124
Options: (No TOS-capability, DC, Upward)
LS Type: Summary Links (Network)
Link State ID: 10.1.0.0 (summary Network Number)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x878F
Length: 28
Network Mask: /22
      TOS: 0  Metric: 11
```

   a. The LSA was created by an ABR due to an **area range** command.

   b. The LSA was created by an ASBR due to a **summary-address** command.

   c. If created by an **area range** command, the best metric for a subordinate subnet on that ABR must have been 11.

   d. None of the other answers is correct.

6. Router R1, an ASBR connected to the Internet and to backbone area 0, has been configured with a **default-information originate** command. Which of the following is true about the effects of this configuration command?

   a. R1 will always create and flood a default route into the OSPF domain.

   b. R1 will create and flood an LSA for prefix/length 0.0.0.0/0 into the OSPF domain if R1's IP routing table has a route to 0.0.0.0/0.

   c. R1 will set a flag on the LSA for the subnet between itself and one of the ISPs, noting this subnet as a default network, regardless of whether R1 has a default route.

   d. R1 will set a flag on the LSA for the subnet between itself and one of the ISPs, noting this subnet as a default network, but only if R1 has a route to 0.0.0.0/0.

7. Which of the following is true about routers internal to a totally NSSA area? (Choose two.)

   a. Routers cannot redistribute external routes into the area.
   b. Routers should have zero Type 3 LSAs in their LSDBs.
   c. Routers should have zero Type 5 LSAs in their LSDBs.
   d. Routers should learn default routes from the ABRs attached to the area.

8. ABR R1 has been configured with an **area 1 stub no-summary** command. Which stubby area type is area 1?

   a. Stub
   b. Totally stubby
   c. NSSA
   d. Totally NSSA

Foundation Topics

Route Filtering

OSPF supports several methods to filter routes. However, the OSPF's internal logic restricts most filtering, requiring that the filtering be done either on an ABR or ASBR. This same internal logic dictates what each type of filtering can do and what it cannot do. So, when thinking about OSPF route filtering, you need to go beyond the concept of matching IP prefix/length information and consider OSPF internals as well. This first major section begins with a discussion of the OSPF internals that impact OSPF route filtering, followed by information about two of OSPF's route filtering tools.

First, consider the difference in how OSPF chooses intra-area versus interarea routes. For intra-area routes, OSPF uses pure link state logic, with full topology information about the area, piecing together the topology map from the Type 1 and Type 2 LSAs. This logic relies on all routers inside the area having an identical copy of the LSDB for that area. With the full topology, the SPF algorithm can be run, finding all possible routes to each subnet.

For interarea routes, OSPF uses distance vector logic. The intra-area SPF calculation includes the calculation of the metric of the best route to reach each ABR in the area. To choose the best interarea route, a router uses distance vector logic of taking its known metric to reach the ABR and adds the metric for that subnet as advertised by the ABR. In particular, no additional SPF calculation is required to find all interarea routes for a given prefix/length, making this logic more like distance vector logic.

Keeping these thoughts in mind, next consider the concept of route filtering inside one area. First, OSPF routers do not advertise routes; instead, they advertise LSAs. Any filtering applied to OSPF messages would need to filter the transmission of LSAs. However, inside one area, all routers must know all LSAs, or the whole SPF concept fails, and routing loops could occur. As a result, OSPF cannot and does not allow the filtering of LSAs inside an area, specifically the Type 1 and Type 2 LSAs that describe the intra-area topology.

OSPF does allow some route filtering, however, taking advantage that OSPF uses distance vector logic with Type 3 LSAs (and the Type 5 LSAs used for external routes). Because of the underlying distance vector logic, an OSPF ABR can be configured to filter Type 3 LSAs, with no risk of creating routing loops. (The same applies for autonomous system border routers [ASBRs] filtering the Type 5 LSAs created for external routes.) As a result of these related concepts, IOS limits OSPF route filtering to the following:

- Filtering Type 3 LSAs on ABRs
- Filtering Type 5 LSAs on ASBRs
- Filtering the routes OSPF would normally add to the IP routing table on a single router

Of these, the second option occurs as an option of the route redistribution process as explained in Chapter 9, "Basic IGP Redistribution," so it will not be covered further in this chapter. The other two topics will be examined next.

Type 3 LSA Filtering

ABRs, by definition, connect to the backbone area and at least one other area. ABRs, as a fundamental part of their role as ABR, create and flood Type 3 Summary LSAs into one area to represent the subnets in the other areas connected to that ABR. Type 3 LSA filtering tells the ABR to filter the advertisement of these Type 3 LSAs.

For example, consider Figure 7-1, which shows a generalized design with two ABR routers. The figure focuses on three subnets in area 0 for which each ABR would normally create and flood a Type 3 Summary LSA into area 1. However, in this case, the engineer has made the following choices:

- On ABR1, filter subnet 3 from being advertised.
- On ABR2, filter both subnet 2 and 3 from being advertised.

The goal of such a filtering plan could be to prevent all area 1 users from reaching subnet 3 and to allow access to subnet 2–but only through ABR1. If ABR1 were to fail, none of the area 1 routers could calculate a route for subnet 2 through ABR2, because ABR2 has not created and flooded a Type 3 LSA for that subnet. The goal for subnet 1 would be to allow each area 1 router to choose the best route through either ABR, while having a redundant route in case one route failed.

To configure type 3 LSA filtering, you use the **area** *number* **filter-list prefix** *name* **in | out** command under **router ospf**. The referenced **prefix-list** matches subnets, with subnets matched by a deny action being filtered, and subnets match with a permit action allowed through as normal. Then OSPF performs the filtering by not flooding the Type 3 LSAs into the appropriate areas. (See Chapter 4's section "IP Prefix List Concepts" for a review of IP prefix lists.)

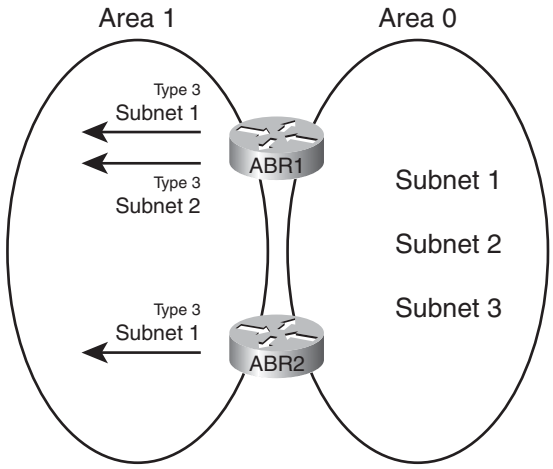


Figure 7-1 *Generic View of Type 3 LSA Filtering*

The trickiest part of the configuration relates to the **in** and **out** parameters at the end of the **area filter-list** router subcommand. These parameters define the direction relative to the area listed in the command, as follows:

- When **in** is configured, IOS filters prefixes being created and flooded *into the configured area*.
- When **out** is configured, IOS filters prefixes coming *out of the configured area*.

The need for the **in** and **out** parameters makes more sense when you consider an ABR connected to at least three areas. Figure 7-2 shows just such a sample, with both the in and out directions represented.

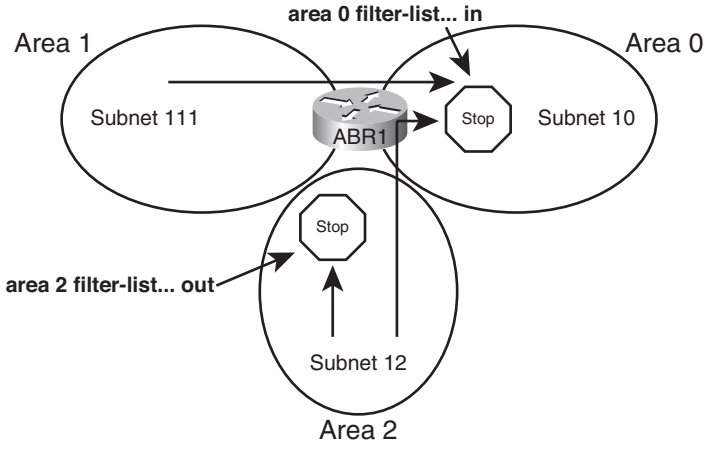


Figure 7-2 *Generic View of Type 3 LSA Filtering*

The **area 0 filter-list... in** command in the figure shows the ABR considers filtering routes from all other areas (area 1 and 2 in this case) when creating and flooding Type 3 LSAs into area 0. The **area 2 filter-list... out** command in the figure shows how the ABR only considers prefixes that exist in area 2. However, in this case, the ABR filters LSAs regardless of the area into which the Type 3 LSA would be advertised.

For example, consider the case of subnet 111, in area 1. Assume that all prefix lists happen to match subnet 111 so that subnet 111 should be filtered. The following list summarizes what happens on ABR1 regarding the potential advertisement of a Type 3 LSA for this subnet being flooded into areas 0 and 2.

- ABR1 filters the subnet 111 LSA from being sent into area 0 due to the **area 0 filter-list... in** command.
- ABR1 does not filter the subnet 111 LSA from being sent into area 2, because there is no **area 1 filter-list... out** command nor **area 2 filter-list... in** command.

As another example, Figure 7-3 shows an example internetwork with three candidate routes to be filtered by ABRs R1 and R2. ABRs R1 and R2 will play the roles of ABR1 and

ABR2 in Figure 7-1, with R1 filtering one of the three subnets, and R2 filtering two of the subnets. Note that R1 and R2 will each use different **in** and **out** keywords as well.

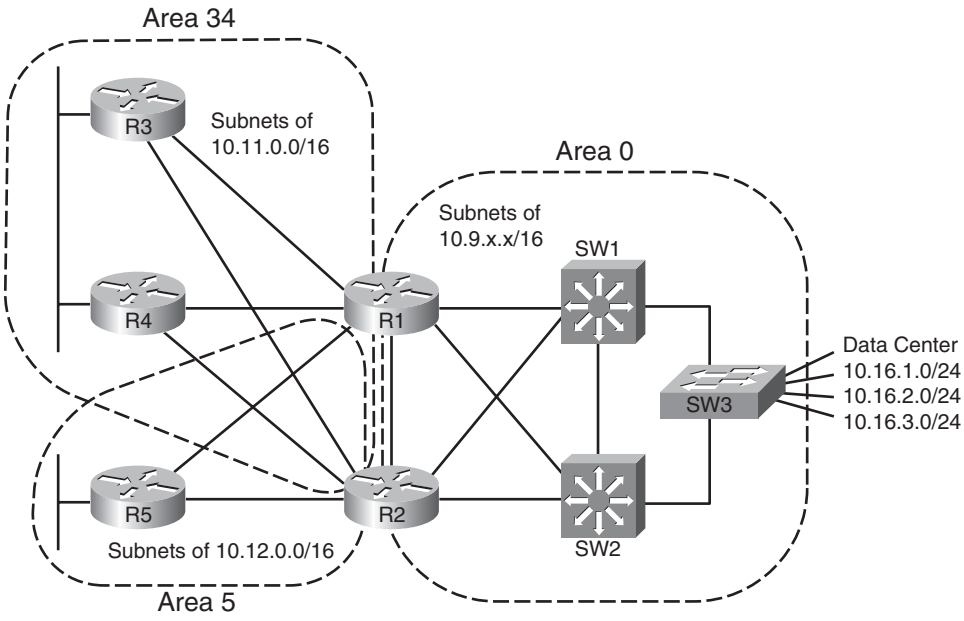


Figure 7-3 *Type 3 LSA Filtering Example*

Example 7-1 shows the configuration on both R1 and R2.

Example 7-1 *WAN1's distribute-list to Filter Manufacturing Routes*

```
! On Router R1:
ip prefix-list filter-into-area-34 seq 5 deny 10.16.3.0/24
ip prefix-list filter-into-area-34 seq 10 permit 0.0.0.0/0 le 32
!
router ospf 1
 area 34 filter-list prefix filter-into-area-34 in
```

```
! On Router R2:
ip prefix-list filter-out-of-area-0 seq 5 deny 10.16.2.0/23 ge 24 le 24
ip prefix-list filter-out-of-area-0 seq 10 permit 0.0.0.0/0 le 32
!
router ospf 2
 area 0 filter-list prefix filter-out-of-area-0 out
```

First, take a closer look at the specifics of the R1 configuration commands. The prefix list on R1 matches exactly route 10.16.3.0/24, with a **deny** action. The second prefix-list command matches all subnets, because the **0.0.0.0/0** parameter matches all subnet numbers, and the **le 32** parameter, combined with the original /0 prefix length, matches all prefix

lengths from /0 to /32. The **area 34... in** command tells R1 to apply this filtering to all Type 3 LSAs that R1 creates and would otherwise flood into area 34. As a result, the area 34 LSDB will not contain a Type 3 LSA for 10.16.3.0/24, as injected by R1.

R2's configuration uses a slightly different prefix list. The filter examines all Type 3 LSAs for subnets in area 0. The first **prefix-list** command matches all prefixes in range 10.16.2.0–10.16.3.255 (per the **10.16.2.0/23** parameter) but specifically for a prefix length of exactly 24. With a deny action, these routes are filtered. The second **prefix-list** command matches all other subnets with the same match all logic seen earlier on R1, using a permit action. R2's **area 0... out** command tells R2 to filter the subnets that R2 learns in area 0 and for which R2 would normally create Type 3 LSAs to flood into all other areas. So, neither area 34 nor area 5 will learn these two filtered subnets (10.16.2.0/24 and 10.16.3.0/24) in Type 3 LSAs from R2.

The end result of this added configuration results in the following Type 3 LSAs for the three subnets shown on the right side of Figure 7-3:

- Two Type 3 LSAs for 10.16.1.0/24 (created by R1 and R2, respectively)
- One Type 3 LSA for 10.16.2.0/24 (created by R1)
- None for 10.16.3.0/24

Example 7-2 confirms the contents of the LSDB in area 34, on Router R3.

Example 7-2 *Area 34 LSDB, as Seen on R3*

```
! On Router R3: gather show ip ospf database, and highlight all the Type 3's.
R3# show ip route 10.16.0.0 255.255.0.0 longer-prefixes
! Legend lines omitted for brevity

      10.0.0.0/8 is variably subnetted, 17 subnets, 3 masks
O IA     10.16.2.0/24 [110/658] via 10.10.13.1, 00:00:32, Serial0/0/0.1
O IA     10.16.1.0/24 [110/658] via 10.10.23.2, 00:41:39, Serial0/0/0.2
                      [110/658] via 10.10.13.1, 00:00:32, Serial0/0/0.1

R3#show ip ospf database ¦ include 10.16
10.16.1.0         1.1.1.1           759          0x80000002 0x008988
10.16.1.0         2.2.2.2           745          0x80000002 0x006BA2
10.16.2.0         1.1.1.1          759         0x80000002 0x007E92
```

The first command in the example lists R3's routes for all subnets whose first two octets are 10.16. Note that R3 has no route to 10.16.3.0/24, because both R1 and R2 filtered the Type 3 LSA. R3 happens to have equal-cost routes for 10.16.1.0/24, which is possible because both R1 and R2 permitted the advertisement of the Type 3 LSA for that subnet. R3 has only one route for 10.16.2.0/24, through R1, because R2 filtered its Type 3 LSA for that prefix.

The second command in Example 7-2 lists all LSAs that include "10.16," which includes the two Type 3 LSAs for 10.16.1.0/24, and the single Type 3 LSA for 10.16.2.0/24.

Finally, note that although the configuration in Example 7-1 showed **area filter-list** commands with both **in** and **out** parameters for variety, the result of R2's **area filter-list... out** command is that does not flood the filtered LSAs to either area 34 or area 5. If the design goals specifically meant to filter only LSAs from being advertised from Area 0 into Area 34, the **area 34 filter-list... in** command should have been used on both routers.

Filtering OSPF Routes Added to the Routing Table

In some cases, an engineer may need to filter a route, but the area design does not work well compared to the filtering goals. For instance, if an area has 20 routers, and the engineer wants to filter the route so that five of the routers do not learn the route, Type 3 LSA filtering cannot be used. Type 3 LSA filtering can only filter the LSA from being flooded throughout the entire area.

The next feature discussed in this section, referenced as *filtering with distribute lists* (based the configuration command it uses), allows individual routers to filter OSPF routes from getting into their respective IP routing tables. This type of filtering injects logic between the SPF algorithm on a router and that same router's IP routing table. This feature does not change the LSDB flooding process, does not change the LSAs added by ABRs or ASBRs, and does not change the SPF algorithm's choice of best route. However, when SPF chooses routes to add to the IP routing table, if a router has been configured with a **distribute-list in** OSPF router subcommand, enabling this feature, that router then filters the routes before adding them to that router's IP routing table. Figure 7-4 shows the general idea.

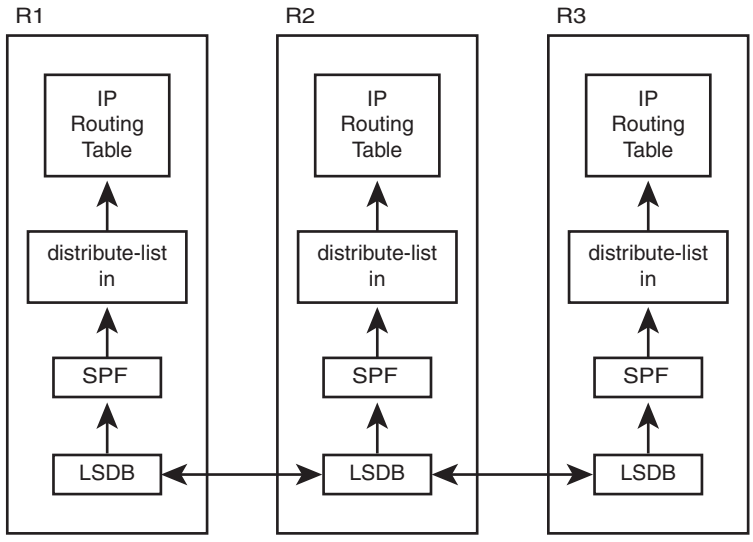


Figure 7-4 *OSPF Filtering with Distribute Lists*

In effect, you could prevent an OSPF route from being added to one or more routers' routing tables, but without risking causing routing loops, because the intra-area LSDB topol-

ogy remains intact. By filtering routes from being added to the IP routing table, you prevent the routers from forwarding packets to the filtered subnets, but presumably that's the intended goal of route filtering.

The mechanics of the **distribute-list** router subcommand has a few surprises, which are summarized in this list:

- The command requires either an **in** or **out** direction. Only the **in** direction works for filtering routes as described in this section.
- The command must refer to either a numbered ACL, named ACL, prefix list, or route map. Regardless, routes matched with a **permit** action are allowed into the routing table, and routes matched with a **deny** action are filtered.
- Optionally, the command can include the **interface** *interface-name-and-number* parameters. The router compares these parameters to the route's outgoing interface.

Example 7-3 shows a sample configuration on Router R3 from Figure 7-3. In this case, all filtering listed in Examples 7-1 and 7-2 has been removed, so no routes or LSAs have been filtered. Then, the engineer adds the **distribute-list** command on R3 to filter the route for 10.16.1.0/24, based on prefix-list filter-1.

Example 7-3 *R3's distribute-list to Filter 10.16.1.0/24*

```
! On Router R3:
ip prefix-list filter-1 seq 5 deny 10.16.1.0/24
ip prefix-list filter-1 seq 10 permit 0.0.0.0/0 le 32
!
router ospf 3
 distribute-list prefix filter-1 in
!
R3#show ip route ospf ¦ include 10.16.1
R3#
R3#show ip ospf database ¦ include 10.16.1.0
10.16.1.0         1.1.1.1           1143         0x80000007 0x007F8D
10.16.1.0        2.2.2.2          1538         0x80000007 0x0061A7
```

Note that the configuration matches only prefix 10.16.1.0/24 with a deny clause and permits all other routes. As a result, OSPF on R3 does not add a route for subnet 10.16.1.0/24 to the IP routing table, as implied by the null output of the **show ip route ospf | include 10.16.1** command. The **show ip ospf database | include 10.16.1** command lists all LSAs that have 10.16.1 in the text output, showing the two Type 3 LSAs for the subnet.

Route Summarization

OSPF allows summarization at both ABRs and ASBRs but not on other OSPF routers. The main reason is again that the LSDB must be the same for all routers in a single area. So, if summarization is needed, the summary prefixes should be created at the edge of an area (ABR or ASBR) and flooded throughout that area. However, the idea of summarizing on a

router internal to an area, hoping that some routers in the area use the summary route, and others in the same area do not, cannot be done with OSPF.

Good planning of route summaries can overcome the restriction of performing the summarization only on ABRs and ASBRs. A good OSPF area design includes consideration of future address summaries, and a good OSPF route summarization design considers the ABR locations. Although it is rare to design a large internetwork from scratch, an addressing plan that assigns all or most subnets in an area from one large address block does make address summarization easier.

OSPF summarization differs slightly on ABRs versus ASBRs. This section first examines route summarizations on ABRs and then ASBRs.

Manual Summarization at ABRs

The more difficult task with OSPF route summarization occurs when planning the design of IP address blocks and OSPF areas. When the IP addressing plan and OSPF design have been completed, if the subnet numbers inside an area happen to be from the same general range, and none of the subnet in that range exist in other OSPF areas, then a reasonable summary route can be created at the ABRs connected to that area. Without first having such a reasonable block of addresses, route summarization may not be a useful option.

After a range of subnets has been chosen for summarization, the parameters in the **area range** command must be planned. This command defines the parameters for the summary route, most notably the origin area from which the subnets exist, and the subnet number/mask that defines the summary route that should be advertised. The generic version of the command is listed next, followed by some notes about the various parameters:

```
area area-id range ip-address mask [cost cost]
```

- The configured area number refers to the area where the subnets exist; the summary will be advertised into all other areas connected to the ABR.
- The ABR compares the summary route's range of addresses with all intra-area OSPF routes, in the origin area, for which the ABR is creating Type 3 LSAs. If at least one subordinate subnet exists (subnets that sit inside the range), then the ABR advertises the summary route as a Type 3 LSA.
- The ABR does not advertise the subordinate subnet's Type 3 LSAs.
- The ABR assigns a metric for the summary route's Type 3 LSA, by default, to match the best metric among all subordinate subnets.
- The **area range** command can also explicitly set the cost of the summary.
- If no subordinate subnets exist, the ABR does not advertise the summary.

For example, Figure 7-3 (earlier in this chapter) lists three subnets on the right side of the figure, noted as Data Center subnets: 10.16.1.0/24, 10.16.2.0/24, and 10.16.3.0/24. ABR R1 could be configured to summarize these routes as 10.16.0.0/22, which includes all three subnets. (10.16.0.0/22 implies a range from 10.16.0.0–10.16.3.255.) The ABRs (R1 and R2)

could be configured to advertise a summary route using the **area 0 range 10.16.0.0 255.255.252.0** router subcommand.

Behind the scenes, ABR route summarization causes the ABR to no longer advertise the subordinate routes' Type 3 LSAs, but to instead advertise one Type 3 LSA for the summary prefix. Figure 7-5 shows this concept on ABR R1, assuming the **area 0 range 10.16.0.0 255.255.252.0** router subcommand has been configured. The three Type 3 LSAs that would normally have been advertised are shown above the ABR, and the one Type 3 LSA for the summary route, which replaces the upper LSAs, is shown under the ABR.

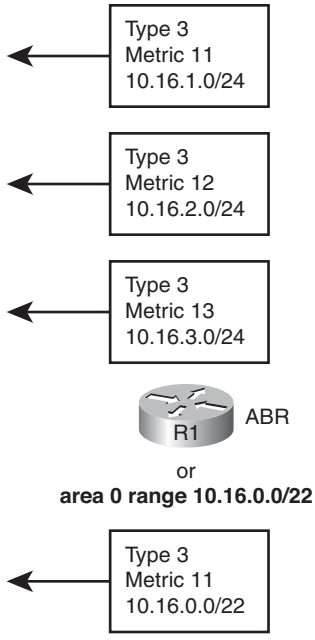


Figure 7-5 *OSPF Area Summarization–Consolidating Type 3 LSAs*

Example 7-4 shows some **show** command output related to this example. All route filtering in earlier examples has been removed, and both R1 and R2 have configured OSPF to summarize 10.16.0.0/22 with the **area 0 range 10.16.0.0 255.255.252.0** router OSPF subcommand. However, in R2's case, the **metric 12** parameter was used.

Example 7-4 *R3's distribute-list to Filter 10.16.1.0/24*

```
! On Router R1, before the summarization:
R1#sh ip route ospf ¦ incl 10.16
O        10.16.2.0/24 [110/12] via 10.10.17.7, 00:00:24, FastEthernet0/0
O        10.16.3.0/24 [110/13] via 10.10.17.7, 00:00:24, FastEthernet0/0
O        10.16.1.0/24 [110/11] via 10.10.17.7, 00:00:34, FastEthernet0/0
```

```
! Next, configuring the summarization:
router ospf 1
 area 0 range 10.16.0.0 255.255.252.0
```

```
! Next, on R2, configuring the same summary
router ospf 2
 area 0 range 10.16.0.0 255.255.252.0 cost 12
```

```
! Next, from R3
R3#show ip ospf database summary 10.16.0.0

             OSPF Router with ID (3.3.3.3) (Process ID 3)

                 Summary Net Link States (Area 34)

  Routing Bit Set on this LSA
  LS age: 124
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.16.0.0 (summary Network Number)
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000001
  Checksum: 0x878F
  Length: 28
  Network Mask: /22
        TOS: 0  Metric: 11

  LS age: 103
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 10.16.0.0 (summary Network Number)
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000001
  Checksum: 0x739E
  Length: 28
  Network Mask: /22
        TOS: 0  Metric: 12

R3#show ip route 10.16.0.0 255.255.0.0 longer-prefixes
! legend omitted for brevity

     10.0.0.0/8 is variably subnetted, 16 subnets, 4 masks
O IA    10.16.0.0/22 [110/658] via 10.10.13.1, 00:03:46, Serial0/0/0.1
```

The example demonstrates the theory of what happens behind the scenes. R3 lists only two Type 3 LSAs related to the 10.16.1.0/24, 10.16.2.0/24, and 10.16.3.0/24 subnets: the Type 3 LSAs created by R1 and R2 for 10.16.0.0/22. However, the output does not denote that this LSA represents a summarized route–it simply looks like yet another Type 3 LSA.

(Any mention of the word "summary" in the output refers to the fact that Type 3 LSAs are called summary LSAs.) In this case, R3's path to reach both R1 and R2 ties, but the LSA for R1's 10.16.0.0/22 summary was injected with metric 11, based on the lowest metric subordinate route on R1, whereas R2's uses the explicitly configured metric 12. As a result, R3's best route for 10.16.0.0/22 uses R1, as shown in the route at the end of the example.

The first **show** command in the example shows R1's metrics for the three subordinate subnets, specifically metrics 11, 12, and 13. As such, R1's summary for 10.16.0.0/22, as shown in R3's **show ip ospf database summary 10.16.0.0** command, confirms that by default R1 gave the summary route's Type 3 LSA the best metric among the component subnets.

Note: Although not discussed in depth here, the optional **not-advertise** option on the **area range** command tells the ABR to not advertise the Type 3 LSA for the summary route, making it possible to do the equivalent of Type 3 LSA filtering with the **area range** command.

Manual Summarization at ASBRs

OSPF defines an ASBR as a router that redistributes routes into OSPF from some other routing source. When redistributing the routes, the ASBR creates a Type 5 External LSA for each redistributed subnet, listing the subnet number as the LSID and listing the mask as one of the fields in the LSA. The LSA also lists the ASBR's RID as the advertising router and a cost metric for the route. For the purposes of route summarization, you can think of a Type 5 LSA as working much like a Type 3 LSA, except for routes learned externally.

Chapter 9 discusses external OSPF routes in more depth, including some additional background on Type 5 LSAs. However, to keep the discussion of OSPF route summarization together, this section describes ASBR route summarization, which has many similarities to summarization by an ABR.

If you add the **summary-address** *prefix mask* OSPF subcommand, OSPF will then attempt to summarize the external routes by creating a Type 5 LSA for the summary route, and by no longer advertising the Type 5 LSAs for the subordinate subnets. When looking for potential subordinate subnets inside the summary, the ASBR looks at all routes being redistributed into OSPF from all outside route sources, and if any subordinate subnets exist, the ASBR performs the route summarization.

Notably, this command works very much like the **area range** command on ABRs, with the main exception being that the **summary-address** command cannot explicitly set the metric of the summary route. The list of features is as follows:

- The ASBR compares the summary route's range of addresses with all routes redistributed into OSPF on that ASBR to find any subordinate subnets (subnets that sit inside the summary route range). If at least one subordinate subnet exists, the ASBR advertises the summary route.
- The ASBR does not advertise the subordinate subnets.
- To create the summary, the ASBR actually creates a Type 5 LSA for the summary route.

- The ASBR assigns the summary route the same metric as the lowest metric route amongst all subordinate subnets.
- If no subordinate subnets exist, the ASBR does not advertise the summary.
- Unlike the **area range** command, the **summary-address** command cannot be used to directly set the metric of the summary route.

The **summary-address** OSPF subcommand defines the summary route on the ASBR, with similar syntax and parameters as compared to the **area range** command seen on ABRs. Table 7-2 lists the two commands for comparison and study.

Key Topic

Table 7-2 *OSPF Route Summarization Commands*

Where Used	Command
ASBR	**summary-address** {{*ip-address mask*} | {*prefix mask*}} [**not-advertise**]
ABR	**area** *area-id* **range** *ip-address mask* [**advertise** | **not-advertise**] [**cost** *cost*]

Default Routes and Stub Areas

Enterprises typically use default routes in two different cases:

- To direct remote-site routers at the edge of the Enterprise network to send all packets toward the core of the Enterprise, with the core routers knowing all the more specific routes to Enterprise destination addresses.
- To direct traffic on all Enterprise routers toward an Internet-facing router so that all traffic destined for the Internet eventually arrives at the Enterprise's Internet-connected routers.

Engineers could achieve both of these goals by using route summarization with the **area range** and **summary-address** commands. For example, consider a case in which the goal is to drive all packets destined for Internet hosts to one of two equal Internet routers for an Enterprise, as shown in Figure 7-6. The design shows two ASBRs connected to the Internet. Both ASBRs could learn routes with BGP. Rather than redistribute all BGP routes into the Enterprise, the ASBRs summarize to the ultimate summary, 0.0.0.0/0. The two OSPF ASBRs flood the Type 5 LSA for a summary route–one from ASBR1 and one from ASBR2–throughout the Enterprise. As a result, all OSPF routers choose a default route, with the packets destined for locations in the Internet eventually reaching one of the two ASBRs, which then forwards the packets into the Internet.

To meet the other design goal for using defaults–to get the routers in an area to use default routing to deliver packets to the ABR–the ABR could use the **area range** command to flood a default route into a single area. Again in Figure 7-6, if the design called for the routers in area 1 to use a default route to reach other destinations in the Enterprise, the ABRs connected to area 1, like ABR1, could use the **area 0 range 0.0.0.0 0.0.0.0** com-

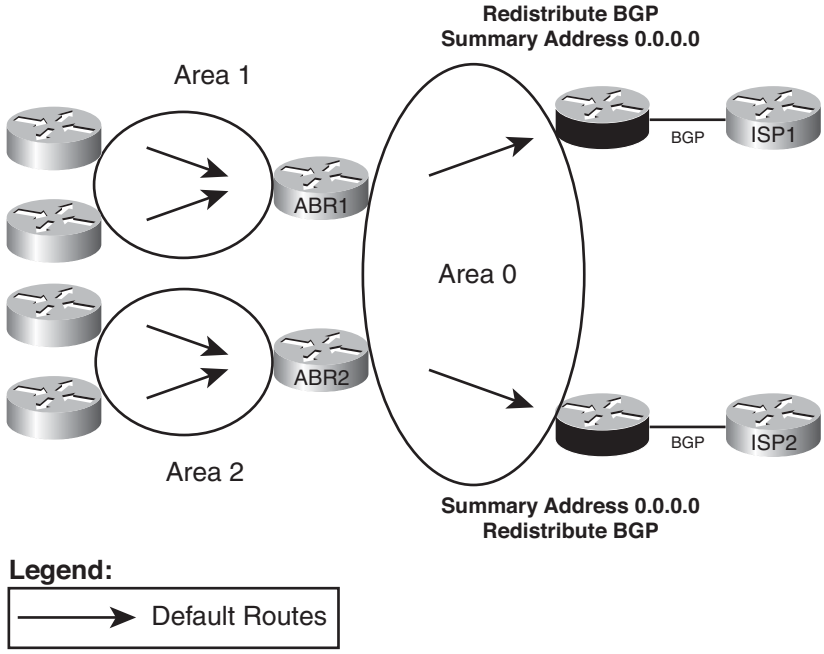


Figure 7-6 *Using ASBR Route Summarization to Advertise Summary Routes*

mand. ABR1 would then advertise a default route into the area, as an LSA type 3, and not advertise any of the other Type 3 LSAs known from area 0. The routers internal to area 1 would use their default route for packets destined to unknown destination addresses, but the ABRs would have full knowledge of the routes inside the Enterprise and know how to forward the packets at that point.

Even though you can use the **summary-address** and **area range** commands, most engineers use other methods to introduce and control default routes inside the OSPF domain. The first tool, the **default-information originate** OSPF subcommand, introduces a default route to be flooded throughout the OSPF domain. As a result, it is most useful for default routing to draw packets toward ASBRs connected to external networks. The other tool, stub areas, focuses on the other common use of default routes, controlling when ABRs flood default routes into a given area. This section examines both topics.

Domain-wide Defaults Using the default-information originate Command

The OSPF subcommand **default-information originate** tells OSPF to create a Type 5 LSA (used for external routes) for a default route–0.0.0.0/0–and flood it like any other Type 5 LSA. In other words, it tells the router to create and flood information about a default route throughout the OSPF domain.

For example, consider a typical dual-homed Internet design, as shown in Figure 7-7. In this case, the Enterprise has two Internet-connected routers, and the engineer wants to use default routing inside the Enterprise to cause all the Enterprise routers to send packets toward either ASBR1 or ASBR2.

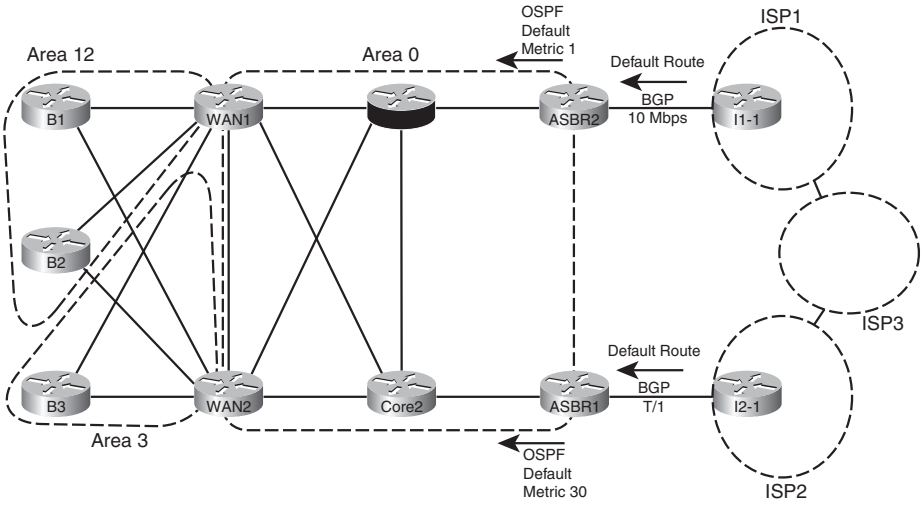


Figure 7-7 *Dual-Homed Internet Design Using Default Routes*

The **default-information originate** command tells the ASBRs to flood a default route into OSPF, but only if the ASBR itself has a default route in its IP routing table. This logic relies on the fact that the ASBRs typically either have a static default route pointing to the connected ISP router, or they learn a default route from the ISP using BGP. (Figure 7-7 each ISP advertising a default route in this case.) All the routers then learn a default route, based on the Type 5 LSAs for 0.0.0.0/0 as flooded by the ASBRs.

Because a router withdraws its OSPF default route when its own IP route to 0.0.0.0/0 fails, OSPF allows the design in Figure 7-7 to fail over to the other default route. When all is well, both ISP1 and ISP2 advertise a default route to the Enterprise using BGP, so both ASBR1 and ASBR2 have a route to 0.0.0.0/0. As shown in the figure, ASBR2 has been configured to advertise its OSPF default with a lower metric (1) than does ASBR1 (metric 30), so the Enterprise routers will forward traffic to the Internet through ASBR2. However, if ISP1 quits advertising that default with BGP, or BGP fails between ASBR2 and ISP1's I1-1 router, ASBR2 will withdraw its OSPF default route. The only remaining OSPF default route will be the one that leads to ASBR1, making use of the backup default route.

The full command syntax, as shown here, provides several optional parameters that impact its operation:

```
default-information originate [always] [metric metric-value] [metric-type type-
value] [route-map map-name]
```

The following list summarizes the features of the **default-information originate** OSPF subcommand:

- With all default parameters, it injects a default route into OSPF, as an External Type 2 route, using a Type 5 LSA, with metric 1, but only if a default route exists in that router's routing table.

- With the **always** parameter, the default route is advertised even if there is no default route in the router's routing table.
- The **metric** keyword defines the metric listed for the default route (default 1).
- The **metric-type** keyword defines whether the LSA is listed as external type 1 or external type 2 (default). (Chapter 9's section "Redistribution into OSPF" discusses OSPF external route types.)
- The decision of when to advertise, and when to withdraw, the default route is based on matching the referenced **route-map** with a permit action.

When configured, OSPF will flood the default route throughout the OSPF routing domain, drawing traffic to each ASBR, as shown earlier in Figure 7-6.

Note: The type of external OSPF route (Type 1 or Type 2) is explained more fully in Chapter 9.

Stubby Areas

As mentioned earlier, the two most common reasons to consider using default routes are to drive all Internet-destined traffic toward the Internet-connected routers in the Enterprise and to drive traffic inside an area toward one of the ABRs in that area. This second design choice allows the routers in an area to use default routes for forwarding packets to ABRs, rather than more specific routes. Using default routes inside an area reduces memory consumption and CPU processing time on the routers inside the area, because the routers in that area can have fewer LSAs in their LSDBs.

The OSPF stub router feature provides engineers a very simple way to enable the function of flooding default routes inside an area, with those default routes driving IP packets back toward the ABRs attached to that area. ABRs in stub areas advertise a default route into the stub area. At the same time, the ABR chooses to not advertise external routes (5 LSAs) into the area, or even instead to no longer advertise interarea routes (in Type 3 LSAs) into the area. As a result, all routers in the stub area can still route to the destinations (based on the default route), but the routers require less memory and processing.

The following list summarizes these features of stub areas for easier study and review:

- ABRs create a default route, using a Type 3 LSA, listing subnet 0.0.0.0 and mask 0.0.0.0, and flood that into the stub area.
- ABRs do not flood Type 5 LSAs into the stub area.
- ABRs may not flood other Type 3 LSAs into the area.
- The default route has a metric of 1 unless otherwise configured using the OSPF subcommand **area** *area-num* **default-cost** *cost*.
- Routers inside stub areas cannot redistribute external routes into the stubby area, because that would require a Type 5 LSA in the area.

- All routers in the area must be configured to be stubby; if not, neighbor relationships cannot form between potential neighbors based on this mismatched configuration.

Figure 7-8 shows a familiar design in which area 34 will become a stub area. The design shows three external routes and lists three of many internal routes inside area 0. The figure shows ABRs R1 and R2 advertising defaults into area 34.

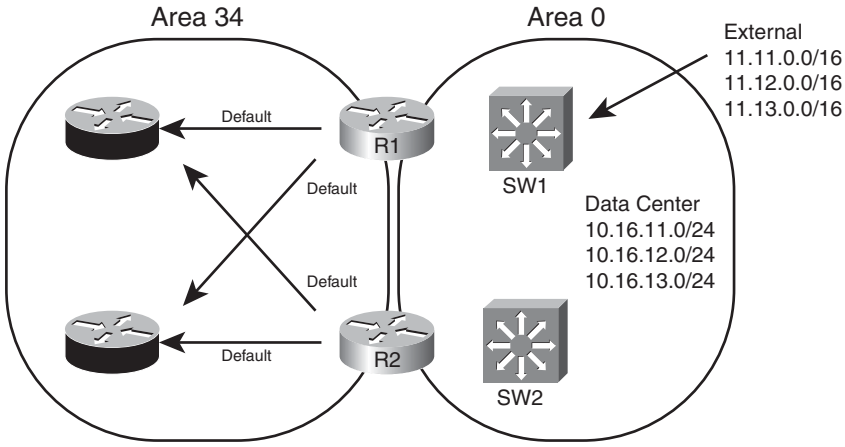


Figure 7-8 *Stubby Area Design*

Figure 7-8 demonstrates the core feature common to all types of stub areas: the ABRs flood a default route into the area. The routers inside the area can then calculate their best default route. Next, the text examines the different types of OSPF areas, before moving on to the details of configuration and verification.

Introducing Stubby Area Types

Even within the realm of stubby areas, four types of stubby areas exist: stub, totally stubby, not-so-stubby areas (NSSA), and totally NSSA.

Two types of stubby areas have the word "totally" as part of the name, and two do not. The differences between those with the word "totally" and those without have to do with whether Type 3 LSAs are flooded into the area. The rules are

- For all types of stubby areas, the ABR always filters Type 5 (external) LSAs.
- For totally stubby and totally NSSA areas, the ABR also filters Type 3 LSAs.
- For stubby and NSSA areas–those without the word "totally" in the name–the ABRs do not filter Type 3 LSAs, advertising Type 3 LSAs as normal.

For example, consider the diagram in Figure 7-8, with area 34 as simply a stub area. As for all types, the ABRs each advertise a default route into area 34. As for all stubby area types, the ABRs filter all Type 5 LSAs, which means that the three Type 5 LSAs for

11.11.0.0/16, 11.12.0.0/16, and 11.13.0.0/16 would not exist in the LSDBs for area 34. Finally, because the area is not a totally stubby area, the ABRs do create and flood Type 3 LSAs for interarea routes as usual, so they flood LSAs for the 10.16.11.0/24, 10.16.12.0/24, and 10.16.13.0/24 subnets listed in the figure.

Next, consider a similar scenario but with a totally stubby area for area 5. As for all stubby area types, the ABRs each advertise a default route into area 5. As for all stubby area types, the ABRs filter all Type 5 LSAs, which means that the three Type 5 LSAs for 11.11.0.0/16, 11.12.0.0/16, and 11.13.0.0/16 would not exist in the LSDBs for area 5. The key difference exists in that the ABRs also would not create and flood Type 3 LSAs for interarea routes as usual, so they would not advertise Type 3 LSAs for the 10.16.11.0/24, 10.16.12.0/24, and 10.16.13.0/24 subnets listed in the figure into area 5.

The other difference in stubby area types relates to whether the name uses NSSA (NSSA or totally NSSA) or not (stubby, totally stubby). Stubby area types that use the NSSA name can redistribute external routes into the area; stubby area types without NSSA in the name cannot.

Configuring and Verifying Stubby Areas

Configuring stub and totally stubby areas requires only three commands, but with at least one command on each router, as listed in Table 7-3:

Note: For totally stubby areas, only the ABR must have the **no-summary** keyword on the **area** *area-id* **stub no-summary** command. However, including this keyword on internal routers does not cause a problem.

Figure 7-9 shows a more detailed view of area 34 from Figure 7-8, so by making area 34 a stub area, ABRs R1 and R2 will not flood Type 3 LSAs into area 34—other than the Type 3 LSA for the default routes. Example 7-4 shows the configuration on Routers R1, R2, and R3 from Figure 7-9.

Table 7-3 *Stub Area Configuration Options*

Action	Configuration Steps
Stubby	**Configure area** *area-id* **stub** on each router in the area
Totally stubby	**Configure area** *area-id* **stub no-summary** command on the ABRs **Configure area** *area-id* **stub**, without the **no-summary** keyword, on all other routers in the area
Set the metric of the default route	**Configure area** *area-id* **default-metric** *metric* on an ABR; can differ from ABR to ABR. Default value is 1.

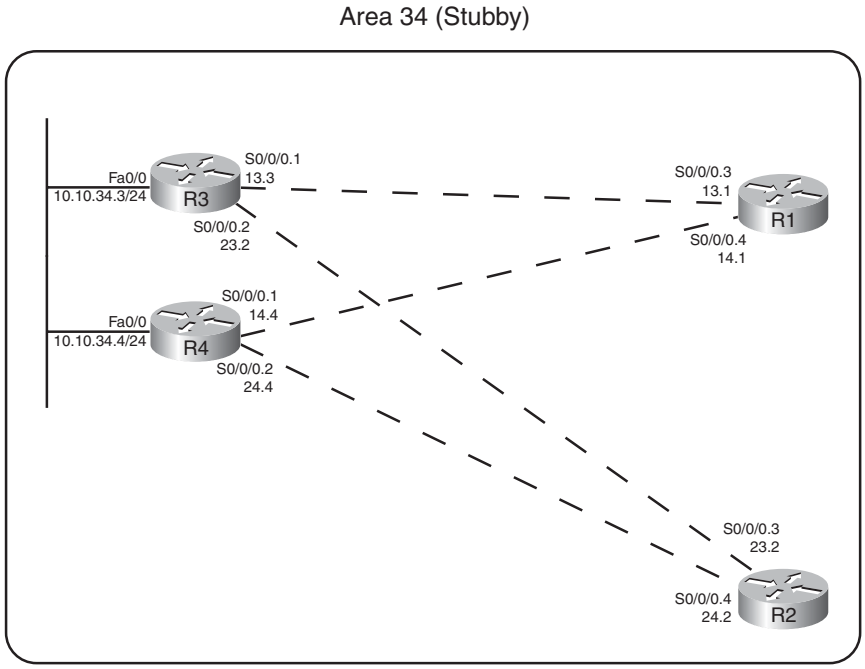


Figure 7-9 *Detailed View of Area 34*

Example 7-4 *Stub Area Configuration*

```
! On Router R1:
router ospf 1
 area 34 stub
 auto-cost reference-bandwidth 1000
!
interface s0/0/0.3 point-to-point
 ip ospf area 34
!
interface s0/0/0.4 point-to-point
  ip ospf area 34
```

```
! On Router R2:
router ospf 2
 area 34 stub
 auto-cost reference-bandwidth 1000
!
interface s0/0/0.3 point-to-point
 ip ospf area 34
!
interface s0/0/0.4 point-to-point
 iip ospf area 34
```

```
! On Router R3:
router ospf 3
 area 34 stub
 auto-cost reference-bandwidth 1000
!
interface s0/0/0.1 point-to-point
 ip ospf area 34
 ip ospf cost 500
!
interface s0/0/0.2 point-to-point
 ip ospf area 34
!
interface fa0/0
 ip ospf area 34
```

With the configuration as shown, both R1 and R2 will inject a default route, represented as a Type 3 LSA, with default metric 1. They will also not flood the Type 5 LSAs into area 34. Example 7-5 confirms these facts, showing the Type 3 LSA for the summary, and the absence of Type 5 LSAs in the output of the **show ip ospf database** command on router R3.

Example 7-5 *Evidence of Type 5's Existing, Disappearing, and Defaults Appearing*

```
! Before making Area 34 stubby:


R3#show ip ospf database ¦ begin AS External
                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
11.11.0.0       7.7.7.7         929         0x80000001 0x00016D 0
12.12.0.0       7.7.7.7         845         0x80000001 0x00E784 0
13.13.0.0       7.7.7.7         835         0x80000001 0x00CE9B 0

! After making area 34 stubby - no output from the next command.

R3#show ip ospf database ¦ begin AS External
R3#

! The database for area 34 now has two Type 3 LSAs for default routes.
R3#show ip ospf database

             OSPF Router with ID (3.3.3.3) (Process ID 3)

                 Router Link States (Area 34)
```

```
! Lines omitted for brevity - skipped to "Summary Net" (Type 3) section

                Summary Net Link States (Area 34)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         1.1.1.1         692         0x80000001 0x0093A6
0.0.0.0         2.2.2.2         686         0x80000001 0x0075C0
10.10.5.0       1.1.1.1         692         0x8000000E 0x00445C
10.10.5.0       2.2.2.2         686         0x8000000F 0x002477
10.10.12.0      1.1.1.1         692         0x8000000E 0x0054AF
10.10.12.0      2.2.2.2         686         0x8000000E 0x0036C9
! Many Type 3 LSA's omitted for brevity's sake
```

Example 7-5 shows the existence of the Type 5 external LSAs before area 34 became a stubby area, and the disappearance of those same LSAs once it was made a stubby area. The **show ip ospf database** command then shows two LSAs that list default routes, one learned from RID 1.1.1.1 (R1), and one learned from RID 2.2.2.2 (R2).

Example 7-6 continues the verification of how stub areas work with three more commands.

Example 7-6 *Three External Routes Before and None Afterward Changing to Stubby*

```
! Next, R3 confirms it thinks area 34 is a stub area
R3#show ip ospf
 Routing Process "ospf 3" with ID 3.3.3.3
 Start time: 00:00:38.756, Time elapsed: 07:51:19.720
! lines omitted for brevity
    Area 34
         Number of interfaces in this area is 3
         It is a stub area
         Area has no authentication
         SPF algorithm last executed 00:11:21.640 ago
         SPF algorithm executed 18 times
         Area ranges are
         Number of LSA 29. Checksum Sum 0x0D3E01
         Number of opaque link LSA 0. Checksum Sum 0x000000
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0

! The next command shows all Type 3 (summary) LSAs of prefix 0.0.0.0

R3#show ip ospf database summary 0.0.0.0

             OSPF Router with ID (3.3.3.3) (Process ID 3)
```

```
                Summary Net Link States (Area 34)

  Routing Bit Set on this LSA
  LS age: 879
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 0.0.0.0 (summary Network Number)
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000001
  Checksum: 0x93A6
  Length: 28
  Network Mask: /0
        TOS: 0  Metric: 1

  LS age: 873
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(Network)
  Link State ID: 0.0.0.0 (summary Network Number)
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000001
  Checksum: 0x75C0
  Length: 28
  Network Mask: /0
        TOS: 0  Metric: 1

! The next command lists statistics of the number of LSAs of each type -
! note a total of 0 Type 5 LSAs, but many Type 3

R3#show ip ospf database database-summary

            OSPF Router with ID (3.3.3.3) (Process ID 3)

Area 34 database summary
  LSA Type      Count    Delete   Maxage
  Router        4        0        0
  Network       1        0        0
  Summary Net   24       0        0
  Summary ASBR  0        0        0
  Type-7 Ext    0        0        0
    Prefixes redistributed in Type-7  0
  Opaque Link   0        0        0
  Opaque Area   0        0        0
  Subtotal      29       0        0
```

```
Process 3 database summary
  LSA Type       Count     Delete    Maxage
  Router         4          0         0
  Network        1          0         0
  Summary Net    24         0         0
  Summary ASBR   0          0         0
  Type-7 Ext     0          0         0
  Opaque Link    0          0         0
  Opaque Area    0          0         0
  Type-5 Ext     0          0         0
      Prefixes redistributed in Type-5  0
  Opaque AS      0          0         0
  Non-self       28
  Total          29        0         0
```

Following are the three commands in Example 7-6 in order:

- **show ip ospf–**Confirms with one (highlighted) line that the router believes that the area is a stub area.
- **show ip ospf database summary 0.0.0.0–**By definition, this command lists all summary (Type 3) LSAs with prefix 0.0.0.0. It lists two such LSAs, created by R1 and R2 (RIDs 1.1.1.1 and 2.2.2.2, respectively), both with metric 1 (the default setting).
- **show ip ospf database database-summary–**This command lists statistics about the numbers of and types of LSAs in the database. The counters show 0 Type 5 LSAs, and a few dozen Type 3s–confirming that the area, while stubby, is not totally stubby.

Configuring and Verifying Totally Stubby Areas

Configuring totally stubby areas requires almost no additional effort as compared with stubby areas. As listed earlier in Table 7-3, the only difference for totally stubby configuration versus stubby configuration is that the ABR's include the **no-summary** keyword on the **area stub** command. (**no-summary** refers to the fact that ABRs in totally stubby areas do not create/flood Type 3 summary LSAs.)

Example 7-7 shows another example configuration, this time with area 34 as a totally stubby area. Additionally, the default routes' metrics have been set so that both R3 and R4 will use R1 as their preferred ABR, by setting R2's advertised summary to a relatively high metric (500). Example 7-7 just shows the changes to the configuration shown in Example 7-4.

Example 7-7 *Totally Stubby Area Configuration*

```
! On Router R1:
router ospf 1
 area 34 stub no-summary
 auto-cost reference-bandwidth 1000
```

```
! On Router R2:
router ospf 2
 area 34 stub no-summary
 area 34 default-cost 500
 auto-cost reference-bandwidth 1000
```

The configuration of a totally stubby area reduces the size of the LSDB in area 34, because the ABRs no longer flood Type 3 LSAs into area 34, as shown in Example 7-8. R3 displays its LSDB, listing only two Summary (Type 3) LSAs–the two default routes advertised by the two ABRs, respectively. No other Type 3 LSAs exist, nor do any external (Type 5) or ASBR summary (Type 4) LSAs.

Also, note that the example lists the OSPF routes known to R3. Interestingly, in the topology shown for area 34, R3 learns only three OSPF routes: the two intra-area routes for the subnets between R4 and the two ABRs, plus the best default route. The default route has a metric of 501, based on R3's S0/0/0.1 interface cost plus the cost 1 listed for R1's Type 3 LSA for the default route.

Example 7-8 *Confirmation of the Effects of a Totally Stubby Area*

```
! On Router R3: show ip ospf database, show ip route ospf, get show ip ospf
database database-summary –
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.10.13.1 to network 0.0.0.0


     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.10.13.0/29 is directly connected, Serial0/0/0.1
O       10.10.14.0/29 [110/657] via 10.10.34.4, 00:57:37, FastEthernet0/0
C       10.10.23.0/29 is directly connected, Serial0/0/0.2
O       10.10.24.0/29 [110/657] via 10.10.34.4, 00:57:37, FastEthernet0/0
C       10.10.34.0/24 is directly connected, FastEthernet0/0
O*IA 0.0.0.0/0 [110/501] via 10.10.13.1, 00:24:35, Serial0/0/0.1


R3#show ip ospf database database-summary

             OSPF Router with ID (3.3.3.3) (Process ID 3)
```

```
! lines omitted for brevity

Process 3 database summary
  LSA Type       Count     Delete    Maxage
  Router         4          0         0
  Network        1          0         0
  Summary Net    2          0         0
  Summary ASBR   0          0         0
  Type-7 Ext     0          0         0
  Opaque Link    0          0         0
  Opaque Area    0          0         0
  Type-5 Ext     0          0         0
      Prefixes redistributed in Type-5  0
  Opaque AS      0          0         0
  Non-self       6
  Total          7          0         0

R3#show ip ospf database ¦ begin Summary
                Summary Net Link States (Area 34)

Link ID          ADV Router       Age          Seq#        Checksum
0.0.0.0          1.1.1.1          1407         0x80000003 0x008FA8
0.0.0.0          2.2.2.2          1506        0x80000004 0x00FF3E
```

Following are the three commands in Example 7-8 in order:

- **show ip route–**It lists a single interarea route–a default route, with destination 0.0.0.0/0. The output also lists this same next-hop information as the gateway of last resort.
- **show ip ospf database database-summary–**The statistics still show no External Type 5 LSAs, just as when the area was stubby, but now shows only 2 Type 3 LSAs, whereas before a few dozen existed.
- **show ip ospf database | begin Summary–**This command shows the output beginning with the Type 3 Summary LSAs. It lists two default route LSAs: one from R1 and one from R2.

Examples 7-6 and 7-8 demonstrate the key differences between stub (they do see Type 3 LSAs) and totally stubby areas (which do not see Type 3 LSAs). Next, this section looks at the different types of not-so-stubby areas.

The Not-So-Stubby Area (NSSA)

Stub and totally stubby areas do not allow external routes to be injected into a stubby area–a feature that originally caused some problems. The problem is based on the fact that stub areas by definition should never learn a Type 5 LSA, and OSPF injects external routes

into OSPF as Type 5 LSAs. These two facts together mean that a stubby area could not normally have an ASBR that was injecting external routes into the stub area.

The not-so-stubby area (NSSA) option for stubby areas overcomes the restriction on external routes. The solution itself is simple: Because stubby areas can have no Type 5 LSAs, later OSPF RFCs defined a newer LSA type (Type 7) that serves the same purpose as the Type 5 LSA, but only for external routes in stubby areas. So, an NSSA area can act just like a stub area, except that routers can inject external routes into the area.

Figure 7-10 shows an example, with four steps. The same stubby area 34 from the last few figures still exists; it does not matter at this point whether area 34 is totally stubby or simply stubby.

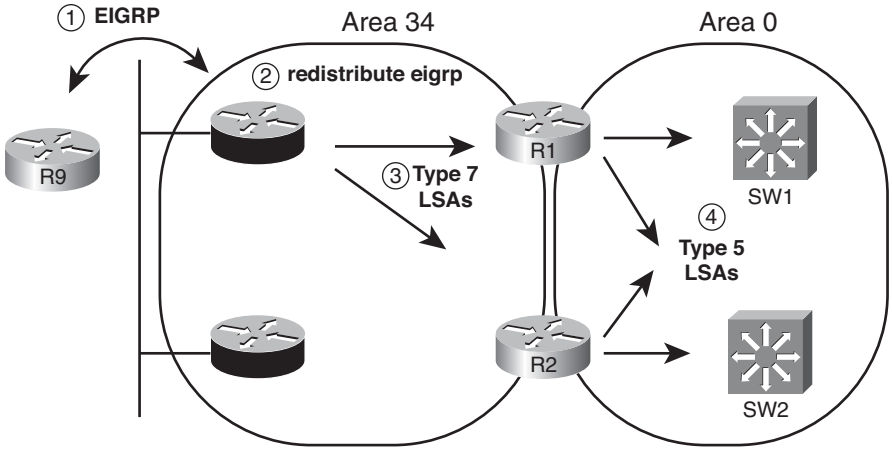


Figure 7-10 *External Routes in an NSSA Area (34)*

The steps labeled in the figure are as follows:

Step 1. ASBR R3 learns routes from some external source of routing information, in this case, EIGRP from R9.

Step 2. An engineer configures route redistribution using the **redistribute** command, taking the routes learned with EIGRP, and injecting them into OSPF.

Step 3. R3 floods Type 7 LSAs throughout stub area 34.

Step 4. ABRs R1 and R2 then create Type 5 LSAs for the subnets listed in the Type 7 LSAs, and flood these Type 5 LSAs into other areas, like area 0.

The configuration of an NSSA area requires only a little effort compared to configuring stubby and totally stubby areas: Just use the **area nssa** command instead of the **area stub** command, with no differences in other parameters. NSSA areas can be simply stubby, filtering only external routes from getting into the area, or totally stubby. The NSSA designation just defines whether routers in that area can redistribute external routes into OSPF.

Example 7-9 shows a sample with the configuration of a totally NSSA area 34 from the network represented in the last four figures. Note that as with the **area stub** command, the **area nssa** command's **no-summary** option is required only on the ABRs.

Example 7-9 *Totally NSSA Area Configuration and Verification*

```
! On Router R1:
router ospf 1
 area 34 nssa no-summary
! On Router R2:
router ospf 2
 area 34 nssa no-summary
 area 34 default-cost 500
! On Router R3:
router ospf 3
 area 34 nssa
! On Router R4:
router ospf 4
 area 34 nssa
```

The same verification steps and commands can be used for NSSA areas as were shown in the earlier examples for stub areas. In particular, the **show ip ospf** command states that the area is an NSSA area. You can also see Type 7 LSAs in the OSPF LSDB after redistribution has been configured, as shown in Chapter 9.

Stubby Area Summary

Table 7-4 summarizes the key points regarding stubby areas.

Key Topic

Table 7-4 *OSPF Stubby Area Types*

Area Type	ABRs flood Type 5 External LSAs into the area?	ABRs flood Type 3 Summary LSAs into the area?	Allows redistribution of external LSAs into the stubby area?
Stub	No	Yes	No
Totally stubby	No	No	No
NSSA	No	Yes	Yes
Totally NSSA	No	No	Yes

Note: Both types of totally stubby areas (totally stubby, totally NSSA) are Cisco proprietary.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 7-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 7-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
When using OSPF, prevent the routers in sites for one division of the company from knowing IP routes for subnets in another division. (3)	
The design shows an Enterprise that uses only OSPF. It lists a goal of keeping the LSDBs and routing tables in each area small. (3)	
The design lists a goal of extremely small LSDBs and IP routing tables on branch office routers–which stub area types work best? (2)	
The design calls for the flooding of a domain-wide default route to draw traffic toward the Internet-connected routers.	

Implementation Plan Peer Review Table

Table 7-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 7-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a design with area 0, with different ABRs connecting area 0 to areas 1, 2, and 3. The configurations show Type 3 LSA filtering into the nonbackbone areas but not in the opposite direction. Could this configuration filter subnets in area 1 from being seen in area 2?	
The design shows the configuration of Type 3 LSA filtering on an internal router in area 1. Could the filter have any effect?	
The plan shows the configuration of the **area range** command on an ABR. What is the metric for the summary route? And in what conditions will the ABR advertise the summary?	
The plan shows the configuration of the **area 1 stub** command for an area mostly located on the west coast of the USA. The company just bought another company whose sites are also on the west coast. What issues exist if you add links from the acquired company into area 1?	
The plan shows the configuration of the **default-information originate always** command on the one router to which the Internet links connect. What happens to default route when Internet link fails? What happens to packets destined for the Internet during this time?	

Create an Implementation Plan Table

To practice skills useful when creating your own OSPF implementation plan, list in Table 7-7 configuration commands related to the configuration of the following features. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 7-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Filter Type 3 LSAs from being sent into an area.	
Filter the OSPF routes calculated on one router from being added to that one router's routing table.	
Configure route summarization on ABRs.	
Configure route summarization on ASBRs.	
Configure the OSPF domain-wide advertisement of a default route.	
Configure stubby or totally stubby areas.	
Configure NSSA or totally NSSA areas.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own OSPF verification plan, list in Table 7-8 all commands that supply the requested information. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 7-8 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display all IP routes for subnets in a range, regardless of prefix length.	
Display the contents of an IP prefix list.	
Display details of all Type 3 LSAs known to a router.	
Display details of all Type 5 external LSAs known to a router.	
Display the metric advertised in a summary route created by the **area range** command.	
Display the metric advertised in a summary route created by the **summary-address** command.	
Discover whether a router resides in a stubby area, and if so, which kind.	
Confirm stubby area concepts by looking at the numbers of Type 3 and Type 5 LSAs known to a router.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review All the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 7-9 lists a reference of these key topics and the page numbers on which each is found.

Table 7-9 *Key Topics for Chapter 7*

Key Topic Element	Description	Page Number
List	Explanations of the features of the **area range** command	232
List	Explanations of the features of the **summary-address** command	235

Complete the Tables and Lists from Memory

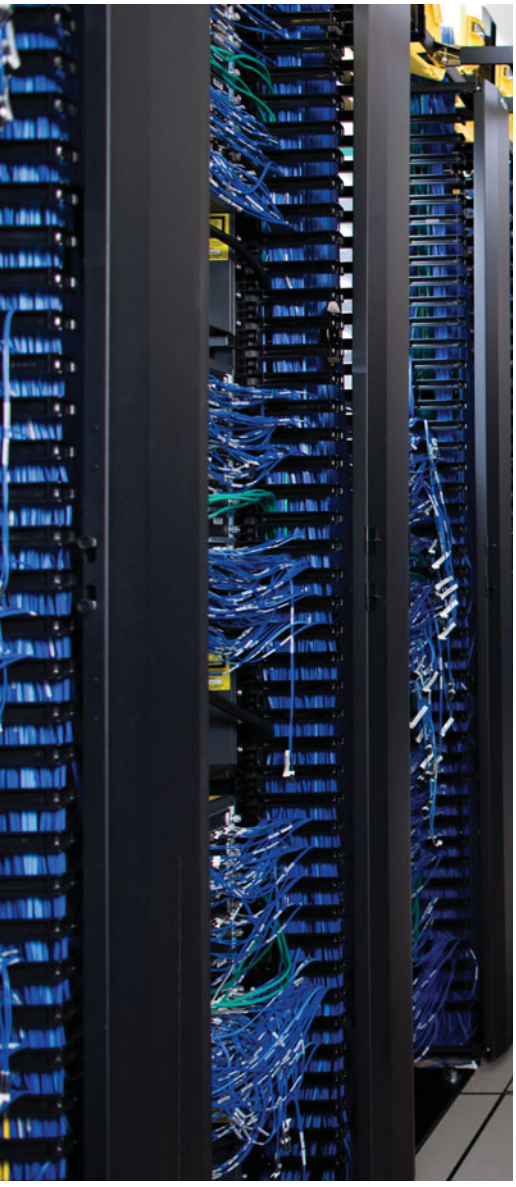
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

Type 3 LSA Filtering, Stub area, Totally stubby area, Not-so-stubby area, Type 5 External LSA

This page intentionally left blank



This chapter covers the following subjects:

Virtual Links: This section examines how engineers can use virtual links to connect separate parts of an area through another area to maintain the requirement that OSPF areas be contiguous.

OSPF over Frame Relay: This section examines the issues related to multipoint Frame Relay designs and configurations, in particular issues related to neighbor discovery or definition, whether the routers use a DR, and issues related to Frame Relay mapping.

CHAPTER 8

OSPF Virtual Links and Frame Relay Operations

This chapter contains two topics that at first glance seem to have little in common. The first topic, OSPF virtual links, gives network designers a tool to overcome an area design issue that occurs from time to time. The second topic, making OSPF work over Frame Relay multipoint interfaces, focuses on the implementation choices with different IOS commands when using OSPF with Frame Relay.

Although unrelated from a technology perspective, these two OSPF topics share that both provide a solution to specific types of problems–problems that occur in a small percentage of OSPF internetworks. Because these features solve particular (but different) problems, rather than include these topics in Chapters 5, 6, or 7–which cover the most central and common OSPF features–we decided to put these less-commonly seen features here in Chapter 8. Hopefully, this separation allowed a little more focus in the other OSPF chapters, while still including these two topics in the book.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these five self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 8-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 8-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Virtual Links	1, 2
OSPF over Frame Relay	3–5

1. Which of the following answers can be verified as true based on the following command output from Router R1?

```
R1#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 4.4.4.4 is up
  Run as demand circuit
  DoNotAge LSA allowed.
```

```
Transit area 1, via interface FastEthernet0/1, Cost of using 3
```

a. R1 is configured with an **area 0 virtual-link 4.4.4.4 cost 3** command.

b. The **ping 4.4.4.4** command on R1 must currently be successful.

c. R1's Fa0/0 OSPF cost is 3.

d. 4.4.4.4 is known to R1 based on a Type 1 LSA in area 1.

2. Several links have been broken so that for the next day or two, what was formerly a contiguous area 0 has been broken into two parts. However, both parts of area 0 have working links into area 1 using routers with RID 1.1.1.1 and 2.2.2.2. Which answers list the command on the router with RID 1.1.1.1 to create a virtual link to help solve this temporary problem?

   a. **area 0 virtual-link 2.2.2.2**

   b. **area 1 virtual-link 2.2.2.2**

   c. **area 0 source-rid 1.1.1.1 dest-rid 2.2.2.2**

   d. **virtual-link transit-area 1 RID 2.2.2.2**

3. Router R1 connects to a Frame Relay cloud using a multipoint subinterface, with ten PVCs associated with the subinterface. What command would make this router not use a DR and require static OSPF neighbor definition?

   a. **ip ospf network broadcast**

   b. **ip ospf network non-broadcast**

   c. **ip ospf network point-to-multipoint**

   d. **ip ospf network point-to-multipoint non-broadcast**

4. Router R1 connects to a Frame Relay cloud using a multipoint subinterface, with ten PVCs associated with the subinterface. What command would make this router not use a DR, and dynamically discover OSPF neighbors?

   a. **ip ospf network broadcast**

   b. **ip ospf network non-broadcast**

   c. **ip ospf network point-to-multipoint**

   d. **ip ospf network point-to-multipoint non-broadcast**

5. Ten routers, R1 through R10, connect in a partial mesh over Frame Relay. For the mesh, R1 and R2 have PVCs connected to all other routers, but Routers R3 through R10 act as branch routers, with only two PVCs–one to R1 and one to R2. The routers use IP subnet 10.1.1.0/24, with addresses 10.1.1.1, 10.1.1.2, and so on, through 10.1.1.10, respectively. The routers all use Inverse ARP to learn Frame Relay mapping information. All routers use a multipoint subinterface with network type point-to-multipoint nonbroadcast. A co-worker's implementation plan lists lots of configuration commands related to this design. The design states that all hosts should be able to ping all other hosts. Which commands are required for proper functioning of OSPF in this case? (Choose two.)

   a. **frame-relay map** commands on R3–R10 referencing the other routers in this group.

   b. Nine OSPF **neighbor** commands on each router.

   c. Nine OSPF **neighbor** commands on R1 and R2, with only two such commands on R3–R10.

   d. R1 and R2 with **ip ospf priority 1** commands to ensure they become DR and BDR.

   e. R3–R10 with **ip ospf priority 0** commands to ensure they do not become DR or BDR.

Foundation Topics

Virtual Links

OSPF area design requires the use of a backbone area, area 0, with each area connecting to area 0 through an ABR. However, in some cases two backbone areas exist; in other cases, a nonbackbone area may not have a convenient point of connection to the backbone area, for example:

Case 1. An existing internetwork needs to add a new area, with a convenient, low-cost connection point with another nonbackbone area. However, that connection does not give the new area any connection to area 0.

Case 2. Even with a well-designed area 0, a combination of link failures might result in a discontiguous backbone area, essentially creating two backbone areas.

Case 3. Two companies could merge, each using OSPF. To merge the OSPF domains, one backbone area must exist. It may be more convenient to connect the two networks using links through an existing nonbackbone area, but that design means two backbone areas, which is not allowed.

Figure 8-1 shows an example of each of the first two cases.

The problems in each case have different symptoms, but the problems all stem from the area design requirements: Each area should be contiguous, and each nonbackbone area should connect to the backbone area through an ABR. When the network does not meet these requirements, engineers could simply redesign the areas. However, OSPF provides an alternative tool called an OSPF virtual link.

Understanding OSPF Virtual Link Concepts

An OSPF virtual link allows two ABRs that connect to the same nonbackbone area to form a neighbor relationship through that nonbackbone area, even when separated by many other routers and subnets. This virtual link acts like a virtual point-to-point connection between the two routers, with that link inside area 0. The routers form a neighbor relationship, inside area 0, and flood LSAs over that link.

For example, consider the topology in Figure 8-2, which shows an example of the third of the three cases described in the beginning of this section. In this case, two companies merged. Both companies had a small office in the same city, so for expediency's sake, they connected the two former Enterprise internetworks through a newly combined local sales office in area 1.

Although adding the link between branch offices may be a cost-effective temporary choice, it creates a design problem: Two backbone areas now exist, and OSPF requires that the backbone area be contiguous. To solve this problem, the engineer configures a virtual link between ABRs C1 and C2. The virtual link exists inside area 0, making area 0 contiguous.

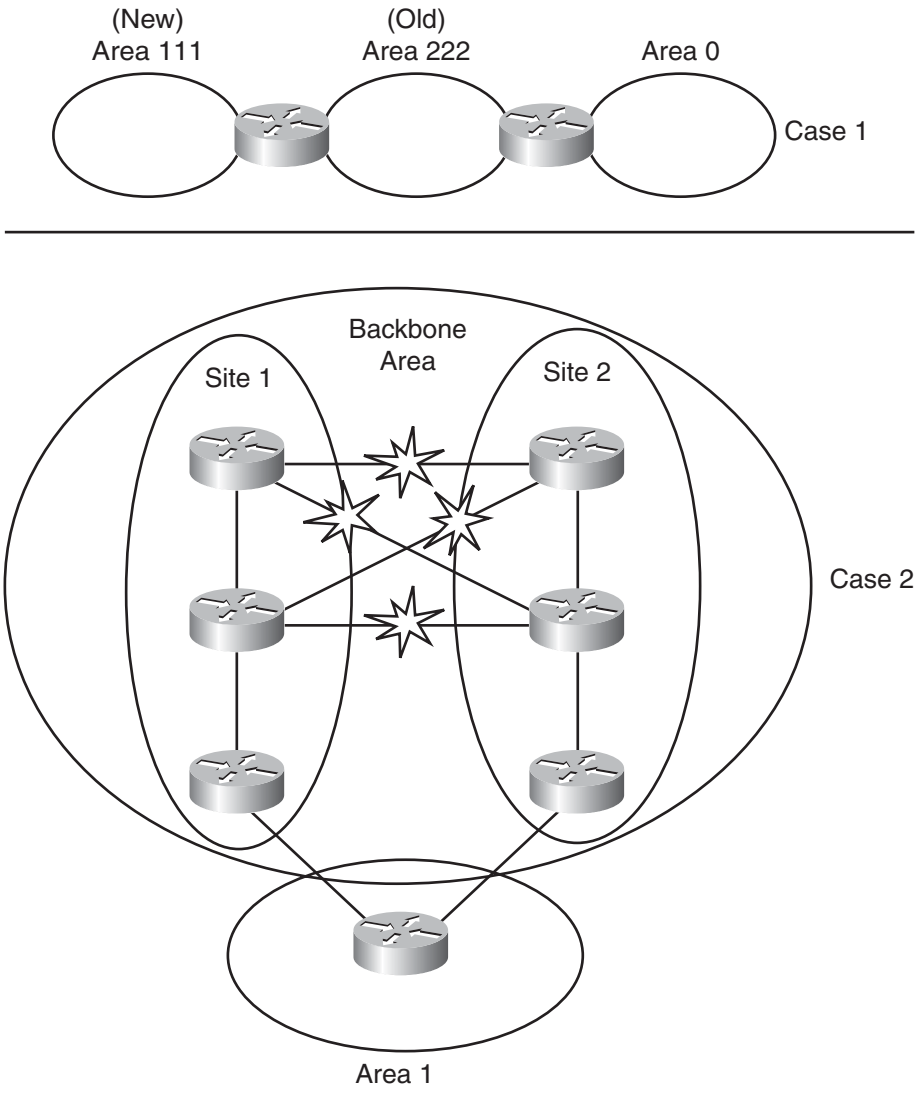


Figure 8-1 *Examples of Area Design Issues*

To define the virtual link, each router configures the other router's RID and a reference to the area through which the virtual link passes (area 1 in this case). The two routers send the usual OSPF message types, encapsulated inside unicast IP packets, with a destination IP address of the router on the other end of the virtual link. Any routers between the two routers that create the virtual link—for instance, the two branch routers in Figure 8-2—just forward these OSPF packets like any other packet. The neighbors on the ends of the virtual link flood their LSDBs to each other so that all routers in both parts of area 0 learn the routes from the other area 0.

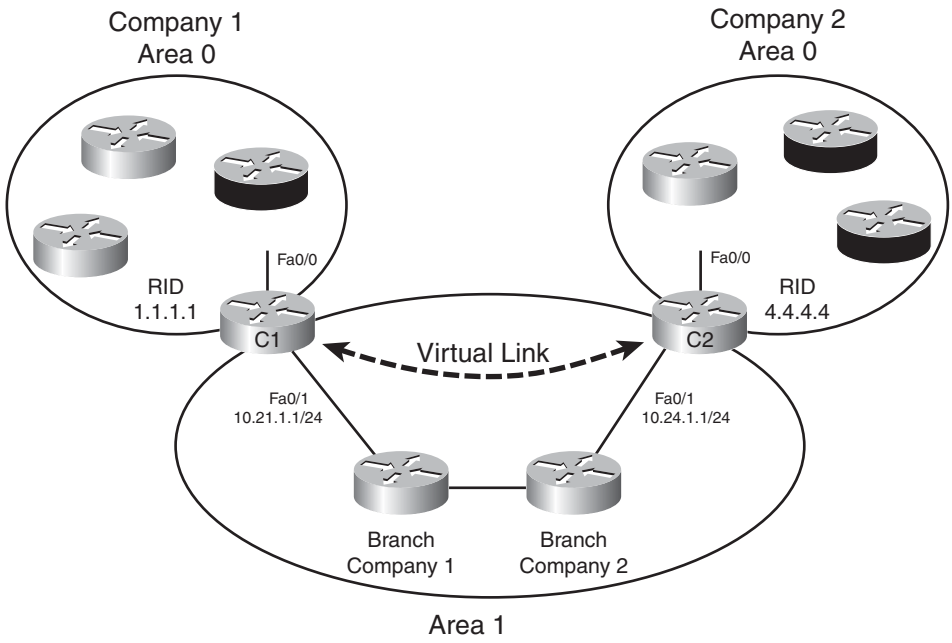


Figure 8-2 *Connecting Two Area 0s with a Virtual Link*

The ABRs connected over a virtual link act mostly like any other ABR, with a couple of differences. The first difference is that ABRs send all OSPF messages as unicasts to the IP address of the router on the other end of the link. Second, the routers also mark the *Do Not Age (DNA)* bit in the LSAs, meaning that all routers on the other side of the virtual link will not expect the LSAs to be reflooded over the virtual link on the usual 30-minute refresh interval. This helps reduce overhead over the virtual link, which often runs over slower links and less-powerful routers. The router also assigns an OSPF cost to the virtual link, just as it would for an interface.

After the virtual link is up, the ABRs' SPF processes can calculate their best routes just like before, using the virtual link as another point-to-point link in area 0. For packets destined to pass from one part of the backbone over the virtual link to the other part of the backbone, the chosen best routes eventually lead the packets to the router with the virtual link. That router, connected to the transit nonbackbone area, has already calculated its next hop based on the LSDB in the transit area (Router C1 and transit area 1 in the example of Figure 8-2). The routers in the transit area choose routes that eventually deliver the packet to the router on the other end of the virtual link (Router C2 in Figure 8-2).

Configuring OSPF Virtual Links with No Authentication

Configuring an OSPF virtual link requires a minor amount of configuration just to get the link working, with several optional configuration items. Most of the optional configuration settings relate to features that would normally be configured on the interface connecting two neighboring routers, but with a virtual link, there is no such interface, so the

parameters must be added to the **area virtual-link** command. The following list summarizes the key configuration options on the **area virtual-link** router subcommand:

- The *remote-RID* in the **area** *area-num* **virtual-link** *remote-RID* command refers to the other router's RID.
- The *area-num* in the **area** *area-num* **virtual-link** *remote-RID* command refers to the transit area over which the packets flow between the two routers.
- The transit area over which the two routers communicate must not be a stubby area.
- The optional configuration of OSPF neighbor authentication parameters, normally configured as interface subcommands, must be configured as additional parameters on the **area virtual-link** command.
- The optional configuration of Hello and Dead intervals, normally configured as interface subcommands, must be configured as additional parameters on the **area virtual-link** command.
- The router assigns the virtual link an OSPF cost as if it were a point-to-point link. The router calculates the cost as the cost to reach the router on the other end of the link, as calculated using the transit area's LSDB.

Example 8-1 shows the configuration of a virtual link on Router C1 and Router C2 shown in Figure 8-2. The configuration shows the virtual link, referencing area 1 as the transit area, with each router referring to the other router's RIDs. The configuration also shows the loopback IP addresses on which the ABR's RIDs are based being advertised into OSPF.

Example 8-1 *OSPF Virtual Link Configuration on Routers C1 and C2*

```
! On Router C1:
router ospf 1
 area 1 virtual-link 4.4.4.4
!
interface fastethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip ospf 1 area 0
!
interface fastethernet0/1
 ip address 10.21.1.1 255.255.255.0
 ip ospf 1 area 1
!
interface loopback 1
 ip address 1.1.1.1 255.255.255.0
 ip ospf 1 area 1
```

```
! On Router C2:
router ospf 4
 area 1 virtual-link 1.1.1.1
```

```
!
interface fastethernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip ospf 4 area 0
!
interface fastethernet0/1
 ip address 10.21.1.1 255.255.255.0
 ip ospf 4 area 1
!
interface loopback 1
 ip address 4.4.4.4 255.255.255.0
 ip ospf 4 area 1
```

Verifying the OSPF Virtual Link

To prove whether the virtual link works, a neighbor relationship between C1 and C2 must reach FULL state, resulting in all routers in both parts of area 0 having the same area 0 LSDB. Example 8-2 shows the working neighbor relationship, plus status information for the virtual link with the **show ip ospf virtual-link** command.

Example 8-2 *OSPF Virtual Link Configuration on Routers C1 and C2*

```
C1#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 4.4.4.4 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 1, via interface FastEthernet0/1, Cost of using 3
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Adjacency State FULL (Hello suppressed)
    Index 1/2, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
!
! next, note that the neighbor reaches FULL state, with no DR elected.

C1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4           0   FULL/  -        -           10.24.1.1       OSPF_VL0
2.2.2.2           1   FULL/DR         00:00:35    10.21.1.2
FastEthernet0/1
```

```
C1#show ip ospf neighbor detail 4.4.4.4
 Neighbor 4.4.4.4, interface address 10.24.1.1
    In the area 0 via interface OSPF_VL0
    Neighbor priority is 0, State is FULL, 6 state changes
    DR is 0.0.0.0 BDR is 0.0.0.0
    Options is 0x32 in Hello (E-bit, L-bit, DC-bit)
    Options is 0x72 in DBD (E-bit, L-bit, DC-bit, O-bit)
    LLS Options is 0x1 (LR)
    Neighbor is up for 00:00:21
    Index 1/2, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
   Last retransmission scan time is 0 msec, maximum is 0 msec
```

The only new command in the example, **show ip ospf virtual-links**, details some items unique to virtual links. In particular, the first highlighted portion shows the assignment of a name to the link (VL0); if multiple were configured, each would have a different number. This virtual link name/number is then referenced inside the LSDB. It also shows that the routers both allow the use of the Do Not Age (DNA) bit, so periodic reflooding will not occur over this virtual link. It lists a cost of 3; as it turns out, each of the three interfaces between Router C1 and C2 have an OSPF cost of 1, so C1's area 1 cost to reach C4 is 3. The output also confirms that the routers have reached a fully adjacent state and is suppressing the periodic Hello messages.

The familiar **show ip ospf neighbor** command lists a few new items as well. Note that the interface refers to the virtual link "OSPF VL0" instead of the interface, because there is no interface between the neighbors. It also lists no dead timer, because the neighbors choose to not use the usual Hello/Dead interval process over a virtual link. (Instead, if all the transit area's routes to reach the router on the other router of the link fail, the virtual link fails.) Finally, the **show ip ospf neighbor detail 4.4.4.4** command shows the interesting phrase "In the area 0 via interface OSPF VL0," confirming that the neighborship does indeed exist in area 0.

Note: OSPF does not require that the RID IP address range be advertised as a route in OSPF. As a result, the RID listed in the **area virtual-link** command may not be pingable, but the virtual link still work.

Configuring Virtual Link Authentication

Because virtual links have no underlying interface on which to configure authentication, authentication is configured on the **area virtual-link** command itself. Table 8-2 shows the variations of the command options for configuring authentication on virtual links. Note that the configuration shown in Table 8-2 may be typed in one longer **area virtual-link** command, as shown in the table, or the authentication type and key can be configured on

separate **area virtual-link** commands. Regardless, IOS stores two different **area virtual-link** commands into the running-config: one command that enables the type of authentication, and one that lists the authentication key, as shown in upcoming Example 8-3.

Key Topic

Table 8-2 *Configuring OSPF Authentication on Virtual Links*

Type (Name)	Type (Number)	Command Syntax for Virtual Links
none	0	**area** *num* **virtual-link** *router-id* **authentication null**
clear text	1	**area** *num* **virtual-link** *router-id* **authentication authentication-key** *key-value*
MD5	2	**area** *num* **virtual-link** *router-id* **authentication message-digest message-digest-key** *key-num* **md5** *key-value*

Example 8-3 shows a modified version of the configuration shown in Example 8-2, now with MD5 authentication configured on both C1 and C2.

Example 8-3 *OSPF Virtual Link Configuration on Routers C1 and C2*

```
! On Router C1 - configuring the authentication type and key
Router ospf 1
 Area 1 virtual-link 4.4.4.4 authentication message-digest message-digest-key 1
md5 fred
```

```
! On Router C2 - configuring the authentication type and key
router ospf 4
 Area 1 virtual-link 1.1.1.1 authentication message-digest message-digest-key 1
md5 fred
!
! The router separated the authentication type and authentication key
! into two separate commands.
C2#show running-config
! line omitted for brevity
router ospf 4
 area 1 virtual-link 1.1.1.1 authentication message-digest
 area 1 virtual-link 1.1.1.1 message-digest-key 1 md5 fred

C2#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 1.1.1.1 is up
! lines omitted for brevity
  Message digest authentication enabled
    Youngest key id is 1
```

This concludes the discussion of OSPF virtual links. As mentioned in the introduction of this chapter, the chapter now changes focus completely to discuss OSPF issues when using Frame Relay.

OSPF over Multipoint Frame Relay

To form OSPF neighbor relationships on LANs and point-to-point WAN interfaces, OSPF simply needs to be enabled on the interface, and the neighboring routers must agree to a set of parameters (as summarized in Chapter 5, Table 5-5.) However, when using Frame Relay in certain configurations, engineers must do more implementation planning and use additional configuration commands to make OSPF work. Additionally, the people monitoring and operating the network must also understand several differences in how OSPF works in such cases and be ready to interpret the output of **show** commands.

The issues described in this section of the chapter arise when the IP subnetting and Frame Relay design places more than two WAN routers into a single subnet. Such a design requires multipoint logic, either configured on the physical serial interface or on a multipoint subinterface.

This section begins by reviewing the IP subnetting design issues that can lead to the broader OSPF issues over Frame Relay. Next, this section examines the issues of running OSPF over Frame Relay, concentrating on the concepts. This final part of this section ends with several OSPF configuration examples using multipoint Frame Relay.

IP Subnetting Design over Frame Relay

When planning the subnets to use over a Frame Relay WAN, the design engineer has several options. The simplest and most obvious choice uses a single subnet for every PVC (as seen throughout the examples in Chapters 5, 6, and 7), with all routers using point-to-point subinterfaces associated with each single PVC. Most companies who still use Frame Relay today still use such a subnetting design.

In some cases, a shortage of IP addresses may make a design engineer choose to use fewer WAN subnets, with less waste. Many companies use private IP addresses inside their Enterprise internetwork–often times specifically class A network 10.0.0.0–and most companies do not have a shortage of IP addresses and subnets available. However, using one subnet per PVC does consume more IP addresses than the alternative–to put more than two routers into a Frame Relay WAN subnet. You get the advantage of fewer IP addresses consumed in the WAN but the penalty of more configuration and complexity with the OSPF implementation.

For example, consider the two internetworks shown in Figure 8-3. On the left, four routers connect in a full mesh of PVCs, whereas on the right, the four routers use a partial mesh. In each case, assume each router uses their s0/0/0 interface to connect to the cloud.

First, consider the differences in IP addressing for the one-subnet-per-PVC choice and the single-subnet choice. The single-subnet design could use just one subnet–for instance, subnet 10.1.1.0/29, with usable address 10.1.1.1–10.1.1.6 (subnet broadcast 10.1.1.7). With four routers, the subnet contains enough IP addresses for each router, plus two additional addresses. Conversely, with a single-subnet-per-PVC, the design would require six subnets,

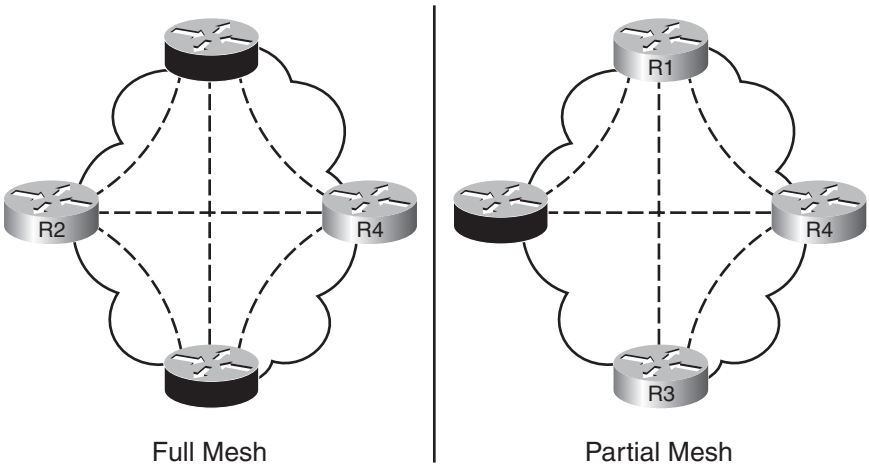


Figure 8-3 *Sample Full and Partial Mesh Frame Relay*

each of which consumes four numbers: subnet number, two usable IP addresses, and a subnet broadcast address. For example, the design could use subnets 10.1.1.0/30, 10.1.1.4/30, 10.1.1.8/30, 10.1.1.12/30, 10.1.1.16/30, and 10.1.1.20.0/30 for a range of consumed numbers of 10.1.1.0–10.1.1.23.

Sample Configuration Using Physical Interfaces

Ignoring OSPF for now, to implement such a design on the internetwork shown in Figure 8-3, a somewhat short configuration could be used on each router. Example 8-4 shows the configuration, with the IP addresses being defined on the physical serial interfaces on each router.

Example 8-4 *OSPF over Point-to-Point Frame Relay Subinterfaces*

```
! Router R1
interface S0/0/0
 encapsulation frame-relay
 ip address 10.2.123.1 255.255.255.248
```

```
! Router R2
interface S0/0/0
 encapsulation frame-relay
 ip address 10.2.123.2 255.255.255.248
```

```
! Router R3
interface S0/0/0
 encapsulation frame-relay
 ip address 10.2.123.3 255.255.255.248
```

```
! Router R4
interface S0/0/0
 encapsulation frame-relay
 ip address 10.2.123.4 255.255.255.248
```

This brief configuration works in part because of many default settings. IOS enables Frame Relay Inverse ARP (InARP) by default, so all four routers learn of the other routers' IP address/DLCI mappings dynamically. All PVCs learned on the interface will be associated with the physical interface.

As a result, after creating this configuration, with a working Frame Relay network, the four routers can **ping** each other in every case for which a PVC exists between the two routers. In other words, in the full mesh case, all four routers should ping the other three routers' IP addresses in the 10.1.1.0/29 subnet. In the partial mesh case, the R1/R3 pair cannot ping each other.

The ping problem between R1 and R3 demonstrates some key concepts related to how Frame Relay mapping works. InARP allows the routers to learn the IP address of any router on the other end of a PVC, but R1/R3 do not have a PVC connecting them in the partial mesh side of Figure 8-3, so InARP does not learn the mapping. Then, when R1 (10.2.123.1) wants to send a packet to 10.2.123.3 (R3), R1 looks at its Frame Relay mapping information and does not find 10.2.123.3. As a result, R1 discards the packet because it does not know what DLCI to use to forward the packet.

Sample Configuration Using Multipoint Subinterfaces

The same effects occur with a valid alternate configuration using a multipoint subinterface, both in the full and partial mesh cases. When using subinterfaces, IOS needs a command under a Frame Relay subinterface that associates each PVC with that subinterface: either a **frame-relay interface-dlci** command or **frame-relay map** command. The **frame-relay interface-dlci** command just associates the DLCI with the subinterface, relying on Frame Relay InARP to discover mappings. The second command (**frame-relay map**) statically configures that mapping, while also associating the DLCI with the subinterface.

Example 8-5 shows an alternative configuration on Router R1, using a multipoint subinterface. All four routers could mimic the same configuration, but that configuration is not shown in the example.

Example 8-5 *OSPF over Multipoint Frame Relay Subinterfaces*

```
! Router R1
interface S0/0/0
 encapsulation frame-relay
!
interface S0/0/0.1 multipoint
 ip address 10.1.123.1 255.255.255.248
 frame-relay interface-dlci 102
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
```

This configuration on R1, with similar configuration on the other three routers, has the same results as those with Example 8-4. When a full mesh exists, all routers should be pingable. With a partial mesh, all routers should be pingable, except those with no PVC between the two routers (R1/R3 in this case).

With those background details in mind, the chapter now examines the challenges that arise when trying to use multipoint.

OSPF Challenges When Using Multipoint

Next, consider OSPF over both the full and partial meshes as described in the last few pages. As it turns out, if you enabled OSPF on the interfaces, but do not add OSPF configuration, OSPF does not form neighborships in either case.

Note: Rather than repeat the long phrase "on physical interfaces or on multipoint interfaces" many times in the coming pages, the rest of this chapter simply refers to both as "multipoint interfaces," because more than one destination may be reachable through these types ofinterfaces.

The default OSPF network type on multipoint interfaces–the nonbroadcast network type–prevents OSPF from working without further configuration. Additionally, depending on the configuration you add to make OSPF work, the network may appear to be working in that the correct routes are learned, but packets cannot be forwarded. Worse yet, packet forwarding could work when testing, but later, when the DR in that subnet failed, packet forwarding could fail, even though paths exist over different PVCs.

To avoid such pitfalls, this section examines the issues. To understand the issues, an engineer needs to know the answer to three questions related to OSPF over Frame Relay multipoint interfaces:

- Do the routers attempt to discover neighbors by sending and receiving multicast OSPF Hello messages, or do the neighbors require static definition?
- Do the routers attempt to elect a DR/BDR?
- Does a partial mesh exist or full mesh?

The next few pages examine these topics one at a time.

Neighbor Discovery or Static Neighbor Definition

The OSPF network type–a per-interface setting–defines whether a router attempts to discover OSPF neighbors on an interface. When the router attempts discovery, it sends multicast (224.0.0.5) OSPF Hello messages out the interface.

The first potential problem occurs when a Frame Relay configuration oversight causes the dynamic OSPF neighbor discovery process to fail. When a router needs to send a broadcast or multicast over Frame Relay, the router must actually send a copy of the packet over each PVC instead, because the Frame Relay network does not have the ability to replicate the broadcast or multicast packet. Routers do send the multicast OSPF Hellos for any PVCs listed in **frame-relay interface-dlci** commands, and for any DLCIs listed in **frame-relay map** commands if the **broadcast** keyword was included. However, if the **frame-relay map** command omits the **broadcast** keyword, that router will not replicate the OSPF multicasts, or any other broadcast or multicast frames for that DLCI.

Alternatively, some OSPF network types tell a router to not attempt automatic discovery of neighbors. In such a case, the router should be configured with the **neighbor** *ip-address* OSPF router subcommand. This command overcomes the issue with neighbor discovery.

The actions to take for the issues related to neighbor discover are somewhat straightforward:

- If the configured network type allows for neighbor discovery using Hellos, and you use InARP, the neighbors should be discovered.
- If the configured network type allows for neighbor discovery, and you use **frame-relay map**, ensure that the **broadcast** keyword is included.
- If the configured network type does not allow for neighbor discovery, statically configure neighbors.

The upcoming samples show cases of using network types that dynamically discover neighbors and cases that show static definition of neighbors.

To Use a Designated Router, or Not

IOS has several available OSPF network types for Frame Relay, some of which tell the router to try to elect a DR/BDR. The reasoning is that when choosing a subnet design like those shown in Figure 8-3, with a single subnet that includes more than two Frame Relay routers, the design looks like a LAN from a Layer 3 perspective. Other options for OSPF network types tell the routers to not elect a DR/BDR.

A potential issue exists when using a DR over Frame Relay. The issue relates to whether the Frame Relay partial mesh includes the right PVCs to support the OSPF messages that flow in the presence of a DR/BDR. As you may recall from Chapter 6, "OSPF Topology, Routes, and Convergence," when a DR exists, all OSPF routers must send and receive OSPF messages with the DR, with messages to the DR being sent to the 224.0.0.6 all-DR's multicast address. Additionally, the DR sends messages to the 224.0.0.5 all-SPF router's multicast address, with these messages intended for all OSPF routers in the subnet. For the same reason, all routers should send and receive messages with the BDR, in case it takes over for the DR. However, two DROther routers do not need to send each other OSPF messages.

For a Frame Relay partial mesh to work when also using an OSPF DR, a PVC must exist between any two routers that need to perform database exchange directly with each other. In short, the following PVCs must exist:

- Between the DR and every other router in the subnet
- Between the BDR and every other router in the subnet

For example, consider the partial mesh shown in Figure 8-4, which mirrors the partial mesh seen on the right side of Figure 8-3. In this case, R1 acts as DR, and R2 as BDR. OSPF actually works fine in this design, until the fated day when R1's serial link fails. When it fails, R2 takes over as DR, but R3 has no way to send and receive OSPF messages with R2. As a result, R3 will no longer learn routes that pass over the WAN.

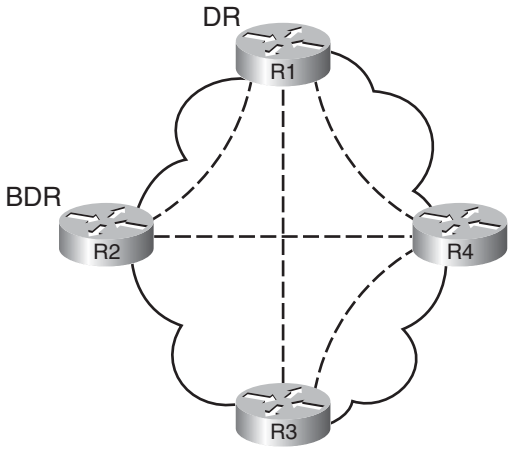


Figure 8-4 *Poor Choices of BDR in a Partial Mesh*

To deal with this particular issue with a partial mesh, either avoid using a DR at all by changing the OSPF network type on all routers in the subnet, or restrict the role of DR/BDR to routers with PVCs connected to all other routers in a subnet. For example, R1 and R4 in Figure 8-4 have three PVCs, one each to the other three routers. By configuring the routers so that R2 and R3 refuse to become DR or BDR, you can avoid the case of a poor choice of DR/BDR.

Summarizing the key points about the issues in this section:

Key Topic

- If using a network type that requires a DR/BDR, restrict the role of DR and BDR to routers that have a PVC connecting to all other routers.
- Configure the OSPF network type to avoid using a DR/BDR.

Mapping Issues with a Partial Mesh

The last issue with OSPF over multipoint Frame Relay topologies relates to the Layer 2 mapping, either created by InARP or statically configured with **frame-relay map** commands. This mapping on a router lists each neighbor's IP address, and that router's DLCI used to reach that router. InARP learns this mapping for each router on the other end of a PVC, but not for other routers where no PVC exists. For example, in Figure 8-4, R2 and R3 would not learn mapping for each other with InARP. Similarly, when configuring the **frame-relay map** command, many engineers think only of statically creating mappings for neighboring routers on the other end of each PVC. In short, the mapping in both cases matches the partial mesh.

OSPFmultipoint interfaces have problems when the Frame Relay mapping matches the partial mesh but does not include mapping to every other router in the subnet–even those routers where no PVC exists. The best way to describe the problem is to start with an example, as seen in Figure 8-5. The figure shows the same old partial mesh, with some IP addresses and DLCIs shown.

R2's IP Routing Table

Subnet	Next-hop
10.3.3.0/26	10.2.14.3

Frame Relay Mapping

IP	DLCI
10.2.14.2	101
10.2.14.3	101
10.2.14.4	104

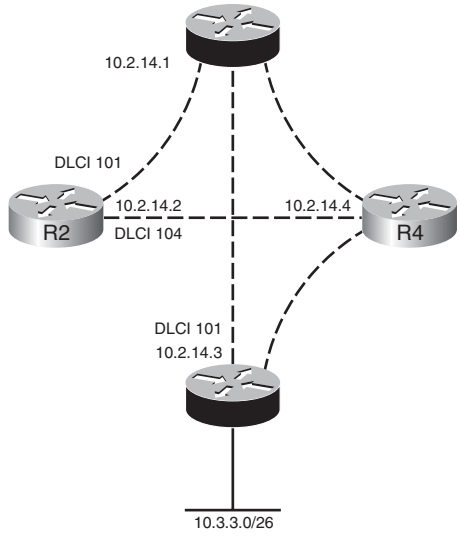


Figure 8-5 *Frame Relay Mapping Issues*

When OSPF works over this Frame Relay WAN, the next-hop address of the learned routes will list the IP address of a router in that WAN subnet–*even if no PVC connects to that next-hop router*. In this case, R2's route for 10.3.3.0/26 lists R3, 10.2.14.3, as next hop. When R2 attempts to add a Frame Relay header to the packet, and assign a DLCI, R2 must have a mapping between that next-hop address (10.2.14.3) and the DLCI to use to reach that next hop. However, no such Frame Relay mapping exists on R2, because R2 was relying on InARP–and InARP works only directly over a PVC and not indirectly.

To solve the problem, R2 needs a mapping for next hop 10.2.14.3 to use a DLCI to another router, allowing that other router to use IP routing to forward the packet to R3. For example, R2 could use DLCI 101, the PVC connecting R2 to R1. By doing so, R2 will encapsulate forwarded packets inside a frame with DLCI 101, sending the packet to R1. R1 will then use IP routing to forward the packet on to R3. Similarly, R3 could use a command that statically mapped 10.2.14.2, R2's IP address, to DLCI 101, which is R3's DLCI used for its PVC to R1.

Summarizing the issue and the solution to this problem:

> For any routers without a direct PVC, statically configure a IP-address-to-DLCI mapping with the other router's next-hop IP address, and a DLCI that does connect to a router that has PVCs with each of the two routers

Configuring and Verifying OSPF Operations on Frame Relay

The concepts related to the issues of running OSPF over Frame Relay can require a bit of thought about things that many of us can ignore when working with a typical OSPF network. Thankfully, the implementation choices are a little more straightforward. When the design dictates more than two routers per subnet over Frame Relay, you need to configure Frame Relay to use either physical interfaces or multipoint subinterfaces. In those cases, only four OSPF network types exist. Then, the choice of OSPF network type answers two big questions: Will the routers try and elect a DR, and will they try to discover neighbors? From there, the rest of the implementation plan flows from the issues already discussed in this chapter.

For reference, Table 8-3 summarizes the four OSPF network types available on serial interfaces that use Frame Relay, either the physical interface or a multipoint subinterface. Note that the setting is per interface and can be set with the **ip ospf network** *type* interface subcommand; the first column in the table lists the exact keyword according to this command.

Key Topic

Table 8-3 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Dynamic Discovery of Neighbors?	Default Hello Interval	Cisco Proprietary?
broadcast	Yes	Yes	10	Yes
nonbroadcast	Yes	No	30	No
point-to-multipoint	No	Yes	30	Yes
point-to-multipoint nonbroadcast	No	No	30	No

All the upcoming examples use the same network diagram as shown in Figure 8-6. After the figure, the text moves on to the examples showing this internetwork configured with three options for network type on Frame Relay serial and multipoint subinterfaces: nonbroadcast, point-to-multipoint, and point-to-multipoint nonbroadcast. (This chapter does not include a sample of using network type broadcast, because the LAN-based examples in Chapter 5 all use a default of network type broadcast.)

Note: The following samples all use multipoint subinterfaces. Also, for each example, all four routers use the same OSPF network type. However, note that IOS has no mechanism to prevent routers from using different interface types or different OSPF network types, but such inconsistency can cause problems and be very difficult to operate. Configurations that mix OSPF network types in the same subnet are outside the scope of this book.

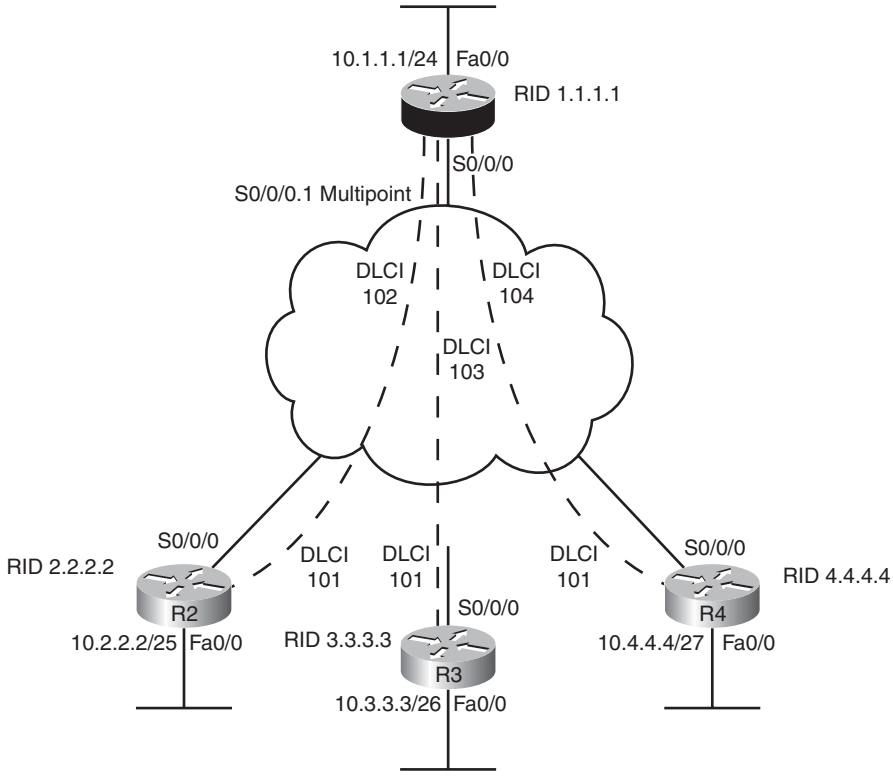


Figure 8-6 *Sample Partial Mesh for Frame Relay Configuration*

Using Network Type Nonbroadcast (NBMA)

By using the (default) network type setting of nonbroadcast, all four routers will elect a DR and BDR. They also will not attempt to multicast Hellos to discover each other, so they need to configure each other in an OSPF **neighbor** commands, as follows:

```
neighbor next-hop-IP [cost cost-value] [priority priority-value]
```

To statically configure an OSPF neighbor, simply use the **neighbor** command in OSPF configuration mode, one command per neighbor. The command references the IP address of the neighbor, specifically the IP address in the subnet to which both the local and neighboring router connect. Also, OSPF requires only this command for two routers connected to the same PVC; routers not connected with a PVC cannot become OSPF neighbors, so no **neighbor** command is required. So, in the design shown in Figure 8-6, router pairs R2-R3, R3-R4, and R2-R4 will not define each other as neighbors with the **neighbor** command.

Next, consider the issue of election of a DR/BDR. The section "To Use a Designated Router, or Not" earlier in this chapter discussed that if a DR is indeed elected, you need to restrict routers so that only routers with PVCs to all other routers are allowed to become DR or BDR. In Figure 8-6, only R1 meets the requirement, so the configuration must limit on R1 to become the DR and simply not use a BDR. (Given the design, a failure of R1

would prevent any further communication on the WAN anyway, so the lack of a BDR does not make the problem worse.)

The OSPF priority defines the priority when choosing a DR and BDR. Configured with the **ip ospf priority** *value* interface subcommand, the higher number (ranges from 0–255) wins. Also, a priority of 0 has special meaning: it prevents that router from becoming the DR or BDR. By configuring OSPF priority of 0 on routers that do not have a PVC connected to each other router (such as R2, R3, and R4 in this case), the other routers (R1 in this case) will be the only routers eligible to become the DR.

Example 8-6 begins by showing the required configuration, defining neighbors and setting the OSPF priority on R2, R3, and R4 to 0. The example shows just R1 and R2; R3's and R4's configuration mirror R2's, other than the IP address settings.

Example 8-6 *OSPF over Multipoint Frame Relay Subinterfaces–Default Configuration*

```
! First, R1's configuration
R1#show running-config int s0/0/0.1
Building configuration...

Current configuration : 169 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.1 255.255.255.248
 frame-relay interface-dlci 102
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
end

R1#show running-config ¦ begin router ospf
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 router-id 1.1.1.1
 neighbor 10.2.123.2
 neighbor 10.2.123.3
 neighbor 10.2.123.4
```

```
! Next, R2's configuration
R2#show running-config interface s0/0/0.1
Building configuration...

Current configuration : 113 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.2 255.255.255.248
 frame-relay interface-dlci 101
 ip ospf priority 0
end
```

```
R2#show running-config ¦ begin router ospf
router ospf 2
 network 10.0.0.0 0.255.255.255 area 0
 router-id 2.2.2.2
 neighbor 10.2.123.1

R2#show ip ospf interface s0/0/0.1
Serial0/0/0.1 is up, line protocol is up
  Internet Address 10.2.123.2/29, Area 0
  Process ID 2, Router ID 2.2.2.2, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DROTHER, Priority 0
  Designated Router (ID) 1.1.1.1, Interface address 10.2.123.1
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
```

Example 8-6 shows the configuration of three neighbors on router R1 (R2, R3, and R4), plus one neighbor on R2 (R1). It does not show the explicit configuration of the OSPF network type because each router uses the default for multipoint subinterfaces of non-broadcast. However, the **show ip ospf interface** command at the end of the output confirms the network type on the multipoint subinterface of R2.

The priority configuration results in R1 becoming DR, and none of the other routers being willing to even become BDR. R1 omits an **ip ospf priority** command, defaulting to 1, while R2 shows an **ip ospf priority 0** command, removing itself from consideration as DR or BDR. The output at the bottom of the page again confirms the result on R2, with its own priority of 0, the result of RID 1.1.1.1 (R1) as DR, but with no BDR elected.

Note: You can also attempt to configure a neighboring router's OSPF priority with the **neighbor** *address* **priority** *priority-value* command.

Although the configuration shown in Example 8-6 takes care of the particulars of defining neighbors and of influencing the choice of DR/BDR, it does not, however, include the configuration for the Frame Relay mapping issue with a partial mesh. All four routers need mapping to the other three router's IP addresses in their common Frame Relay subnet. In this particular case, R1 learns all the mappings with InARP, because it has a PVC connected to all the other routers. However, R2, R3, and R4 have learned all OSPF routes but cannot forward traffic to each other because they have not learned the mappings.

Example 8-7 shows the background regarding this issue, along with an example solution on R2 using **frame-relay map** commands. Note that comments inside the example show the progression of proof and changes to solve the problem.

Example 8-7 *The Need for* ***frame-relay map*** *Commands*

```
! First, R2 has routes for the LAN subnets off R1, R3, and R4, each referring
! to their respective routers as the next-hop IP address
R2#show ip route ospf
      10.0.0.0/8 is variably subnetted, 5 subnets, 5 masks
O        10.4.4.0/27 [110/65] via 10.2.123.4, 00:03:52, Serial0/0/0.1
O        10.3.3.0/26 [110/65] via 10.2.123.3, 00:06:02, Serial0/0/0.1
O        10.1.1.0/24 [110/65] via 10.2.123.1, 00:06:12, Serial0/0/0.1

! Next, R2 successfully pings R1's LAN IP address

R2#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

! Next, R2 fails when pinging R3's LAN IP address

R2#ping 10.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)


! R2 has mapping only for 10.2.123.1 in the Frame Relay subnet
R2#show frame-relay map
Serial0/0/0.1 (up): ip 10.2.123.1 dlci 101(0x65,0x1850), dynamic,
               broadcast,
               CISCO, status defined, active

! Next, the engineer adds the mapping for R3 and R4, with DLCI 101 as the
! PVC, which connects to R1

R2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int s0/0/0.1
R2(config-subif)#frame-relay map ip 10.2.123.3 101 broadcast
R2(config-subif)#frame-relay map ip 10.2.123.4 101 broadcast
R2(config-subif)#^Z
!
! Not shown, similar mapping configured on R3 and R4
```

```
! R2 now has the necessary mappings
R2#show frame-relay map
Serial0/0/0.1 (up): ip 10.2.123.4 dlci 101(0x65,0x1850), static,
               broadcast,
               CISCO, status defined, active
Serial0/0/0.1 (up): ip 10.2.123.3 dlci 101(0x65,0x1850), static,
               broadcast,
               CISCO, status defined, active
Serial0/0/0.1 (up): ip 10.2.123.1 dlci 101(0x65,0x1850), dynamic,
               broadcast,
               CISCO, status defined, active

! Same ping now works

R2#ping 10.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/8 ms
```

Although the example demonstrates the original problem and then the solution, it does also demonstrate a particularly difficult challenge for a verification plan. Before adding the mapping, OSPF appeared to be working on R2, because R2 had learned all the routes it should have learned. However, packet forwarding failed due to the lack of the correct Frame Relay mappings. When reviewing verification plans for the exam, take extra care when using Frame Relay, and do not rely solely on seeing OSPF routes as proof that connectivity exists between two routers.

Using Network Type Point-to-Multipoint

Network type *point-to-multipoint* tells routers to act oppositely compared to the nonbroadcast type: to not elect a DR/BDR and to dynamically discover neighbors. Both options allow the routers to simply not configure additional commands. As a result, the only configuration requirement, beyond enabling OSPF on the multipoint subinterface or physical serial interface, is to configure any static Frame Relay mappings if a partial mesh exists.

Example 8-8 shows a completed OSPF configuration on Routers R1 and R2 in Figure 8-6. All the routers change their network type using the **ip ospf network point-to-multipoint** interface subcommand. The other big change as compared to using the nonbroadcast network type (as seen in Examples 8-6 and 8-7) is the absence of configuration. (Example 8-8 does show all OSPF-related configuration on R1 and R2.) As with Example 8-6, the example omits the R3 and R4 configuration, but the configuration follows the same conventions as on R2. Also, note that the routers without PVCs to each other router still require the **frame-relay map** commands (included in the example) for the same reasons as shown in Example 8-7.

Example 8-8 *Configuring Multipoint OSPF Using Network Type Point-to-Multipoint*

```
! First, R1's configuration
R1#show run int s0/0/0.1
Building configuration...

Current configuration : 169 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.1 255.255.255.248
 frame-relay interface-dlci 102
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
 ip ospf network point-to-multipoint
end
R1#show run ¦ begin router ospf
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 router-id 1.1.1.1
```

```
! Next, on R2:
R2#show run int s0/0/0.1
Building configuration...

Current configuration : 169 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.2 255.255.255.248
 frame-relay interface-dlci 101
 frame-relay map ip 10.2.123.3 101 broadcast
 frame-relay map ip 10.2.123.4 101 broadcast
 ip ospf network point-to-multipoint
end
R2#show run ¦ begin router ospf
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 router-id 2.2.2.2
```

Example 8-9 confirms a few details about the operation with the point-to-multipoint network type. Taken from R1, the **show ip ospf interfaces S0/0/0.1** command confirms the network type, and it also confirms full adjacency with the other three routers. Also, compared to the same command on Router R2 near the end of Example 8-6, note that this command does not even mention the concept of a DR nor BDR, because this network type does not expect to use a DR/BDR.

Example 8-9 *Confirming the Operation of Network Type Point-to-Multipoint*

```
! First, R1's configuration
R1#show ip ospf interface s0/0/0.1
Serial0/0/0.1 is up, line protocol is up
  Internet Address 10.2.123.1/29, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
    Hello due in 00:00:23
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 4/4, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 3, Adjacent neighbor count is 3
    Adjacent with neighbor 2.2.2.2
    Adjacent with neighbor 4.4.4.4
    Adjacent with neighbor 3.3.3.3
  Suppress hello for 0 neighbor(s)
```

Using Network Type Point-to-Multipoint Nonbroadcast

OSPF network type *point-to-multipoint nonbroadcast* tells routers to act like the similarly named point-to-multipoint type by not electing a DR/BDR. However, the difference lies in neighbor discovery–the keyword "nonbroadcast" implies that the routers cannot broadcast (or multicast) to discover neighbors.

The configuration uses a mix of the commands seen for the nonbroadcast and point-to-multipoint network types. Essentially, the configuration of priority for the purpose of influencing the DR/BDR election can be ignored, because no DR/BDR will be elected. However, the routers need **neighbor** commands to predefine neighbors. Also, regardless of network type, for those routers without direct PVCs, Frame Relay mapping must be added.

This particular OSPF network type does have some design advantages if the design also uses multipoint subinterfaces. The required **neighbor** commands also have an optional **cost** parameter so that the cost associated with each neighbor can be different. With point-to-multipoint, the router considers the dynamically discovered neighbors to be reachable with a cost equal to the cost of the associated multipoint subinterface, so the cost cannot be set per neighbor.

Example 8-10 shows the configurations on R1 and R2 for this case, with the configuration of R3 and R4 mirroring R2s.

Example 8-10 *Configuring Multipoint OSPF Using Network Type Point-to-Multipoint Nonbroadcast*

```
! First, R1's configuration
R1#show run int s0/0/0.1
Building configuration...

Current configuration : 169 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.1 255.255.255.248
 frame-relay interface-dlci 102
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
 ip ospf network point-to-multipoint non-broadcast

end
R1#show run ¦ begin router ospf
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 router-id 1.1.1.1
 neighbor 10.2.123.2
 neighbor 10.2.123.3
 neighbor 10.2.123.4
```

```
! Next, on R2:
R2#show run int s0/0/0.1
Building configuration...

Current configuration : 169 bytes
!
interface Serial0/0/0.1 multipoint
 ip address 10.2.123.2 255.255.255.248
 frame-relay interface-dlci 101
 frame-relay map ip 10.2.123.3 101 broadcast
 frame-relay map ip 10.2.123.4 101 broadcast
 ip ospf network point-to-multipoint non-broadcast

end
R2#show run ¦ begin router ospf
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 router-id 2.2.2.2
 neighbor 10.2.123.1
```

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 8-4 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 8-4 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A merger design plan shows two companies with OSPF backbone areas. How can the two areas 0 be connected? (2)	
The subnetting design over Frame Relay calls for a set of 20 routers to use a single WAN subnet. What Frame Relay interface types make sense?	
The design shows 20 routers connected to Frame Relay, in a partial mesh, with one subnet. What OSPF network type would avoid the use of DR/BDR?	
The design shows 20 routers connected to Frame Relay, in a full mesh, with one subnet. What OSPF network type would allow dynamic neighbor discovery?	

Implementation Plan Peer Review Table

Table 8-5 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 8-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The subnetting design over Frame Relay calls for a set of 20 routers to use a single WAN subnet. What three general issues exist that requires extra configuration?	
The plan shows a virtual link between R1 and R2, through area 99, with the connection using R1's Fa0/0 interface. The configuration shows MD5 authentication configuration as interface subcommands on Fa0/0. Will those commands apply to the virtual link?	
The plan shows a sample OSPF configuration with five routers, Frame Relay connected, with one subnet, each using multipoint subinterfaces. A full mesh exists. The configuration shows network type point-to-multipoint nonbroadcast. What OSPF configuration is needed beyond the enabling of OSPF on the interfaces?	
Using the same scenario as the previous row but with network type nonbroadcast, and a partial mesh, answer the same questions.	

Create an Implementation Plan Table

To practice skills useful when creating your own OSPF implementation plan, list in Table 8-6 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 8-6 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Create a virtual link through transit area X.	
Configure MD5 authentication on a virtual link.	
Configure clear-text authentication on a virtual link.	
Configure null authentication on a virtual link.	
Configure the OSPF network type on a Frame Relay multipoint interface.	
Statically configure OSPF neighbors.	
Influence OSPF priority of the local router.	
Influence OSPF priority of a neighboring router.	
Add static Frame Relay mappings.	

Choosing Commands for a Verification Plan Table

To practice skills useful when creating your own OSPF verification plan, list in Table 8-7 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 8-7 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display the name and status of a virtual link.	
Display the RID of a neighbor on a virtual link.	
Display the OSPF authentication type and youngest key for MD5 authentication on a Virtual Link.	
Display the OSPF network type for an interface.	
Display a router's own OSPF priority.	
Display the mapping of neighbor IP address and Frame Relay DLCI.	
Display neighbors known on an interface.	

Author's note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review All the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 8-8 lists a reference of these key topics and the page numbers on which each is found.

Table 8-8 *Key Topics for Chapter 8*

Key Topic Element	Description	Page Number
List	Summary of options on the **area virtual-link** command	263
Table 8-2	Summary of OSPF authentication methods and commands for virtual links	266
List	Recommendations related to multipoint Frame Relay and either neighbor discovery or neighbor definition	271
List	Recommendations related to multipoint Frame Relay when a DR is elected	272
Paragraph	Recommendation with a partial mesh in multipoint Frame Relay regarding static Frame Relay mapping	273
Table 8-3	List of OSPF network types used for Frame Relay physical interfaces and multipoint subinterfaces	274

Complete the Tables and Lists from Memory

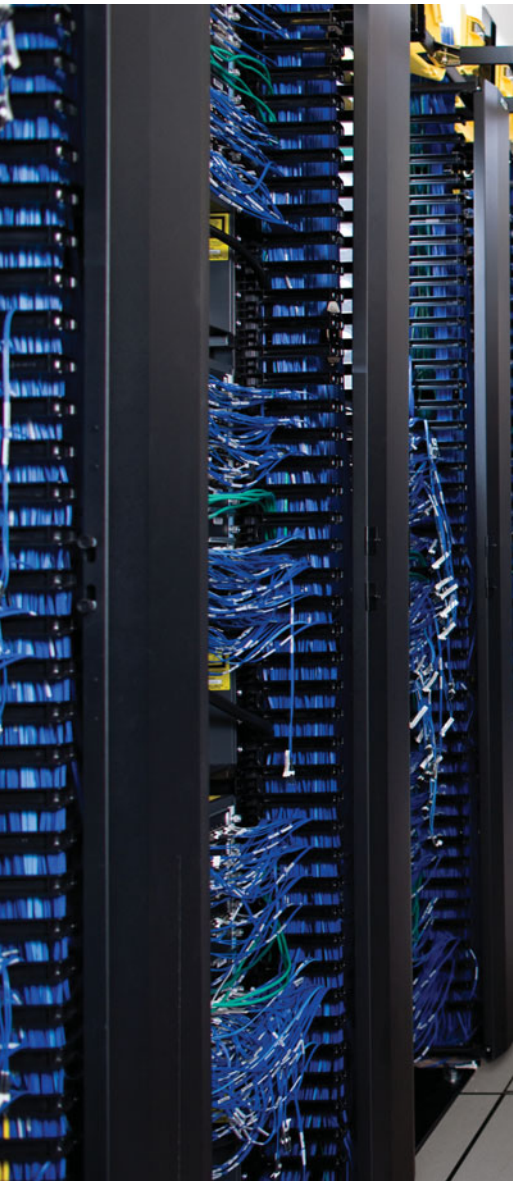
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

Virtual Link, OSPF network type, Full mesh, Partial mesh, Frame Relay mapping, Frame Relay Inverse ARP (InARP), Multipoint subinterface

This page intentionally left blank



This chapter covers the following subjects:

Route Redistribution Basics: This section discusses the reasons why designers might choose to use route redistribution, and how routing protocols redistribute routes from the IP routing table.

Redistribution in EIGRP: This section discusses the mechanics of how IOS redistributes routes from other sources into EIGRP.

Redistribution in OSPF: This section discusses the mechanics of how IOS redistributes routes from other sources into OSPF.

CHAPTER 9

Basic IGP Redistribution

This chapter begins Part 4 of this book: "Path Control." Path control refers to a general class of tools and protocols that Layer 3 devices use to learn, manipulate, and use IP routes. EIGRP and OSPF certainly fit into that category.

The three chapters in Part 4 examine path control features that go beyond the base function of learning IP routes. In particular, Chapter 9, "Basic IGP Redistribution," and Chapter 10, "Advanced IGP Redistribution," examine how routers can exchange routes between routing protocols through route redistribution. Chapter 9 begins by discussing the mechanics of what happens when the routes are redistributed. Chapter 10 then goes further to discuss how to also filter and summarize routes when redistributing–yet another path control function–and discusses some issues and solutions when multiple routers redistribute the same routes. Finally, Chapter 11, "Policy-Based Routing and IP Service Level Agreement," discusses a few smaller path control topics, including IP Service-Level Agreement (SLA) and Policy-Based Routing (PBR).

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 9-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 9-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Route Redistribution Basics	1–2
Redistribution into EIGRP	3–5
Redistribution into OSPF	6–8

1. Which of the following answers is the least likely reason for an engineer to choose to use route redistribution?

    a. To exchange routes between merged companies

b. To give separate control over routing to different parts of one company

c. To support multiple router vendors

d. To knit together an OSPF area if the area becomes discontiguous

2. For a router to successfully redistribute routes between OSPF and EIGRP, which of the following are true? (Choose two.)

a. The router must have one routing protocol configured, but configuration for both routing protocols is not necessary.

b. The router must have at least one working link connected to each routing domain.

c. The **redistribute** command must be configured under EIGRP to send the routes to OSPF.

d. The **redistribute** command should be configured under OSPF to take routes from EIGRP into OSPF.

3. Process EIGRP 1 is redistributing routes from process OSPF 2. Which two of the following methods may be used to set the metrics of the redistributed routes? (Choose 2)

a. Let the metrics default.

b. Set the metric components using the **redistribute** command's **metric** keyword.

c. Set the metric components using the **default-metric** router subcommand.

d. Set the integer (composite) metric using the **redistribute** command's **metric** keyword.

4. Examine the following excerpt from the **show ip eigrp topology 10.2.2.0/24** command on router R1. Which answer can be verified as definitely true based on this output?

```
External data:
  Originating router is 10.1.1.1
  AS number of route is 1
  External protocol is OSPF, external metric is 64
  Administrator tag is 0 (0x00000000)
```

a. R1 is the router that redistributed the route.

b. R1's metric to reach subnet 10.2.2.0/24 is 64.

c. The route was redistributed on a router that has a **router ospf 1** command configured.

d. R1 is redistributing a route to prefix 10.2.2.0/24 into OSPF.

5. Router R1 has a connected route for 10.1.1.0/24 off interface Fa0/0. Interface Fa0/0 has been enabled for OSPF due to a **router ospf 1** and **network 10.0.0.0 0.0.0.255 area 0** command. R1 also has EIGRP configured, with the **redistribute ospf 1 metric 1000 100 10 1 1500** command configured under EIGRP. Which one of the following is true?

a. R1 will not redistribute 10.1.1.0/24 into EIGRP, because R1 knows it as a connected route and not as an OSPF route.

b. For any OSPF routes redistributed into EIGRP, the metric components include a value equivalent to 1 Mbps of bandwidth.

c. For any OSPF routes redistributed into EIGRP, the metric components include a value equivalent to 100 microseconds of delay.

d. No subnets of network 10.0.0.0 will be redistributed due to the omission of the **subnets** parameter.

6. Process OSPF 1 is redistributing routes from process OSPF 2. Which of the following methods may be used to set the metrics of the redistributed routes? (Choose two.)

a. Let the metrics default.

b. Use each redistributed route's OSPF metric using the **redistribute** command's **metric transparent** keywords.

c. Set the metric using the **default-metric** router subcommand.

d. Redistribution is not allowed between two OSPF processes.

7. Examine the following excerpt from the **show ip ospf database asbr-summary** command on router R1 (RID 1.1.1.1). Which answer can be verified as definitely true based on this output?

```
LS Type: Summary Links (AS Boundary Router)
Link State ID: 9.9.9.9 (AS Boundary Router address)
Advertising Router: 3.3.3.3
LS Seq Number: 8000000D
Checksum: 0xE43A
Length: 28
Network Mask: /0
       TOS: 0  Metric: 100
```

a. The output describes the contents of a Type 5 LSA.

b. 3.3.3.3 identifies a router as being the router performing redistribution.

c. R1's metric for its best route to reach the router with RID 9.9.9.9 is 100.

d. The router with RID 3.3.3.3's metric for its best route to reach the router with RID 9.9.9.9 is 100.

8. Router R1 sits inside OSPF area 1. Router R2 redistributes an E1 route into OSPF for prefix 2.2.2.0/24, with external metric 20. Router R22 redistributes an E2 route for the same prefix/length, external metric 10. Under what conditions will R1 choose as its best route the route through R22?

a. R1 will always choose the route through R22.

b. As long as R1's best internal OSPF cost to reach R22 is less than 10.

c. As long as R1's best internal OSPF cost to reach R22 is less than 20.

d. R1 will never choose the route through R22 if the E1 route through R2 is available.

Foundation Topics

Route Redistribution Basics

Most internetworks use a single IGP to advertise and learn IP routes. However, in some cases, more than one routing protocol exists inside a single enterprise. Also, in some cases, the routes learned with an IGP must then be advertised with BGP, and vice versa. In such cases, engineers often need to take routing information learned by one routing protocol and advertise those routes into the other routing protocol–a function provided by the IOS route redistribution feature.

This section examines the basics of route redistribution.

The Need for Route Redistribution

The potential need for route redistribution exists when a route learned through one source of routing information, most typically one routing protocol, needs to be distributed into a second routing protocol domain. For example, two companies might merge, with one company using EIGRP and the other using OSPF. The engineers could choose to immediately migrate away from OSPF to instead use EIGRP exclusively, but that migration would take time and potentially cause outages. Route redistribution allows those engineers to connect a couple of routers to both routing domains, and exchange routes between the two routing domains, with a minimal amount of configuration and with little disruption to the existing networks.

Figure 9-1 shows just such a case, with R1 performing redistribution by using its knowledge of subnet 1 from the EIGRP domain and advertising a route for subnet 1 into the OSPF domain. Note that the opposite should also occur, with the OSPF domain's subnet 2 being redistributed into the EIGRP domain.

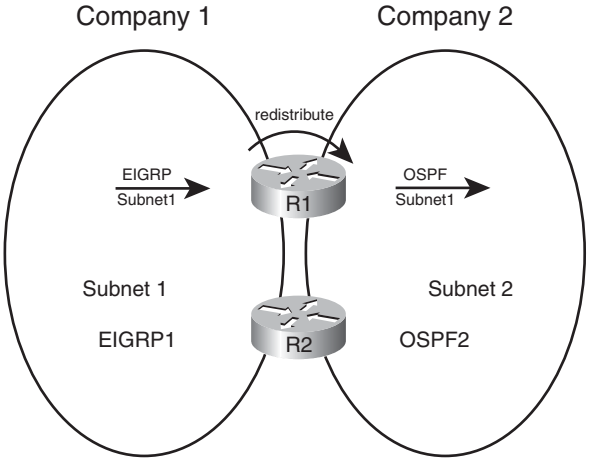


Figure 9-1 *Typical Use of Redistribution*

The main technical reason for needing redistribution is straightforward: An internetwork uses more than one routing protocol, and the routes need to be exchanged between those routing domains, at least temporarily. The business reasons vary widely but include the following:

- Mergers when different IGPs are used.
- Mergers when the same IGP is used.
- Momentum (The Enterprise has been using multiple routing protocols a long time.)
- Different company divisions under separate control for business or political reasons.
- Connections between partners.
- To allow multivendor interoperability (OSPF on non-Cisco, EIGRP on Cisco, for instance).
- Between IGPs and BGP when BGP is used between large segments of a multinational company.
- Layer 3 WAN (MPLS).

The list begins with two entries for mergers just to make the point that even if both merging companies use the same IGP, redistribution may still be useful. Even if both companies use EIGRP, they probably use a different AS number in their EIGRP configuration (with the **router eigrp** asn command). In such a case, to have all routers exchange routing information with EIGRP, all the former company's routers would need to migrate to use the same ASN as the first company. Such a migration may be simple, but it still requires disruptive configuration changes in a potentially large number of routers. Redistribution could be used until a migration could be completed.

Although useful as an interim solution, many permanent designs use redistribution as well. For example, it could be that a company has used different routing protocols (or different instances of the same routing protocol) in different divisions of a company. The network engineering groups may remain autonomous, and manage their own routing protocol domains, using redistribution to exchange routes at a few key connecting points between the divisions. Similarly, partner companies have separate engineering staffs, and want autonomy for managing routing, but also need to exchange routes for key subnets to allow the partnership's key applications to function. Figure 9-2 depicts both of these cases.

The last two cases in the previous list each relate to BGP in some way. First, some large corporations actually use BGP internal to the company's internetwork, redistributing routes from IGPs. Each large autonomous division of the company can design and configure their respective routing protocol instance, redistribute into BGP, and then redistribute out of BGP into other divisions. Also, when an Enterprise uses an MPLS VPN service, the MPLS provider's provider edge (PE) router typically redistributes customer routes with BGP inside the MPLS provider's MPLS network. Figure 9-3 shows samples of both these cases. In each of these cases, a given prefix/length (subnet/mask) is typically distributed into BGP at one location, advertised over a BGP domain, and redistributed back into some IGP.

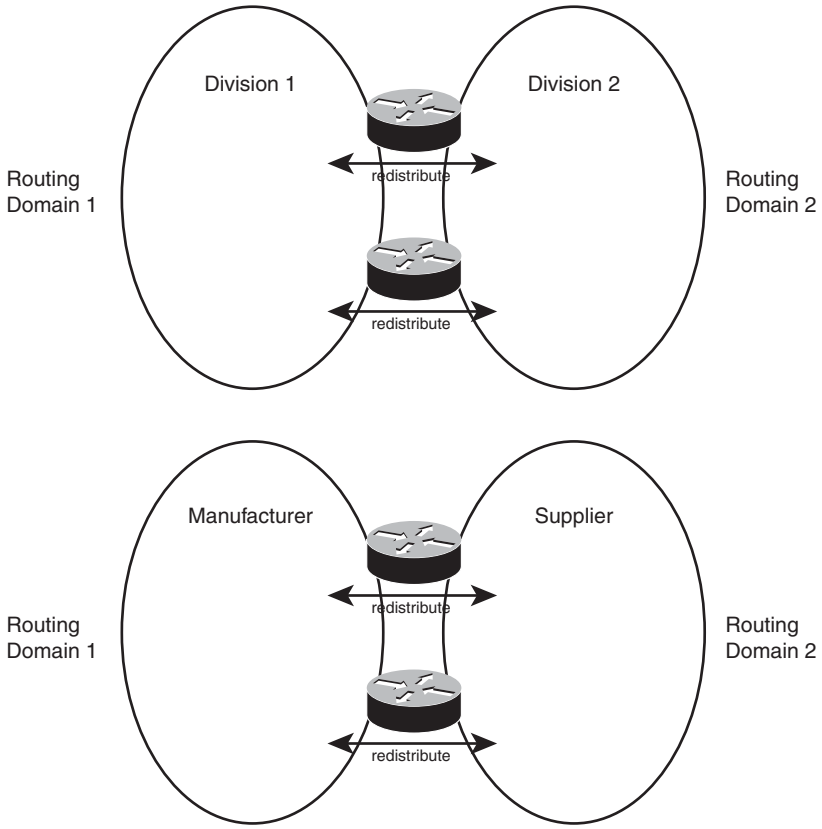


Figure 9-2 *Permanent Uses for Route Redistribution*

Note: Although Enterprises often use BGP to exchange routes with ISPs, redistribution is often not used. Chapter 13, "External BGP," discusses the issues and alternatives for how to advertise Enterprise routes into BGP.

Redistribution Concepts and Processes

Route redistribution requires at least one router to do the following:

Key Topic

Step 1. Use at least one working physical link with each routing domain.

Step 2. A working routing protocol configuration for each routing domain.

Step 3. Additional redistribution configuration for each routing protocol, specifically the **redistribute** command, which tells the routing protocol to take the routes learned by another source of routing information and to then advertise those routes.

The first two steps do not require any new knowledge or commands, but the third step represents the core of the redistribution logic and requires some additional background

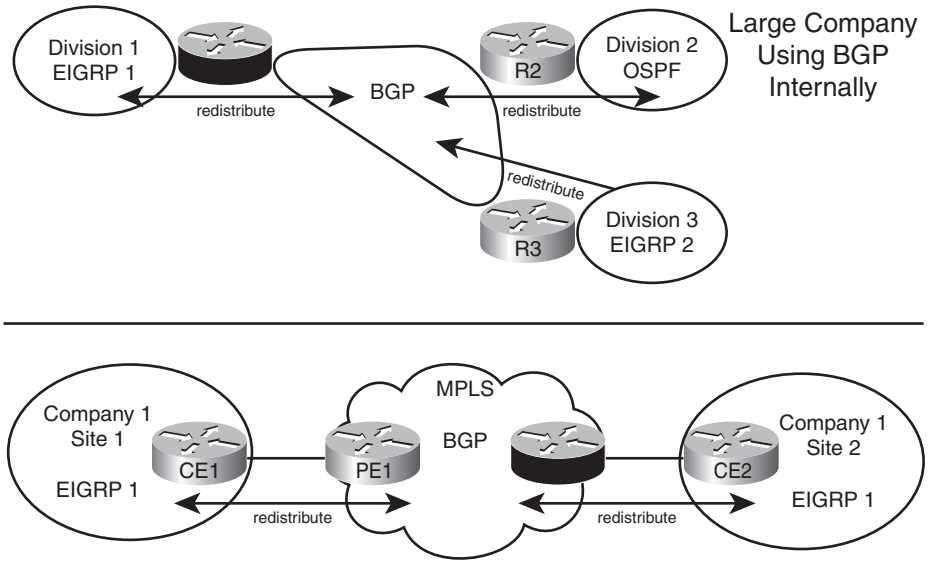


Figure 9-3 *Using Redistribution to Pass Routes Using BGP*

information. To appreciate the third step, Figure 9-4 shows an example router, RD1, that has completed Steps 1 and 2. RD1 uses EIGRP on the left, OSPF on the right, and has learned some routes with each routing protocol (Steps 1 and 2). However, no redistribution has been configured yet.

The goal for redistribution in this case is to have EIGRP advertise about subnets 11, 12, and 13, which exist inside the OSPF domain, and have OSPF advertise about subnets 1, 2, and 3, which exist inside the EIGRP domain. To do that, EIGRP must put topology information about subnets 11, 12 and 13 into its EIGRP topology table, and OSPF must put topology information about subnets 1, 2, and 3 into its topology table. However, OSPF's topology table has a lot of different information in it compared to EIGRP's topology table. OSPF has LSAs, and EIGRP does not. EIGRP lists the components of the composite metric and the neighbor's reported distance (RD)–but OSPF does not. In short, EIGRP and OSPF differ significantly for the contents of their topology tables.

Because the details of various routing protocols' topology tables differ, the redistribution process does not use the topology tables when redistributing routes. Instead, redistribution uses the one table that both routing protocols understand: the IP routing table. Specifically, the IOS **redistribute** command takes routes from the IP routing table and passes those routes to a routing protocol for redistribution. The **redistribute** command, configured inside a routing protocol configuration mode, redistributes routes into that routing protocol from some other source. Figure 9-5 spells it out with an example, which focuses on the internal logic of Router RD1 as shown in Figure 9-4.

Starting on the left of the figure, RD1's EIGRP 1 process configuration lists the **redistribute ospf 2** command. This command tells RD1 to look in the IP routing table, take all OSPF routes added to the IP routing table by the OSPF 2 process on RD1, and put those routes into EIGRP's topology table. Conversely, the **redistribute eigrp 1** command

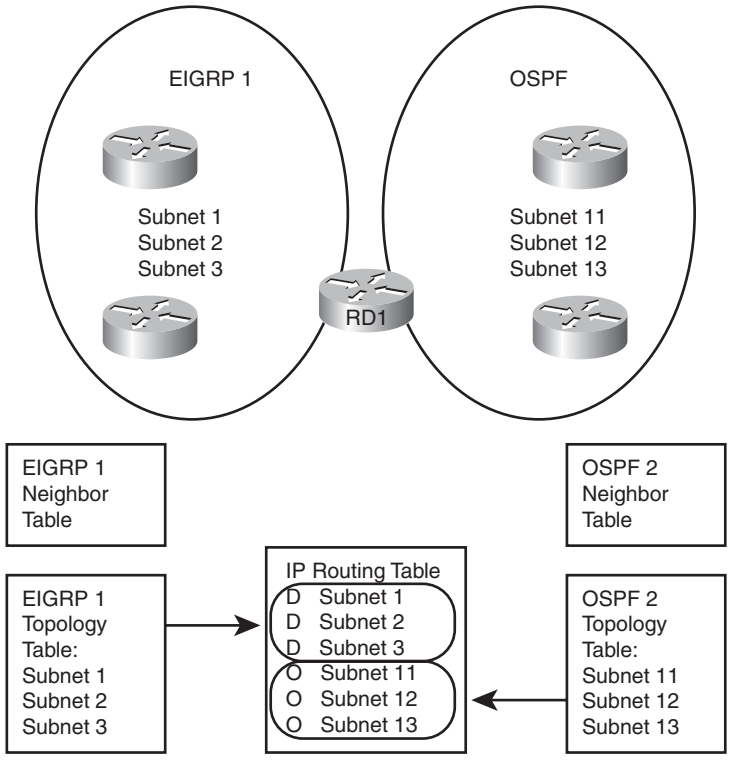


Figure 9-4 *Routing Protocol Tables on a Router Doing Redistribution*

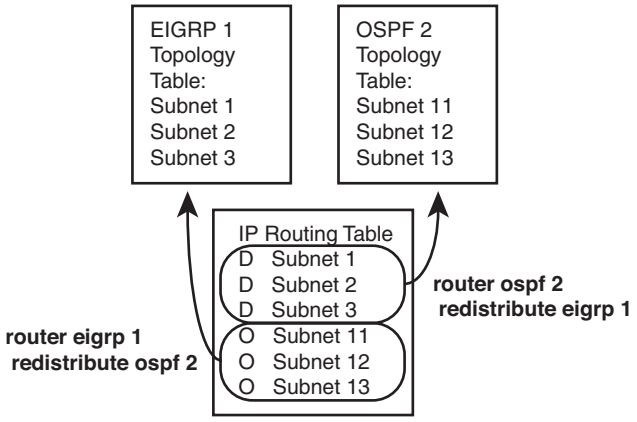


Figure 9-5 *Mutual Redistribution Between OSPF and EIGRP on Router RD1*

configured on the OSPF process tells RD1 to take IP routes from the IP routing table, if learned by EIGRP process 1, and add those routes to OSPF 2's topology table.

The process works as shown in Figure 9-5, but the figure leaves out some important details regarding the type of routes and the metrics used. For EIGRP, the EIGRP topology

table needs more than the integer metric value held by the IP routing table–it needs values for the components of the EIGRP composite metric. EIGRP can use default settings that define the metric components for all routes redistributed into EIGRP, or the engineer can set the metric components in a variety of ways, as covered in several locations later in this chapter.

Like EIGRP, OSPF treats the redistributed routes as external routes. OSPF creates an LSA to represent each redistributed subnet–normally a Type 5 LSA, but when redistributed into an NSSA area, the router instead creates a Type 7 LSA. In both cases, OSPF needs an integer metric to assign to the external route's LSA; the redistribution configuration should include the OSPF cost setting, which may or may not match the metric listed for the route in the redistributing router's IP routing table.

The last concept before moving on to the configuration options is that the **redistribute** command tells the router to take not only routes learned by the source routing protocol, but also connected routes on interfaces enabled with that routing protocol–including passive interfaces. Example 9-1 later in this chapter demonstrates this concept.

Redistribution into EIGRP

This section looks at the specifics of how EIGRP performs redistribution–that is, how EIGRP takes routes from other routing sources, such as OSPF, and advertises them into EIGRP. Before moving into the specifics, however, note that the redistribution as discussed in this chapter does not include any filtering or summarization. In real life, engineers often use both route filtering and route summarization at the redistribution point on a router. For the sake of making the underlying concepts clear, this chapter focuses on the mechanics of redistribution, without filtering, or summarization, or any other changes to the redistributed routes. Chapter 10 then looks at the fun options for manipulating routes at the redistribution point.

This section begins with a couple of short discussions of reference information. The first topic summarizes the parameters of the main configuration command, the EIGRP **redistribute** command, and its parameters. Next, the baseline configuration used in the upcoming samples is listed, including all EIGRP and OSPF configuration, but no redistribution configuration. With those details listed for reference, the rest of this section examines the configuration of redistribution into EIGRP.

EIGRP **redistribute** Command Reference

First, for reference, the following lines show the generic syntax of the **redistribute** command when used as a **router eigrp** subcommand. Note that the syntax differs slightly depending on the routing protocol into which routes will be redistributed. Following that, Table 9-2 lists the options on the command with a brief description.

```
redistribute protocol [process-id | as-number] [metric bw delay reliability load
mtu ] [match {internal | nssa-external | external 1 | external 2}]  [tag tag-
value] [route-map name]
```

Key Topic

Table 9-2 *Commonly Used OSPF Terms*

Option	Description
protocol	The source of routing information. Includes **RIP**, **OSPF**, **EIGRP**, **IS-IS**, **BGP**, connected, and static.
process-id, as-number	If redistributing a routing protocol that uses a process ID or ASN on the **router** global config command, use this parameter to refer to that process or ASN value.
metric	A keyword after which follows the four metric components (bandwidth, delay, reliability, link load), plus the MTU associated with the route.
match	If redistributing from OSPF, this keyword lets you match internal OSPF routes, external (by type), and NSSA external routes, essentially filtering which routes are redistributed.
tag	Assigns a unitless integer value to the routes redistributed by this command—tags which can be later matched by other routers using a route-map.
route-map	Apply the logic in the referenced route-map to filter routes, set metrics, and set route tags.

Baseline Configuration for EIGRP Redistribution Examples

The best method to see the results of redistribution is to use examples, so this section explains the sample internetwork used in the upcoming EIGRP redistribution examples. Figure 9-6 shows the sample internetwork. In this case, the EIGRP domain on the left uses subnets of class B network 172.30.0.0, and the OSPF domain on the right uses subnets of class B network 172.16.0.0. Note that all OSPF subnets reside in area 0 in this example internetwork, although that is not a requirement.

The internetwork uses a single router (RD1) to perform redistribution, just to avoid some interesting issues that occur when multiple routers redistribute the same routes. (Chapter 10 discusses these issues in some depth.) Example 9-1 shows the configuration on RD1, listing the IP addresses of the four active serial interfaces shown in Figure 9-6, plus the complete but basic EIGRP and OSPF configuration—but without any redistribution configured yet.

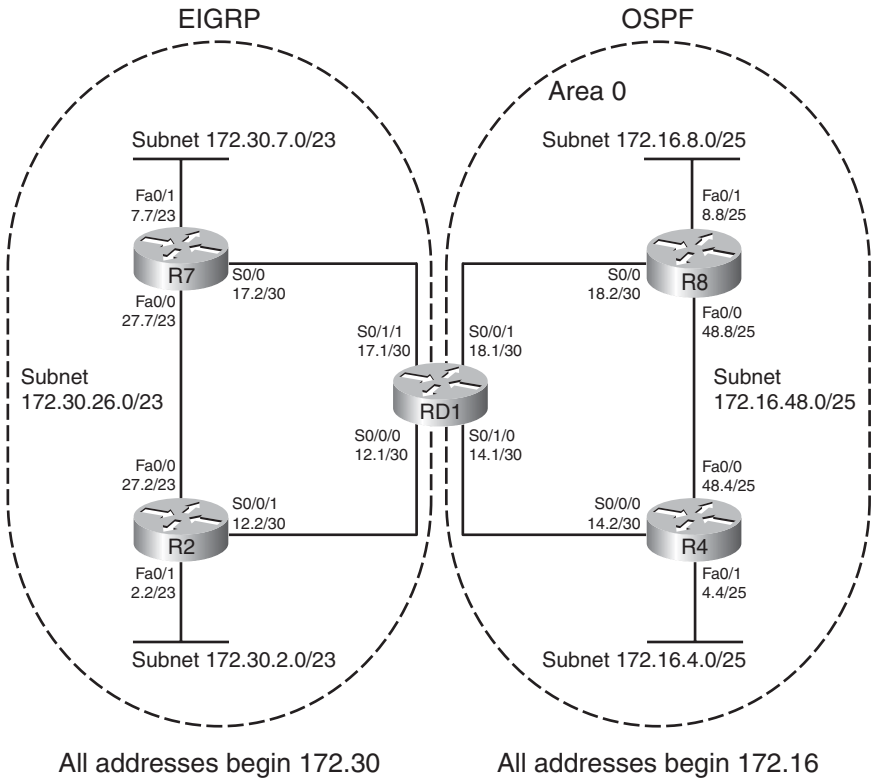


Figure 9-6 *Sample Internetwork Used for Redistribution Examples*

Example 9-1 *Configuration on Router RD1 Before Adding Redistribution Configuration*

```
interface Serial0/0/0
 ip address 172.30.12.1 255.255.255.252
 clock rate 1536000
!
interface Serial0/0/1
 ip address 172.16.18.1 255.255.255.252
 clock rate 1536000
!
interface Serial0/1/0
 ip address 172.16.14.1 255.255.255.252
 clock rate 1536000
!
interface Serial0/1/1
 ip address 172.30.17.1 255.255.255.252
 clock rate 1536000
!
router eigrp 1
 network 172.30.0.0
 no auto-summary
```

```
!
router ospf 2
 router-id 1.1.1.1
 network 172.16.0.0 0.0.255.255 area 0
```

Configuring EIGRP Redistribution with Default Metric Components

For the internetwork of Figure 9-6, a reasonable design goal would be to redistribute EIGRP routes into OSPF, and OSPF routes into EIGRP. This section examines the case of redistributing the routes into EIGRP from OSPF.

First, consider the EIGRP **redistribute** command. For those unfamiliar with the command, it may not be obvious of the direction of redistribution. A better command name might have been "take-routes-from," because the first parameter after the command tells IOS from where to get the routes.

For example, consider the configuration in Example 9-2, which was added to RD1's existing configuration in Example 9-1. The configuration uses only required parameters, namely a reference to the source from which routes should be redistributed. Because the configuration places this command in EIGRP configuration mode, the command tells IOS to redistribute the routes into EIGRP 1, from OSPF 2 in this case.

Example 9-2 *Minimal Configuration for Redistribution from OSPF into EIGRP*

```
RD1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RD1(config)#router eigrp 1
RD1(config-router)#redistribute ospf 2
RD1(config-router)#^Z
```

IOS does accept the configuration; unfortunately, IOS does not actually redistribute routes from OSPF into EIGRP in this case. EIGRP does not have a default setting for the metric components to use when redistributing into EIGRP from OSPF. To confirm these results, examine the output in Example 9-3, which lists **show** command output from RD1 when configured as shown in the previous example. Note that that RD1's EIGRP topology table lists only routes for class B network 172.30.0.0, which all sit inside the EIGRP domain; none of the routes from class B network 172.16.0.0, which exist inside the OSPF domain, have been added to RD1's EIGRP topology table.

Example 9-3 *Redistribution Did Not Work on RD1*

```
RD1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.30.17.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.30.17.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/1/1
P 172.30.26.0/23, 2 successors, FD is 2172416
        via 172.30.12.2 (2172416/28160), Serial0/0/0
```

```
         via 172.30.17.2 (2172416/28160), Serial0/1/1
P 172.30.2.0/23, 1 successors, FD is 2172416
         via 172.30.12.2 (2172416/28160), Serial0/0/0
         via 172.30.17.2 (2174976/30720), Serial0/1/1
P 172.30.6.0/23, 1 successors, FD is 2172416
         via 172.30.17.2 (2172416/28160), Serial0/1/1
         via 172.30.12.2 (2174976/30720), Serial0/0/0
P 172.30.12.0/30, 1 successors, FD is 2169856
        via Connected, Serial0/0/0
```

To complete the configuration of redistribution into EIGRP, Router RD1 needs to set the metric values. EIGRP can set the metrics for redistributed routes in three ways, as summarized in Table 9-3.

Table 9-3 *Methods of Setting EIGRP Metrics When Redistributing into EIGRP*

Function	Command
Setting the default for all **redistribute** commands	The **default-metric** bw delay reliability load mtu EIGRP subcommand
Setting the component metrics applied to all routes redistributed by a single **redistribute** command	The **metric** bw delay reliability load mtu parameters on the **redistribute** command
Setting different component metrics to different routes from a single route source	Use the **route-map** parameter on the **redistribute** command, matching routes and setting metric components.

Note: EIGRP does have a default metric when redistributing from another EIGRP process, in which case it takes the metric from the source of the routing information. In all other cases, the metric must be set using one of the methods in Table 9-3.

If the metrics do not matter to the design, which is likely when only a single redistribution point exists as in Figure 9-6, either of the first two methods listed in Table 9-3 is reasonable. The first method, using the **default-metric** command in EIGRP configuration mode, sets the metric for all routes redistributed into EIGRP, unless set by one of the other methods. Alternatively, the second method, which uses additional parameters on the **redistribute** command, sets the metric for all routes redistributed because of that one **redistribute** command. Finally, if the **redistribute** command also refers to a route map, the route map can use the set metric command to set the metric components for routes matched by the route map clause, overriding the metric settings in the **default-metric** command or with the **metric** keyword on the **redistribute** command.

Example 9-4 shows the addition of the **default-metric 1000 33 255 1 1500** command to RD1's configuration. This command sets the bandwidth to 1000 (Kbps), the delay to 33 (tens-of-microseconds, or 330 microseconds), reliability to 255 (a value between 1–255,

255 is best), load to 1 (a value between 1–255, 1 is best), and MTU of 1500. Note that even though EIGRP ignores the last three parameters by default when calculating integer metrics, you still must configure these settings for the commands to be accepted.

Example 9-4 *Redistributed Routes in RD1*

```
RD1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RD1(config)#router eigrp 1
RD1(config-router)#default-metric 1000 33 255 1 1500
RD1(config-router)#^Z
```

Because this example uses a single **redistribute** command for the EIGRP 1 process, you could have used the **redistribute ospf 2 metric 1000 33 255 1 1500** command and ignored the **default-metric** command to achieve the same goal.

Verifying EIGRP Redistribution

As shown earlier in Figure 9-5, redistribution takes routes from the routing table and places the correct information for those subnets into the redistributing router's topology table. The redistributing router then advertises the routes from its topology table as it would for other routes. To verify that redistribution works, Example 9-5 shows the proof that RD1 indeed created entries in its EIGRP topology table for the five subnets in the OSPF domain.

Example 9-5 *Verifying RD1 Added EIGRP Topology Data for Five OSPF Subnets*

```
RD1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.30.17.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

! Note - all lines for class B network 172.30.0.0 have been omitted for brevity
P 172.16.48.0/25, 1 successors, FD is 2568448
        via Redistributed (2568448/0)
P 172.16.18.0/30, 1 successors, FD is 2568448
        via Redistributed (2568448/0)
P 172.16.14.0/30, 1 successors, FD is 2568448
        via Redistributed (2568448/0)
P 172.16.8.0/25, 1 successors, FD is 2568448
        via Redistributed (2568448/0)
P 172.16.4.0/25, 1 successors, FD is 2568448
        via Redistributed (2568448/0)
RD1#show ip eigrp topology 172.16.48.0/25
IP-EIGRP (AS 1): Topology entry for 172.16.48.0/25
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2568448
```

```
Routing Descriptor Blocks:
172.16.18.2, from Redistributed, Send flag is 0x0
    Composite metric is (2568448/0), Route is External
    Vector metric:
      Minimum bandwidth is 1000 Kbit
      Total delay is 330 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 0
    External data:
      Originating router is 172.30.17.1 (this system)
      AS number of route is 2
      External protocol is OSPF, external metric is 65
      Administrator tag is 0 (0x00000000)
```

The **show** command output lists several interesting facts, including

- On Router RD1, which performed the redistribution, the EIGRP topology table lists the outgoing interface as "via redistributed."
- All the redistributed routes have the same feasible distance (FD) calculation (2568448), because all use the same component metrics per the configured **default-metric** command.
- RD1's two connected subnets in the OSPF 2 domain–subnets 172.16.14.0/30 and 172.16.18.0/30–were also redistributed, even though these routes are connected routes in RD1's routing table.
- The output of the **show ip eigrp topology 172.16.48.0/25** command confirms the component metrics match the values configured on the **default-metric** command.
- The bottom of the output of the **show ip eigrp topology 172.16.48.0/25** command lists information about the external source of the route, including the routing source (OSPF) and that source's metric for the route (65). It also lists the phrase "(this system)," meaning that the router on which the command was issued (RD1 in this case) redistributed the route.

The third item in the list–the fact that RD1 redistributed some connected routes–bears further consideration. The **redistribute ospf 2** command tells EIGRP to redistribute routes learned by the OSPF 2 process. However, it also tells the router to redistribute connected routes for interfaces on which process OSPF 2 has been enabled. Back in Example 9-1, the configuration on RD1 lists a **network 172.16.0.0 0.0.255.255 area 0** command, enabling OSPF 2 on RD1's S0/0/1 and S0/1/0 interfaces. As such, the redistribution process also redistributed those routes.

Stated more generally, when the **redistribute** command refers to another IGP as the routing source, it tells the router to redistribute the following:

- All routes in the routing table learned by that routing protocol
- All connected routes of interfaces on which that routing protocol is enabled

Although Example 9-5 shows the evidence that Router RD1 added the topology data to its EIGRP topology database, it did not show any routes. Example 9-6 shows the IP routing tables on both RD1 and Router R2, a router internal to the EIGRP domain. R2's routes forward the packets toward the redistributing router, which in turn has routes from the OSPF domain with which to forward the packet to the destination subnet.

Example 9-6 *Verification of IP Routes on RD1 and R2*

```
! First, on RD1
RD1#show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 5 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
  Redistributing via eigrp 1


O        172.16.48.0/25 [110/65] via 172.16.18.2, 00:36:25, Serial0/0/1
                        [110/65] via 172.16.14.2, 00:36:25, Serial0/1/0
C        172.16.18.0/30 is directly connected, Serial0/0/1
C        172.16.14.0/30 is directly connected, Serial0/1/0
O        172.16.8.0/25 [110/65] via 172.16.18.2, 00:36:25, Serial0/0/1
O        172.16.4.0/25 [110/65] via 172.16.14.2, 00:36:25, Serial0/1/0
```

```
! Next, on Router R2
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
D EX     172.16.48.0/25 [170/3080448] via 172.30.12.1, 00:25:15, Serial0/0/1
D EX     172.16.18.0/30 [170/3080448] via 172.30.12.1, 00:25:15, Serial0/0/1
D EX     172.16.14.0/30 [170/3080448] via 172.30.12.1, 00:25:15, Serial0/0/1
D EX     172.16.8.0/25 [170/3080448] via 172.30.12.1, 00:25:15, Serial0/0/1
D EX     172.16.4.0/25 [170/3080448] via 172.30.12.1, 00:25:15, Serial0/0/1
     172.30.0.0/16 is variably subnetted, 5 subnets, 2 masks
D        172.30.17.0/30 [90/2172416] via 172.30.27.7, 00:25:15, FastEthernet0/0
```

```
C        172.30.26.0/23 is directly connected, FastEthernet0/0
C        172.30.2.0/23 is directly connected, FastEthernet0/1
D        172.30.6.0/23 [90/30720] via 172.30.27.7, 00:25:15, FastEthernet0/0
C       172.30.12.0/30 is directly connected, Serial0/0/1
```

Beginning with the output for R2, in the second half of the example, R2 knows routes for all five subnets in class B network 172.16.0.0, listing all as external EIGRP routes. The routes all use R2's link connected to RD1. Also, note that the administrative distance (AD) is set to 170, rather than the usual 90 for EIGRP routes. EIGRP defaults to use AD 90 for internal routes and AD 170 for external routes. Chapter 10 shows cases in which this default helps prevent routing loops when multiple redistribution points exist.

RD1 has routes for all routes in the OSPF domain as well, but as either connected or OSPF-learned routes.

Redistribution into OSPF

As you might expect, OSPF redistribution has several similarities and differences as compared to redistribution into EIGRP. Unlike EIGRP, OSPF does have useful default metrics for redistributed routes, but OSPF does use the same general methods to configure metrics for redistributed routes. Like EIGRP, OSPF flags redistributed routes as being external. Unlike EIGRP, OSPF creates LSAs to represent each external route, and OSPF must then apply some much different logic than EIGRP to calculate the best route to each external subnet.

This section examines the OSPF redistribution process and configuration. It also discusses background on three OSPF LSA Types—Types 4, 5, and 7—all created to help OSPF distribute information so that routers can calculate the best route to each external subnet.

OSPF **redistribute** Command Reference

First, for reference, the following lines show the generic syntax of the **redistribute** command when used as a **router ospf** subcommand. Note that the syntax differs slightly depending on the routing protocol into which routes will be redistributed. Following that, Table 9-4 lists the options on the command with a brief description:

```
redistribute protocol [process-id | as-number] [metric metric-value] [metric-type
type-value] [match {internal | external 1 | external 2 | nssa-external}] [tag
tag-value] [route-map map-tag] [subnets]
```

Table 9-4 *Parameters on the OSPF* **redistribute** *Command*

Option	Description
protocol	The source of routing information. Includes **RIP**, **OSPF**, **EIGRP**, **IS-IS**, **BGP**, **connected**, and **static**.
process-id, as-number	If redistributing a routing protocol that uses a process-id or ASN on the **router** global config command, use this parameter to refer to that process or ASN value.

Table 9-4 *Parameters on the OSPF* **redistribute** *Command*

Option	Description
metric	Defines the cost metric assigned to routes redistributed by this command, unless overridden by a referenced route map.
metric-type {**1** | **2**}	Defines the external metric type for the routes redistributed by this command: **1** (E1 routes) or **2** (E2 routes).
match	If redistributing from another OSPF process, this keyword lets you match internal OSPF routes, external (by type), and NSSA external routes, essentially filtering which routes are redistributed.
tag	Assigns a unitless integer value to the routes redistributed by this command—a tag that can be later matched by other routers using a route-map.
route-map	Apply the logic in the referenced route-map to filter routes, set metrics, and set route tags.
subnets	Redistribute subnets of classful networks. Without this parameter, only routes for classful networks are redistributed. (This behavior is unique to the OSPF **redistribute** command.)

Configuring OSPF Redistribution with Minimal Parameters

The **redistribute** subcommand under **router ospf** has many optional settings. To better appreciate some of these settings, this section first examines the results when using all defaults, using as few parameters as possible. Following the discussion of the behavior with defaults, the next examples add the parameters that complete the redistribution configuration.

Redistribution into OSPF uses the following defaults:

- When taking from BGP, use a default metric of 1.
- When taking from another OSPF process, take the source route's metric.
- When taking from all other sources, use a default metric of 20.
- Create a Type 5 LSA for each redistributed route (external) if not inside an NSSA area; create a Type 7 LSA if inside an NSSA area.
- Use external metric type 2.
- Redistribute only routes of classful (class A, B, and C) networks, and not routes for subnets.

To demonstrate OSPF redistribution, this section uses an example that uses the same internetwork shown in Figure 9-6, including the baseline configuration shown in Example 9-1, and the EIGRP redistribution configuration shown in Examples 9-2 and 9-4. Essentially, the upcoming OSPF examples begin with Router RD1 including all the configurations seen in all the earlier examples in this chapter. According to those examples, OSPF has been correctly configured on the routers on the right side of Figure 9-6, EIGRP has been configured on the left, and the configuration of redistribution of OSPF routes into EIGRP has been completed. However, no redistribution into OSPF has been configured yet.

For perspective, before showing the redistribution into OSPF, Example 9-7 reviews the OSPF configuration before adding the redistribution configuration, along with **show** commands listing RD1's IP routing table entries and its OSPF LSDB.

Example 9-7 *Router RD1 Routing Protocol Configuration, Before Redistribution into OSPF*

```
RD1#show run
! lines omitted for brevity
router eigrp 1
 redistribute ospf 2
 network 172.30.0.0
 default-metric 1000 33 255 1 1500
 no auto-summary
!
router ospf 2
 router-id 1.1.1.1
 log-adjacency-changes
 network 172.16.0.0 0.0.255.255 area 0

RD1#show ip route 172.30.0.0
Routing entry for 172.30.0.0/16, 5 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
  Redistributing via eigrp 1

C        172.30.17.0/30 is directly connected, Serial0/1/1
D        172.30.26.0/23 [90/2172416] via 172.30.17.2, 01:08:50, Serial0/1/1
                        [90/2172416] via 172.30.12.2, 01:08:50, Serial0/0/0
D        172.30.2.0/23 [90/2172416] via 172.30.12.2, 01:08:50, Serial0/0/0
D        172.30.6.0/23 [90/2172416] via 172.30.17.2, 01:08:50, Serial0/1/1
C        172.30.12.0/30 is directly connected, Serial0/0/0
RD1#show ip ospf database

             OSPF Router with ID (1.1.1.1) (Process ID 2)

                 Router Link States (Area 0)
```

```
Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1425        0x80000007 0x007622 4
4.4.4.4         4.4.4.4         1442        0x8000000D 0x00B1E9 4
8.8.8.8         8.8.8.8         1466        0x80000006 0x00640E 4

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
172.16.48.4     4.4.4.4         1442        0x80000004 0x007E07
```

```
! The following occurs on OSPF internal router R4

R4#show ip route 172.30.0.0
% Network not in table
```

The output in Example 9-7 shows several important points relative to the upcoming redistribution configuration. First, by design, the EIGRP domain contains subnets of network 172.30.0.0; router RD1 knows routes for five subnets in this range. RD1 has four LSAs: three Type 1 Router LSAs (one each for routers RD1, R4, and R8), plus one Type 2 network LSAs (because only one subnet, 172.16.48.0/25, has elected a DR). Because the design for this internetwork puts all OSPF routers in area 0, no Type 3 summary LSAs exist in RD1's LSDB. Also, because no routers have redistributed external routes into OSPF yet, no Type 5 external nor Type 7 NSSA external routes are listed, either.

By adding the **redistribute eigrp 1** command in OSPF configuration mode, OSPF tries to redistribute routes from EIGRP–but with no success. The reason is that by omitting the **subnets** parameter, OSPF will only redistribute routes for entire classful subnets, and only if such a route is listed in the IP routing table. Example 9-8 shows the results.

Example 9-8 *Redistributing into OSPF from EIGRP 1, all Default Settings*

```
RD1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RD1(config)#router ospf 2
RD1(config-router)#redistribute eigrp 1
% Only classful networks will be redistributed
RD1(config-router)#^Z
RD1#
RD1#show ip ospf database

            OSPF Router with ID (1.1.1.1) (Process ID 2)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         6           0x80000008 0x007A1B 4
```

```
4.4.4.4          4.4.4.4          1782         0x8000000D 0x00B1E9 4
8.8.8.8          8.8.8.8          1806         0x80000006 0x00640E 4

                 Net Link States (Area 0)

Link ID          ADV Router       Age          Seq#       Checksum
172.16.48.4      4.4.4.4          1782        0x80000004 0x007E07
```

IOS even mentions that only classful routes will be redistributed. As seen in Example 9-7, no route exists for the exact class B network prefix of 172.30.0.0/16, and by default, OSPF does not redistribute any subnets inside that range, as noted in the informational message in Example 9-8. So, the OSPF database on Router RD1 remains unchanged.

By changing the configuration to use the **redistribute eigrp 1 subnets** command, OSPF indeed redistributes the routes, as shown in Example 9-9.

Example 9-9 *Redistributing from EIGRP into OSPF, with Subnets*

```
RD1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RD1(config)#router ospf 2
RD1(config-router)#redistribute eigrp 1 subnets
RD1(config-router)#^Z
RD1#
May 12 12:49:48.735: %SYS-5-CONFIG_I: Configured from console by console
RD1#show ip ospf database
! omitting the Type 1 and 2 LSA output for brevity

                 Type-5 AS External Link States

Link ID          ADV Router       Age          Seq#       Checksum Tag
172.30.2.0       1.1.1.1          3            0x80000001 0x008050 0
172.30.6.0       1.1.1.1          3            0x80000001 0x005478 0
172.30.12.0      1.1.1.1          3            0x80000001 0x0005C3 0
172.30.17.0      1.1.1.1          3            0x80000001 0x00CDF5 0
172.30.26.0      1.1.1.1          3            0x80000001 0x007741 0
```

```
! The following occurs on router R4
R4#show ip route 172.30.0.0
Routing entry for 172.30.0.0/16, 5 known subnets
  Variably subnetted with 2 masks

O E2     172.30.17.0/30 [110/20] via 172.16.14.1, 00:01:10, Serial0/0/0
O E2     172.30.26.0/23 [110/20] via 172.16.14.1, 00:01:11, Serial0/0/0
O E2     172.30.2.0/23 [110/20] via 172.16.14.1, 00:01:11, Serial0/0/0
O E2     172.30.6.0/23 [110/20] via 172.16.14.1, 00:01:11, Serial0/0/0
O E2    172.30.12.0/30 [110/20] via 172.16.14.1, 00:01:11, Serial0/0/0
```

After adding the **subnets** option, router RD1 redistributes the five routes from the EIGRP domain. Of particular interest:

- If you look back to Example 9-7's **show ip route** command output from Router RD1, you see three EIGRP-learned routes, plus two connected routes, inside the EIGRP domain. Example 9-9's two **show** commands in Example 9-9 confirm that OSPF redistributes the three EIGRP-learned routes, plus the two connected subnets on which EIGRP is enabled (172.30.12.0/30 and 172.30.17.0/30).
- The **show ip ospf database** command in Example 9-9 lists R1 (RID 1.1.1.1) as the advertising router of the five new Type 5 LSAs, because RD1 (with RID 1.1.1.1) created each Type 5 LSA.
- Per OSPF internal router R4's **show ip route 172.30.0.0** command at the end of Example 9-9, the external metric type is indeed E2, meaning external type 2.
- Per that same command on router R4, the metric for each route is 20. The reasoning is that the default metric is 20 when redistributing from EIGRP into OSPF, and with an E2 route, internal OSPF costs are not added to the cost of the route.

That last point regarding the external route type requires a little more discussion. OSPF defines external routes as either an external type 1 (E1) or external type 2 (E2) route. By default, the OSPF **redistribute** command creates Type 2 routes, noting this external route type in the Type 5 LSA. The difference between the two lies in how OSPF calculates the metrics for E1 and E2 routes.

The next section completes the discussion of how OSPF can set the metrics when redistributing routes–or more specifically, the metric as listed in the Type 5 LSA created for that subnet. Following that, the text takes a detailed look at how OSPF calculates the best route for E2 routes. Later, a different section titled "Redistributing into OSPF as E1 Routes" discusses the same subject, but for E1 routes.

Setting OSPF Metrics on Redistributed Routes

As mentioned earlier, no matter the source of the redistributed route, OSPF has a default metric to use. However, OSPF can set the metrics for redistributed routes using the same options used for EIGRP. Table 9-5 summarizes the defaults and metric setting options for redistribution into OSPF.

Key Topic

Table 9-5 *Summary of Metric Values When Redistributing into OSPF*

Function	Command or Metric Values
Default if no metric configuration exists	Cost 1 for routes learned from BGP. If redistributed from another OSPF process, use the source route's OSPF cost. Cost 20 for all other route sources.
Setting the default for all **redistribute** commands	The **default-metric** cost OSPF subcommand.

Table 9-5 *Summary of Metric Values When Redistributing into OSPF*

Function	Command or Metric Values
Setting the metric for one route source	The **metric** cost parameters on the **redistribute** command.
Setting different metrics for routes learned from a single source	Use the **route-map** parameter on the **redistribute** command.

LSAs and Metrics for External Type 2 Routes

To appreciate how OSPF calculates the possible routes for each E2 route, you need to take a moment to think about the Type 5 LSA in more detail. First, by definition, the router that performs the redistribution into OSPF becomes an autonomous system border router (ASBR) because it injects external routes into OSPF. For each such route, that ASBR creates a Type 5 LSA for that subnet. The Type 5 LSA includes the following fields:

- **LSID:** The subnet number
- **Mask:** The subnet mask
- **Advertising router:** The RID of the ASBR injecting the route
- **Metric:** The metric as set by the ASBR
- **External Metric Type:** The external metric type, either 1 or 2

When created, the ASBR floods the Type 5 LSA throughout the area. Then, if any ABRs exist, the ABRs flood the Type 5 LSAs into any normal (nonstubby) areas. (Note that ABRs must not forward Type 5 LSAs into any type of stubby area, instead relying on default routes.) Figure 9-7 shows a sample flooding of the Type 5 LSA for EIGRP subnet 172.30.27.0/23 as an E2 route.

When flooded, OSPF has little work to do to calculate the metric for an E2 route, because by definition, the E2 route's metric is simply the metric listed in the Type 5 LSA. In other words, the OSPF routers do not add any internal OSPF cost to the metric for an E2 route.

Because routers ignore internal cost when calculating E2 external route metrics, whenever an alternative route can be calculated, the metrics tie. For example, in Figure 9-7, Router R4 has two possible physical routes to ASBR RD1–one directly to RD1, and one through R8. The cost for both routes to external subnet 172.30.26.0/23 will be 20, because that is the cost RD1 assigned to the route (actually, the Type 5 LSA) when redistributing the route.

To avoid loops, OSPF routers use a tiebreaker system to allow a router to choose a best external route. The logic differs slightly depending on whether the router in question resides in the same area as the ASBR (intra-area), or in a different area (interarea), as discussed in under the next two headings.

Determining the Next-Hop for Type 2 External Routes–Intra-area

When a router finds multiple routes for the same E2 destination subnet, it chooses the best route based on the lowest cost to reach any ASBR(s) that advertised the lowest E2 metric. For example, if five ASBRs all advertised the same subnet as an E2 route, and two ASBRs

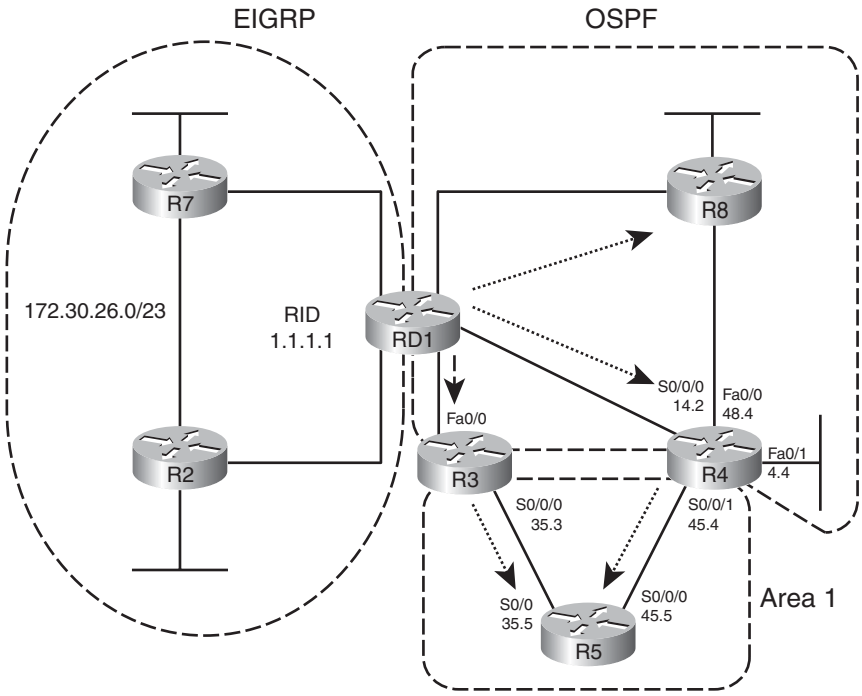


Figure 9-7 *Flooding of Type 5 LSAs*

advertised a metric of 10, and the other three advertised a metric of 20, either of the first two ASBRs could be used. Then, the router calculates its lowest cost route to reach the ASBR and uses the next-hop IP address and outgoing interface listed in that route.

The following list spells out the mechanics of the calculation used to break the tie when multiple equal-cost E2 routes exist for a particular subnet:

Key Topic

Step 1. Find the advertising ASBR(s) as listed in the Type 5 LSA(s) for Type 5 LSAs.

Step 2. Calculate the lowest cost route to reach any of the ASBR(s) based on the intra-area LSDB topology.

Step 3. Use the outgoing interface and next hop based on the best route to reach the ASBR (as chosen at Step 2).

Step 4. The route's metric is unchanged–it is still simply the value listed in the Type 5 LSA.

For example, use Router R4 in Figure 9-7 as an example and the E2 route for 172.30.26.0/23. Before using these four steps, R4 calculated two possible routes for 172.16.26.0/23: an E2 route directly to RD1, and another route through R8. Both routes use metric 20 in this case, so the routes tie. Because of the tie, R4 proceeds with these steps as follows:

Step 1. R4 looks in the Type 5 LSA, and sees RID 1.1.1.1 (RD1) is the advertising ASBR.

Step 2. R4 then looks at its area 0 LSDB entries, including the Type 1 LSA for RID 1.1.1.1, and calculates all possible area 0 routes to reach 1.1.1.1.

Step 3. R4's best route to reach RID 1.1.1.1 happens to be through its S0/0/0 interface, to next-hop RD1 (172.16.14.1), so R4's route to 172.16.26.0/23 uses these details.

Step 4. The route lists metric 20, as listed in the Type 5 LSA.

Figure 9-8 shows the interface costs Router R4 will use, based on its LSDB, to calculate the cost for two possible routes to reach ASBR RD1. Again using subnet 172.30.26.0/23 as an example, RD1 first looks at the Type 5 external LSA and sees RID 1.1.1.1 as the advertising ASBR. R4 then calculates the costs based on its intra-area LSDB–but we can perform the equivalent by adding the interface costs seen in Figure 9-8. Example 9-10 lists the external Type 5 LSAs, highlighting subnet 172.30.26.0/23, and the interface costs on both R4 and R8 as seen in the figure.

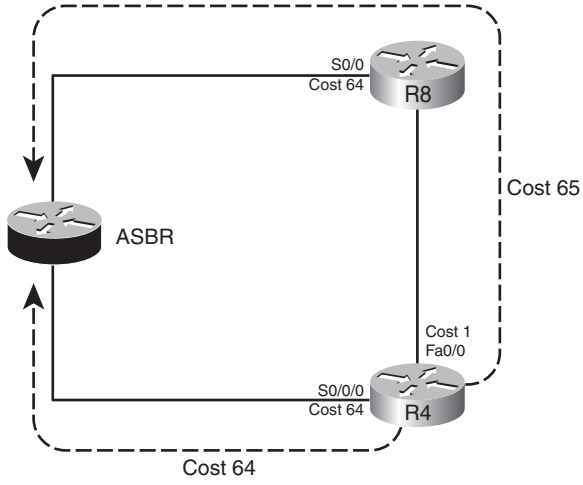


Figure 9-8 *R4's Cost to Reach ASBR RD1*

Example 9-10 *Verifying OSPF External Routes–Intra-area*

```
R4#show ip ospf database ¦ begin Ext
                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
172.30.2.0      1.1.1.1         189         0x80000002 0x007E51 0
172.30.6.0      1.1.1.1         189         0x80000002 0x005279 0
172.30.12.0     1.1.1.1         189         0x80000002 0x0003C4 0
172.30.17.0     1.1.1.1         189         0x80000002 0x00CBF6 0
172.30.26.0     1.1.1.1         189         0x80000002 0x007542 0

R4#show ip ospf database external 172.30.26.0
```

```
            OSPF Router with ID (4.4.4.4) (Process ID 4)

                Type-5 AS External Link States

  Routing Bit Set on this LSA
  LS age: 175
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 172.30.26.0 (External Network Number )
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000001
  Checksum: 0x7741
  Length: 36
  Network Mask: /23
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0

R4#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Se0/0/0      4     0               172.16.14.2/30     64    P2P   1/1
Fa0/1        4     0               172.16.4.4/25      1     DR    0/0
Fa0/0        4     0               172.16.48.4/25     1     DR    1/1
Se0/0/1      4     1               172.16.45.4/25     64    P2P   1/1
! Next output occurs on R8

R8#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Fa0/1        8     0               172.16.8.8/25      1     DR    0/0
Se0/0        8     0               172.16.18.2/30     64    P2P   1/1
Fa0/0        8     0               172.16.48.8/25     1     BDR   1/1
```

Determining the Next-hop for Type 2 External Routes–Interarea

When a router exists in a different area than the ASBR, the issues remain the same, but the tie-breaker calculation of choosing the least cost route to reach the ASBR changes. If a router finds multiple routes to reach a single E2 subnet, some or all may tie based on metric, because the metric is based solely on the external cost as defined by the ASBR. (If multiple ASBRs redistribute routes for the same prefix, each ASBR can assign a different metric.) A router then chooses the best route based on the least-cost route to reach an ASBR that has advertised the lowest E2 cost for the subnet.

When the ASBR is in a different area, the calculation of the cost to reach the ASBR requires more information, and even an additional LSA type, as compared with the

intra-area calculation. To calculate their best route to reach the ASBR, a router in another area adds the cost to reach an ABR between the areas, plus that ABR's cost to reach the ASBR. To make more sense of that concept, Figure 9-9 shows a portion of Figure 9-7, with costs highlighted, assuming the OSPF reference bandwidth is also using default settings.

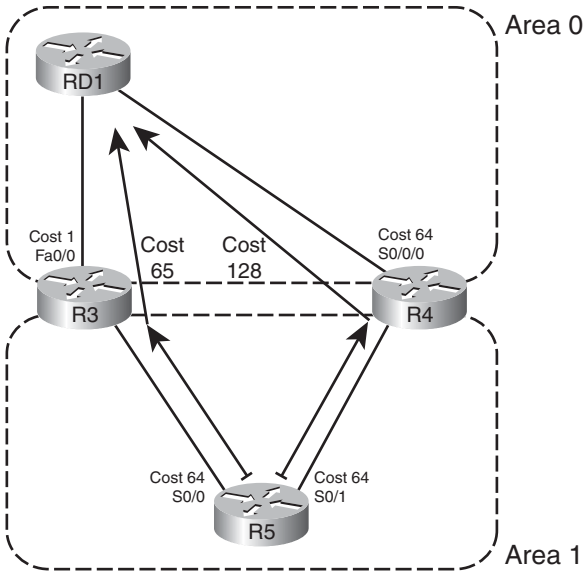


Figure 9-9 *R5's Cost to Reach ASBR RD1*

R5 has two possible routes shown in Figure 9-9 to reach ASBR RD1. On the left, the path through R3 has a total cost of 65. To the right, the router through ABR R4 has a total cost of 128. R5 then chooses the route through R3 as the best route based on the least cost to reach the ASBR.

For humans, when you have a figure and know all costs, the calculation of the costs of the two routes is simple. However, for routers, the calculation occurs in two parts:

Step 1. Calculate the cost to reach the ABR, based on the local area's topology database.

Step 2. Add the cost from the ABR to the ASBR, as listed in a Type 4 LSA.

ABRs create this new type of LSA—the Type 4 Summary ASBR LSA—to support the logic mentioned at Step 2. The Type 4 ASBR LSA lists the RID of the ASBR, and the RID of the ABR that created and flooded the Type 4 LSA. Most importantly, the Type 4 LSA lists that ABR's cost to reach the ASBR. In effect, the LSA makes an announcement like this: "I am ABR X, I can reach ASBR Y, and my cost to reach that ASBR is Z." In short, it allows the second part of the computation.

ABRs create Type 4 LSAs in reaction to receiving an external LSA from some ASBR. When an ABR forwards a Type 5 LSA into an area, the ABR looks at the RID of the ASBR

that created the Type 5 LSA. The ABR then creates a Type 4 LSA listing that ASBR, and the cost to reach that ASBR, flooding that Type 4 LSA into the neighboring areas.

For example, using Figure 9-9 again, R3 would create and flood a Type 4 Summary ASBR LSA into area 1. R3's Type 4 LSA lists ASBR 1.1.1.1 (RD1), ABR 3.3.3.3 (itself), and cost 1 (R3's cost to reach 1.1.1.1). Similarly, in that same example, ABR R4 would create another Type 4 ASBR Summary LSA. This LSA also lists ASBR 1.1.1.1 (RD1), but with advertising ABR 4.4.4.4 (R4), and lists cost 64 (R4's cost to reach 1.1.1.1).

R5, internal to area 1, then calculates the cost for each competing route by adding R5's intra-area cost to reach the respective ABRs (Step 1 in the previous list), to the cost listed in the corresponding Type 4 LSAs (Step 2 in the previous list). When R5 calculates two possible routes to reach external subnet 172.30.26.0/23, R5 finds routes both have a metric of 20, so R5 tries to break the tie by looking at the cost to reach the ASBR over each route. To do so, R5 examines each route, adding its intra-area cost to reach the ABR to the ABR's cost to reach the ASBR (as listed in the Type 4 LSA). In this case, R5 finds the route through R3 has the lower cost (65), so R5 uses outgoing interface s0/0 for its route to 172.30.26.0/23.

Example 9-11 lists the **show** command output that demonstrates the same example. Again focusing on R5's route for 172.30.26.0/23, the example first shows R5's LSDB, beginning with the Summary ASBR LSAs. More discussion follows the example.

Example 9-11 *Redistributing from EIGRP into OSPF, with Subnets*

```
R5#show ip ospf database ¦ begin ASB
                Summary ASB Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
1.1.1.1         3.3.3.3         956         0x8000000D 0x00E43A
1.1.1.1         4.4.4.4         1044        0x8000000B 0x00439A

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
172.30.2.0      1.1.1.1         1185        0x8000000B 0x006C5A 0
172.30.6.0      1.1.1.1         1185        0x8000000B 0x004082 0
172.30.12.0     1.1.1.1         1185        0x8000000B 0x00F0CD 0
172.30.17.0     1.1.1.1         1185        0x8000000B 0x00B9FF 0
172.30.26.0     1.1.1.1         1185        0x8000000B 0x00634B 0

R5#show ip ospf database asbr-summary

            OSPF Router with ID (5.5.5.5) (Process ID 5)

                Summary ASB Link States (Area 1)

  Routing Bit Set on this LSA
```

```
  LS age: 984
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(AS Boundary Router)
  Link State ID: 1.1.1.1 (AS Boundary Router address)
  Advertising Router: 3.3.3.3
  LS Seq Number: 8000000D
  Checksum: 0xE43A
  Length: 28
  Network Mask: /0
        TOS: 0  Metric: 1

  LS age: 1072
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links(AS Boundary Router)
  Link State ID: 1.1.1.1 (AS Boundary Router address)
  Advertising Router: 4.4.4.4
  LS Seq Number: 8000000B
  Checksum: 0x439A
  Length: 28
  Network Mask: /0
        TOS: 0  Metric: 64

R5#show ip ospf border-routers
OSPF Process 5 internal Routing Table
Codes: i - Intra-area route, I - Inter-area route

i 4.4.4.4 [64] via 172.16.45.4, Serial0/1, ABR, Area 1, SPF 6
I 1.1.1.1 [65] via 172.16.35.3, Serial0/0, ASBR, Area 1, SPF 6
i 3.3.3.3 [64] via 172.16.35.3, Serial0/0, ABR, Area 1, SPF 6

R5#show ip route 172.30.0.0
Routing entry for 172.30.0.0/16, 5 known subnets
  Variably subnetted with 2 masks

O E2     172.30.17.0/30 [110/20] via 172.16.35.3, 05:48:42, Serial0/0
O E2     172.30.26.0/23 [110/20] via 172.16.35.3, 05:48:42, Serial0/0
O E2     172.30.2.0/23 [110/20] via 172.16.35.3, 05:48:42, Serial0/0
O E2     172.30.6.0/23 [110/20] via 172.16.35.3, 05:48:42, Serial0/0
O E2    172.30.12.0/30 [110/20] via 172.16.35.3, 05:48:42, Serial0/0
```

The **show ip ospf database | begin ASB** command's output lists two Type 4 LSAs. (The command itself lists the summary of R5's OSPF LSDB, beginning with the section that lists Type 4 LSAs.) Both Type 4 LSAs list ASBR RD1's RID of 1.1.1.1 as the LSID, but they each list difference advertising routers: 3.3.3.3 (R3) and 4.4.4.4 (R4). In that same command, the output lists five Type 5 LSAs for the five subnets in the EIGRP domain, each with advertising router 1.1.1.1 (RD1).

The next command, **show ip ospf database asbr-summary**, lists the same two Type 4 LSAs seen in the previous command, but in detail. The first lists ASBR 1.1.1.1 (RD1), with ABR 3.3.3.3 (R3), and a cost of 1. The second lists ASBR 1.1.1.1, but with ABR 4.4.4.4 (R4), and a cost of 64. The costs list the respective ABR's cost to reach ASBR 1.1.1.1.

The third command, **show ip ospf border-routers**, lists a line for every ABR and ASBR known to the local router. It lists whether the router is inside the same area or in another area, the RID of the ABR or ASBR, and this router's best route to reach each ABR and ASBR. This command essentially shows the answer to the question "which route to ASBR 1.1.1.1 is best." Finally, the last command lists R5's IP route for 172.30.26.0, with the same next-hop and outgoing interface information as seen in the entry for RID 1.1.1.1 in the output of the **show ip ospf border-routers** command.

Redistributing into OSPF as E1 Routes

OSPF's external metric type feature allows engineers a design tool for influencing the choice of best route. E2 routes work well when the design needs to choose the best route based on the external metric–in other words, the metric as perceived outside the OSPF domain. E2 routes ignore the internal OSPF cost (except when breaking ties for best route), so when OSPF compares two E2 routes for the same subnet, that first choice to pick the lowest-metric route is based on the external metric only.

OSPF routers calculate the metrics of E1 routes by adding the internal cost to reach the ASBR to the external cost defined on the redistributing ASBR. As a result, engineer can influence the choice of routes based on the combination of the external and internal OSPF cost simply by redistributing a route as an E1 route instead of as an E2 route. To take advantage of this feature, the **redistribute** command simply needs to set the metric type.

Example 9-12 shows the simple change to the redistribution configuration on RD1 (as shown earlier in Example 9-9) to make all routes redistributed from EIGRP into OSPF be E1 routes. The example also lists output from R4 demonstrating the metric, which is based on the (default) external metric (20) plus R4's best internal metric to reach ASBR 1.1.1.1 (64).

Example 9-12 *Redistributing from EIGRP into OSPF, with Subnets*

```
RD1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
RD1(config)#router ospf 2
RD1(config-router)#redistribute eigrp 1 subnets metric-type 1
RD1(config-router)#^Z
RD1#
```

```
! Moving to router R4
R4#show ip route 172.30.0.0
Routing entry for 172.30.0.0/16, 5 known subnets
  Variably subnetted with 2 masks

O E1    172.30.17.0/30 [110/84] via 172.16.14.1, 00:00:06, Serial0/0/0
O E1    172.30.26.0/23 [110/84] via 172.16.14.1, 00:00:06, Serial0/0/0
```

```
O E1    172.30.2.0/23 [110/84] via 172.16.14.1, 00:00:06, Serial0/0/0
O E1    172.30.6.0/23 [110/84] via 172.16.14.1, 00:00:06, Serial0/0/0
O E1    172.30.12.0/30 [110/84] via 172.16.14.1, 00:00:06, Serial0/0/0

R4#show ip ospf border-routers

OSPF Process 4 internal Routing Table
Codes: i - Intra-area route, I - Inter-area route

i 1.1.1.1 [64] via 172.16.14.1, Serial0/0/0, ASBR, Area 0, SPF 16
i 3.3.3.3 [65] via 172.16.14.1, Serial0/0/0, ABR, Area 0, SPF 16
i 3.3.3.3 [128] via 172.16.45.5, Serial0/0/1, ABR, Area 1, SPF 8
```

Note that for routers in a different area than the ASBR, the calculation of metric follows the same general logic used when breaking ties for E2 routes. Generally, the computation adds three items:

- The best intra-area cost to reach the ABR (per that area's LSDB)
- The cost from that ABR to the ASBR (per Type 4 LSA)
- The external cost for the route (per Type 5 LSA)

For example, Figure 9-9 shows that R5's best cost to reach ASBR RD1 was out S0/0, to R3 next, with cost 65. Adding the external cost of 20, R5's best route will have a metric of 85. R5 calculates that cost by adding the following:

- The intra-area cost to ABR R3 (64), by analyzing the area 1 LSDB entries
- R3's cost to reach ASBR 1.1.1.1, as listed in its Type 4 LSA (1)
- The external cost as listed in the Type 5 LSA (20)

A Brief Comparison of E1 and E2 Routes

OSPF defines two types of external routes to give network designers two slightly different tools with which to calculate the best route to reach destination external to OSPF. For E1 routes, both the external cost and internal OSPF cost matters to the choice of best route. For E2 routes, only the external cost matters to the choice of best route (unless a tie needs to be broken.)

The benefits of the different external route types apply mostly to when multiple ASBRs advertise the same subnet. For example, imagine two ASBRs, ASBR1 and ASBR2, between OSPF and another routing domain. If the goal is to always send traffic through ASBR1, you could use E2 routes and set the metric for ASBR1's redistributed routes to a lower metric than ASBR2. Because routers ignore the internal metrics when calculating the E2 metrics, then every router will choose ASBR1 as the better ASBR. Conversely, if the goal were to balance the traffic, and make each router pick the closest ASBR, both ASBRs could set

the same metric to their redistributed routes, but make the routers Type E1. As a result, routers closer to each ASBR choose best routes based on the lower OSPF internal costs.

Also, note that for a given prefix/length, OSPF always prefers an E1 route over an E2 route.

External Routes in NSSA Areas

Routes may be redistributed into OSPF on any OSPF router, with a few exceptions. The router may be internal to Area 0, like router RD1 in the many examples earlier in this chapter. It can also be an ABR connected to several areas. It can be a router internal to a non-backbone area as well.

Of the four types of stubby areas, two do not allow redistribution into the area, and two do allow redistribution–even though none of the stubby area types allow Type 5 LSAs. OSPF does not allow routers in stubby and totally stubby areas to inject external routes. However, routers in not-so-stubby areas–NSSA areas–can redistribute routes, while still holding to the restriction of having no Type 5 LSAs.

OSPF supports the injection of external routes into NSSA areas by defining the Type 7 AS External LSA. This LSA type essentially replaces the Type 5 LSA's role, but only inside the NSSA area. Figure 9-10 shows a conceptual view.

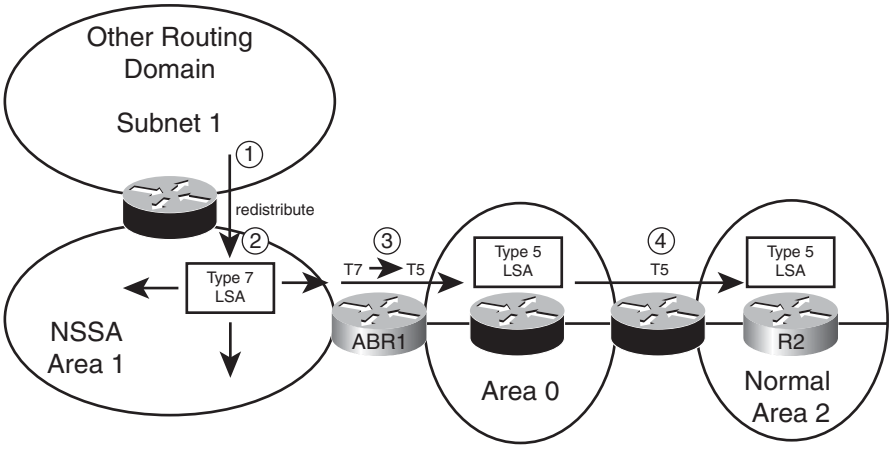


Figure 9-10 *Process of Adding and Converting Type 7 LSAs*

Following the steps in the figure:

Step 1. The ASBR attached to NSSA area 1 redistributes a route for subnet 1, creating a Type 7 LSA.

Step 2. The ASBR floods the Type 7 LSA throughout NSSA area 1.

Step 3. ABR1 converts the Type 7 LSA to a Type 5 LSA when forwarding into other areas (area 0 in this case).

Step 4. ABR2, connected to another normal area, forwards the Type 5 LSA for subnet 1 into normal area 2.

Example 9-13 demonstrates the concept using area 1 from Figures 9-7 and 9-9. Area 1 has been converted to be an NSSA area. R5 has been configured to redistribute connected routes. This feature allows a router to inject connect routes into a routing domain without having to enable the routing protocol on the corresponding interfaces. In this case, R5 will redistribute subnet 10.1.1.0/24, a connected route added by R5 using interface loopback0.

Example 9-13 *Redistributing from EIGRP into OSPF, with Subnets*

```
! R5's new configuration here:
interface loopback0
 ip address 10.1.1.1 255.255.255.0
router ospf 5
 redistribute connected subnets

R5#show ip ospf database ¦ begin Type-7
                Type-7 AS External Link States (Area 1)

Link ID          ADV Router      Age          Seq#       Checksum Tag
10.1.1.0         5.5.5.5         26           0x80000001 0x00E0A6 0

R5#show ip ospf database nssa-external

            OSPF Router with ID (5.5.5.5) (Process ID 5)

                Type-7 AS External Link States (Area 1)

  LS age: 69
  Options: (No TOS-capability, Type 7/5 translation, DC)
  LS Type: AS External Link
  Link State ID: 10.1.1.0 (External Network Number )
  Advertising Router: 5.5.5.5
  LS Seq Number: 80000001
  Checksum: 0xE0A6
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 172.16.45.5
        External Route Tag: 0
```

```
! Moving to router R8
R8#show ip ospf database ¦ begin Type-7

R8#show ip ospf database ¦ begin External
```

```
                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.1.1.0        4.4.4.4         263         0x80000001 0x009302 0
172.30.2.0      1.1.1.1         1655        0x8000000E 0x00665D 0
172.30.6.0      1.1.1.1         1655        0x8000000E 0x003A85 0
172.30.12.0     1.1.1.1         1655        0x8000000E 0x00EAD0 0
172.30.17.0     1.1.1.1         1655        0x8000000E 0x00B303 0
172.30.26.0     1.1.1.1        1655        0x8000000E 0x005D4E 0
```

The example begins with configuration on R5, followed by **show** commands on both Router R5 and R4. In particular, the **show ip ospf database | begin Type-7** command on R5 skips output until the heading for Type 7 LSAs, listing one such LSA. The LSA lists the subnet number (10.1.1.0) as the LSID, and the ASBR's RID (5.5.5.5, or R5). The next command, show detailed output from the **show ip ospf database nssa-external** command on R5 shows the details in the Type 7 LSA, including the LSA cost of 20–the same default used when injecting routes as Type 5 LSAs.

The second half of the output, on Router R8, starts with another **show ip ospf database | begin Type-7** command—the same exact command seen earlier in the example on R5. The null output in this command confirms that R8 has no Type 7 LSAs. However, the final command in the example confirms that R8 does have a Type 5 external LSA for subnet 10.1.1.0, with a listing of R4 (4.4.4.4) as the advertising router. This LSA does not list R5's RID of 5.5.5.5 as the advertising router, because R5 did not create this Type 5 LSA. Instead, R4 created this Type 5 LSA when R4 reacted to learning the Type 7 LSA inside area 1.

Finally, Example 9-14 shows a few interesting items about the IP routing table with NSSA areas. Routers inside the NSSA area use a different code in the output of **show ip route** to denote NSSA external routes as compared with normal external routes. The example shows R4's IP routing table, which lists an N2 route. This means that it is external Type 2, but inside an NSSA area, and using a Type 7 AS external LSA. The second part of the example shows R8's route for the same subnet. Because R8 is inside a non-NSSA area, R8 knows of subnet 10.1.1.0/24 because of a type 5 LSA, so R8 lists the route as an E2 route.

Example 9-14 *Redistributing from EIGRP into OSPF, with Subnets*

```
! R4's output here:
R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
! lines omitted for brevity

     10.0.0.0/24 is subnetted, 1 subnets
O N2    10.1.1.0 [110/20] via 172.16.45.5, 00:10:54, Serial0/0/1
```

```
! R8, in area 0, next
R8#show ip route ¦ begin 10.0.0.0
     10.0.0.0/24 is subnetted, 1 subnets
O E2    10.1.1.0 [110/20] via 172.16.48.4, 00:10:24, FastEthernet0/0
```

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 9-6 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 9-6 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A design shows Router R1 as being connected to both an EIGRP and OSPF routing domain, with all external EIGRP routes using a particular set of component EIGRP metrics. How can these metrics be set? (3)	
A design shows Router R1 as being connected to two different EIGRP domains, with redistribution planned. Can the design cause the routers to calculate metrics based on both the metric assigned when redistributing and the internal EIGRP topology?	
The same design as in the previous row is shown, except describe whether or not the design can cause the routers to calculate metrics based solely on the metric components assigned when redistributing.	

Table 9-6 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A design shows Router R1 as being connected to two different OSPF domains, with redistribution planned, and all routes calculated by including internal and external OSPF distance.	
The same design as in the previous row is shown, except that all external route metrics are based solely on external metrics.	

Implementation Plan Peer Review Table

Table 9-7 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 9-7 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A design shows Router R1 as being connected to both an EIGRP and OSPF routing domain. What default metrics will be used by the **redistribute** command for each routing protocol, if not set in R1's configuration?	
A plan shows redistribution between two EIGRP domains. What must be done to use the source route's original component metrics?	
A plan shows redistribution between two OSPF domains. What must be done to use the source route's original component metrics?	
The plan shows the **redistribute eigrp 2** command to redistribute from EIGRP 2 into OSPF. What other optional parameters are required to ensure redistribution of 10.1.1.0/24 from EIGRP?	
R1 has two connected interfaces in the EIGRP 2 domain and knows dozens of EIGRP routes. The plan shows the **redistribute eigrp 2 subnets** under an OSPF process. What else must be done to redistribute the two connected subnets inside the EIGRP domain?	

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 9-8 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 9-8 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configuring redistribution into EIGRP from OSPF. (List all parameters you can recall.)	
Configuring redistribution into EIGRP from OSPF. (List all parameters you can recall.)	
Setting default metrics for all **redistribute** commands, redistributing into EIGRP.	
Setting default metrics for all redistribute commands, redistributing into OSPF.	

Choosing Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 9-9 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 9-9 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display a brief version of the EIGRP topology table, listing external routes.	
Display the EIGRP topology table, including notations identifying external routes.	
For external EIGRP routes, display the source of the route, external metric, and IP address of the router that redistributed the route.	
Identify external EIGRP-learned IP routes.	
Display a brief version of the OSPF topology table, listing Type 5 External LSAs.	
Display all OSPF Type 4 LSAs.	
Display all OSPF Type 5 LSAs.	
Display all OSPF Type 7 LSAs.	
Display the external route type for an OSPF external route.	
Display OSPF cost for each interface, briefly.	

On an internal router, display any same-area ABRs' costs to reach any ASBRs.	
On an internal router, display that router's best cost to reach an ASBR.	
Display the metric for all currently best external OSPF routes.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 9-10 lists a reference of these key topics and the page numbers on which each is found.

Table 9-10 *Key Topics for Chapter 9*

Key Topic Element	Description	Page Number
List	Requirements for redistribution in a router	294
Table 9-2	Parameters on the EIGRP **redistribute** command	298
Table 9-3	Commands and options to set metrics when redistributing into EIGRP	301
List	Rules from what is redistributed from an IGP	304
Table 9-4	Parameters on the OSPF **redistribute** command	305
List	Defaults of the OSPF **redistribute** command	306
Table 9-5	Commands and options to set metrics when redistributing into OSPF	310
List	Tiebreaker rules for choosing the best E2 routes	312
List	Rules for calculating the metric of an interarea E1 route	319

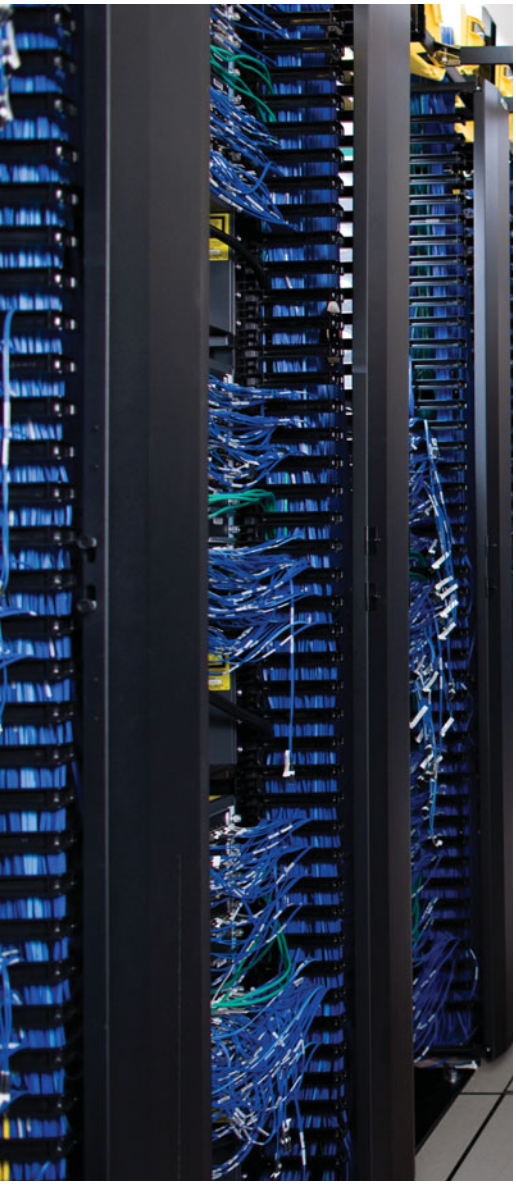
Complete the Tables and Lists from Memory

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

Define Key Terms

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

Redistribution, External route, Type 4 Summary ASBR LSA, Type 5 External LSA, Type 7 AS External LSA, External Type 1, External Type 2



This chapter covers the following subjects:

Redistribution with Route Maps and Distribution Lists: This section focuses on the functions available using route maps and distribute lists on the same router that performs redistribution into either EIGRP or OSPF.

Issues with Multiple Redistribution Points: This section examines the domain loop problem that can occur when multiple routers redistribute routes between the same two routing domains. This section also examines various solutions, including the setting of large metrics, setting the administrative distance, and using route tags.

CHAPTER 10

Advanced IGP Redistribution

This chapter progresses past the basic mechanics of route redistribution (Chapter 9, "Basic IGP Redistribution") to examine two major topics. The first part of this chapter looks at the methods by which a router can manipulate the routes being redistributed beyond the settings of the metrics. This manipulation includes the filtering of routes and the setting of other values that can be associated with a route during the redistribution process.

The second part of the chapter examines a variety of design issues that occur when multiple redistribution points exist between routing domains. Many designs use multiple redistribution points for redundancy and even for load sharing. This redundancy creates some additional complexity. (This complexity has long been a favorite topic for the CCIE Routing lab exam.) This chapter also shows methods of dealing with the design issues, including the manipulation of metrics, administrative distance, and route tags.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 10-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 10-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Redistribution with route maps and distribute lists	1-4
Issues with multiple redistribution points	5-8

1. Router R1 has been configured with the **redistribute ospf 1 route-map fred** command under **router eigrp 1**. The route map named fred needs to be configured to match routes to determine which routes are redistributed into EIGRP. Which of the following answers lists an item that cannot be matched by route map fred?

    a. Subnet number

    b. Next-hop router IP address of the route

c. Whether the route is an E1 or E2 route

d. The route's tag

e. The number of router hops between the router and the subnet

2. Router R1 refers to route-map fred when redistributing from EIGRP into OSPF. The entire route-map is listed next. Which of the following answers must be true based on the configuration as shown?

```
route-map fred deny 10
 match ip address one
route-map fred deny 20
 match ip address two
route-map fred permit 100
```

a. The third route map clause will allow any routes not already filtered by the first two clauses.

b. Routes permitted by ACL "two" will be redistributed.

c. Routes denied by ACL "one" will be redistributed.

d. All routes will be filtered.

3. On Router R1, process EIGRP 1 is redistributing routes from process OSPF 2, calling route-map fred with the **redistribute ospf 2** command. R1 has learned intra-area routes for 10.1.1.0/24 and 10.1.2.0/24 in part due to the Type 2 LSAs known for each subnet. The route map filters route 10.1.1.0/24 and allows 10.1.2.0/24 through. Which of the following commands on router R1 list subnet 10.1.1.0/24? (Choose two.)

a. **show ip route**

b. **show ip eigrp topology**

c. **show ip ospf topology**

d. **show ip eigrp topology 10.1.1.0/24**

4. Router R1 is redistributing between two OSPF processes. Given the configuration shown, which includes all commands in the route map named fred, which of the following answers is true regarding the redistribution into OSPF process 1?

```
router ospf 1
 redistribute ospf 2 external 2 route-map fred
!
route-map fred permit 10
 match ip address 1
 set metric-type type-1
```

a. No routes are redistributed because a route cannot be both E1 and E2.

b. Only OSPF E2 routes in the OSPF 2 domain will be considered for redistribution.

c. Inside the OSPF 2 domain, any formerly E2 routes will become E1 routes.

d. Routes permitted by ACL 1 will be redistributed, regardless of whether the routes are E1 or E2 routes.

5. Which of the following is not true regarding IOS default settings for administrative distance?

   a. EIGRP internal: 90
   b. OSPF external: 110
   c. EIGRP external: 90
   d. RIP: 120
   e. OSPF internal: 110

6. A network includes a RIPv2 domain, an EIGRP domain, and an OSPF domain. Each pair of routing domains has multiple routers redistributing routes between the pair of domains. The design requires that the redistribution configuration avoid matching based on prefix/length because of the trouble in maintaining such configurations. Which one of the following tools can be used in all three routing domains to attempt to prevent domain loops? (This book uses the term *domain loop* to refer to the long routes that might be chosen for routes when redistribution exists–for example, a route may forward packets from the EIGRP domain, to the OSPF domain, back to EIGRP, and then to subnet X in the RIP domain.)

   a. Setting route tags
   b. Setting the default administrative distance differently for internal and external routes
   c. Setting administrative distance differently per route
   d. Setting metrics much higher for all external routes than for all internal routes

7. A co-worker is developing an implementation plan for a design that uses OSPF 2 and RIPv2 routing domains, with two routers redistributing between the two domains. The co-worker asks your help in choosing how to prevent domain loops by setting administrative distance. (This chapter uses the term *domain loop* to refer to the long routes that might be chosen for routes when redistribution exists–for example, a route may forward packets from the EIGRP 1 domain, to OSPF2, back to EIGRP 1, and then to subnet X in the RIP domain.) Assuming all other related settings use defaults, which of the following would solve the domain loop problem?

   a. The **distance ospf intra-area 80 inter-area 80** OSPF subcommand
   b. The **distance ospf external 80** OSPF subcommand
   c. The **distance ospf intra-area 180 inter-area 180** OSPF subcommand
   d. The **distance ospf external 180** OSPF subcommand

8. Router R1 sets a route tag for subnet 10.1.1.0/24 when redistributing from OSPF into EIGRP. Which of the following unit is assigned to the route tag?

   a. Kilobits/second.
   b. Tens of microseconds.
   c. Cost.
   d. Hop count.
   e. No unit is assigned.

Foundation Topics

Redistribution with Route Maps and Distribute Lists

In some cases, a redistribution design calls for all routes to be redistributed, all with the same metric, and all with the same external route type (if applicable). However, in other cases, the metrics may need to be set differently for different routes. Additionally, some designs require that only a subset of the routes should be redistributed, for instance, when only a few key subnets need to be exposed for connections from a partner. And with routing protocols that have different types of external routes, such as OSPF and IS-IS, the design may or may not allow all redistributed routes to be of the same external route type.

All these features require a tool by which IOS can identify the routes that need to be treated differently, whether given different metrics, filtered, and assigned a different external route type. IOS provides such a feature by allowing a reference to a **route-map** from the **redistribute** command. In particular, the route-map can perform the following:

Key Topic

- Identify the subset of the routes to filter or change based on the route's prefix/length, plus many other factors.
- Make filtering choices about which routes are redistributed, and which are not.
- Set the metric to different values based on information matchable by the route-map.
- Set the type of External route for different redistributed routes, for example, OSPF Type 1 for some routes, Type 2 for others.
- Set a route tag, a unitless integer value that can later be matched with a **route-map** at another redistribution point.

This section examines the mechanics of using the **redistribute... route-map** command option to filter routes and set the metrics, along with a few other small features.

Note: Chapter 4's section "Filtering by Using route-maps" describes the logic behind the **route-map** command, so this chapter simply reviews the logic as needed. Refer to Chapter 4, "EIGRP Route Summarization and Filtering," for more detail on route maps.

Overview of Using route-maps with Redistribution

The **redistribute** command has two mechanisms that allow filtering of routes:

- The **match {internal | external 1 | external 2 | nssa-external}** parameters
- The **route-map** *map-name* option

Of these two options, the first applies only when redistributing from OSPF, and matches routes solely based on the types of routes listed here. However, the **route-map** referenced

by the **redistribute** command has many options for identifying routes by matching various facts about the route.

To identify the routes, route-maps use the **match** subcommand. The **match** command can refer to ACLs and prefix-lists to match anything matchable by those tools, plus match other facts more directly. Table 10-2 lists the **match** command options that matter when using route-maps for IGP redistribution.

Table 10-2 *match Command Options for Redistribution*

match Command	Description
match interface *interface-type interface-number* [*... interface-type interface-number*]	Looks at outgoing interface of routes
*__match ip address__ {[*access-list-number* | *access-list-name*] | **prefix-list** *prefix-list-name*}	Examines route destination prefix and prefix length
*__match ip next-hop__ {*access-list-number* | *access-list-name*}	Examines route's next-hop address
*__match ip route-source__ {*access-list-number* | *access-list-name*}	Matches advertising router's IP address
match metric *metric-value* [**+-** deviation]	Matches route's metric, or a range (plus/minus the configured deviation)
match route-type {**internal** | **external** [**type–1** | **type–2**] | **level–1** | **level–2**}	Matches route type
match tag *tag-value* [*...tag-value*]	Matches the route tag, which requires that another router has earlier set the tag

* Can reference multiple numbered and named ACLs on a single **match** command.

A route-map referenced by the **redistribute** command always attempts to filter routes. If the route-map matches a particular route with a particular route-map clause, and the action in that clause is **permit**, then the route is redistributed. However, if the first route-map clause matched by a packet has a **deny** action, the packet is filtered–in other words, not redistributed. In short, the logic matches the same logic as described in Chapter 4 for route-maps referenced by the **distribute-list** command. (Chapter 4 shows how to filter routes inside EIGRP, without redistributing.)

Additionally, for routes not filtered by the route-map, the route-map can set other values (like the route's metric) using the aptly-named **set** command. Table 10-3 lists the various route-map **set** subcommands that can be used to set the values used for routes redistributed into IGPs.

Key Topic

Table 10-3 *set Command Options for Redistribution into IGPs*

set Command	Description
set metric *metric-value*	Sets the route's metric for OSPF, RIP, and IS-IS
set metric *bandwidth delay reliability loading mtu*	Sets the EIGRP route's metric values
set metric-type {**type–1** | **type–2**}	Sets type of route for OSPF
set tag *tag-value*	Sets the unitless tag value in the route

Filtering Redistributed Routes with Route Maps

As usual, the best way to understand the configuration, and the methods to verify the results, is to use an example. In this case, the same internetwork used throughout Chapter 9 is used, but with some more routes added. Figure 10-1 shows some of the detail of the internetwork.

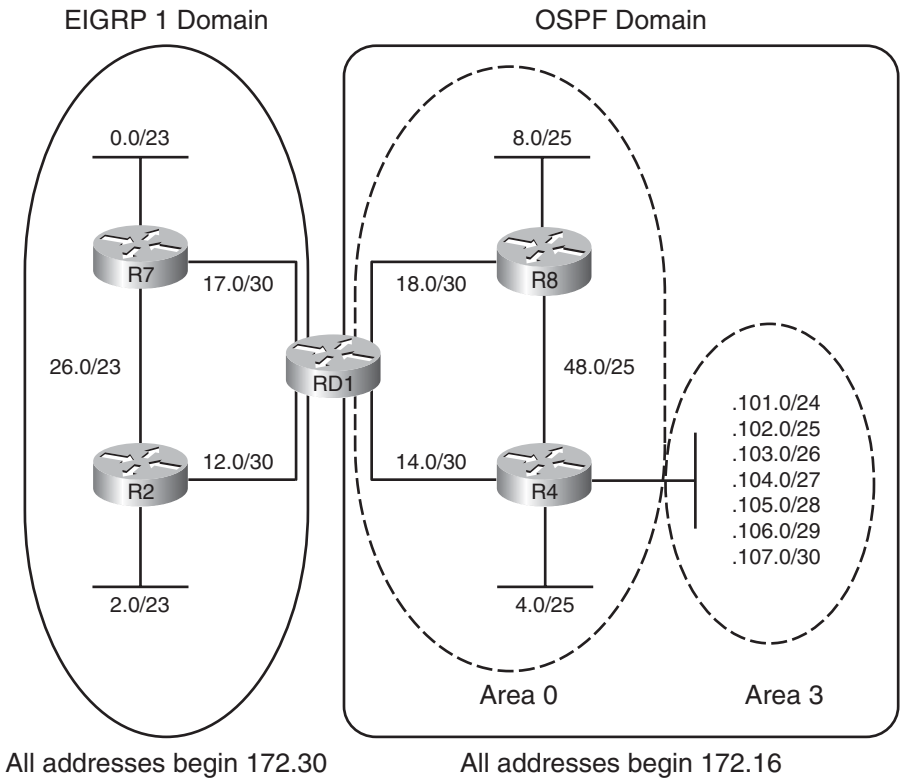


Figure 10-1 *Sample Internetwork Used for Redistribution Route Map Examples*

The internetwork has been preconfigured with mainly defaults, as follows:

- EIGRP works well on the left side of Figure 10-1.
- OSPF works well on the right side.
- Mutual redistribution has been configured on router RD1, with no filtering.
- All routes use these metric settings: EIGRP (1500 10 255 1 1500), OSPF (20).

Example 10-1 shows the routing protocol configuration on Router RD1 at the beginning of the example.

Example 10-1 *Initial Configuration–Mutual Redistribution, No Filtering*

```
RD1#show run
! lines omitted for brevity
router eigrp 1
 redistribute ospf 2
 network 172.30.0.0
 default-metric 1500 10 255 1 1500
 auto-summary
!
router ospf 2
 router-id 1.1.1.1
 log-adjacency-changes
 redistribute eigrp 1 subnets
 network 172.16.0.0 0.0.255.255 area 0
```

Configuring Route Filtering with Redistribution

The configuration shown in Example 10-1 shows mutual redistribution with no filtering. The next example extends that same configuration to now use a **route-map** that should filter routes being redistributed from OSPF process 2 into EIGRP AS 1. Any routes not mentioned in Table 10-4, but shown in Figure 10-1, should be redistributed.

Table 10-4 *Parameters Used in Route Filtering Example*

Prefixes	Action
172.16.101.0/24	deny
172.16.102.0/25	permit
172.16.103.0/26	
172.16.104.0/27	deny
172.16.105.0/28	
172.16.106.0/29	permit
172.16.107.0/30	

The route-map simply needs to match the routes to be filtered with a route-map clause that has a **deny** action and match the routes to not be filtered with a clause with a **permit** action. Example 10-2 shows two such potential solutions, with **route-map** names option1 and option2. The general style of the two options, both of which work, is as follows:

- **Option 1:** Begin with a match of the routes to be filtered, using extended IP ACLs, with a **deny** action so the routes are filtered. Then use a **permit** clause with no **match** command at all, matching and allowing through all remaining routes.
- **Option 2:** Begin with a match of the routes to be allowed, matching with prefix lists, with a **permit** action. Then use the implicit deny all at the end of the route-map to filter unwanted routes.

Example 10-2 *Redistribution Filtering Configuration Example*

```
! This ACL matches subnet 172.16.101.0, with mask 255.255.255.0
ip access-list extended match-101
 permit ip host 172.16.101.0 host 255.255.255.0

! This ACL matches subnets 172.16.104.0 and 172.16.105.0, with masks
! 255.255.255.224 and 255.255.255.240, respectively.
ip access-list extended match-104-105
 permit ip host 172.16.104.0 host 255.255.255.224
 permit ip host 172.16.105.0 host 255.255.255.240
!
! This prefix list matches the five subnets in area 0
ip prefix-list match-area0-permit seq 5 permit 172.16.14.0/30
ip prefix-list match-area0-permit seq 10 permit 172.16.18.0/30
ip prefix-list match-area0-permit seq 15 permit 172.16.8.0/25
ip prefix-list match-area0-permit seq 20 permit 172.16.4.0/25
ip prefix-list match-area0-permit seq 25 permit 172.16.48.0/25
!
! This prefix list matches the two sets of two area 3 subnets that will
! be permitted to be redistributed
ip prefix-list match-area3-permit seq 5 permit 172.16.102.0/23 ge 25 le 26
ip prefix-list match-area3-permit seq 10 permit 172.16.106.0/23 ge 29 le 30
```

```
! The first alternative route-map:
route-map option1 deny 10
 match ip address match-101
!
```

```
route-map option1 deny 20
 match ip address match-104-105
!
route-map option1 permit 100
```

```
! The second alternative route-map:
route-map option2 permit 10
 match ip address prefix-list match-area3-permit
!
route-map option2 permit 20
 match ip address prefix-list match-area0-permit
```

```
! Finally, the configuration shows the enablement of option 1.
Router eigrp 1
 Redistribute ospf 2 route-map option1
```

Route-map option1 takes the approach of denying the redistribution of some routes, and then allowing the rest through. The last clause in this route map, with sequence number 100, does not have a **match** command at all, meaning that it will match any and all routes. The **permit** action on this last clause overrides the implied deny all at the end of the route-map.

The ACLs referenced by **route-map option1** show some particular interesting features for matching routes. With an extended ACL, IOS compares the source IP address parameter to the subnet number of the route and the destination IP address to the subnet mask of the route. For example, the **permit ip host 172.16.1.0 host 255.255.255.0** command matches the specific route for subnet 172.16.101.0, specifically with mask 255.255.255.0.

Route-map option2 takes the opposite approach compared to option1, for no other reason than to just show an alternative. It uses two different prefix lists to match the routes—one for subnets in area 0, all of which are redistributed, another for subnets in area 3 that should be allowed through the redistribution process. Alternatively, all routes could have been matched with a single prefix list, with a single **permit** clause in the option2 route-map.

Finally, the very end of the example shows the syntax of the **redistribute** command, with **route-map option1** enabled.

Verifying Redistribution Filtering Operations

The redistribution process takes routes from the IP routing table of a router and adds the appropriate entries to the destination routing protocol's topology table. The filtering process prevents some of the routes from being added to the topology table, so an examination of the destination routing protocol's topology table shows whether the filtering

worked correctly. Additionally, the routing tables of other routers in the destination routing domain can be checked.

A good redistribution verification plan should check that the correct routes are filtered and confirm that no extra routes are filtered. In a production environment, that work might be laborious. With the example shown in Figure 10-1 and Example 10-2, verification takes a little less time due to the relatively small number of routes and that the subnets in the OSPF domain all begin with 172.16.

Example 10-3 shows an abbreviated version of the EIGRP topology table on Router RD1. The **show ip route 172.16.0.0** command lists the 12 OSPF subnets that currently exist in the OSPF domain (as shown in Figure 10-1). The **show ip eigrp topology | include 172[.]16** command lists only routes that include text "172.16," listing only nine subnets–and omitting the three subnets that should have been filtered, which confirms that the filtering worked.

Note: The brackets in the **show ip eigrp topology | include 172[.]16** command tell IOS to treat the period as a literal, searching for the text "172.16" in the command output, instead of treating the period as a wildcard in an IOS regular expression.

Example 10-3 *Verifying Redistribution Filtering*

```
RD1#show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 12 known subnets
  Attached (2 connections)
  Variably subnetted with 7 masks
  Redistributing via eigrp 1

O       172.16.48.0/25 [110/65] via 172.16.18.2, 03:25:56, Serial0/0/1
                       [110/65] via 172.16.14.2, 03:24:09, Serial0/1/0
C       172.16.18.0/30 is directly connected, Serial0/0/1
C       172.16.14.0/30 is directly connected, Serial0/1/0
O       172.16.8.0/25 [110/65] via 172.16.18.2, 03:25:56, Serial0/0/1
O       172.16.4.0/25 [110/65] via 172.16.14.2, 03:24:49, Serial0/1/0
O IA    172.16.104.0/27 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.105.0/28 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.106.0/29 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.107.0/30 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.101.0/24 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.102.0/25 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0
O IA    172.16.103.0/26 [110/65] via 172.16.14.2, 03:24:44, Serial0/1/0

RD1#show ip eigrp topology ¦ include 172[.]16
P 172.16.48.0/25, 1 successors, FD is 1709056
P 172.16.18.0/30, 1 successors, FD is 1709056
P 172.16.14.0/30, 1 successors, FD is 1709056
```

```
P 172.16.8.0/25, 1 successors, FD is 1709056
P 172.16.4.0/25, 1 successors, FD is 1709056
P 172.16.106.0/29, 1 successors, FD is 1709056
P 172.16.107.0/30, 1 successors, FD is 1709056
P 172.16.102.0/25, 1 successors, FD is 1709056
P 172.16.103.0/26, 1 successors, FD is 1709056
```

Besides examining the topology tables on the router doing the redistribution, a **show ip route** command on other routers inside the EIGRP domain, like R2, could be used to confirm the presence and absence of the routes according to the plan. However, the routing table on the redistributing router will list the routes as learned from the original routing domain.

Any ACLs or prefix lists used to match packets can also be used as a gauge to tell if the correct statements matched routes. The **show ip access-list** *[number|name]* and **show ip prefix-list detail** *[name]* commands list counters that increment each time IOS matches a route for redistribution. Particularly when first using the ACL or prefix list, these commands can confirm which statements have been matched. The counters do increment each time the router considers whether to redistribute a route. In particular, when a route fails, and the redistributing router removes the route from the routing table, and then later adds the route to the routing table again, the counters for matching the ACL or prefix list will increment. Example 10-4 shows an example of each command, and the appropriate counters.

Example 10-4 *Verifying Redistribution Filtering*

```
RD1#show access-list
Extended IP access list match-101
    10 permit ip host 172.16.101.0 host 255.255.255.0 (1 match)
Extended IP access list match-104-105
    10 permit ip host 172.16.104.0 host 255.255.255.224 (1 match)
    20 permit ip host 172.16.105.0 host 255.255.255.240 (1 match)
RD1#show ip prefix-list detail match-area-0-permit
ip prefix-list match-area0-permit:
   count: 5, range entries: 0, sequences: 5 - 25, refcount: 3
   seq 5 permit 172.16.14.0/30 (hit count: 6, refcount: 1)
   seq 10 permit 172.16.18.0/30 (hit count: 5, refcount: 1)
   seq 15 permit 172.16.8.0/25 (hit count: 4, refcount: 2)
   seq 20 permit 172.16.4.0/25 (hit count: 3, refcount: 3)
   seq 25 permit 172.16.48.0/25 (hit count: 2, refcount: 2)
```

Setting Metrics when Redistributing

Setting a different metric for different redistributed routes requires only a minor amount of additional configuration. The redistributing router still needs a route-map and still needs to match the routes. Additionally, to set the metric for routes matched by a particular clause, the route-map needs the **set metric** route-map subcommand. When redistribut-

ing into EIGRP, this command has five parameters (bandwidth, delay, reliability, load, and MTU). When redistributing into OSPF or RIP, a single integer metric is used.

Configuring the Metric Settings

Continuing with the same internetwork shown in Figure 10-1, and with the same filtering goals summarized earlier in Table 10-4, Table 10-5 further defines the goals from redistribution from OSPF into EIGRP in this internetwork. The same routes will be filtered, but now the metrics of the allowed routes will be set differently as listed in the table.

Table 10-5 *Parameters Used in Metric and Tag Setting Example*

Prefix	Action	Metric (Bandwidth, delay, reliability, load, MTU)
172.16.101.0	deny	N/A
172.16.102.0 172.16.103.0	permit	1000 44 255 1 1500
172.16.104.0 172.16.105.0	deny	N/A
172.16.106.0 172.16.107.0	permit	100 4444 255 1 1500
All others	permit	1500 10 255 1 1500

The requirements in Table 10-5 list three different sets of metrics for the redistributed routes. To implement this design, the route-map needs at least three clauses: one for each set of routes for which the metric should differ. The example route-maps listed earlier in Example 10-2 do not happen to separate the three groups of allowed routes into different route-map clauses, so a new route-map will be used. Example 10-5 shows the new configuration. Note that it does make use of one of the old IP prefix-lists, namely **match-area0-permit**.

Example 10-5 *Route-map to Set Metrics According to Table 10-5*

```
! First, two new prefix lists are added – one to match subnets 102 and 103,
! and another to match subnets 106 and 107.


ip prefix-list match-102-103 seq 5 permit 172.16.102.0/23 ge 25 le 26
!
ip prefix-list match-106-107 seq 5 permit 172.16.106.0/23 ge 29 le 30

! The following is a repeat of the prefix list that matches the five routes
! in area 0
ip prefix-list match-area0-permit seq 5 permit 172.16.14.0/30
ip prefix-list match-area0-permit seq 10 permit 172.16.18.0/30
```

```
ip prefix-list match-area0-permit seq 15 permit 172.16.8.0/25
ip prefix-list match-area0-permit seq 20 permit 172.16.4.0/25
ip prefix-list match-area0-permit seq 25 permit 172.16.48.0/25

! A new route map to filter and set metrics, with three clauses


route-map set-metric permit 10
 match ip address prefix-list match-area0-permit
!
route-map set-metric permit 20
 match ip address prefix-list match-102-103
 set metric 1000 44 255 1 1500
!
route-map set-metric permit 30
 match ip address prefix-list match-106-107
 set metric 100 4444 255 1 1500


!
router eigrp 1
 default-metric 1500 10 255 1 1500
 redistribute ospf 2 route-map set-metric
```

The new route-map has three explicitly configured clauses, two of which explicitly set the metric values using the **set metric** command. However, the first clause (sequence number 10), which matches routes for the five subnets inside area 0, does not use a **set metric** command to set the metric. Instead, because this route map clauseomits the **set metric** command, routes that match this clause use the **metric** keyword on the **redistribute** command, or if not listed, the metrics as defined by the **default-metric** EIGRP subcommand. In this case, because the **redistribute** command does not list a **metric** keyword, routes matched by this clause (sequence number 30) use the metric values listed in the **default-metric** command.

Verifying the Metric Settings

Verifying the metrics again requires an examination of the EIGRP topology table. In this case, Example 10-6 displays a couple of views of RD1's EIGRP topology table, focusing on routes to 172.16.102.0/25 and 172.16.106.0/29. The configuration in Example 10-5 earlier set the metrics to different values, and next the output in Example 10-6 shows the differences:

Example 10-6 *Verifying Metrics as Set During Redistribution*

```
RD1#show ip eigrp topology 172.16.102.0/25
IP-EIGRP (AS 1): Topology entry for 172.16.102.0/25
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1709056
```

```
  Routing Descriptor Blocks:
  172.16.14.2, from Redistributed, Send flag is 0x0
      Composite metric is (2571264/0), Route is External
      Vector metric:
        Minimum bandwidth is 1000 Kbit
        Total delay is 440 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      External data:
        Originating router is 172.30.17.1 (this system)
        AS number of route is 2
        External protocol is OSPF, external metric is 65
        Administrator tag is 0 (0x00000000)

RD1#show ip eigrp topology 172.16.104.0/25
% IP-EIGRP (AS 1): Route not in topology table

RD1#show ip eigrp topo 172.16.106.0/29
IP-EIGRP (AS 1): Topology entry for 172.16.106.0/29
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1709056
  Routing Descriptor Blocks:
  172.16.14.2, from Redistributed, Send flag is 0x0
      Composite metric is (26737664/0), Route is External
      Vector metric:
        Minimum bandwidth is 100 Kbit
        Total delay is 44440 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      External data:
        Originating router is 172.30.17.1 (this system)
        AS number of route is 2
        External protocol is OSPF, external metric is 65
        Administrator tag is 0 (0x00000000)
!
RD1#show ip prefix-list detail match-102-103
ip prefix-list match-102-103:
   count: 1, range entries: 1, sequences: 5 - 5, refcount: 2
   seq 5 permit 172.16.102.0/23 ge 25 le 26 (hit count: 14, refcount: 1)
```

Although you could use variations of the **show ip route** command to verify the new metrics, because the redistribution process sets the EIGRP component metrics, the **show ip eigrp topology** command displays much more useful verification information.

Setting the External Route Type

When redistributing into OSPF, IOS automatically sets the external route type to external Type 2 (E2). When redistributing into OSPF, IOS can set the type to E1 or E2 by using the **set metric-type {type-1 | type-2}** route-map subcommand. When a **redistribute** OSPF subcommand references such a route-map, the routes matched by the route-map clause with the **set metric-type** command will be designated as that external type in the Type 5 LSA created for that subnet.

Note that the **redistribute** command also allows the **match {internal | external 1 | external 2 | nssa-external}** parameters, but these parameters do not set the type or route. Instead, these parameters match existing routes as part of the process of deciding which routes to redistribute.

Redistribution Filtering with the **distribute-list** Command

Using a route-map as referenced on the **redistribute** command provides many features. You can filter routes, assign different metrics for different routes, and assign external route types. You can even assign route tags as discussed later in the section "Preventing Domain Loops by Filtering on Route-tag Using Distribute Lists." However, if the plan calls for route filtering only when redistributing, but none of the other functions supplied by a route-map are needed, and you can match all the routes with a single ACL or prefix list, then IOS supports a second style of route filtering configuration using the **distribute-list** command.

This book has reviewed two uses of the **distribute-list** command in earlier chapters (Chapters 4 and 8), both of which show how to filter routes inside a single routing domain. For example, Chapter 4 shows how to filter EIGRP routes using the **distribute-list** command, both for routing updates received in and for routing updates sent out by a router. The **distribute-list** command refers to the direction, and to either an ACL or IP prefix-list, allowing the routes matched with a permit and filtering routes matched with a deny action.

The **distribute-list** command can be configured to refer to the routing process from which routes are redistributed and cause the router to filter routes taken from that process. To do so, the command must use the **out** direction, and it must refer to the routing process from which routes are redistributed. For example, **distribute-list 1 out ospf 2**, configured under an EIGRP process, tells EIGRP to apply ACL 1 to routes redistributed from the OSPF 2 process. For another example, under an OSPF process, the **distribute-list prefix fred out eigrp 1** command tells OSPF to apply IP prefix list fred to routes redistributed from the EIGRP 1 process.

Finally, one note about internals of how this command works. The filtering takes place as the routes are redistributed. As a result, routes filtered by the **distribute-list** command prevent the routes from being added to the topology table of the destination routing protocol. So, the same verification commands seen in earlier examples, with focus on the topology tables, can be used to show whether the filtering worked. Also, the counters in the **show ip access-list** and **show ip prefix-list detail** commands also increment to show whether the filtering worked.

Issues with Multiple Redistribution Points

The use of a single router to redistribute routes means that a single failure could cause hosts in different routing domains to fail. The redistributing router could simply fail, or interfaces or links on that router could fail. To avoid that single point of failure, most redistribution designs call for a minimum of two routers performing redistribution, particularly in cases where the redistribution function will be somewhat permanent.

The existence of two or more redistribution points between the same two routing domains introduces some complexity and caveats. The issues revolve around the concept that a route in one domain can be advertised into another domain, and then back into the original routing domain. In fact, the CCIE Routing and Switching exam has included such tricky redistribution issues as a central part of the exam for many years.

Figure 10-2 shows one of the issues when using multiple redistribution points. In this case, the arrowed lines show the best route from a router in domain 2 to reach a subnet also in domain 2. However, the route actually passes through domain 1.

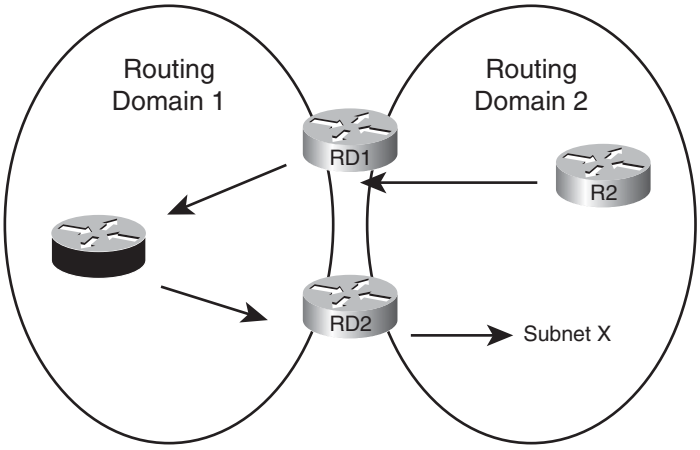


Figure 10-2 *A Domain Loop*

Figure 10-2 shows the long route that goes from R2, through RD1, to R1, back into routing domain 2 through RD2. This long route occurs because of the routing advertisements that flow in the opposite direction: advertised by RD2 into routing domain 1 and then by RD1 back into routing domain 2. The problem occurs when the twice-redistributed route for subnet X is redistributed back into the original domain with a relatively low metric. The twice-redistributed route then has a better metric than the route that was advertised only internal to that routing domain.

This section examines how to prevent this "domain loop" problem when using multiple redistribution points. Interesting, this problem does not occur, at least with default settings, when EIGRP is one of the two routing protocols. So this section begins with examples of RIP and OSPF redistribution, showing how to prevent this domain looping problem, and then showing why EIGRP accomplishes this same feat with default settings.

Note: I know of no industry standard name for the problem shown in Figure 10-2. For the duration of this chapter, I refer to it simply as the *domain loop* problem.

Preventing Routing Domain Loops with Higher Metrics

One easy method of preventing the domain loop problem is to assign purposefully high metric values when redistributing routes. For example, consider the case shown in Figure 10-3, with a RIP domain on the left, and OSPF on the right. In this case, the two routers doing the redistribution (RD1 and RD2) assign OSPF metric 500 when redistributing routes into OSPF, and metric 5 when redistributing routes into RIP.

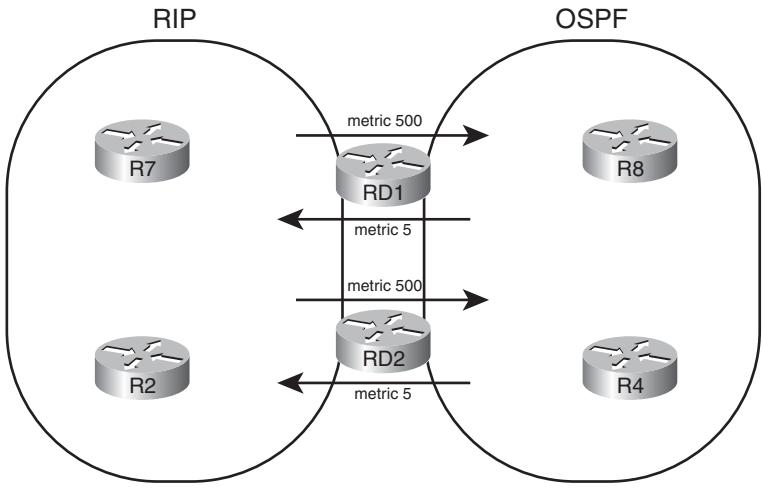


Figure 10-3 *Defeating Domain Loops by Using Very Large Metrics*

First, focus on routes inside the RIP domain. This design prevents the domain loop problem—routes that send packets from the RIP domain, into OSPF, and back again—if the normal intra-domain RIP routes never exceed a hop count of 4. Then, all routes redistributed from RIP into OSPF, and then back into RIP, will at least have a metric of 5. As a result, the route advertisements that looped back into the RIP domain will always have less desirable metrics than the RIP advertisements from within the RIP domain.

The same concept applies to OSPF. For routes completely internal to the OSPF domain, if the highest cost is 499, then the redistribution of external routes as metric 500 makes prevents the domain loop. For example, a subnet that exists in the OSPF domain could be advertised into RIP by RD1, and then re-advertised by RD2 back into the OSPF domain—but with a metric that begins at 500. Again, assuming all the normal OSPF routes that were not reintroduced as external routes have a cost of less than 500, the domain loop problem is defeated.

Note that OSPF actually defeats the domain loop problem without using the higher metrics. OSPF always prefers internal routes over E1 routes, and E1 routes over E2 routes, before even considering the metrics.

Preventing Routing Domain Loops with Administrative Distance

Each router associates an *administrative distance* (AD) with every route it considers to be added to the routing table. When a router must consider multiple routes from different sources for the exact same prefix/length, the first item considered by the router is not the metric, but rather the AD. The lower the AD, the better the route.

Note that the AD is a local setting on a router and cannot be advertised to neighboring routers.

Each routing source has a default AD according to IOS. In some cases, a given routing source has different defaults for different types of routes inside that routing source. For example, EIGRP has a separate setting for EIGRP internal routes (AD 90) than EIGRP external routes (AD 170). Table 10-6 lists the default settings.

Key Topic

Table 10-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
On-Demand Routing (ODR)	160
EIGRP (external)	170
iBGP	200
Unreachable	255

EIGRP Default AD Defeats Loop from EIGRP to OSPF to EIGRP

The default AD settings for EIGRP takes care of the domain loop problem when redistributing between EIGRP and OSPF. First, consider an EIGRP and OSPF domain with two redistribution points (Routers RD1 and RD2), as shown in Figure 10-4. The figure shows a general idea of route advertisements for subnet X, which exists in the EIGRP domain. (Note: to reduce clutter, the figure shows only route advertisements that affect router RD2's logic; the same issue exists on both redistributing routers.)

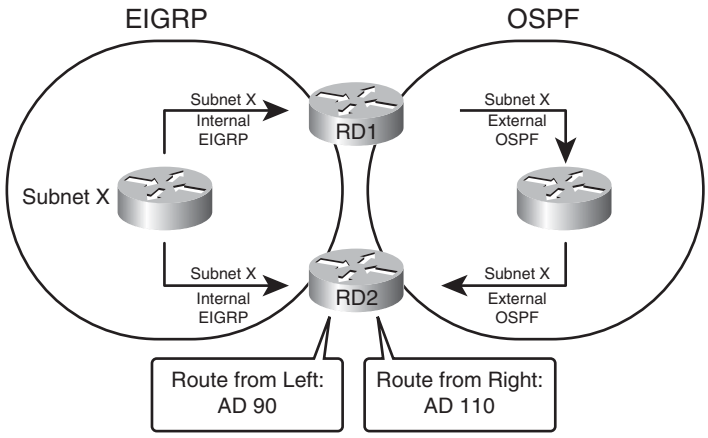


Figure 10-4 *Subnet X: Internal EIGRP, External OSPF, on Router RD2*

Key Topic

Router RD2 hears about a route for subnet X as an internal EIGRP route (default AD 90) on the left. RD2 also hears about the subnet as an external OSPF route on the right (default AD 110). As a result, RD2 will do a couple of things that are important to this discussion:

- RD2 considers the internal EIGRP route as the best route, because of the lower AD, and places that route in its own IP routing table.
- RD2 does not redistribute a route for subnet X, from OSPF back to EIGRP, because RD2 does not have an OSPF route for subnet X.

The second point is particularly important but easily missed. Remember that routers use the IP routing table as the basis for route redistribution. Both RD1 and RD2 redistribute routes in both directions between both domains. However, a route must be in the routing table before it can be redistributed. Because RD2's route for subnet X will list its EIGRP route, RD2's redistribution from OSPF into EIGRP will not redistribute a route for subnet X. Because RD2 will not advertise a route for subnet X from OSPF back into EIGRP, the domain loop has been prevented.

EIGRP Default AD Defeats Loop from OSPF to EIGRP to OSPF

The reverse case–routes taken from OSPF, advertised into EIGRP, and then advertised back into OSPF–is the more interesting possible domain loop case. However, the default EIGRP AD settings still defeat the domain loop issue. Figure 10-5 shows an example similar to Figure 10-4, but this time with subnet Y in the OSPF domain. As before, the focus of the figure is on the routing advertisements that reach Router RD2, with other details omitted to reduce clutter.

In this case, Router RD2 hears about a route for subnet Y as an external EIGRP route (default AD 170) and an internal OSPF route (default AD 110). As a result, RD2 chooses the OSPF internal route as the best route and adds that to RD2's routing table. Because RD2 does not have an EIGRP route for subnet Y, RD2 will not redistribute a route for subnet Y from EIGRP into OSPF, again defeating the domain loop problem.

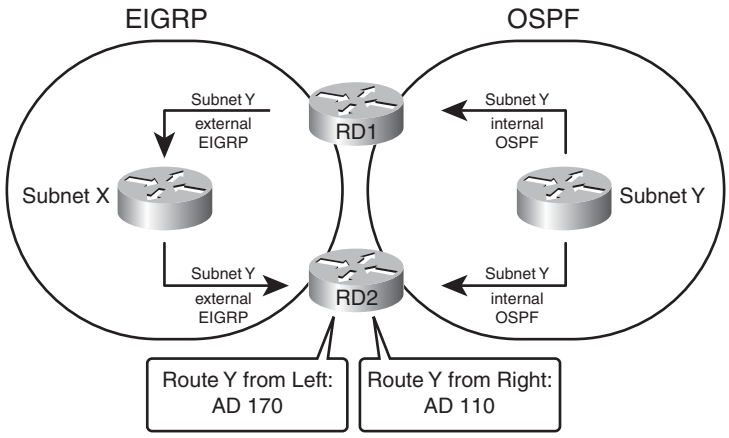


Figure 10-5 *Avoiding Domain Loops from OSPF to EIGRP to OSPF*

Setting AD per Route Source for Internal and External Routes

The reason that the default EIGRP AD settings work well can be summarized generically as follows:

> For each of the two routing protocols, the AD used for internal routes for one routing protocol is better than the AD used for external routes by the other routing protocol.

When comparing EIGRP's and OSPF's defaults, both of the generic criteria are met:

- EIGRP internal AD 90 < OSPF external AD 110
- OSPF internal AD 110 < EIGRP external AD 170

Likewise, when redistributing between EIGRP and RIP:

- EIGRP internal AD 90 < RIP external AD 120
- RIP internal AD 120 < EIGRP external AD 170

Note: RIP does not have a concept of internal and external routes; the preceding references refer to internal routes as routes that exist inside the RIP domain, and external as routes that exist outside the RIP domain.

When redistributing between OSPF and RIP, the default AD settings do not defeat the domain loop problem. However, IOS supports the definition of different AD settings for all routing protocols. With EIGRP, the internal and external AD settings can be overridden, although the defaults work well for the preventing of domain loops. OSPF can be configured to use a different AD for external routes, intra-area routes, and interarea routes. RIP, which does not have a concept of internal and external routes, can only be set with a single AD value. Table 10-7 shows the router subcommands to set the AD values, per route category.

Table 10-7 *Setting AD Values with the* ***distance*** *Command*

Routing Protocol	Command
RIP	**distance** *ad-value*
EIGRP	**distance eigrp** *internal-ad external-ad*
OSPF	**distance ospf {external** *ad-value***} {intra-area** *ad-value***} {inter-area** *ad-value***}**

To defeat the OSPF-RIP domain loop problem by setting AD, just configure the AD for OSPF external routes using the **distance ospf external** *ad-value* command in OSPF configuration mode. The actual AD value does not matter much, but it should be higher than RIP's AD on that same router. For example, the **distance ospf external 130** command in OSPF configuration mode results in the following, assuming all other AD values default:

- RIP internal AD 120 < OSPF external AD 130
- OSPF internal AD 110 < RIP external AD 120

Domain Loop Problems with More than Two Routing Domains

With only two routing domains, the domain loop solutions seen so far–setting higher metrics and setting AD—can deal with the domain loop problems. However, with three or more routing domains, setting metrics and AD values does not always solve the domain loop problem. In particular, problems can occur when three or more routing domains connect in sequence, as shown in Figure 10-6.

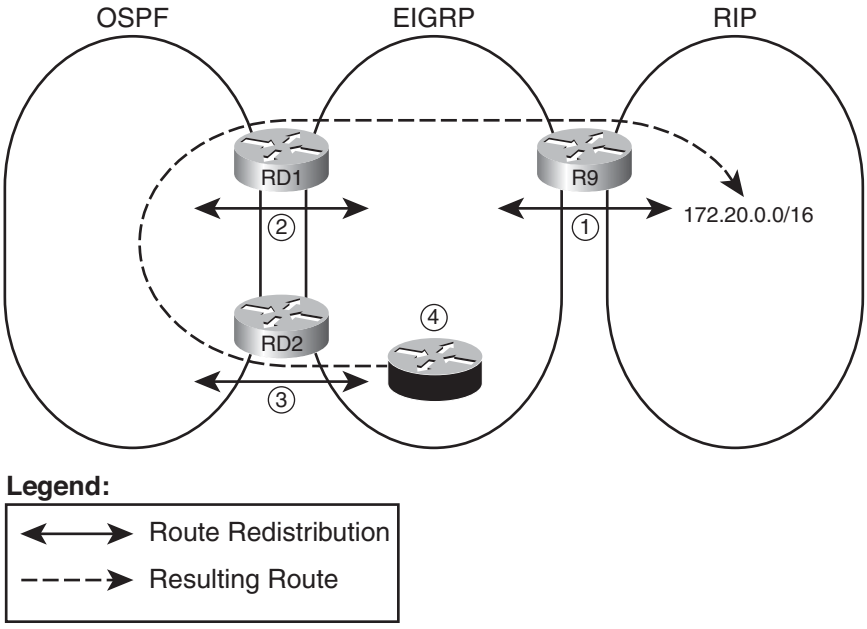


Figure 10-6 *Inefficient Routing with Looped Routing Advertisements*

The steps noted in the figure are as follows:

Step 1. Router R9 advertises a route for network 172.20.0.0/16 from the RIP domain into the EIGRP domain where the route is treated with (default) AD 170 as an external route.

Step 2. Router RD1 redistributes this EIGRP external route into OSPF where it is treated as an E2 route, AD 110, by default.

Step 3. Router RD2 uses the AD 110 E2 route, rather than the AD 170 EIGRP external route, as its best route for 172.20.0.0/16. As a result, RD2 can then redistribute that OSPF route back into EIGRP as an external route.

Step 4. Router R4 learns of two external routes for 172.20.0.0/16, and the routes tie based on AD (170). R4 may have a better EIGRP metric through RD2, depending on the metrics used at redistribution, preferring this long route through the OSPF domain as shown.

This is just one example case for such problems, but the problem exists because the obviously better route and the longer domain loop route are both external routes. The two competing routes tie on AD as a result. In the earlier cases, with only two routing domains, this problem does not occur.

Several solutions exist for such problems. None of the solutions require a lot of extra configuration, other than that some of the solutions require ACLs or prefix lists that match the prefixes from the various routing domains. The next three sections address each option, namely: using per-route AD settings, filtering routes based on prefix/length, and using route tags.

Using Per-route Administrative Distance Settings

As seen in Table 10-7, you can use the **distance** router subcommand to set the AD value per routing protocol, per type (internal and external). The **distance** command also supports another syntax in which the router sets the AD for individual routes based on the following criteria:

- The router that advertised the routing information
- Optionally, for the prefixes/lengths of the routes as matched by a referenced ACL

The syntax of the command in this case is

```
distance distance ip-adv-router wc-mask [acl-number-or-name]
```

In this command, the required parameters match the neighboring router that advertises a route. The router with the **distance** command configured compares the advertising router's IP address to the range of addresses implied by the *ip-adv-router* and *wc-mask* parameters of the command, as if these were parameters in an ACL. For routes advertised by a matching neighbor, that router then applies the AD listed in the command.

Optionally, the **distance** command can also refer to an ACL. If included, that router compares the ACL to the prefix/length of each route learned from any matched neighbors and uses the listed AD only for routes permitted by the ACL.

For example, consider the problem in Figure 10-6. Assuming the design calls for all hosts to have reachability to 172.20.0.0/16, the route must be redistributed by R9 into the EIGRP domain. For the best availability, this route should be redistributed from EIGRP into OSPF at both redistribution points (RD1 and RD2). The unfortunate long route choice by Router R4 in the figure occurs at what is listed as Step 3 in that figure, with Router RD2 using AD to determine that its external OSPF route for 172.20.0.0/16 (AD 110) is better than its EIGRP external route (AD 170) for that same prefix.

One solution would be to cause RD2 to use a higher AD–specifically higher than the 170 AD used for EIGRP external routes–for prefix 172.20.0.0/16 as learned with OSPF. A **distance** command on RD2 could solve the problem.

Upcoming Examples 10-7 and 10-8, plus Figure 10-7, demonstrate both the domain loop problem in this same case, along with the solution. First, Figure 10-7 shows a more detailed topology for reference. (Note that unlike several other examples in this chapter and in Chapter 9, this example shows EIGRP on the right, and OSPF on the left.) Then, Example 10-7 shows the relevant configuration and a few related **show** commands on Router RD2 before using the **distance** command to prevent the problem. This example shows Router R4 using the longer path through the OSPF domain on the left. Finally, Example 10-8 shows the configuration of the **distance** command and resulting solution.

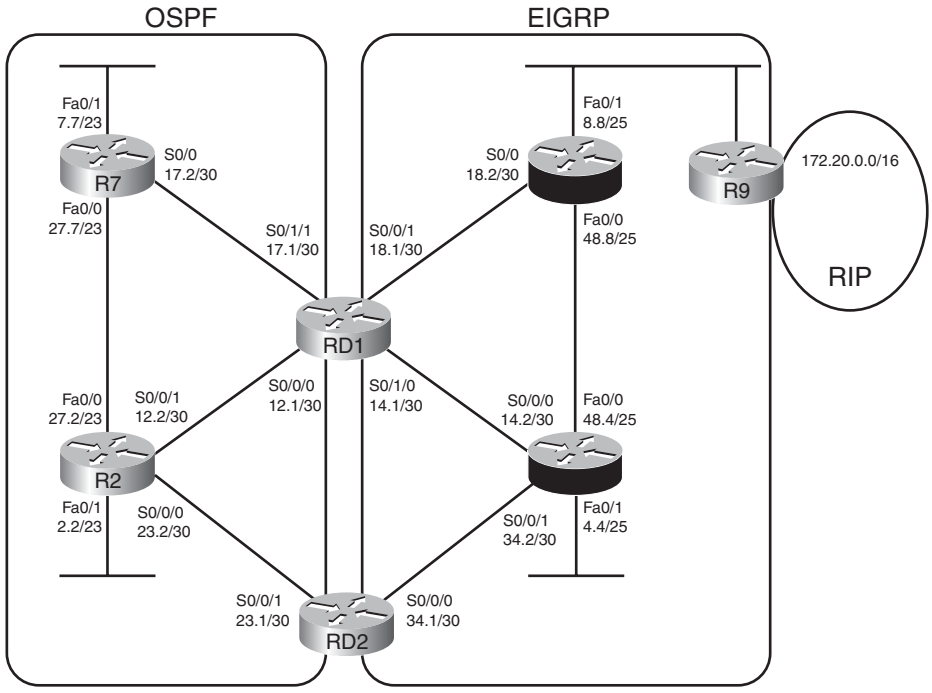


Figure 10-7 *Detailed View of Internetwork*

Example 10-7 *Long Route from RD2, into OSPF, for 172.20.0.0/16*

```
! The following is the routing protocol configuration on RD2
router eigrp 1
 redistribute ospf 2 metric 1000 200 255 1 1500
 network 172.16.0.0
 no auto-summary
!
router ospf 2
 router-id 3.3.3.3
 log-adjacency-changes
 redistribute eigrp 1 subnets
 network 172.30.0.0 0.0.255.255 area 0

! Next, the long route for 172.20.0.0/16 is listed. This route goes from
! RD2 back into the OSPF domain; interface S0/0/1 connects to router R2.
RD2#show ip route ¦ include 172.20.0.0
O E2 172.20.0.0/16 [110/20] via 172.30.23.2, 00:06:57, Serial0/0/1


! Next, the source of this routing information is listed under the
! text "Known via". RD2's current route is learned by OSPF.
RD2#show ip route 172.20.0.0
Routing entry for 172.20.0.0/16
  Known via "ospf 2", distance 110, metric 20, type extern 2, forward metric 128
  Redistributing via eigrp 1
  Advertised by eigrp 1 metric 1000 200 255 1 1500
  Last update from 172.30.23.2 on Serial0/0/1, 00:07:04 ago
  Routing Descriptor Blocks:
  * 172.30.23.2, from 1.1.1.1, 00:07:04 ago, via Serial0/0/1
      Route metric is 20, traffic share count is 1

! RD2 does know a working (successor) route for the same prefix,
! but prefers the lower-AD route (110) through OSPF.
RD2#show ip eigrp topology ¦ section 172.20.0.0
P 172.20.0.0/16, 1 successors, FD is 2611200
        via Redistributed (2611200/0)
```

The comments inside Example 10-7 detail the current state, with the longer route, as shown in Figure 10-6. Most important, note the "Known via..." text in the output of the **show ip route 172.20.0.0** command. This output specifically states the source of the route that is currently in the routing table.

Next, Example 10-8 shows the configuration on RD2 to solve this problem by setting RD2's AD for that specific route and additional **show** commands.

Example 10-8 *Configuring Per-Route AD on Router RD2*

```
RD2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
RD2(config)#router ospf 2
RD2(config-router)# distance 171 1.1.1.1 0.0.0.0 match-172-20
RD2(config-router)#!
RD2(config-router)#ip access-list standard match-172-20
RD2(config-std-nacl)# permit host 172.20.0.0
RD2(config-std-nacl)#^Z
RD2#

! Now the best route for 172.20.0.0 is known from EIGRP 1.
RD2#show ip route 172.20.0.0
Routing entry for 172.20.0.0/16
  Known via "eigrp 1", distance 170, metric 3635200, type external
  Redistributing via ospf 2, eigrp 1
  Advertised by ospf 2 subnets
  Last update from 172.16.34.2 on Serial0/0/0, 00:08:01 ago
! lines omitted for brevity

! The next command lists the matching logic of the distance command.
RD2#show ip protocols ¦ section ospf
Routing Protocol is "ospf 2"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.30.23.1
  It is an autonomous system boundary router
  Redistributing External Routes from,
    eigrp 1, includes subnets in redistribution
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.30.0.0 0.0.255.255 area 0
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1              171      00:00:35
    2.2.2.2              110      00:00:35
    7.7.7.7              110      00:00:35
  Distance: (default is 110)
    Address         Wild mask        Distance  List
    1.1.1.1         0.0.0.0          171       match-172-20
  Redistributing: ospf 2, eigrp 1
```

The configuration, although short, has one possibly counterintuitive twist. The IP address of the neighboring router, referenced in the **distance** command in OSPF configuration mode, will be compared to the *OSPF RID of the OSPF router that owns the LSA*. In this case, Router RD1 creates the Type 5 LSA for 172.20.0.0, and RD1's RID happens to be 1.1.1.1. RD2's **distance 171 1.1.1.1 0.0.0.0 match-172-20** command tells OSPF to look for LSAs owned by exactly RID 1.1.1.1, and if the prefix is permitted by the match-172-20 ACL, apply AD 171 to this route.

The **show ip route 172.20.0.0** command verifies that Router RD1 now prefers its AD 170 EIGRP route for 172.20.0.0/16. The highlighted portions of this command now refers to routing source EIGRP 1, with the outgoing interface of S0/0/0, which connects RD2 into the EIGRP domain. Because RD2 no longer has an OSPF route for 172.20.0.0/16, RD2 will not redistribute such an OSPF route back into EIGRP, defeating the domain loop problem.

Note: A complete solution requires all redistributing routers to perform this kind of configuration, for all such routes from the third routing domain.

Although this example shows the OSPF version of the **distance** command, one notable difference exists between the OSPF version and the RIP and EIGRP **distance** commands. When used as a RIP or EIGRP subcommand, the **distance** command matches the interface IP address of the neighboring router that advertises the route.

Preventing Domain Loops by Filtering on Subnet While Redistributing

The next tool prevents domain loops by filtering the routes based on prefix. Figure 10-8 shows the idea from a redistribution design perspective.

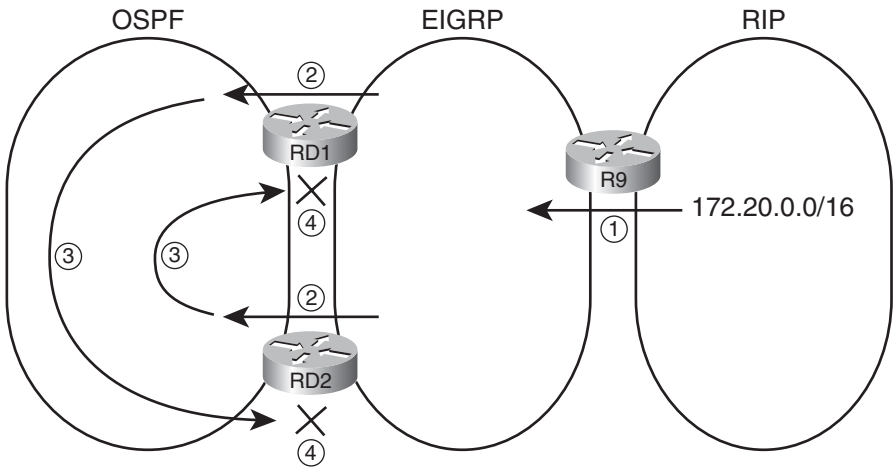


Figure 10-8 *Preventing Domain Loops with Route Filtering*

Following are the steps as listed in the figure:

Step 1. Router R9 advertises a route for network 172.20.0.0/16 from the RIP domain into the EIGRP domain.

Step 2. Routers RD1 and RD2 both redistribute this EIGRP external route into OSPF.

Step 3. Both RD1 and RD2 flood the route advertisement for the OSPF external route throughout the OSPF domain.

Step 4. Both RD1 and RD2 apply a route-map to their redistribution from OSPF into EIGRP, filtering routes with prefix 172.20.0.0.

The configuration itself uses the same methods and commands as included earlier in the section "Filtering Redistributed Routes with Route Maps."

Interestingly, this design does prevent the long routes, as shown earlier in Figure 10-6, but it does leave the possibility of a long route on a redistributing router. For example, if using all default AD settings, RD2 still learns an OSPF (default AD 110) route for 172.20.0.0 from RD1, so it may choose as best route the OSPF route through RD1 as the best route. Setting the AD for OSPF external routes to something larger than EIGRP's external AD of 170 would prevent this particular problem as well.

Preventing Domain Loops by Filtering on route-tag Using Distribute Lists

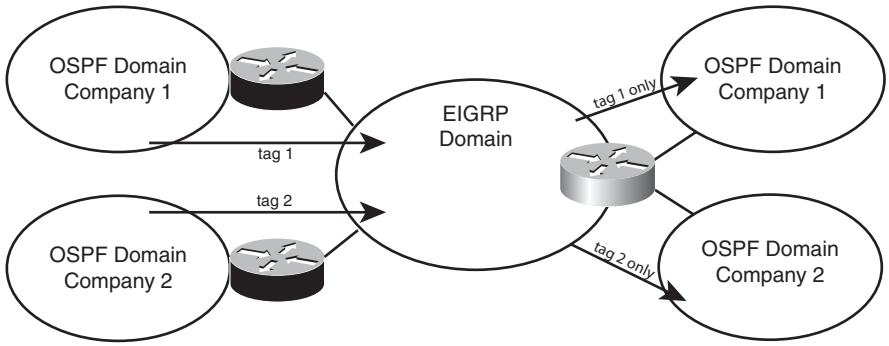
Route tags, the last tool shown in this chapter for preventing the domain loop problem, has a much broader use than just preventing redistribution problems.

A route tag is a unitless 32-bit integer that most routing protocols can assign to any given route. The assignment of a tag occurs when some IOS function adds the tag–for instance, it can be assigned by a route-map referenced by a routing protocol **distribute-list** or **redistribute** command. That tag follows the route advertisement, even through the redistribution process. At some later point in the flooding of routing information, other IOS tools, typically other route-maps, can match routes with a given route tag to make a decision.

In some cases, the idea of a route tag creates a mental block because it has no one specific purpose. The network engineer chooses the purpose of any particular route tag; the purpose has not been predetermined by a particular protocol. The folks that created the routing protocol provided us all with a nice, convenient place to add the equivalent of a post-it note to each route; it's up to us to decide what the note means.

Figure 10-9 shows one common use of route tags other than for solving the domain loop problem. In the figure, one large company that uses EIGRP (the middle of the figure) bought two smaller companies, both of whom use OSPF. The larger company wants to connect both small companies into the larger network, but they want to prevent hosts in the two smaller companies from knowing routes to the other smaller company. The figure shows only left-to-right advertisements of routes to reduce the clutter.

The two routers on the left each redistribute routes from the smaller companies into the EIGRP. The routers apply a route tag of 1 to each route from OSPF domain 1, and a tag of 2 to routes redistributed from OSPF domain 2. The actual numbers do not matter, as long as they are unique. On the right, the routers know that the routes from OSPF domain 1



Key Topic

Figure 10-9 *Using Route Tags to Determine Routing Domain Origin*

have route tag 1, and only these routes should be redistributed into the other part of OSPF domain 1. So, when redistributing into OSPF domain 1, the route-map makes a comparison of the route tag (command **match tag 1**) and allows only those routes. Similarly, when redistributing into OSPF domain 2, the **match tag 2** command would be used, redistributing only routes with tag 2.

To use route tags to prevent domain loop problems, you can use the following strategy:

Key Topic

- Choose and set a tag value that identifies routes taken from domain X and advertised into domain Y.
- When redistributing in the opposite direction (from domain Y into domain X), match the tag value and filter routes with that tag.

For example, consider the case shown in Figure 10-10. The figure shows the usual RD1 and RD2 between two routing domains, with EIGRP on the right in this case and OSPF on the left. The engineer then planned to use route tag 11 to mean "routes taken from EIGRP and redistributed into OSPF." The figure shows one direction of potential loops: From EIGRP through RD1, through OSPF, and back to EIGRP through RD2. However, the same concept would apply to the other direction.

The first step (noted with a circled 1 in the figure) is the usual redistribution, but with a route-map that tags all routes redistributed from EIGRP into OSPF with tag 11. RD2 learns these routes with OSPF. At Step 2, RD2 tries to redistribute the routes but chooses to filter all routes that have a tag value of 11. As a result, none of the routes learned from EIGRP are re-advertised back into EIGRP. Example 10-9 shows configuration that matches Figure 10-10.

Example 10-9 *RD1 and RD2 Configuration with Route Tags to Prevent Domain Loops*

```
! The following is the routing protocol configuration on RD1
router ospf 2
 router-id 3.3.3.3
 log-adjacency-changes
 redistribute eigrp 1 subnets route-map set-tag-11
 network 172.30.0.0 0.0.255.255 area 0
```

```
!
route-map set-tag-11 permit 10
 set tag 11
! The following is the routing protocol configuration on RD2
router eigrp 1
 redistribute ospf 2 metric 1000 200 255 1 1500 route-map stop-tag-11
 network 172.16.0.0
 no auto-summary
!
route-map stop-tag-11 deny 10
 match tag 11
!
route-map stop-tag-11 permit 20
```

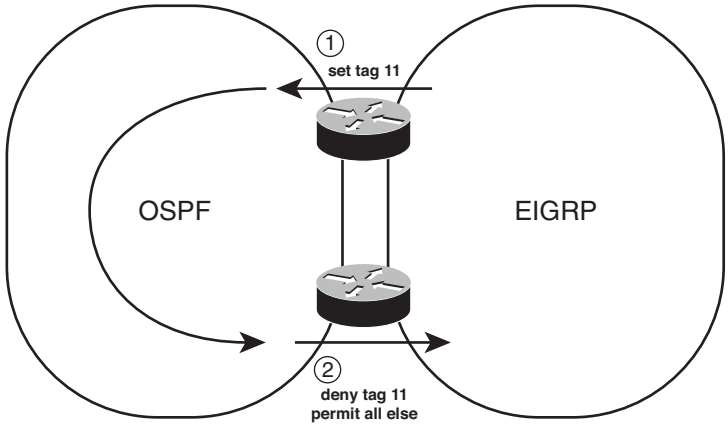


Figure 10-10 *Using Route Tags to Prevent Domain Loop Problems*

Key Topic

First, note that the configuration does rely on a couple of default route-map actions that bear some review. In the set-tag-11 route-map, only one route-map clause exists, and that clause has no **match** commands. A route-map clause with no **match** commands matches all routes, so all routes are assigned tag 11. In the stop-tag-11 route-map, the first clause lists a **deny** action, meaning that all routes matched by that clause (all with tag 11) are filtered. All other routes, for example those routes for subnets native to the OSPF domain, match the second clause (line number 20), because that second clause does not have a **match** command.

Example 10-9 shows the configuration that tags routes coming from EIGRP into OSPF and then filters routes with that same tag as they go from OSPF into EIGRP. For a complete solution, the reverse case would also need to be configured, using a different route tag value.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 10-8 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? For any configuration items, a general description can be used, without concern about the specific parameters.

Table 10-8 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Filter routes when redistributing. (2)	
Set different metrics for different routes redistributed from one routing source.	
Set some OSPF routes as E1, and some as E2, when redistributed from one routing source.	
The design shows multiple redistribution points with two routing domains, with a need to prevent domain loops. (3)	
The design shows multiple redistribution points with more than two routing domains, and a need to prevent domain loops. (2)	

Implementation Plan Peer Review Table

Table 10-9 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 10-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A design shows an OSPF and EIGRP routing domain, with multiple redistributing routers, with no obvious configuration to prevent routing domain loops. What default AD values exist, and do they prevent any problems?	
The same question as the previous row, except with RIP and OSPF domains.	
The same question as the previous row, except with RIP and EIGRP domains.	
A plan shows redistribution between EIGRP and OSPF on two routers. The configuration for OSPF on one router lists **redistribute eigrp 1 subnets** and **distribute-list 1 out**. Will this configuration attempt to filter routes? Is a route-map option required to filter when redistributing?	
A partially complete plan shows three different routing domains, with multiple redistribution points between each pair of routing domains. The configuration shows large ACLs matching various subnets and setting AD per-route using the **distance** command. What alternative method might be easier to maintain as the network changes?	

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 10-10 configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 10-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Filtering routes on redistribution from OSPF into EIGRP, allowing only routes permitted by ACL 1 (2 methods).	
Filtering routes on redistribution from EIGRP into OSPF, allowing only routes permitted by prefix list barney (2 methods).	
Configuring the route map that will set metric components to 1000, 200, 255, 1, and 150, for routes permitted by ACL 1, and filter all other routes.	
Set OSPF's administrative distance for all internal routes to 110, and all external routes to 180.	

Table 10-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Set EIGRP's administrative distance for routes learned from neighbor 1.1.1.1 to 190, only for subnets between 10.1.0.0 – 10.1.255.255.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 10-11 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 10-11 *Verification Plan Memory Drill*

Information Needed	Command
Confirm that OSPF routes were redistributed from the IP routing table into that same router's EIGRP topology table.	
Display the number of matches in an ACL used for redistribution filtering.	
Display the number of matches in an IP prefix list used for redistribution filtering.	
Display the configuration of a route map.	
Display the component metrics of a route redistributed into EIGRP.	
Confirm the absence or presence of a route that could have been redistributed from OSPF into EIGRP.	
Confirm the absence or presence of a route that could have been redistributed from EIGRP into OSPF.	
Display an IP route's administrative distance.	
Display the administrative distance settings for EIGRP.	
Display the administrative distance settings for OSPF.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 10-12 lists a reference of these key topics and the page numbers on which each is found.

Table 10-12 *Key Topics for Chapter 10*

Key Topic Element	Description	Page Number
List	A summary of functions that can be performed by a route-map referenced by a **redistribute** command	332
Table 10-2	The **match** route-map subcommands applicable to OSPF and EIGRP redistribution	333
Table 10-3	The **set** route-map subcommands applicable to OSPF and EIGRP redistribution	334
Table 10-6	Default administrative distance settings in IOS	346
Figure 10-4	Conceptual view of how default EIGRP administrative distance prevents domain loops–EIGRP internal versus OSPF external	347
Figure 10-9	Example use of route tags for purpose other than preventing domain loops	356
List	Recommendations for how to use route tags to prevent the domain loop problem	356
Figure 10-10	Example of how to prevent domain loops using route tags	357

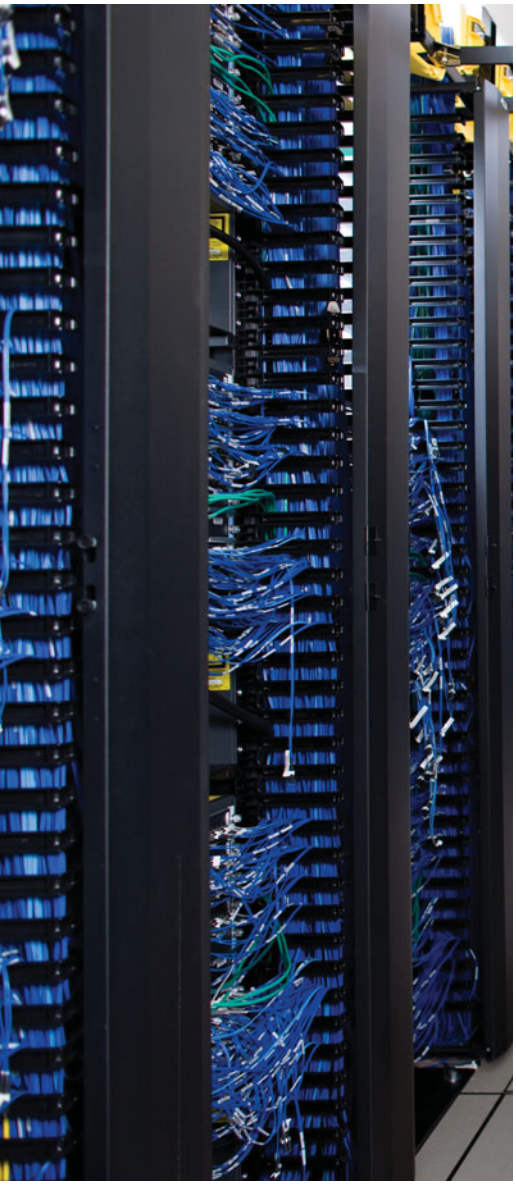
Complete the Tables and Lists from Memory

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

Define Key Terms

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

Domain loop, administrative distance, route tag



This chapter covers the following subjects:

Policy-Based Routing: This section describes the IOS Policy-Based Routing (PBR) feature, which allows a router to make packet forwarding decisions on criteria other than the packet's destination address as matched with the IP routing table.

IP Service Level Agreement: This section gives a general description of the IP Service Level Agreement (IP SLA) feature, with particular attention to how it can be used to influence when a router uses a static route and when a router uses PBR.

CHAPTER 11

Policy-Based Routing and IP Service Level Agreement

The term *path control* has many fine shades of meaning, depending on the context. The broadest typical use of the term refers to any and every function that influences where a router forwards a packet. With that definition, path control includes practically every topic in this book. In other cases, the term *path control* refers to tools that influence the contents of the routing table, most typically referring to routing protocols.

This chapter examines two path control topics that fit only into the broader definition of the term. The first, Policy-Based Routing (PBR), also sometimes called simply Policy Routing, influences the IP data plane, changing the forwarding decision a router makes, but without first changing the IP routing table. The second tool, IP Service-Level Agreement (SLA), monitors network health and reachability. The router can then choose when to use routes, and when to ignore routes, based on the status determined by IP SLA.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these six self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 11-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 11-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Policy-Based Routing	1-3
IP Service-Level Agreement	4-6

1. Policy-Based Routing (PBR) has been enabled on Router R1's interface F0/0. Which of the following is true regarding how PBR works? (Choose two.)

   a. Packets entering F0/0 will be compared based on the PBR route map.

   b. Packets exiting F0/0 will be compared based on the PBR route map.

   c. IOS ignores the PBR forwarding directions when the packet matches a route map **deny** clause.

d. IOS ignores the PBR forwarding directions when the packet matches a route map **permit** clause.

2. Examine the following configuration on Router R1. R1's **show ip route 172.16.4.1** command lists a route with outgoing interface S0/1/1. Host 172.16.3.3 uses telnet to connect to host 172.16.4.1. What will Router R1 do with the packets generated by host 172.16.3.3 because of the telnet, assuming the packets enter R1's F0/0 interface? (Choose two.)

```
interface Fastethernet 0/0
 ip address 172.16.1.1 255.255.255.0
 ip policy route-map Q2
!
route-map Q2 permit
 match ip address 101
 set interface s0/0/1
!
access-list 101 permit tcp host 172.16.3.3 172.16.4.0 0.0.0.255
```

a. The packet will be forwarded out S0/0/1, or not at all.

b. The packet will be forwarded out S0/0/1 if it is up.

c. The packet will be forwarded out S0/1/1 if it is up.

d. The packet will be forwarded out S0/1/1 if it is up, or if it is not up, out s0/0/1.

e. The packet will be forwarded out S0/0/1 if it is up, or if it is not up, out s0/1/1.

3. The following output occurs on Router R2. Which of the following statements can be confirmed as true based on the output from R2?

```
R2# show ip policy
Interface       Route map
Fa0/0           RM1
Fa0/1           RM2
S0/0/0          RM3
```

a. R2 will forward all packets that enter Fa0/0 per the PBR configuration.

b. R2 will use route map RM2 when determining how to forward packets that exit interface Fa0/1.

c. R2 will consider using PBR for all packets exiting S0/0/0 per route map RM3.

d. R2 will consider using PBR for all packets entering S0/0/0 per route map RM3.

4. Which of the following are examples of traffic that can be created as part of an IP Service-Level Agreement operation? (Choose two.)

a. ICMP Echo

b. VoIP (RTP)

c. IPX

d. SNMP

5. The following configuration commands exist only in an implementation plan document. An engineer does a copy/paste of these commands into configuration mode on Router R1. Which of the following answers is most accurate regarding the results?

```
ip sla 1
icmp-echo 1.1.1.1 source-ip 2.2.2.2
ip sla schedule 1 start-time now life forever
```

   a. The SLA operation will be configured but will not start until additional commands are used.
   b. The SLA operation is not completely configured so it will not collect any data.
   c. The SLA operation is complete and working, collecting data into the RTTMON MIB.
   d. The SLA operation is complete and working but will not store the data in the RTTMON MIB without more configuration.

6. The following output occurs on Router R1. IP SLA operation 1 uses an ICMP echo operation type, with default frequency of 60 seconds. The operation pings from address 1.1.1.1 to address 2.2.2.2. Which of the following answers is true regarding IP SLA and object tracking on R1?

```
R1# show track
Track 2
  IP SLA 1 state
  State is Up
    3 changes, last change 00:00:03
  Delay up 45 secs, down 55 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 6
  Tracked by:
    STATIC-IP-ROUTING 0
```

   a. The tracking return code fails after the SLA operation results in an ICMP echo failure three times.
   b. The tracking return code fails after the SLA operation results in an ICMP echo failure one time.
   c. After the tracking object fails, the tracking object moves back to an up state 45 seconds later in all cases.
   d. After moving to a down state, the tracking object moves back to an OK state 45 seconds after the SLA operation moves to an OK state.

Foundation Topics

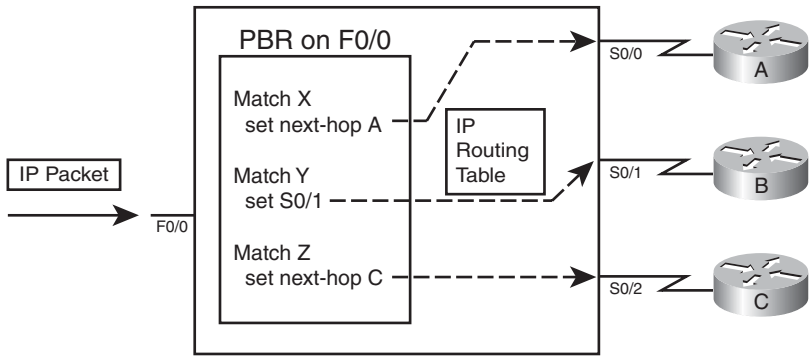
Policy-Based Routing

When a packet arrives at the incoming interface of a router, the router's data plane processing logic takes several steps to process the packet. The incoming packet actually arrives encapsulated inside a data link layer frame, so the router must check the incoming frame's Frame Check Sequence (FCS) and discard the frame if errors occurred in transmission. If the FCS check passes, the router discards the incoming frame's data link header and trailer, leaving the Layer 3 packet. Finally, the router does the equivalent of comparing the destination IP address of the packet with the IP routing table, matching the longest-prefix route that matches the destination IP address.

Note: Most routers today default to use Cisco Express Forwarding (CEF), which causes the router to match the destination of IP packets with the CEF table, instead of the IP routing table. IOS derives the CEF table from the information in the IP routing table, with much faster table lookup as compared with using the routing table directly.

Policy-Based Routing (PBR) overrides the router's natural destination-based forwarding logic. PBR intercepts the packet after de-encapsulation on the incoming interface, before the router performs the CEF table lookup. PBR then chooses how to forward the packet using criteria other than the usual matching of the packet's destination address with the CEF table.

PBR chooses how to forward the packet by using matching logic defined through a route map, which in turn typically refers to an IP ACL. That same route map also defines the forwarding instructions–the next-hop IP address or outgoing interface–for packets matched by the route map. Figure 11-1 shows the general concept, with PBR on interface F0/0 overriding the usual routing logic, forwarding packets out three different outgoing interfaces.



Key Topic

Figure 11-1 *PBR Concepts*

To perform the actions shown in Figure 11-1, the engineer configures two general steps:

Step 1. Create a route map with the logic to match packets, and choose the route, as shown on the left side of the figure.

Step 2. Enable the route map for use with PBR, on an interface, for packets entering the interface.

The rest of this section focuses on the configuration and verification of PBR.

Matching the Packet and Setting the Route

To match packets with a route map enabled for PBR, you use the familiar route-map **match** command. However, you have two **match** command options to use:

```
match ip address
match length min max
```

The **match ip address** command uses the same familiar logic as seen in several other chapters of this book. This command can reference standard and extended ACLs. Any item matchable by an ACL can be matched in the route map. The **match length** command allows you to specify a range of lengths, in bytes.

When a route map clause (with a permit action) matches a packet, the **set** command defines the action to take regarding how to forward the packet. The four **set** command options define either the outgoing interface or the next-hop IP address, just like routes in the IP routing table. Table 11-2 lists the options, with some explanations.

Table 11-2 *Choosing Routes Using the PBR* **set** *Command*

Command	Comments
set ip next-hop *ip-address*[. . . *ip-address*]	Next-hop addresses must be in a connected subnet; forwards to the first address in the list for which the associated interface is up.
set ip default next-hop *ip-address*[. . . *ip-address*]	Same logic as previous command, except policy routing first attempts to route based on the routing table.
set interface *interface-type interface-number* [. . . *interface-type interface-number*]	Forwards packets using the first interface in the list that is up.
set default interface *interface-type interface-number* [. . . *interface-type interface-number*]	Same logic as previous command, except policy routing first attempts to route based on the routing table.

Note that two of the commands allow the definition of a next-hop router, and two allow the definition of an outgoing interface. The other difference in the commands relates to whether the command includes the **default** keyword. The section "How the **default** Keyword Impacts Logic Ordering" later in this chapter describes the meaning of the **default** keyword.

After the route map has been configured with all the clauses to match packets, and set their outgoing interface or next-hop address, the only remaining step requires the **ip policy route-map** *name* command to enable PBR for packets entering an interface.

PBR Configuration Example

To tie the concepts together, Figure 11-2 shows a sample internetwork to use in a PBR example. In this case, EIGRP on R1 chooses the upper route to reach the subnets on the right, because of the higher bandwidth on the upper link (T1) as compared with the lower link (64 Kbps).

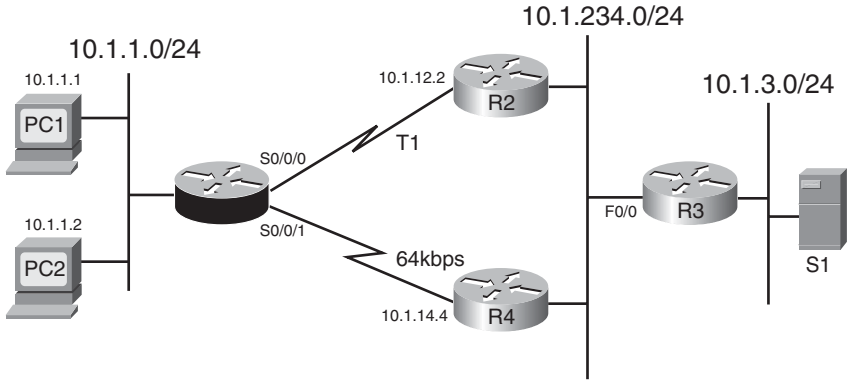


Figure 11-2 *Network Used in PBR Example*

For this example, the PBR configuration matches packets sent from PC2 on the left to server S1 in subnet 10.1.3.0/24 on the right. PBR on R1 routes these packets out S0/0/1 to R4. These packets will be routed over the lower path–out R1's S0/0/1 to R4–instead of through the current through R2, as listed in R1's IP routing table.

Example 11-1 *R1 PBR Configuration*

```
interface Fastethernet 0/0
 ip address 10.1.1.9 255.255.255.0
 ip policy route-map PC2-over-low-route
!
route-map PC2-over-low-route permit
 match ip address 101
 set ip next-hop 10.1.14.4
!
access-list 101 permit ip host 10.1.1.2 10.1.3.0 0.0.0.255
```

The configuration enables PBR with F0/0's **ip policy route-map PC2-over-low-route** command. The referenced route map matches packets that match ACL 101; ACL 101 matches packets from PC2 only, going to subnet 10.1.3.0/24. The route-map clause uses a permit action, which tells IOS to indeed apply PBR logic to these matched packets. (Had the

route-map command listed a deny action, IOS would simply route the packet as normal–it would not filter the packet.) Finally, for packets matched with a permit action, the router forwards the packet based on the **set ip next-hop 10.1.14.4** command, which tells R1 to forward the packet to R4 next.

Note that for each packet entering F0/0, PBR either matches the packet with a route map permit clause, or matches the packet with a route map deny clause. All route maps have an implicit deny clause at the end that matches all packets not already matched by the route map. PBR processes packets that match a permit clause using the defined **set** command. For packets matched by a deny clause, PBR lets the packet go through to the normal IP routing process.

To verify the results of the policy routing, Example 11-2 shows two **traceroute** commands: one from PC1 and one from PC2. Each shows the different paths. (Note that the output actually comes from a couple of routers configured to act as hosts PC1 and PC2 for this example.)

Example 11-2 *Confirming PBR Results Using* **traceroute**

```
! First, from PC1 (actually, a router acting as PC1):
PC1# trace 10.1.3.99
Type escape sequence to abort.
Tracing the route to 10.1.3.99

  1 10.1.1.9 4 msec 0 msec 4 msec
  2 10.1.12.2 0 msec 4 msec 4 msec
  3 10.1.234.3 0 msec 4 msec 4 msec
  4 10.1.3.99 0 msec *  0 msec
```

```
! Next, from PC2
PC2# trace 10.1.3.99

Type escape sequence to abort.
Tracing the route to 10.1.3.99

  1 10.1.1.9 4 msec 0 msec 4 msec
  2 10.1.14.4 8 msec 4 msec 8 msec
  3 10.1.234.3 8 msec 8 msec 4 msec
  4 10.1.3.99 4 msec *  4 msec
```

The output differs only in the second router in the end-to-end path–R2's 10.1.12.2 address as seen for PC1's packet and 10.1.14.4 as seen for PC2's packet.

The verification commands on the router doing the PBR function list relatively sparse information. The **show ip policy** command just shows the interfaces on which PBR is enabled and the route map used. The **show ip route-map** command shows overall statistics for the number of packets matching the route map for PBR purposes. The only way to verify the types of packets that are policy routed is to use the **debug ip policy** command, which can be dangerous on production routers, given its multiple lines of output per

packet, or to use **traceroute**. Example 11-3 lists the output of the **show** and **debug** commands on Router R1 with the **debug** output being for a single policy routed packet.

Example 11-3 *Verifying PBR on Router R1*

```
R1# show ip policy
Interface       Route map
Fa0/0           PC2-over-low-route

R1# show route-map
route-map PC2-over-low-route, permit, sequence 10
  Match clauses:
    ip address (access-lists): 101
  Set clauses:
    ip next-hop 10.1.14.4
  Policy routing matches: 12 packets, 720 bytes
R1# debug ip policy
*Sep 14 16:57:51.675: IP: s=10.1.1.2 (FastEthernet0/0), d=10.1.3.99, len 28,
policy match
*Sep 14 16:57:51.675: IP: route map PC2-over-low-route, item 10, permit
*Sep 14 16:57:51.675: IP: s=10.1.1.2 (FastEthernet0/0), d=10.1.3.99 (Serial0/0/1),
len 28, policy routed
*Sep 14 16:57:51.675: IP: FastEthernet0/0 to Serial0/0/1 10.1.14.4
```

How the **default** Keyword Impacts PBR Logic Ordering

The example in the previous section showed a **set** command that did not use the **default** keyword. However, the inclusion or omission of this keyword significantly impacts how PBR works. This parameter in effect tells IOS whether to apply PBR logic *before* trying to use normal destination-based routing, or whether to first try to use the normal destination-based routing, relying on PBR's logic only if the destination-based routing logic fails to match a nondefault route.

First, consider the case in which the **set** command omits the **default** parameter. When IOS matches the associated PBR route map permit clause, IOS applies the PBR logic first. If the **set** command identifies an outgoing interface that is up, or a next-hop router that is reachable, IOS uses the PBR-defined route. However, if the PBR route (as defined in the **set** command) is not working–because the outgoing interface is down or the next hop is unreachable using a connected route–then IOS next tries to route the packet using the normal destination-based IP routing process.

Next, consider the case in which the **set** command includes the **default** parameter. When IOS matches the associated PBR route map permit clause, IOS applies the normal destination-based routing logic first, with one small exception: it ignores any default routes. So, the router first tries to route the packet as normal, but if no nondefault route matches the packet's destination address, then the router forwards the packet as directed in the **set** command.

For example, for the configuration shown in Example 11-1, by changing the **set** command to **set ip default next-hop 10.1.14.4**, R1 would have first looked for (and found) a working route through R2, and forwarded packets sent by PC2 over the link to R2. Summarizing:

- Omitting the **default** parameter gives you logic like this: "Try PBR first, and if PBR's route does not work, try to route as usual."
- Including the **default** parameter gives you logic like this: "Try to route as usual while ignoring any default routes, but if normal routing fails, use PBR."

Additional PBR Functions

Primarily, PBR routes packets received on an interface, but using logic other than matching the destination IP address and the CEF table. This section briefly examines three additional PBR functions.

Applying PBR to Locally Created Packets

In some cases, it may be useful to use PBR to process packets generated by the router itself. However, PBR normally processes packets that enter the interface(s) on which the **ip policy route-map** command has been configured, and packets generated by the router itself do not actually enter the router through some interface. To make IOS process locally created packets using PBR logic, configure the **ip local policy route-map** *name* global command, referring to the PBR route map at the end of the command.

The section "Configuring and Verifying IP SLA" later in this chapter shows an example use of this command. IP SLA causes a router to create packets, so applying PBR to such packets can influence the path taken by the packets.

Setting IP Precedence

Quality of service (QoS) refers to the entire process of how the network infrastructure can choose to apply different levels of service to different packets. For example, a router may need to keep delay and jitter (delay variation) low for VoIP and Video over IP packets, because these interactive voice and video calls only work well when the delay and jitter are held very low. So, the router might let VoIP packets bypass a long queue of data packets that might be waiting to exit an interface, giving the voice packet better (lower) delay and jitter.

Most QoS designs mark each packet with a different value inside the IP header, for the purpose of identifying groups of packets–a service class–that should get a particular QoS treatment. For instance, all VoIP packets could be marked with a particular value so that the router can then find those marked bits, know that the packet is a VoIP packet due to that marking, and apply QoS accordingly.

Although the most commonly used QoS marking tool today is Class-Based Marking, in the past, PBR was one of the few tools that could be used for this important QoS function of marking packets. PBR still supports marking; however, most modern QoS designs ignore PBR's marking capabilities.

Before discussing PBR's marking features, a little background about the historical view of the IP header's type of service (ToS) byte is needed. The IP header originally defined a ToS

byte whose individual bits have been defined in a couple of ways over the years. One such definition used the first three bits in the ToS byte as a three-bit IP Precedence (IPP) field, which could be used for generic QoS marking, with the higher values generally implying a better QoS treatment. Back in the 1990s, the ToS byte was redefined as the Differentiated Services (DS) byte, with the first six bits defined as the Differentiated Service Code Point (DSCP). Most QoS implementations today revolve around setting the DSCP field.

PBR supports setting the older QoS marking fields—the IP Precedence (IPP) and the entire ToS byte—using commands **set ip precedence** *value* and **set ip tos** *value*, respectively, in the route map. To configure packet marking, configure PBR as normal, but add a **set** command that defines the field to be marked and the value.

PBR with IP SLA

Besides matching a packet's length, or matching a packet with an ACL, PBR can also react to some dynamic measurements of the health of the IP network. To do so, PBR relies on the IP Service-Level Agreement (IP SLA) tool. In short, if the IP SLA tool measures the network's current performance, and the performance does not meet the defined threshold, PBR chooses to not use a particular route. The second half of this chapter discusses IP SLA, with the section "Configuring and Verifying IP SLA" demonstrating how PBR works with IP SLA.

IP Service-Level Agreement

The IOS IP Service-Level Agreement (IP SLA) feature measures the ongoing behavior of the network. The measurement can be as simple as using the equivalent of a ping to determine if an IP address responds, or as sophisticated as measuring the jitter (delay variation) of VoIP packets that flow over a particular path. To use IP SLA, an engineer configures IP SLA operations on various routers, and the routers will then send packets, receive responses, and gather data about whether a response was received, and the specific characteristics of the results, such as delay and jitter measurements.

IP SLA primarily acts as a tool to test and gather data about the network. Network management tools can then collect that data and report whether the network reached SLAs for the network. Many network management tools support the ability to configure IP SLA from the management tools' graphical interfaces. When configured, the routers gather the results of the operations, storing the statistics in the IOS RTTMON MIB. Management applications can later gather the statistics from this MIB on various routers and report on whether the business SLAs were met based on the gathered statistics. For example, you can use the Cisco Works Internetwork Performance Monitor (IPM) feature to configure and monitor the data collected using IP SLA operations.

So, why bother with a pure network management feature in this book focused on IP routing? Well, you can configure static routes and PBR to use IP SLA operations, so that if the operation shows a failure of a particular measurement or reduced performance of the measurement below a configured threshold, the router stops using either the static route or PBR logic. This combination of features provides a means to control when the static and PBR paths are used and when they are ignored.

This section begins with discussion of IP SLA as an end to itself. Following that, the topic of SLA object tracking is added, along with how to configure static routes and PBR to track IP SLA operations so IOS knows when to use, and when to ignore, these routes.

Understanding IP SLA Concepts

IP SLA uses the concept of an *operation*. Each operation defines a type of packet that the router will generate, the destination and source address, and other characteristics of the packet. The configuration includes settings about the time-of-day when the router should be sending the packets in a particular operation, the types of statistics that should be gathered, and how often the router should send the packets. Also, you can configure a router with multiple operations of different types.

For example, a single IP SLA operation could define the following:

- Use ICMP echo packets.
- Measure the end-to-end round trip response time (ICMP echo).
- Send the packets every 5 minutes, all day long.

Note: For those of you who have been around IOS for a while, IP SLA may sound familiar. IP SLA has origins in earlier IOS features, including the *Response Time Reporter (RTR)* feature. The RTR feature uses the **rtr** command and uses the term *probe* to refer to what IP SLA refers to as an *operation*.

All the SLA operations rely on the router sending packets and some other device sending packets back. Figure 11-3 shows the general idea and provides a good backdrop to discuss some related issues.

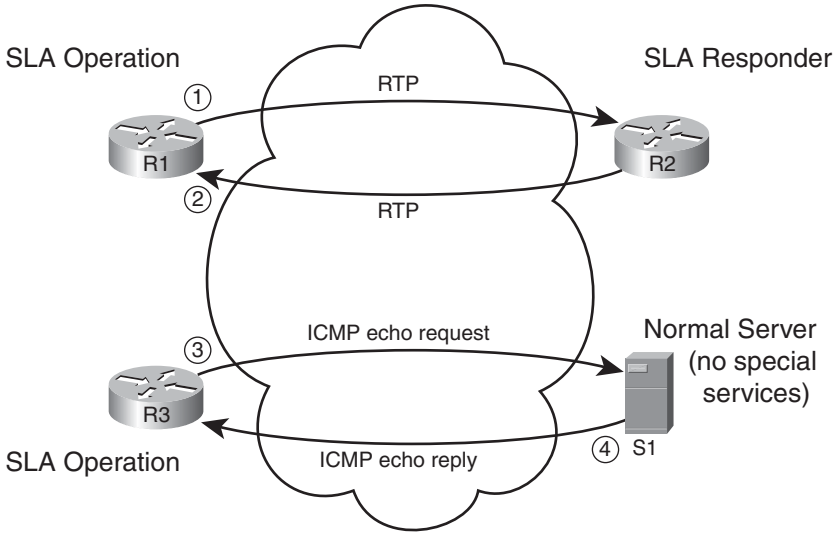


Figure 11-3 *Sending and Receiving Packets with IP SLA*

Key Topic

An IP SLA operation can cause the router to send packets to any IP address, whether on a router or a host. When sending to a host, as in the bottom part of the figure, the host does not need any special software or configuration–instead, the host just acts as normal. That means that if an SLA operation sends packets to a host, the router can only use operation types that send packets that the host understands. For instance, the router could use ICMP echo requests, TCP connection requests, or even HTTP GET requests to a web server, because the server should try to respond to these requests.

The operation can also send packets to another router, which gives IP SLA a wider range of possible operation types. If the operation sends packets to which the remote router would normally respond, like ICMP echo requests, the other router needs no special configuration. However, IP SLA supports the concept of the IP SLA responder, as noted in Figure 11-3 for R2. By configuring R2 as an IP SLA responder, it responds to packets that a router would not normally respond to, giving the network engineer a way to monitor network behavior without having to place devices around the network just to test the network.

For example, the operation could send Real-time Transport Protocol (RTP) packets–packets that have the same content as VoIP–as shown in Figure 11-3 as Step 1. Then the IP SLA responder function on R2 can reply back as if a voice call exists between the two routers, as shown in Step 2 of that figure.

A wide range of IP SLA operations exist. The following list summarizes the majority of the available operation types, just for perspective:

- ICMP (echo, jitter)
- RTP (VoIP)
- TCP connection (establishes TCP connections)
- UDP (echo, jitter)
- DNS
- DHCP
- HTTP
- FTP

Configuring and Verifying IP SLA

This book describes IP SLA configuration in enough depth to get a sense for how it can be used to influence static routes and policy routing. To that end, this section examines the use of an ICMP echo operation, which requires configuration only on one router, with no IP SLA responder. The remote host, router, or other device replies to the ICMP Echo Requests just like any other ICMP echo requests.

The general steps to configure an ICMP-based IP SLA operation are as follows:

Key Topic

Step 1. Create the IP SLA Operation, and assign it an integer operation number, using the **ip sla** *sla-ops-number* global configuration command.

Step 2. Define the operation type and the parameters for that operation type. For ICMP echo, you define the destination IP address or hostname, and optionally, the source IP address or hostname, using the **icmp-echo** {destination-ip-address | destination-hostname} [source-ip {ip-address | hostname} | source-interface interface-name] SLA operation subcommand.

Step 3. (Optional) Define a (nondefault) frequency that the operation should send the packets, in seconds, using the **frequency** *seconds* IP SLA subcommand.

Step 4. Schedule when the SLA will run, using the **ip sla schedule** *sla-ops-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*}] [**ageout** *seconds*] [**recurring**] global command.

Example 11-4 shows the process of configuring an ICMP Echo operation on Router R1 from Figure 11-2. The purpose of the operation is to test the PBR route through R4. In this case, the operation will be configured as shown in Figure 11-4, with the following criteria:

- Send ICMP Echo Requests to server S1 (10.1.3.99).
- Use source address 10.1.1.9 (R1's F0/0 IP address).
- Send these packets every 60 seconds.
- Start the operation immediately, and run it forever.
- Enable PBR for locally generated packets, matching the IP SLA operation with the PBR configuration so that the SLA operation's packets flow over the lower route.

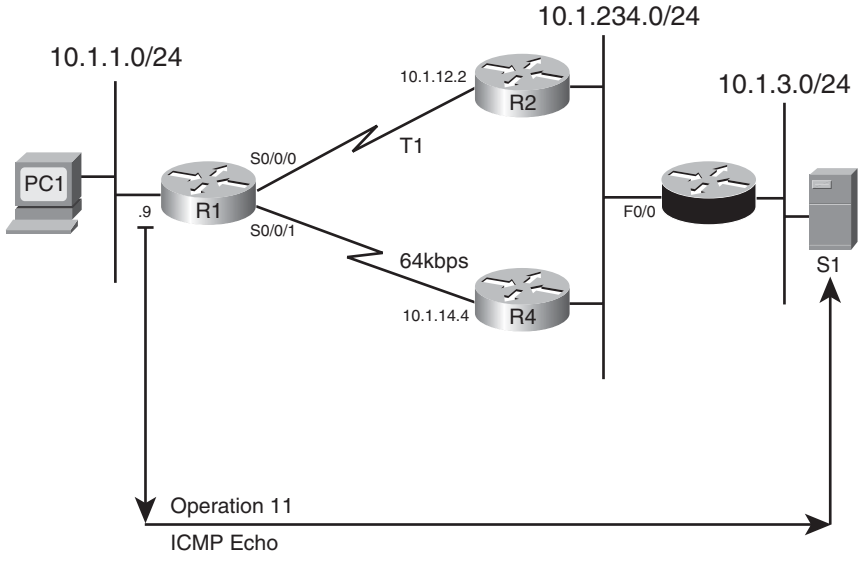


Figure 11-4 *Concept of IP SLA Operation on R1*

Example 11-4 *Configuring an ICMP Echo Operation on Router R1*

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip sla 11
R1(config-ip-sla)#icmp?
icmp-echo  icmp-jitter

R1(config-ip-sla)# icmp-echo 10.1.3.99 source-ip 10.1.1.9
R1(config-ip-sla)# frequency 60
R1(config-ip-sla)# exit
R1(config)# ip sla schedule 11 start-time now life forever

! Changes to the PBR configuration below
R1(config)# access-list 101 permit ip host 10.1.1.9 host 10.1.3.99
R1(config)# ip local policy PC2-over-low-route
R1(config)# end
```

First, focus on the pure IP SLA configuration, located from the beginning of the example up through command **ip sla schedule**. The configuration creates IP SLA operation 11. The parameters on the **icmp-echo** command act as if you used an extended ping from the command line, specifying both the source and destination IP address. The last command directly related to IP SLA, the **ip sla schedule** command, enables the operation now, and runs the operation until the network engineer takes some action to disable it, in some cases by removing the operation with the **no ip sla** *sla-ops-number* command.

The last two commands in the example show a change to the earlier PBR configuration so that the SLA operation's packets flow over the lower route. The **ip local policy PC2-over-low-route** global command tells R1 to process packets generated by R1, including the IP SLA operation packets, using PBR. The addition of the **access-list 101** command to the configuration shown earlier in Example 11-1 makes the route map match the source and destination address of the SLA operation. That former route map's **set** command sent the packets over the link to R4.

IP SLA supports a couple of particularly useful verification commands: **show ip sla configuration** and **show ip sla statistics**. The first command confirms all the configuration settings for the operation, and the second lists the current statistics for the operation. Example 11-5 shows examples of each on R1, after the configuration shown in Example 11-4.

Example 11-5 *Verification of an IP SLA Operation*

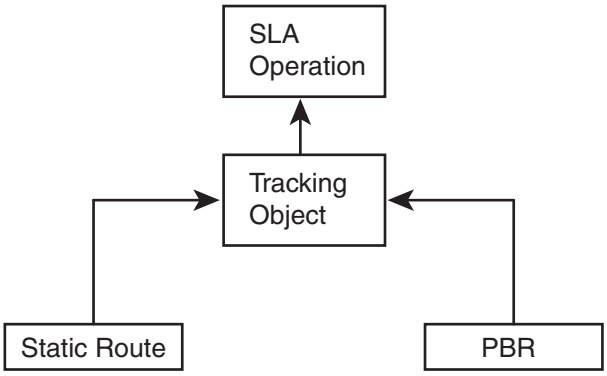
```
R1# show ip sla configuration
IP SLAs Infrastructure Engine-II
Entry number: 11
Owner:
Tag:
Type of operation to perform: echo
Target address/Source address: 10.1.3.99/10.1.1.9
```

```
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:

R1# show ip sla statistics 11
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
        Latest RTT: 8 milliseconds
Latest operation start time: *19:58:08.395 UTC Mon Sep 14 2009
Latest operation return code: OK
Number of successes: 22
Number of failures: 0
Operation time to live: Forever
```

The highlighted lines in the output of the **show ip sla configuration** command correspond to the values explicitly configured in Example 11-4. The more interesting output exists in the output of the **show ip sla statistics 11** command, which lists the statistics only for operation 11. In this case, 22 intervals have passed, showing 22 ICMP Echo Requests as successful with no failures. It also lists the latest round trip time (RTT). Finally, it lists the return code of the most recent operation (OK in this case)–a key value used by SLA tracking.

Tracking SLA Operations to Influence Routing

As previously mentioned, you can configure both static routes and PBR to be used only when an SLA operation remains successful. The configuration to achieve this logic requires the configuration of a tracking object and cross references between the static route, PBR, and IP SLA, as shown in Figure 11-5.



Key Topic **Figure 11-5** *Configuration Relationships for Path Control Using IP SLA*

The tracking object looks at the IP SLA operation's most recent return code to then determine the tracking state as either "up" or "down." Depending on the type of SLA operation, the return code may be a simple toggle, with "OK" meaning that the last operation worked. The tracking object would then result in an "up" state if the SLA operation resulting in an "OK" return code. Other SLA operations that define thresholds have more possible return codes. The tracking operation results in an "up" state if the IP SLA operation is within the configured threshold.

One of the main reasons that IOS requires the use of this tracking object is so that the routes do not flap. Route flapping occurs when a router adds a route to the routing table, then quickly removes it; conditions change, so the router soon adds the route to the table again; and so on. If a static route tracked the IP SLA object directly, the SLA object's return code could change each time the operation ran, causing a route flap. The tracking object concept provides the ability to set a delay of how soon after a tracking state change the tracking object should change state. This feature gives the engineer a tool to control route flaps.

This section shows how to configure a tracking object for use with both a static route and with PBR.

Configuring a Static Route to Track an IP SLA Operation

To configure a static route to track an IP SLA, you need to configure the tracking object and then configure the static route with the **track** keyword. To do so, use these steps:

Key Topic **Step 1.** Use the **track** *object-number* **ip sla** *sla-ops-number* **[state | reachability]** global command.

Step 2. (Optional) Configure the delay to regulate flapping of the tracking state by using the **delay {down** *seconds* | **up** *seconds*} command in tracking configuration mode.

Step 3. Configure the static route with the **ip route** *destination mask* {*interface* | *next-hop*} **track** *object-number* global command.

Example 11-6 shows the configuration of tracking object 2, using the same design shown in Figure 11-2 and 11-4. In this case, the configuration adds a static route for subnet 10.1.234.0/24, the LAN subnet to which R2, R3, and R4 all connect. EIGRP chooses a route over R1's S0/0/0 as its best route, but this static route uses S0/0/1 as the outgoing interface.

Example 11-6 *Configuring a Static Route with Tracking IP SLA*

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# track 2 ip sla 11 state
R1(config-track)# delay up 90 down 90
R1(config-track)# exit
R1(config)# ip route 10.1.234.0 255.255.255.0 s0/0/1 track 2
R1(config)# end
```

The configuration begins with the creation of the tracking object number 2. As with IP SLA operation numbers, the number itself is unimportant, other than that the **ip route** command refers to this same number with the **track 2** option at the end of the command. The tracking object's delay settings have been made at 90 seconds.

The **show track** command lists the tracking object's configuration plus many other details. It lists the current tracking state, the time in this state, the number of state transitions, and the other entities that track the object (in this case, a static route).

Example 11-7 shows what happens when the IP SLA operation fails, causing the static route to be removed. The example starts with the configuration shown in Example 11-6, along with the SLA operation 11 as configured in Example 11-4. The following list details the current operation and what happens sequentially in the example:

Step 1. Before the text seen in Example 11-7, the current IP SLA operation already sends packets using PBR, over R1's link to R4, using source IP address 10.1.1.9 and destination 10.1.3.99 (server S1).

Step 2. At the beginning of the next example, because the IP SLA operation is working, the static route is in R1's IP routing table.

Step 3. An ACL is configured on R4 (not shown) so that the IP SLA operation fails.

Step 4. A few minutes later, R1 issues a log message stating that the tracking object changed state from up to down.

Step 5. The example ends with several commands that confirm the change in state for the tracking object, and confirmation that R1 now uses the EIGRP learned route through R2.

Note: This example uses the **show ip route ... longer-prefixes** command because this command lists only the route for 10.1.234.0/24, which is the route that fails over in the example.

Example 11-7 *Verifying Tracking of Static Routes*

```
! Next - Step 2 Step 2 Step 2 Step 2
R1# show ip route 10.1.234.0 255.255.255.0 longer-prefixes
! Legend omitted for brevity

      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
S        10.1.234.0/24 is directly connected, Serial0/0/1
R1# show track
Track 2
  IP SLA 11 state
  State is Up
    1 change, last change 01:24:14
  Delay up 90 secs, down 90 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 7
  Tracked by:
    STATIC-IP-ROUTING 0
!

! Next, Step 3 Step 3 Step 3
! Not shown - SLA Operations packets are now filtered by an ACL on R4
! Sometime later...
!

! Next - Step 4 Step 4 Step 4 Step 4
R1#
*Sep 14 22:55:43.362: %TRACKING-5-STATE: 2 ip sla 11 state Up->Down

! Final Step - Step 5 Step 5 Step 5 Step 5
R1# show track
Track 2
  IP SLA 11 state
  State is Down
  2 changes, last change 00:00:15
  Delay up 90 secs, down 90 secs
  Latest operation return code: No connection
  Tracked by:
    STATIC-IP-ROUTING 0
R1# show ip route 10.1.234.0 255.255.255.0 longer-prefixes
! Legend omitted for brevity
```

```
     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D       10.1.234.0/24 [90/2172416] via 10.1.12.2, 00:00:25, Serial0/0/0
```

Configuring PBR to Track an IP SLA

To configure PBR to use object tracking, use a modified version of the **set** command in the route map. For instance, the earlier PBR configuration used the following **set** command:

```
set ip next-hop 10.1.14.4
```

Instead, use the **verify-availability** keyword, as shown in this command:

```
set ip next-hop verify-availability 10.1.14.4 1 track 2
```

When the tracking object is up, PBR works as configured. When the tracking object is down, PBR acts as if the **set** command does not exist. That means that the router will still attempt to route the packet per the normal destination-based routing process.

The output of the related verification commands does not differ significantly when comparing the configuration of tracking for static routes versus PBR. The **show track** command lists "ROUTE-MAP" instead of "STATIC-IP-ROUTING," but the details of the **show track**, **show ip sla statistics**, and object tracking log message seen in Example 11-7, remain the same.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 11-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 11-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design calls for traffic destined for one server in subnet 10.1.1.0/24 to be sent over a different route than the IGP-learned route for 10.1.1.0/24. (2)	
Same requirement as the previous row, except that only a subset of the source hosts should have their packets take a different route than the IGP-learned route.	
The design requires that a static route be used, but only when a particular database server is reachable.	

Implementation Plan Peer Review Table

Table 11-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 11-7 all configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 11-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A plan lists two PBR route maps–one that uses the **default** keyword in its **set** command, and the other that does not. What is the fundamental difference?	
A plan shows a route map enabled for policy routing, and the route map matches some packets with a deny route-map clause. What does IOS do with those packets?	
The plan document shows a PBR route map with the command **set ip dscp ef**. Does PBR support marking? And can it mark DSCP?	
The plan shows an IP SLA operation number 5, with a static route configured with the **track 5** parameters. What issues might exist with the linkages between these commands?	
The IP SLA configuration shows an IP SLA operation that uses ICMP Echo, with the destination IP address of a server. What must be done on the server to support this operation?	
Same scenario as the previous row, except the destination address is on a router.	
Same scenario as the previous row, except the operation generates RTP packets to measure voice jitter.	

Table 11-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure the matching logic in a PBR route map (two options).	
Configure the next-hop IP address in a PBR route map (two options).	
Configure the outgoing interface in a PBR route map (two options).	
Enable PBR on an interface.	
Enable PBR for packets created by the router.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 11-8 all commands that supply the requested information. You may want to record your answers

outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 11-8 *Verification Plan Memory Drill*

Information Needed	Command
List interfaces on which PBR is enabled and the route-map used.	
Display the configuration of a route map.	
Generate debug messages for each packet that matches PBR.	
Display the configuration of an SLA operation.	
Show the measurements from an SLA operation.	
Display the status of a tracking object.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter, but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 11-9 lists a reference of these key topics and the page numbers on which each is found.

Key Topic

Table 11-9 *Key Topics for Chapter 10*

Key Topic Element	Description	Page Number
Figure 1-1	PBR concepts	366
Table 11-2	List of PBR **set** commands to define the route	367
List	Comparisons of PBR logic when including/omitting the **set** command's **default** keyword	371
Figure 1-3	IP SLA concepts	373
List	Configuration checklist for IP SLA	374
Figure 11-5	Configuration relationships for IP SLA and object tracking	378
List	Configuration checklist for object tracking	378

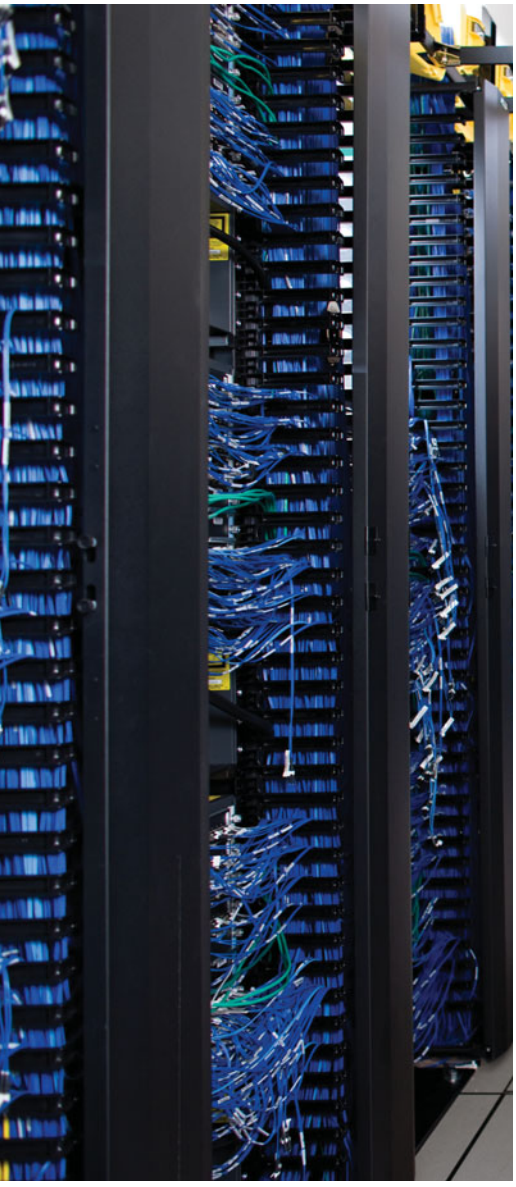
Complete the Tables and Lists from Memory

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

Definitions of Key Terms

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

Policy-Based Routing, IP Service-Level Agreement, tracking object, path control, ToS, IP Precedence, SLA Operation



This chapter covers the following subjects:

The Basics of Internet Routing and Addressing: This section reviews the use of public and private IP addresses in the Internet, both in theory and practice.

Introduction to BGP: This section introduces several basic concepts about BGP, including the concept of autonomous system numbers (ASN), path attributes (PA), and both internal and external BGP.

Outbound Routing Toward the Internet: This section examines the options and tradeoffs for how to influence outbound routes from an Enterprise toward the Internet.

CHAPTER 12

Internet Connectivity and BGP

Enterprises almost always use some IGP routing protocol. Sure, Enterprises could instead choose to exclusively use static routes throughout their internetworks, but they typically do not. Using an IGP requires much less planning, configuration, and ongoing effort compared to using static routes. Routing protocols take advantage of new links without requiring more static route configuration, and the routing protocols avoid the misconfiguration issues likely to occur when using a large number of static routes.

Similarly, when connecting to the Internet, Enterprises can use either static routes or a routing protocol, namely Border Gateway Protocol (BGP). However, the decision to use BGP instead of static routes does not usually follow the same logic that leads engineers to almost always use an IGP inside the Enterprise. BGP might not be necessary or even useful in some cases. To quote Jeff Doyle, author of two of the most respected books on the subject of IP routing, "Not as many internetworks need BGP as you might think" (from his book Routing TCP/IP, Volume II).

This chapter examines the facts, rules, design options, and some perspectives on Internet connectivity for Enterprises. Along the way, the text examines when static routes may work fine, how BGP might be useful in some cases, and the cases for which BGP can be of the most use. Chapter 13, "External BGP," discusses the configuration and verification of BGP for basic operation, but with no overt attempt to influence BGP's choice of best paths. Chapter 14, "Internal BGP and BGP Route Filtering," then discusses the need for Internal BGP (iBGP), along with BGP route filtering. Finally, Chapter 15, "BGP Path Control," completes this book's coverage of BGP by examining the tools by which BGP can be made to choose different routes, and some basic configuration to manipulate the choices of best route.

This chapter begins with a section that reviews some CCNA-level topics from the perspective of Enterprise connectivity to the Internet. In particular, this first section examines public and private IPv4 addressing, NAT, and how the Internet uses the public IPv4 address space. Next, the chapter introduces BGP as a routing protocol, emphasizing that most Enterprises choose to use BGP when needing to influence the choice of best route. The final section of this chapter examines different design categories for Internet connections, emphasizing the path control options gained by using BGP in those designs.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these seven self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 12-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 12-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
The Basics of Internet Routing and Addressing	1–2
Introduction to BGP	3–5
Outbound Routing Toward the Internet	6, 7

1. Which of the following are considered private IPv4 addresses? (Choose two.)
   - **a.** 192.16.1.1
   - **b.** 172.35.1.1
   - **c.** 225.0.0.1
   - **d.** 127.0.0.1
   - **e.** 10.1.1.1
2. Class C network 200.1.1.0/24 was allocated to an ISP that operated primarily in Asia. That ISP then assigned this entire Class C network to one of its Asian customers. Network 200.1.2.0/24 has yet to be assigned to any ISP. Which of the following is most likely to be true?
   - **a.** 200.1.2.0/24 could be assigned to any registrar or ISP in the world.
   - **b.** 200.1.2.0/24 will be assigned in the same geography (Asia) as 200.1.1.0/24.
   - **c.** 200.1.2.0/24 cannot be assigned as public address space.
   - **d.** Routers inside North American ISPs increase their routing table size by 1 as a result of the customer with 200.1.1.0/24 connecting to the Internet.
3. Router R1, in ASN 11, learns a BGP route from BGP peer R22 in ASN 22. R1 and then uses BGP to advertise the route to R2, also in ASN 11. What ASNs would you see in the BGP table on R2 for this route?
   - **a.** 22
   - **b.** 11
   - **c.** 1
   - **d.** None

4. Which of the following are most likely to be used as an ASN by a company that has a registered public 16-bit ASN? (Choose two.)

   a. 1
   b. 65,000
   c. 64,000
   d. 64,550

5. Which of the following statements is true about a router's eBGP peers that is not also true about that same router's iBGP peers?

   a. The eBGP peer neighborship uses TCP.
   b. The eBGP peer uses port 180 (default).
   c. The eBGP peer uses the same ASN as the local router.
   d. The eBGP peer updates its AS_Path PA before sending updates to this router.

6. Which of the following is the primary motivation for using BGP between an Enterprise and its ISPs?

   a. To influence the choice of best path (best route) for at least some routes
   b. To avoid having to configure static routes
   c. To allow redistribution of BGP routes into the IGP routing protocol
   d. To monitor the size of the Internet BGP table

7. The following terms describe various design options for Enterprise connectivity to the Internet. Which of the following imply that the Enterprise connects to two or more ISPs? (Choose two.)

   a. Single Homed
   b. Dual Homed
   c. Single Multihomed
   d. Dual Multihomed

Foundation Topics

The Basics of Internet Routing and Addressing

The original design for the Internet called for the assignment of globally unique IPv4 addresses for all hosts connected to the Internet. The idea is much like the global telephone network, with a unique phone number, worldwide, for all phone lines, cell phones, and the like.

To achieve this goal, the design called for all organizations to register and be assigned one or more public IP networks (Class A, B, or C). Then, inside that organization, each address would be assigned to a single host. By using only the addresses in their assigned network number, each company's IP addresses would not overlap with other companies. As a result, all hosts in the world would have globally unique IP addresses.

The assignment of a single classful network to each organization actually helped keep Internet routers' routing tables small. The Internet routers could ignore all subnets used inside each company, and instead just have a route for each classful network. For instance, if a company registered and was assigned Class B network 128.107.0.0/16, and had 500 subnets, the Internet routers just needed one route for that whole Class B network.

Over time, the Internet grew tremendously. It became clear by the early 1990s that something had to be done, or the growth of the Internet would grind to a halt. At the then-current rate of assigning new networks, all public IP networks would soon be assigned, and growth would be stifled. Additionally, even with routers ignoring the specific subnets, the routing tables in Internet routers were becoming too large for the router technology of that day. (For perspective, more than 2 million public Class C networks exist, and two million IP routes in a single IP routing table would be considered quite large–maybe even too large–for core routers in the Internet even today.)

To deal with these issues, the Internet community worked together to come up with both some short-term and long-term solutions to two problems: the shortage of public addresses and the size of the routing tables. The short-term solutions to these problems included

- Reduce the number of wasted public IP addresses by using classless IP addressing when assigning prefixes—assigning prefixes/lengths instead of being restricted to assigning only Class A, B, and C network numbers.
- Reduce the need for public IP addresses by using Port Address Translation (PAT, also called NAT overload) to multiplex more than 65,000 concurrent flows using a single public IPv4 address.
- Reduce the size of IP routing tables by making good choices for how address blocks are allocated to ISPs and end users, allowing for route summarization on a global scale.

This section examines some of the details related to these three points, but this information is not an end to itself for the purposes of this book. The true goal is to understand outbound routing (from the Enterprise to the Internet), and the reasons why you may or may not need to use a dynamic routing protocol such as Border Gateway Protocol (BGP) between the Enterprise and the Internet.

Public IP Address Assignment

The Internet Corporation for Assigned Network Numbers (ICANN, www.icann.org) owns the processes by which public IPv4 (and IPv6) addresses are allocated and assigned. A related organization, the Internet Assigned Numbers Authority (IANA, www.iana.org) carries out many of ICANN policies. These organizations define which IPv4 addresses can be allocated to different geographic regions, in addition to managing the development of the Domain Name System (DNS) naming structure and new Top Level Domains (TLD), such as .com.

ICANN works with several other groups to administer a public IPv4 address assignment strategy that can be roughly summarized as follows:

Step 1. ICANN and IANA group public IPv4 addresses by major geographic region.

Step 2. IANA allocates those address ranges to Regional Internet Registries (RIR).

Step 3. Each RIR further subdivides the address space by allocating public address ranges to National Internet Registries (NIR) or Local Internet Registries (LIR). (ISPs are typically LIRs.)

Step 4. Each type of Internet Registry (IR) can assign a further subdivided range of addresses to the end user organization to use.

Figure 12-1 shows an example that follows the same preceding four-step sequence. In this example, a company in North America needs a subnet with six hosts, so the ISP assigns a /29 prefix (198.133.219.16/29). Before that happens, however, the process gave this company's ISP (NA-ISP1, an ISP in North America) the right to assign that particular prefix.

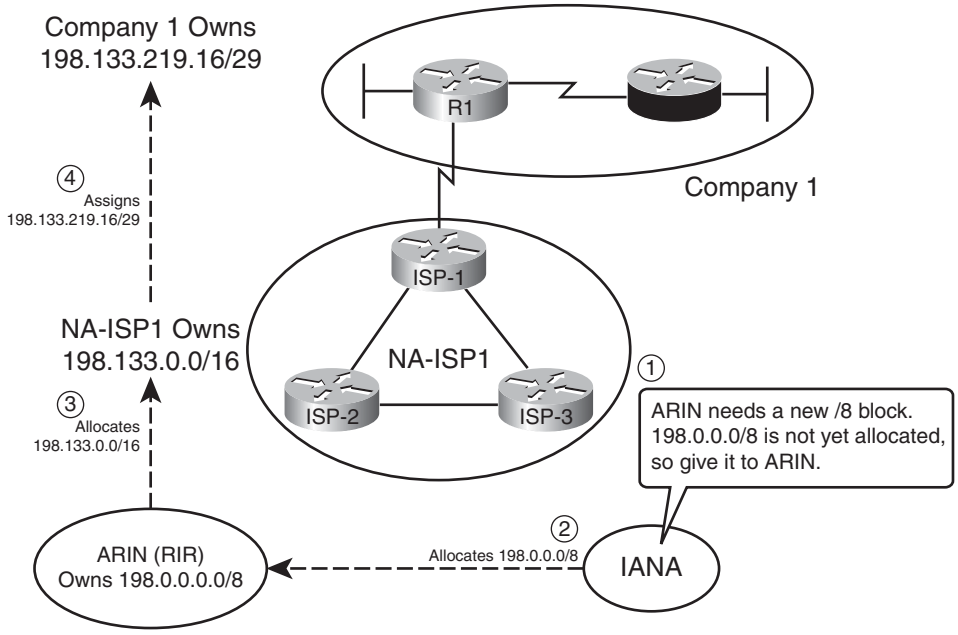


Figure 12-1 *Conceptual View of Public IPv4 Address Assignment*

Key Topic

The process starts with ICANN and IANA. These organizations maintain a set of currently unallocated public IPv4 addresses. (See http://www.iana.org/numbers/, and look for the IPv4 addresses link, to see the current list.) When ARIN, the RIR for North America, notices that it is running out of IPv4 address space, ARIN requests a new public address block. IANA examines the request, finds a currently unallocated public address block (Step 1 in the figure), and allocates the block to ARIN (Step 2 in the figure).

Next, an ISP named NA-ISP1 (shorthand for North American ISP number 1) asks ARIN for a prefix assignment for a /16 sized address block. After ARIN ensures that NA-ISP1 meets some requirements, ARIN assigns a prefix of 198.133.0.0/16 (Step 3 in the figure). Then, when Company1 becomes a customer of NA-ISP1, NA-ISP1 can assign a prefix to Company 1 (198.133.219.16/29 in this example, Step 4).

Although the figure shows the process, the big savings for public addresses occurs because the user of the IP addresses can be assigned a group much smaller than a single Class C network. In some cases, companies only need one public IP address; in other cases, they may need only a few, as with Company1 in Figure 12-1. This practice allows IRs to assign the right-sized address block to each customer, reducing waste.

Internet Route Aggregation

Although the capability to assign small blocks of addresses helped extend the IPv4 public address space, this practice also introduced many more public subnets into the Internet, driving up the number of routes in Internet routing tables. At the same time, the number of hosts connected to the Internet, back in the 1990s, was increasing at a double-digit rate–per month. Internet core routers could not have kept up with the rate of increase in the size of the IP routing tables.

The solution was, and still is today, to allocate numerically consecutive addresses–addresses that can be combined into a single route prefix/length—by geography and by ISP. These allocations significantly aid route summarization.

For example, continuing the same example shown in Figure 12-1, Figure 12-2 shows some of the routes that can be used in ISPs around the globe based on the address assignment shown in Figure 12-1.

First, focus on the routers shown in Europe and South America. Routers outside North America can use a route for prefix 198.0.0.0/8, knowing that IANA assigned this prefix to be used only by ARIN, which manages IP addresses in North America. The underlying logic is that if the routers outside North America can forward the packet into North America, then the North American routers will have more specific routes. The single route for 198.0.0.0/24 shown in Europe and South America can be used instead of literally millions of subnets deployed to companies in North America such as Company1.

Next, consider routers in North America, specifically those outside the NA-ISP1 network. Figure 12-2 shows one such ISP, named NA-ISP2 (North American ISP number 2), on the left. This router can learn one route for 198.133.0.0/16, the portion of the 198.0.0.0/8 block assigned to NA-ISP1 by IANA. Routers in NA-ISP2 can forward all packets for destinations inside this prefix to NA-ISP1, rather than needing a route for all small address blocks

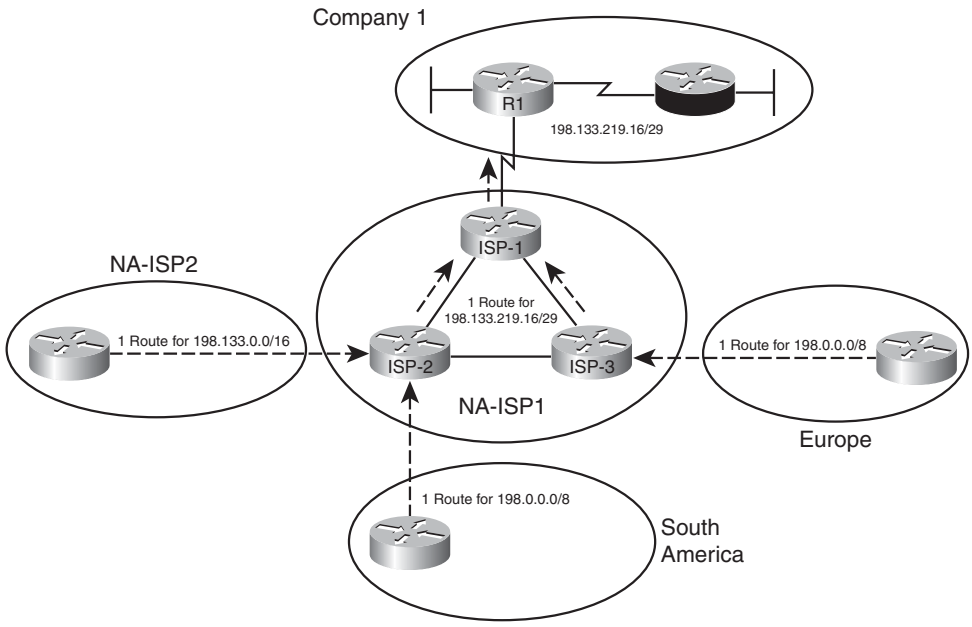


Figure 12-2 *IPv4 Global Route Aggregation Concepts*

assigned to individual Enterprises such as Company1. This significantly reduces the number of routes required on NA-ISP2 routers.

Finally, inside NA-ISP1, its routers need to know to which NA-ISP1 router to forward packets to for that particular customer. So, the routes listed on NA-ISP1's routers lists a prefix of 198.133.219.16/29. As a result, packets are forwarded toward router ISP-1 (located inside ISP NA-ISP1), and finally into Company 1.

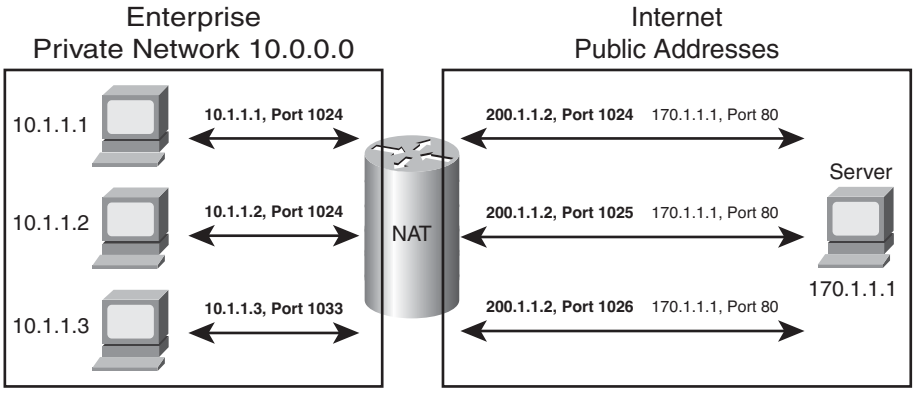
The result of the summarization inside the Internet allows Internet core routers to have a much smaller routing table–on the order of a few hundred thousand routes instead of a few tens of millions of routes. For perspective, website www.potaroo.net, a website maintained by Geoff Huston, who has tracked Internet growth for many years, lists statistics for 300,000 BGP routes in Internet routers back in August 2009.

The Impact of NAT/PAT

Although classless public IP address assignment does help extend the life of the IPv4 address space, NAT probably has a bigger positive impact because it enables an Enterprise to use such a small number of public addresses. NAT allows an Enterprise to use private IP addresses for each host, plus a small number of public addresses. As packets pass through a device performing NAT–often a firewall, but it could be a router–the NAT function translates the IP address from the private address (called an *inside local* address by NAT) into a public address (called an *inside global* address).

Note: For the purposes of this book, the terms NAT, PAT, and NAT overload are used synonymously. There is no need to distinguish between static NAT, dynamic NAT without overload, and dynamic NAT with overload (also called PAT).

NAT reduces the need for public IPv4 addresses to only a few addresses because of how NAT can multiplex flows using different TCP or UDP port numbers. Figure 12-3 shows a sample that focuses on a router performing NAT. The figure shows an Enterprise network on the left, with the Enterprise using private Class A network 10.0.0.0/8. The Internet sits on the right, with the NAT router using public IP address 200.1.1.2.



Dynamic NAT Table, With Overloading

Inside Local	Inside Global
10.1.1.1:1024	200.1.1.2:1024
10.1.1.2:1024	200.1.1.2:1025
10.1.1.3:1033	200.1.1.2:1026

Figure 12-3 *IPv4 Global Route Aggregation Concepts*

The figure shows how the Enterprise, on the left, can support three flows with a single public IP address (200.1.1.2). The NAT feature dynamically builds its translation table, which tells the router what address/port number pairs to translate. The router reacts when a new flow occurs between two hosts, noting the source IP address and port number of the Enterprise host on the left, and translating those values to use the public IP address (200.1.1.2) and an unused port number in the Internet. Note that if you collected the traffic using a network analyzer on the right side of the NAT router, the IP addresses would include 200.1.1.2 but not any of the network 10.0.0.0/8 addresses. Because the combination of the IP address (200.1.1.2 in this case) and port number must be unique, this one IP address can support 2^{16} different concurrent flows.

Private IPv4 Addresses and Other Special Addresses

When allocating the public IPv4 address space, IANA/ICANN restricts themselves in several ways. Of course, the private IP address ranges cannot be assigned to any group for

use in the public Internet. Additionally, several other number ranges inside the IPv4 address space, as summarized in RFC 3330, are reserved for various reasons. Tables 12-2 and 12-3 list the private addresses and other reserved values, respectively, for your reference.

Table 12-2 *Private IP Address Reference*

Number of Classful Networks	Range of Classful Networks	Prefix for Entire Range
(1) Class A:	10.0.0.0	10.0.0.0/8
(16) Class B:	172.16.0.0 through 172.31.0.0	172.16.0.0/12
(256) Class C:	192.168.0.0 through 192.168.255.0	192.168.0.0/16

Table 12-3 *Reserved Values in IPv4 Address Range (RFC 3330)*

Value or Range	Reason
0.0.0.0/8	Used for self-identification on a local subnet.
127.0.0.0/8	Loopback testing.
169.254.0.0/16	This "link local" block is used for default IPv4 address assignment when DHCP process fails.
192.0.2.0/24	Reserved for use in documentation and example code.
192.88.99.0/24	Used for IPv6 to IPv4 relay (6to4 relay) (RFC 3068).
198.18.0.0/15	Benchmark testing for Internet devices (RFC 2544).

Table 12-3 lists other reserved ranges of IPv4 addresses that IANA will not allocate in the public Internet.

In summary, every Enterprise that connects to the Internet must use at least one public IP address, and often several public IP addresses. Although some companies do have a large public IPv4 address block–often obtained before the shortage of public IPv4 addresses in the early to mid-1990's–most companies have a small address block, which then requires the use of NAT/PAT. These details have some impact on whether BGP is useful in a given case.

Introduction to BGP

Border Gateway Protocol (BGP) advertises, learns, and chooses the best paths inside the global Internet. When two ISPs connect, they typically use BGP to exchange routing information. Collectively, the ISPs of the world exchange the Internet's routing table using BGP. And Enterprises sometimes use BGP to exchange routing information with one or more ISPs, allowing the Enterprise routers to learn Internet routes.

One key difference when comparing BGP to the usual IGP routing protocols is BGP's robust best path algorithm. BGP uses this algorithm to choose the best BGP path (route) using rules that extend far beyond just choosing the route with the lowest metric. This more complex best path algorithm gives BGP the power to let engineers configure many different settings that influence BGP best path selection, allowing great flexibility in how routers choose the best BGP routes.

BGP Basics

BGP, specifically BGP Version 4 (BGPv4), is the one routing protocol in popular use today that was designed as an Exterior Gateway Protocol (EGP) instead of as an Interior Gateway Protocol (IGP). As such, some of the goals of BGP differ from those of an IGP such as OSPF or EIGRP, but some of the goals remain the same.

First, consider the similarities between BGP and various IGPs. BGP does need to advertise IPv4 prefixes, just like IGPs. BGP needs to advertise some information so that routers can choose one of many routes for a given prefix as the currently best route. As for the mechanics of the protocol, BGP does establish a neighbor relationship before exchanging topology information with a neighboring router.

Next, consider the differences. BGP does not require neighbors to be attached to the same subnet. Instead, BGP routers use a TCP connection (port 179) between the routers to pass BGP messages, allowing neighboring routers to be on the same subnet, or to be separated by several routers. (It is relatively common to see BGP neighbors who do not connect to the same subnet.) Another difference lies in how the routing protocols choose the best route. Instead of choosing the best route just by using an integer metric, BGP uses a more complex process, using a variety of information, called BGP *path attributes*, which are exchanged in BGP routing updates much like IGP metric information.

Table 12-4 summarizes some of these key comparison points.

Key Topic

Table 12-4 *Comparing OSPF and EIGRP Logic to BGP*

OSPF/EIGRP	BGP
Forms neighbor relationship before sending routing information	Same.
Neighbors typically discovered using multicast packets on the connected subnets	Neighbor IP address is explicitly configured and may not be on common subnet.
Does not use TCP	Uses a TCP connection between neighbors (port 179).

Table 12-4 *Comparing OSPF and EIGRP Logic to BGP*

OSPF/EIGRP	BGP
Advertises prefix/length	Advertises prefix/length, called *Network Layer Reachability Information (NLRI.)*
Advertises metric information	Advertises a variety of path attributes (PA) that BGP uses instead of a metric to choose the best path.
Emphasis on fast convergence to the truly most efficient route	Emphasis on scalability; may not always choose the most efficient route.
Link state (OSPF) or distance vector (EIGRP) logic	Path vector logic (similar to distance vector).

BGP ASNs and the AS_SEQ Path Attribute

BGP uses BGP path attributes (PA) for several purposes. PAs define information about a path, or route, through a network. Some BGP PAs describe information that can be useful in choosing the best BGP route, using the best path algorithm; BGP also uses some other PAs for other purposes besides choosing the best path. This chapter focuses on one particular PA that routers use when choosing the best path, and using this PA to help prevent loops. Later, in Chapter 15, the text explores PAs in more detail, in the context of how to use those PAs as a tool to achieve some design goal.

By default, if no BGP PAs have been explicitly set, BGP routers use the BGP AS_PATH (autonomous system path) PA when choosing the best route among many competing routes. The AS_Path PA itself has many subcomponents, only some of which matter to the depth of the CCNP coverage of the topic. However, the most obvious component of AS_Path, the AS_Seq (AS Sequence), can be easily explained with an example when the concept of an autonomous system number (ASN) has been explained.

The integer BGP ASN uniquely identifies one organization that considers itself autonomous from other organizations. Each company whose Enterprise network connects to the Internet can be considered to be an autonomous system and can be assigned a BGP ASN. (IANA/ICANN also assigns globally unique ASNs.) Additionally, each ISP has an ASN, or possibly several, depending on the size of the ISP.

When a router uses BGP to advertise a route, the prefix/length is associated with a set of PAs, including the AS_Path. The AS_Path PA associated with a prefix/length lists the ASNs that would be part of an end-to-end route for that prefix as learned using BGP. In a way, the AS_Path implies information like this: "If you use this path (route), the path will go through this list of ASNs."

BGP uses the AS_Path to perform two key functions:

- Choose the best route for a prefix based on the shortest AS_Path (fewest number of ASNs listed).
- Prevent routing loops.

An example can help demonstrate the concept. This example, and some others in this chapter, use the design shown in Figure 12-4. This network has five ASNs: three ISPs and two customers.

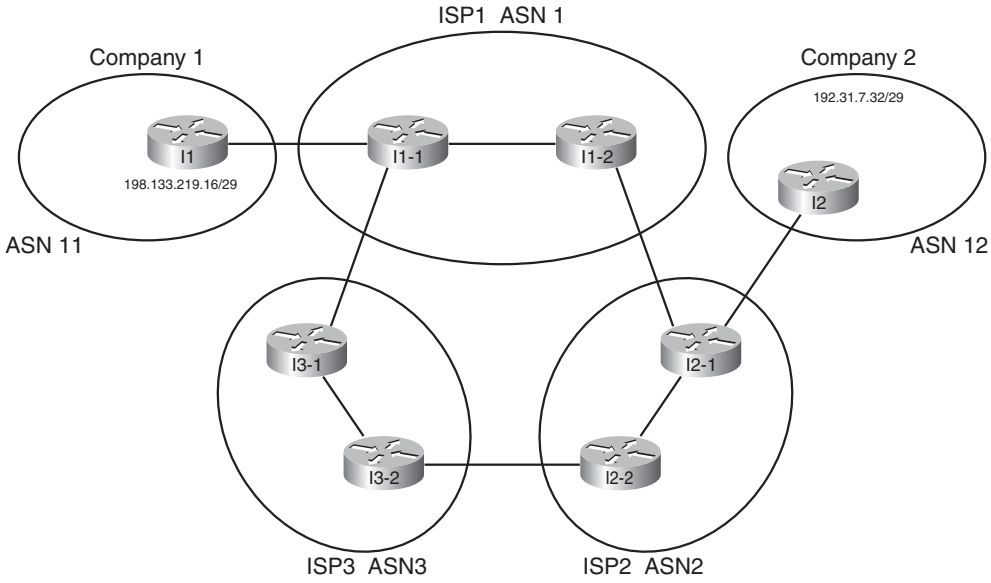


Figure 12-4 *Sample Portion of the Internet*

Figure 12-4 shows only a couple of routers in each ISP, and it also does not bother to show much of the Enterprise networks for the two companies. However, the diagram does show enough detail to demonstrate some key BGP concepts. For the sake of discussion, assume each line between routers represents some physical medium that is working. Each router will use BGP, and each router will form BGP neighbor relationships with the routers on the other end of each link. For example, ISP1's I1-2 router will have a BGP neighbor relationship with Routers I1-1 and with I2-1.

With that in mind, consider Figure 12-5, which shows the advertisement of BGP updates for prefix 192.31.7.32/29 to the other ASNs:

The figure shows four steps, as follows:

Step 1. I2, in ASN 12, advertises the route outside ASN 12. So, I2 adds its own ASN (12) to the AS_Path PA when advertising the route.

Step 2. The routers inside ASN 2, when advertising the route outside ASN 2, add their own ASN (2) to the AS_Path PA when advertising the route. Their advertised AS_Path is then (12,2).

Step 3. Router I3-1, inside ASN 3, had previously learned about the route for 192.31.7.32/29 from ASN 2, with AS_Path (12,2). So, I3-1 advertises the route to ASN 1, after adding its own ASN (3) to the AS_Path so that the AS_Path (12, 2, 3).

Step 4. Similarly, Router I1-1, inside ASN 1, advertises the route to ASN3. Because ASN 3 is a different ASN, I1-1 adds its own ASN (1) to the AS_Path PA so that the AS_Path lists ASNs 12, 2, and 1.

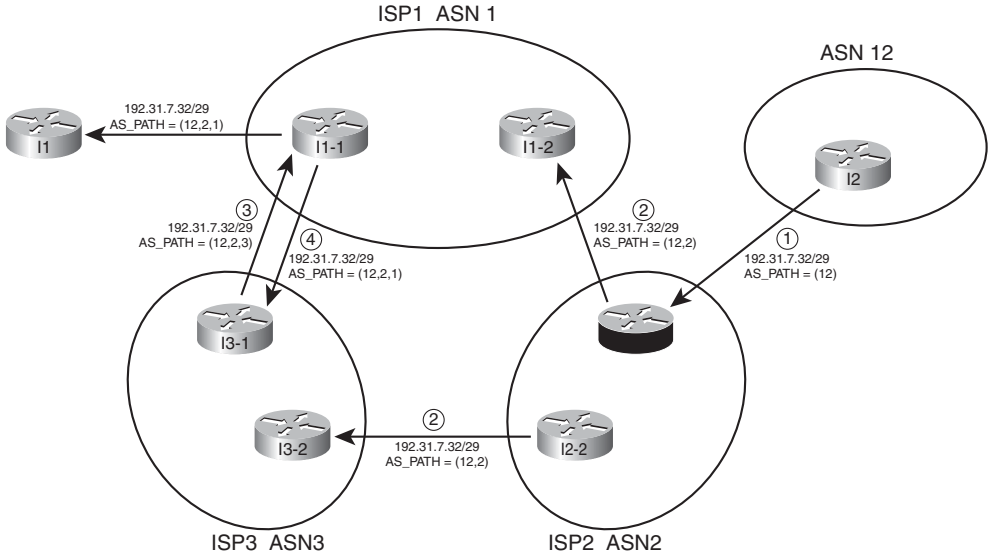


Figure 12-5 *Advertisement of NLRI to Demonstrate AS_Path*

Now, step back from the details, and consider the two alternative routes learned collectively by the routers in ASN 1:

- 192.31.7.32/29, AS_Path (12,2)
- 192.31.7.32/29, AS_Path (12,2,3)

Because the BGP path selection algorithm uses the shortest AS_Path, assuming no other PAs have been manipulated, the routers in ASN 1 use the first of the two paths, sending packets to ASN 2 next, and not using the path through ASN 3. Also, as a result, note the advertisement from ASN 1 into ASN 11 lists an AS_Path that reflects the best path selection of the routers inside ASN 1, with the addition of ASN 1 to the end of the AS_Path of the best route (12,2,1).

BGP routers also prevent routing loops using the ASNs listed in the AS_Path. When a BGP router receives an update, and a route advertisement lists an AS_Path with its own ASN, the router ignores that route. The reason is that because the route has already been advertised through the local ASN, to believe the route and then advertise it further might cause routing loops.

Internal and External BGP

BGP defines two classes of neighbors (peers): internal BGP (iBGP) and external BGP (eBGP). These terms use the perspective of a single router, with the terms referring to whether a BGP neighbor is in the same ASN (iBGP) or a different ASN (eBGP).

A BGP router behaves differently in several ways depending on whether the peer (neighbor) is an iBGP or eBGP peer. The differences include different rules about what must be true before the two routers can become neighbors, different rules about which routes the BGP best path algorithm chooses as best, and even some different rules about how the routers update the BGP AS_Path PA. These differences will be highlighted in the context of other features throughout this chapter and in Chapters 13, 14, and 15, but for now, it is interesting to examine the differences related to AS_Path.

When advertising to an eBGP peer, a BGP router updates the AS_Path PA, but it does not do so when advertising to an iBGP peer. For example, Figure 12-6 shows the same design, with the same route advertisement, as in Figure 12-5. However, in this case, all the BGP connections have been listed as either iBGP or eBGP.

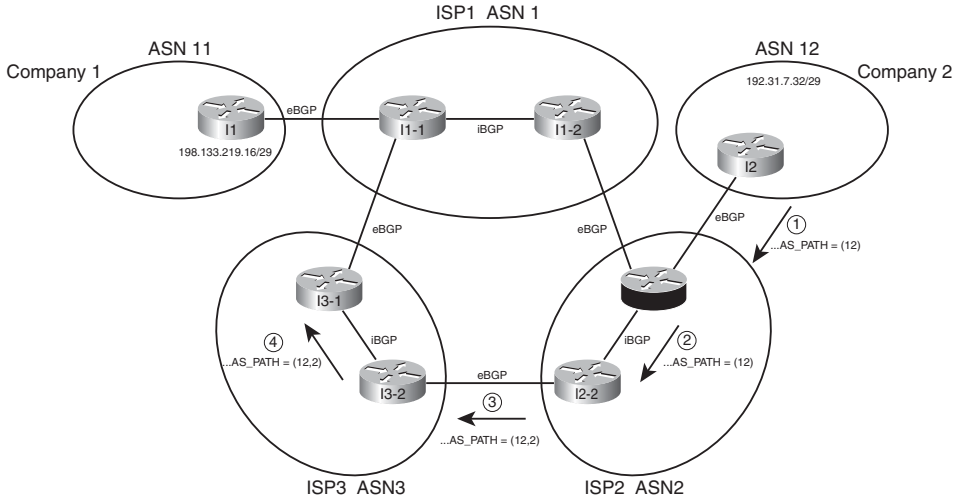


Figure 12-6 *iBGP, eBGP, and Updating AS_Path for eBGP Peers*

The figure highlights the route advertisement from ASN 12, over the lower path through ASN 2 and 3. Note that at Step 1, Router I2, advertising to an eBGP peer, adds its own ASN to the AS_Path. At Step 2, Router I2-1 is advertising to an iBGP peer (I2-2), so it does not add its own ASN (2) to the AS_Path. Then, at Step 3, Router I2-2 adds its own ASN (2) to the AS_Path before sending an update to eBGP peer I3-2, and so on.

Public and Private ASNs

For the Internet to work well using BGP, IANA administers the assignment of ASNs much like it does with IP address prefixes. One key reason why ASNs must be assigned as unique values is that if ASNs are duplicated, the BGP loop prevention process can actually prevent parts of the Internet from learning about a route. For example, consider Figure 12-7, with the same design as in the last few figures–but this time with a duplicate ASN.

In this figure, both ISP1 and Company 1 use ASN 12. The example BGP updates begin as in Figures 12-5 and 12-6, with Company 1 advertising its prefix. Routers inside ISP1 re-

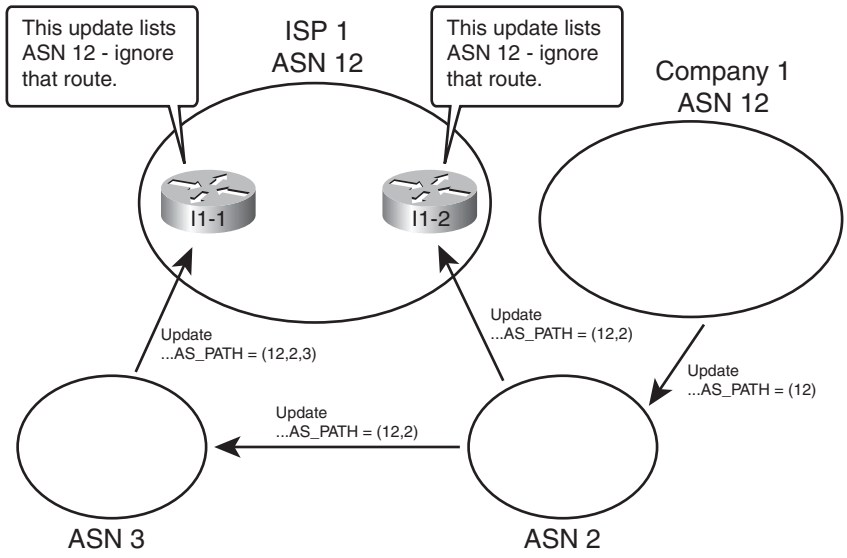


Figure 12-7 *Duplicate ASN (12) Preventing Route Advertisement*

Key Topic

ceive BGP updates that list the same prefix used by Company 1, but both Updates list an AS_Path that includes ASN 12. Because ISP1 thinks it uses ASN 12, ISP1 thinks that these BGP Updates should be ignored as part of the BGP loop prevention process. As a result, customers of ISP1 cannot reach the prefixes advertised by routers in Company 1.

To prevent such issues, IANA controls the ASN numbering space. Using the same general process as for IPv4 addresses, ASNs can be assigned to different organizations. The 16-bit BGP ASN implies a decimal range of 0 through 65,535. Table 12-5 shows some of the details of IANA's current ASN assignment conventions.

Table 12-5 *16-Bit ASN Assignment Categories from IANA*

Key Topic

Value or Range	Purpose
0	Reserved
1 through 64,495	Assignable by IANA for public use
64,496 through 65,511	Reserved for use in documentation
64,512 through 65,534	Private use
65,535	Reserved

Like the public IPv4 address space has suffered with the potential for complete depletion of available addresses, the public BGP ASN space has similar issues. To help overcome this issue, the ASN assignment process requires that each AS justify whether it truly needs a publicly unique ASN or whether it can just as easily use a private ASN. Additionally, RFC

5398 reserves a small range of ASNs for use in documentation so that the documents can avoid the use of ASNs assigned to specific organizations.

Private ASNs allow the routers inside an AS to participate with BGP, while using the same ASN as many other organizations. Most often, an AS can use a private AS in cases where the AS connects to only one other ASN. (Private ASNs can be used in some cases of connecting to multiple ASNs as well.) The reason is that with only one connection point to another ASN, loops cannot occur at that point in the BGP topology, so the need for unique ASNs in that part of the network no longer exists. (The loops cannot occur due to the logic behind the BGP best path algorithm, coupled with that BGP only advertises the best path for a given prefix.)

Outbound Routing Toward the Internet

The single biggest reason to consider using BGP between an Enterprise and an ISP is to influence the choice of best path (best route). The idea of choosing the best path sounds appealing at first. However, because the majority of the end-to-end route exists inside the Internet, particularly if the destination is 12 routers and a continent away, it can be a challenge to determine which exit point from the Enterprise is actually a better route.

As a result, Enterprises typically have two major classes of options for outbound routing toward the Internet: default routing and BGP. Using default routes is perfectly reasonable, depending on the objectives. This section examines the use of default routes toward the Internet, and some of the typical Enterprise BGP designs and how they can be used to influence outbound routes toward the Internet.

Comparing BGP and Default Routing for Enterprises

Chapter 4, "EIGRP Route Summarization and Filtering," section "Default Routing to the Internet Router," introduced the concept of using default routes on branch office routers, with static routes and redistribution on the WAN edge routers. With this design, the branch router could use a default static route pointing toward the core of the network. The WAN edge routers then needed static routes for the subnets at each branch, with the WAN edge routers advertising these branch subnets into the core using an IGP.

The branch office default routing design results in less processing on the routers, less memory consumption, and no IGP overhead on the link between the branch and WAN distribution routers. In particular, the branch routers can have a single or a few default routes, instead of potentially hundreds of routes for specific prefixes, all with the same next-hop information.

The same general concept of using defaults and static routes at Enterprise branches can be applied to the Enterprise network and its connections to one or a few ISPs. Similar to a branch router, an entire Enterprise often has only a few connections to the Internet. If one of those connections is considered better than the others, then all packets sent from the Enterprise toward the Internet would normally follow that one Internet link, for all Internet destinations. Likewise, the ISPs, similar to WAN distribution routers in this analogy, could configure static routes for the Enterprise's public IP address prefix and then use BGP in the Internet to advertise those routes. Figure 12-8 shows the general idea.

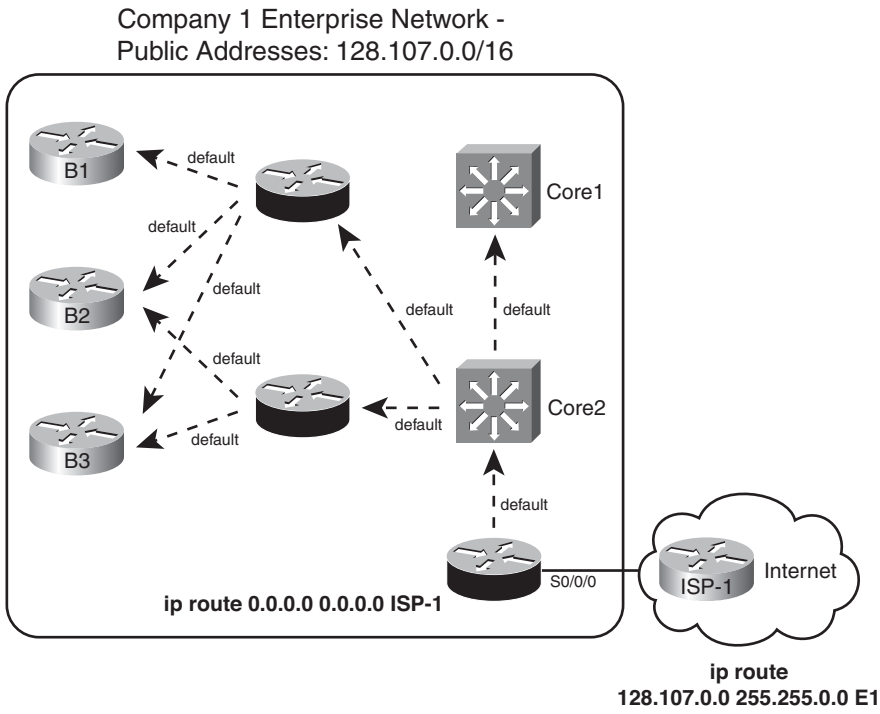


Figure 12-8 *Use of Static Default into the Internet*

Although the Enterprise could choose to use BGP in this case, such a decision is not automatic. First, the alternative of using static routes, as shown in the figure, does not require a lot of work. The Enterprise network engineer just needs to configure a default route and advertise it throughout the Enterprise; the dashed lines in the figure represent the advertisement of the default route with the Enterprise's IGP. (See Chapter 4's section "Default Routing to the Internet Router," and Chapter 7, "OSPF Route Summarization, Filtering, and Default Routing," section "Domain-wide Defaults Using the **default-information originate** Command," for a review of how to flood the static route inside an EIGRP or OSPF domain, respectively.)

In addition to the configuration on the Enterprise router (E1), the ISP network engineer has to configure static routes for that Enterprise's public IP address range, and redistributing those routes into BGP and advertising throughout the Internet. The figure shows a static route for Company 1's 128.107.0.0/16 public address range. Additionally, this prefix would need to be injected into BGP for advertising into the rest of the Internet.

Instead of using static default routes, you could enable BGP between E1 and ISP-1. Running BGP could mean that the Enterprise router requires significant memory and more processing power on the router. The design may also require other Enterprise routers besides the Internet-connected routers to know the BGP routes, requiring additional routers to have significant CPU and memory. Finally, although you can configure BGP to choose one route over another using PAs, the advantage of choosing one path over another may

not be significant. Alternatively, you could ask the ISP to advertise only a default route with BGP.

Now that you have seen a few of the reasons why you may be fine using static routes instead of BGP, consider why you might want to use BGP. First, it makes most sense to use BGP when you have at least two Internet connections. Second, BGP becomes most useful when you want to choose one outbound path over another path for particular destinations in the Internet. In short, when you have multiple Internet connections, and you want to influence some packets to take one path and some packets to take another, consider BGP.

The rest of this chapter examines different cases of Internet connectivity and weighs the reasons why you might choose to use BGP. For this discussion, the perspective of the Enterprise network engineer will be used. As such, outbound routing is considered to be routes that direct packets from the Enterprise toward the Internet, and inbound routing refers to routes that direct packets into the Enterprise from the Internet.

To aid in the discussion, this section examines four separate cases:

- Single homed (1 link per ISP, 1 ISP)
- Dual homed (2+ links per ISP, 1 ISP)
- Single multihomed (1 link per ISP, 2+ ISPs)
- Dual multihomed (2+ links per ISP, 2+ ISPs)

Note: The terms in the preceding list may be used differently depending on what book or document you read. For consistency, this book uses these terms in the same way as the Cisco authorized ROUTE course associated with the ROUTE exam.

Single Homed

The single-homed Internet design uses a single ISP, with a single link between the Enterprise and the ISP. With single-homed designs, only one possible next-hop router exists for any and all routes for destinations in the Internet. As a result, no matter what you do with BGP, all learned routes would list the same outgoing interface for every route, which minimizes the benefits of using BGP.

Single-homed designs often use one of two options for routing to and from the Internet:

- Use static routes (default in the Enterprise, and a static for the Enterprise's public address range at the ISP).
- Use BGP, but only to exchange a default (ISP to Enterprise) and a route for the Enterprise's public prefix (Enterprise to ISP).

The previous section, "Comparing BGP and Defaults for Enterprises," already showed the main concepts for the first option. For the second option, the concept still uses the IGP's mechanisms to flood a default throughout the Enterprise, causing all packets to go toward the Internet facing router. Instead of static routes, however, the following must happen:

- The ISP router uses BGP to advertise a default route to the Enterprise.

- You must configure the IGP on the Enterprise's Internet-facing router to flood a default route (typically only if the default route exists in that router's routing table).
- You must configure BGP on the Enterprise router and advertise the Enterprise's public prefix toward the ISP.

Both options–using static defaults and BGP learned defaults–have some negatives. Some packets for truly nonexistent destinations flow through the Enterprise to the Internet-facing router (E1 in the example of Figure 12-8), and over the link to the Internet, before being discarded for lack of a matching route. For example, if the Enterprise used private network 10.0.0.0/8 internally, packets destined for addresses in network 10.0.0.0/8 that have not yet been deployed will match the default route and be routed to the Internet.

To avoid wasting this bandwidth by sending packets unnecessarily, a static route for 10.0.0.0/8, destination null0, could be added to the Internet-facing router but not advertised into the rest of the Enterprise. (This type of route is sometimes called a *discard route*.) This route would prevent the Internet-facing router from forwarding packets destined for network 10.0.0.0/8 into the Internet.

Dual Homed

The dual-homed design has two (or more) links to the Internet, but with all links connecting to a single ISP. This type of design can use a pair of routers, two pairs, or a combination, as shown in the three cases in Figure 12-9.

Comparing the dual-homed case to the single-homed design, the second link gives the Enterprise a choice. The Enterprise router(s) could choose between one of two links, and in the case with two Enterprise routers, the choice of a different link also means the choice of sending packets to a different router.

Each of the cases shown in Figure 12-9 is interesting, but the case with two Enterprise routers provides the most ideas to consider. When considering whether to use BGP in this case, and if so, how to use it, first think about whether you want to influence the choice of outbound route. The common cases when using defaults works well, ignoring BGP, are:

- To prefer one Internet connection over another for all destinations, but when the better ISP connection fails, all traffic re-routes over the secondary connection.
- To treat both Internet connections as equal, sending packets for some destinations out each path. However, when one fails, all traffic re-routes over the one still-working path.

The text now examines each option, in order, including a discussion of how to choose the best outbound routing using both partial and full BGP updates.

Preferring One Path over Another for All Destinations

When the design calls for one of the two Internet connections to always be preferred, regardless of destination, BGP can be used, but it is not required. With a goal of preferring one path over another, the routers can use default routes into the Internet.

To demonstrate the concept, Figure 12-10 shows a dual-homed design, this time with two routers (E1 and E2) connected to the Internet. Each router has a single link into the single ISP. (Using the terminology from the ROUTE class, dual homed means two or more links

Key Topic

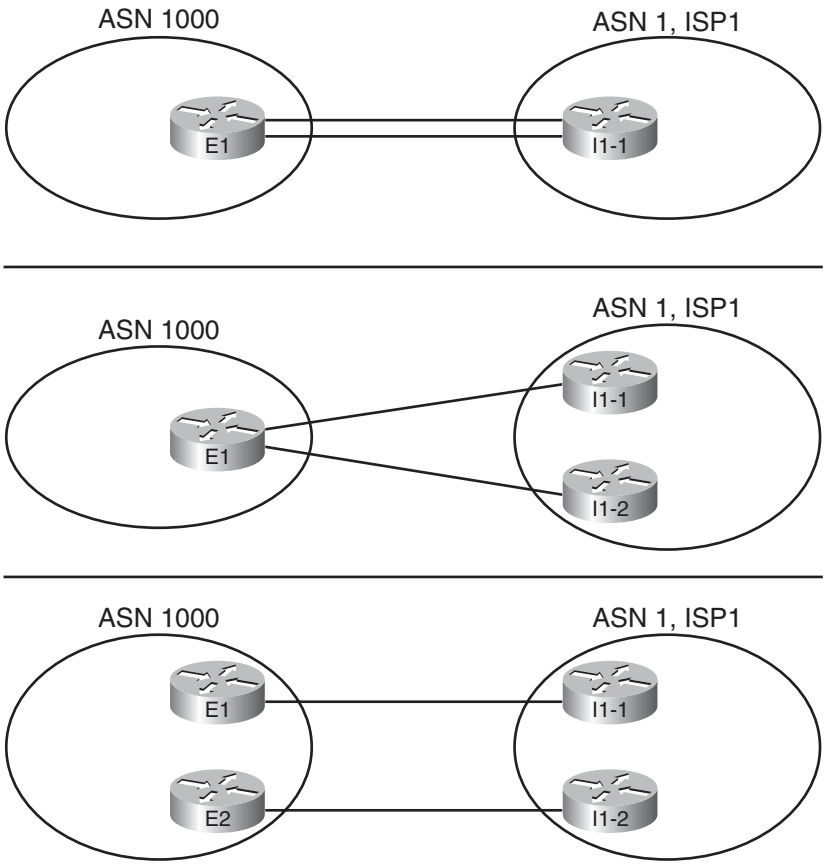


Figure 12-9 *Dual-Homed Design Options*

but to a single ISP; dual multihomed means two or more links each to two or more different ISPs.) Figure 12-10 shows the routes that result from using default routes to forward all traffic toward Router E1.

Figure 12-10 shows that all routers forward the Internet-destined packets toward Router E1, because this router has the faster Internet connection to ISP1 (100 Mbps in this case). Again in this example, the other connection from Router E2 to ISP3 uses a 10 Mbps link.

To make this design work, with failover, both E1 and E2 need to advertise a default route into the Enterprise, but the route advertised by the primary router (E1) needs to have metrics set so that it is always the better of the two routes. For example, with EIGRP, E1 can configure a static default route with Router I1-1 as the next hop, but with very high bandwidth and very low delay upon redistribution into EIGRP. Conversely, E2 can create a default for Router I1-2 as the next-hop router, but with a low bandwidth but high delay. Example 12-1 shows the configuration of the static default route on both E1 and E2, with the **redistribute** command setting the metrics.

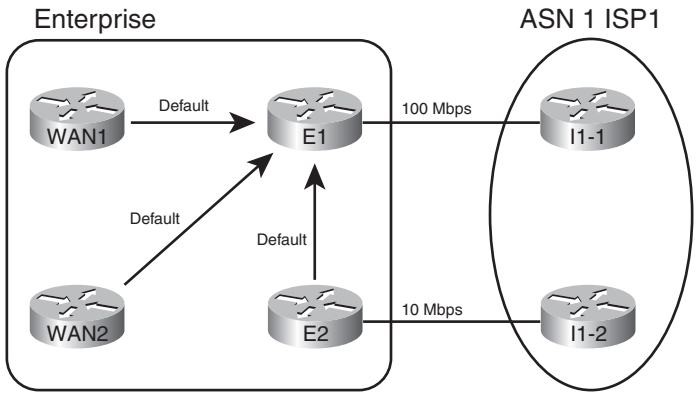


Figure 12-10 *Dual-Homed Design, Using Defaults to Favor One Link*

Example 12-1 *Default Routing on Router E1*

```
! Configuration on router E1 – note that the configuration uses
! a hostname instead of I1-1's IP address
ip route 0.0.0.0 0.0.0.0 I1-1
router eigrp 1
 redistribute static metric 100000 1 255 1 1500
```

```
! Configuration on router E2 - note that the configuration uses
! a hostname instead of I2-1's IP address
ip route 0.0.0.0 0.0.0.0 I2-1
router eigrp 1
 redistribute static metric 10000 100000 255 1 1500
```

Note: With EIGRP as the IGP, do not forget that the delay setting must be set higher to avoid cases where some routers forward packets toward the secondary Internet router (E2). The reason is that EIGRP uses constraining bandwidth, so a high setting of bandwidth at the redistribution point on E1 may or may not cause more remote routers to use that route.

A slightly different approach can be taken in other variations of the dual-homed design, as seen back in Figure 12-9. The first two example topologies in that figure show a single router with two links to the same ISP. If the design called to using one link as the preferred link, and the engineer decided to use default routes, that one router would need two default routes. To make one route be preferred, that static default route would be assigned a better administrative distance (AD) than the other route. For example, the commands **ip route 0.0.0.0 0.0.0.0 I1-1 3** and **ip route 0.0.0.0 0.0.0.0 I1-2 4** could be used on Router E1 in Figure 12-9, giving the route through I1-1 a lower AD (3), preferring that route. If the link to I1-1 failed, the other static default route, through I1-2, would be used.

Choosing One Path over Another Using BGP

The big motivation to use BGP occurs when you want to influence which link is used for certain destinations in the Internet. To see such a case, consider Figure 12-11, which adds

Company 3 to the design. In this case, Company 3 uses prefix 192.135.250.0/28 as its public address range. Company 3 may be located closer to I1-2 inside ISP1 than to Router I1-1, and in such cases, the BGP design calls for making the packets flow over the route as shown.

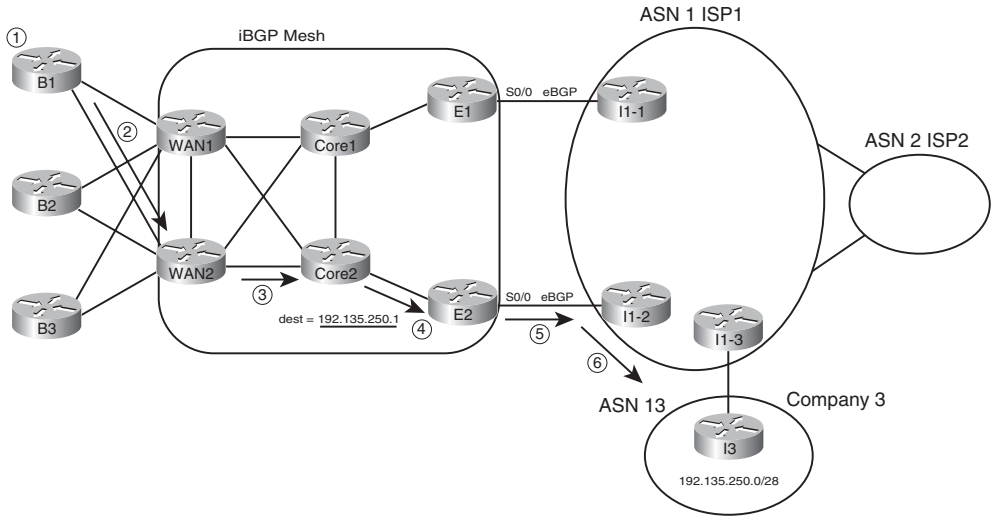


Figure 12-11 *Preferring One Outbound Link over Another*

Two notable actions must take place for this design to work, beyond the basic configuration of the eBGP peers as shown. First, the engineers at the Enterprise and ISP must agree as to how to make BGP specify a prefix as being best reached through a particular link. In this particular case, the routes advertised by I1-2 for prefix for 192.135.250.0/28 must have BGP PA settings that appear better than those learned from I1-1. In this case, you cannot just rely on the default of checking the AS_Path length, because the AS_Path length should tie, because I1-1 and I1-2 are in the same ASN. So when planning with the engineers of ISP1, the Enterprise network engineer must discuss what kinds of prefixes that might work better through I1-1, which would be better through I1-2, and how the ISP might set PA values to which the Enterprise routers (E1 and E2) can react. (Chapter 15 discusses some of the options to influence the outbound routes.)

The second big consideration occurs inside the Enterprise network with a need to run BGP between multiple routers. So far in this chapter, the Enterprise routers all used default routes to send packets to the Internet-facing routers, and only those routers knew Internet routes. However, for the design of Figure 12-11 to work, E1 and E2 must communicate BGP routes using an iBGP connection. And because packet forwarding between E1 and E2 goes through other routers (such as Core1 and Core2), those routers typically also need to run BGP. You might even decide to run BGP on the WAN routers as well. By doing so, the core routers know the best BGP routes; for instance, they all know that the better

route for Company 3's 192.135.250.0/28 public address space is through E2, so the packet is forwarded to E2. The following list outlines the logic matching Figure 12-11:

Step 1. A host at Branch B1 sends a packet to 192.135.250.1.

Step 2. Router B1 matches its default route, forwarding the packet to Router WAN2.

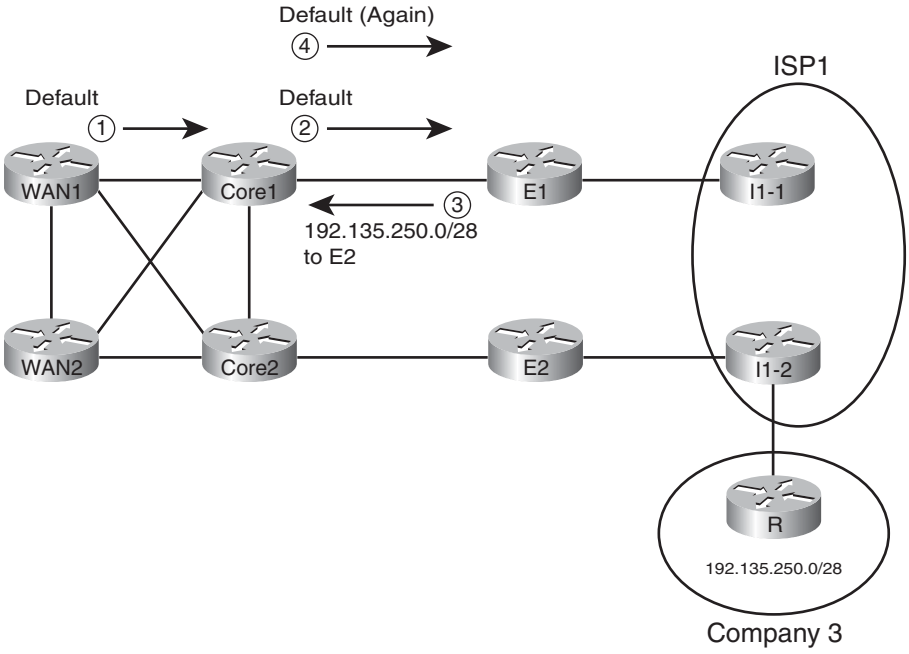
Step 3. WAN2 matches its iBGP-learned route for 192.135.250.0/28, forwarding to Core2.

Step 4. Core2 matches its iBGP-learned route for 192.135.250.0/28, forwarding to E2.

Step 5. E2 matches its eBGP-learned route for 192.135.250.0/28, forwarding to I1-2.

Step 6. The routers in ISP1 forward the packet to Router I3, in Company 3.

The routers in the core of the Enterprise need to run BGP because without it, routing loops can occur. For example, if WAN1, WAN2, Core1, and Core2 did not use BGP, and relied on default routes, their default would drive packets to either E1 or E2. Then, E1 or E2 might send the packets right back to Core1 or Core2. (Note that there is no direct link between E1 and E2.) Figure 12-12 shows just such a case.



Key Topic

Figure 12-12 *A Routing Loop Without BGP in the Enterprise Core*

In this case, both E1 and E2 know that E2 is the best exit point for packets destined to 192.135.250.0/28 (from Figure 12-11). However, the core routers use default routes, with WAN1 and Core1 using defaults that send packets to E1. Following the numbers in the figure

Step 1. WAN1 gets a packet destined for 192.135.250.1 and forwards the packet to Core1 based on its default route.

Step 2. Core1 gets the packet and has no specific route, so it forwards the packet to E1 based on its default route.

Step 3. E1's BGP route tells it that E2 is the better exit point for this destination. To send the packet to E2, E1 forwards the packet to Core1.

Step 4. Core1, with no knowledge of the BGP route for 192.135.250.0/28, uses its default route to forward the packet to E1, so the packet is now looping.

A mesh of iBGP peerings between at least E1, E2, Core1, and Core2 would prevent this problem.

Partial and Full BGP Updates

Unfortunately, Enterprise routers must pay a relatively large price for the ability to choose between competing BGP routes to reach Internet destinations. As previously mentioned, the BGP table in the Internet core is at approximately 300,000 routes as of the writing of this chapter in 2009. To make a decision to use one path instead of another, an Enterprise router must know about at least some of those routes. Exchanging BGP information for such a large number of routes consumes bandwidth. It also consumes memory in the routers and requires some processing to choose the best routes. Some samples at Cisco.com show BGP using approximately 70 MB of RAM for the BGP table on a router with 100,000 BGP-learned routes.

To make matters a bit worse, in some cases, several Enterprise routers may also need to use BGP, as shown in the previous section. Those routers also need more memory to hold the BGP table, and they consume bandwidth exchanging the BGP table.

To help reduce the memory requirements of receiving full BGP updates (BGP updates that include all routes), ISPs give you three basic options for what routes the ISP advertises:

- **Default route only:** The ISP advertises a default route with BGP, but no other routes.
- **Full updates:** The ISP sends you the entire BGP table.
- **Partial updates:** The ISP sends you routes for prefixes that might be better reached through that ISP, but not all routes, plus a default route (to use instead of the purposefully omitted routes as needed).

If all you want to do with a BGP connection is use it by default, then you can have the ISP send just a default route. If you are willing to take on the overhead of getting all BGP routes, then asking for full updates is reasonable. However, if you want something in between, the partial updates option is useful.

BGP partial updates give you the benefit of choosing the best routes for some destinations, while limiting the bandwidth and memory consumption. With partial updates, the ISP advertises routes for prefixes that truly are better reached through a particular link. However, for prefixes that may not be any better through that link, the ISP does not advertise those prefixes with BGP. Then the Enterprise routers can use the better path based on

the routes learned with BGP, and use a default route for the prefixes not learned with BGP. For example, previously in Figure 12-11, Router I1-2 could be configured to only advertise routes for those such as 192.135.250.0/28, from Company 3 in that figure–in other words, only routes for which Router I1-2 had a clearly better route than the other ISP1 routers.

Single Multihomed

A single-multihomed topology means a single link per ISP, but multiple (at least 2) ISPs. Figure 12-13 shows a couple of single-multihomed designs, each with two ISPs:

The single-multihomed design has some similarities with both the single-homed and dual-homed designs previously seen in this section. The single-multihomed design on the top of the figure, which uses a single router, acts like the single-homed design for default routes in the Enterprise. This design can flood a default route throughout the Enterprise, drawing traffic to that one router, because only one router connects to the Internet. With the two-router design on the lower half of Figure 12-13, defaults can still be used in the

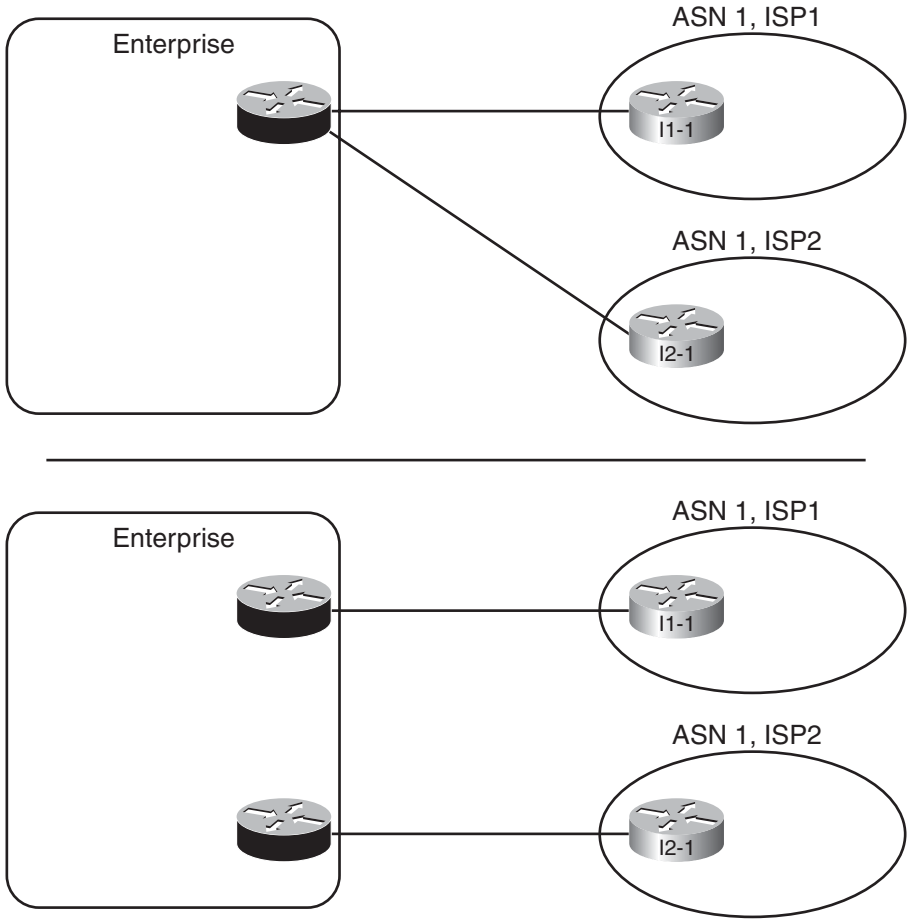


Figure 12-13 *Single-Multihomed Designs*

Enterprise to draw traffic to the preferred Internet connection (if one is preferred) or to balance traffic across both.

The single-multihomed design works like the dual-homed design in some ways because two (or more) links connect the Enterprise to the Internet. With two links, the Internet design might call for the use of defaults, always preferring one of the links. The design engineer might also choose to use BGP, learn either full or partial updates, and then favor one connection over another for some of the routes.

Figure 12-14 shows these concepts with a single-multihomed design, with default routes in the Enterprise to the one Internet router (E1).

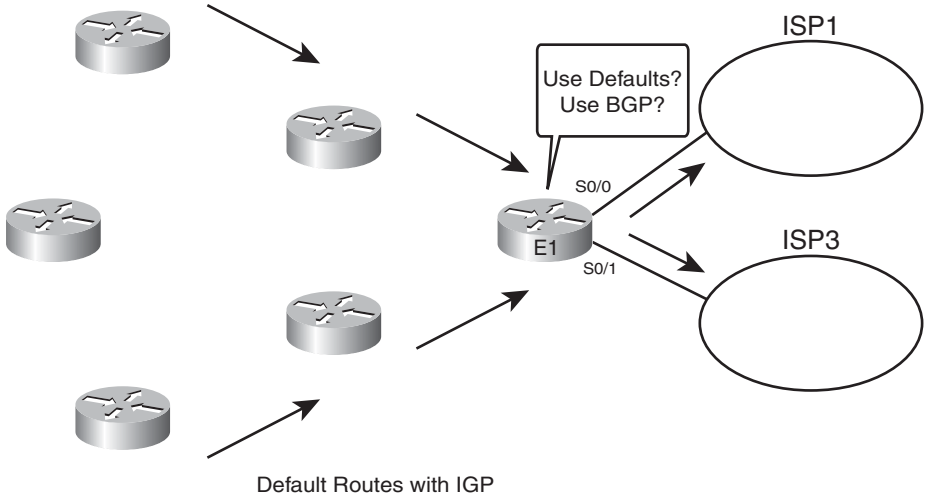
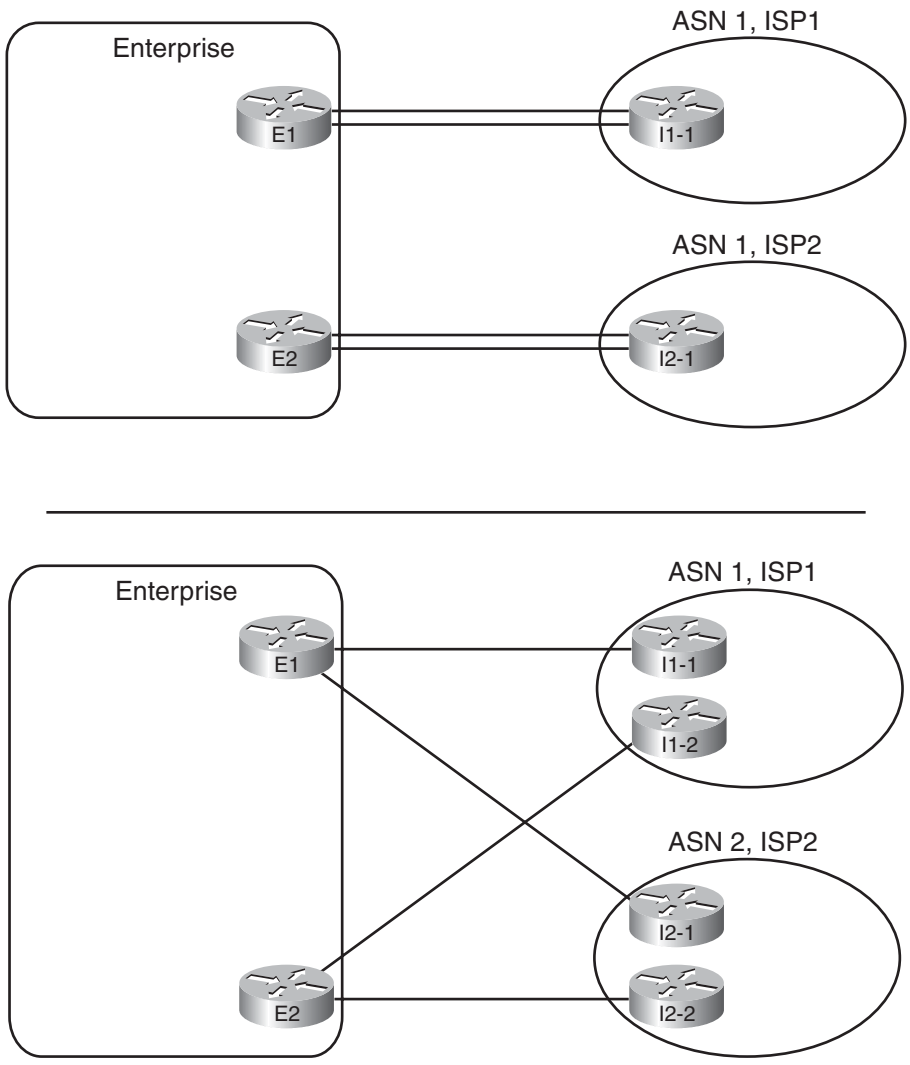


Figure 12-14 *Outbound Routing with a Single-Multihomed Design*

Dual Multihomed

The last general category of Internet access topologies is called dual multihomed. With this design, two or more ISPs are used, with two or more connections to each. A number of different routers can be used. Figure 12-15 shows several examples.

Figure 12-15 does not show all design options, but because at least two ISPs exist, and at least two connections per ISP, much redundancy exists. That redundancy can be used for backup, but most often, BGP is used to make some decisions about the best path to reach various destinations.



Key Topic

Figure 12-15 *Dual-Multihomed Options*

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 12-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 12-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A design shows a single router connected to the Internet as part of a single-homed Internet design. It lists sections for Enterprise routing toward the Internet-facing router(s) in the Enterprise, and another section for choosing routes on the Internet-facing router into the Internet. List the reasonable options.	
Use the same criteria as the previous item in this table, except the single Enterprise router connected to the Internet now has two links to the same ISP (dual homed).	
Use the same criteria as the previous item, except use two routers with one link each to the same ISP (dual homed).	
Same criteria as previous row, but with a single multihomed with two routers.	

Implementation Plan Peer Review Table

Table 12-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 12-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a single router in a dual-homed Internet design, with the router using BGP over each link to that same ISP. What criteria would impact your choice of accepting only default routes, or partial updates, or full updates, using BGP in this case? (3)	
The plan shows four Enterprise routers with BGP configuration, with two of those routers with links to two different ISPs. Which connections are eBGP? iBGP?	

Create an Implementation Plan Table

This chapter does not focus on implementation or verification, but it did review one concept about static routes, as listed in Table 12-7.

Table 12-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configuring multiple static default routes, each with different administrative distance settings	

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 12-8 lists a reference of these key topics and the page numbers on which each is found.

Table 12-8 *Key Topics for Chapter 12*

Key Topic Element	Description	Page Number
Figure 12-1	Public IP Address Assignment model	391
Table 12-4	Comparisons of OSPF and EIGRP to BGP	396
List	Two key functions for BGP AS_Path	397
Figure 12-5	BGP process to update AS_Path when advertising NLRI	399
Figure 12-7	Demonstration of how AS_Path can be used to prevent loops	401
Table 12-5	16-bit BGP ASN assignment ranges	401

List	Description of terms single homed, dual homed, single multihomed, and dual multihomed	404
Figure 12-9	Common dual-homed designs	406
Figure 12-12	Example routing loop due to lack of iBGP in the Enterprise core	409
List	Three options for the routes received from an ISP	410
Figure 12-15	Common dual-multihomed options	413

Complete the Tables and Lists from Memory

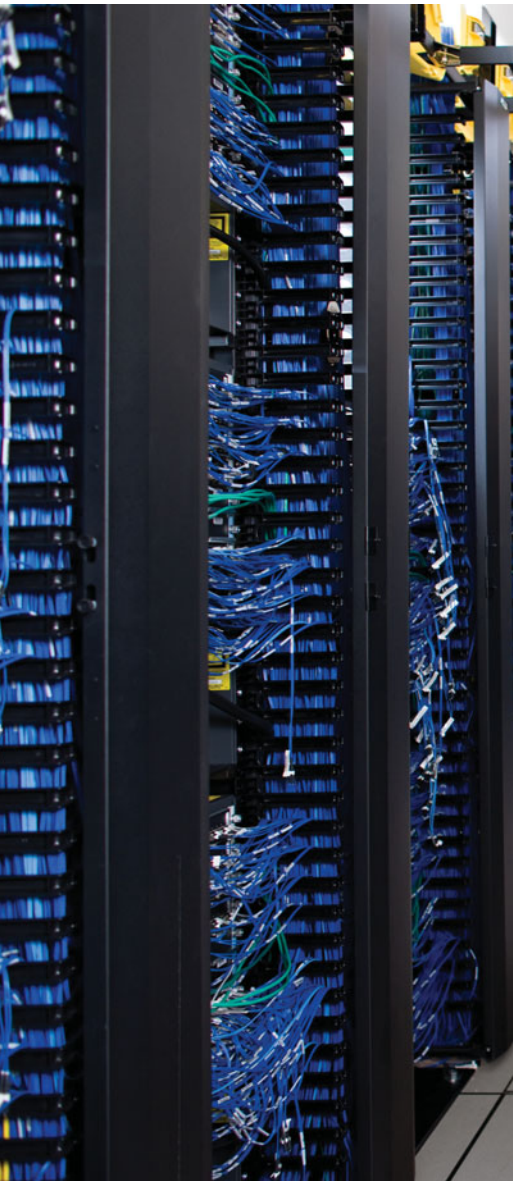
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

public IP address, private IP address, Network Address Translation (NAT), Port Address Translation (PAT), AS Sequence, Path Attribute (PA), AS path, public ASN, private ASN, default route, single homed, dual homed, single multihomed, dual multihomed, Autonomous System Number (ASN)

This page intentionally left blank



This chapter covers the following subjects:

External BGP for Enterprises: This section examines the required configuration for external BGP connections, plus a few optional but commonly used configuration settings. It also examines the commands used to verify that eBGP works.

Verifying the BGP Table: This section discusses the contents of the BGP table, particularly the routes learned using eBGP connections.

Injecting Routes into BGP for Advertisement to the ISPs: This section shows how you can configure an eBGP router to advertise the public IP address range used by an Enterprise.

CHAPTER 13

External BGP

BGP configuration differs slightly when comparing the configuration of an External BGP (eBGP) peer and an Internal BGP (iBGP) peer. Many small differences in operation exist as well.

This chapter focuses on external BGP configuration and verification. The chapter begins with a discussion of the fundamentals of BGP configuration that applies to both internal and external peers. It then discusses the reasons why eBGP peers may or may not benefit from additional optional configuration settings.

After the eBGP neighborships have been established, the BGP routers exchange routes. The chapter examines the verification commands used to see the BGP routes learned from eBGP neighbors by examining the BGP table. The chapter closes with a discussion of how to configure the **network** command and to configure route redistribution to make a BGP router advertise an Enterprise's public IP prefix to its eBGP peers.

Chapter 14, "Internal BGP and BGP Route Filtering," discusses the particulars of iBGP configuration and verification, plus BGP route filtering (for both iBGP and eBGP).

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 13-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 13-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
External BGP for Enterprises	1–4
Verifying the BGP Table	5–7
Injecting Routes into BGP for Advertisement to the ISPs	8

1. Enterprise Router R1, in ASN 1, connects to ISP Router I1, ASN 2, using eBGP. The single serial link between the two routers uses IP addresses 10.1.1.1 and 10.1.1.2, respectively. Both routers use their S0/0 interfaces for this link. Which of the following commands would be needed on R1 to configure eBGP? (Choose two.)

   a. **router bgp 2**
   b. **router bgp 1**
   c. **neighbor 10.1.1.2 remote-as 2**
   d. **neighbor 10.1.1.2 Update-source 10.1.1.1**
   e. **neighbor 10.1.1.2 Update-source S0/0**

2. Enterprise Router R1, in ASN 1, connects to ISP Router I1, ASN 2, using eBGP. There are two parallel serial links between the two routers. The implementation plan calls for each router to base their BGP TCP connection on their respective loopback1 interfaces, with IP addresses 1.1.1.1 and 2.2.2.2, respectively. Which of the following commands would not be part of a working eBGP configuration on Router R1?

   a. **router bgp 1**
   b. **neighbor 2.2.2.2 remote-as 2**
   c. **neighbor 2.2.2.2 update-source loopback1**
   d. **neighbor 2.2.2.2 multihop 2**

3. The following output, taken from a **show ip bgp** command on Router R1, lists two neighbors. In what BGP neighbor state is neighbor 1.1.1.1?

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
1.1.1.1         4     1      60      61       26    0    0 00:45:01
0
2.2.2.2         4     3     153     159       26    0    0 00:38:13
1
```

   a. Idle
   b. Opensent
   c. Active
   d. Established

4. The following output was taken from the **show ip bgp** command on Router R2. In this case, which of the following commands are most likely to already be configured on R2? (Choose two.)

```
BGP router identifier 11.11.11.11, local AS number 11
...
Neighbor         V     AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
1.1.1.1          4      1       87       87         0    0    0 00:00:06 Idle
(Admin)
2.2.2.2          4      3      173      183        41    0    0 00:58:47
2
```

   a. **router bgp 11**
   b. **neighbor 1.1.1.1 remote-as 11**
   c. **neighbor 2.2.2.2 prefix-limit 1**
   d. **neighbor 1.1.1.1 shutdown**

5. Which of the following answers is most true about the BGP Update message?

   a. It lists a set of path attributes, along with a list of prefixes that use those PAs.
   b. It lists a prefix/length, plus the PA settings for that prefix.
   c. It lists withdrawn routes, but never in the same Update message as newly advertised routes.
   d. A single Update message lists at most a single prefix/length.

6. The following output occurs on Router R1. Which of the following cannot be determined from this output?

```
R1# show ip route 180.1.1.0 255.255.255.240
Routing entry for 180.1.1.0/28
  Known via "bgp 2", distance 20, metric 0
  Tag 3, type external
  Last update from 192.168.1.2 00:10:27 ago
  Routing Descriptor Blocks:
  * 192.168.1.2, from 192.168.1.2, 00:10:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3
```

   a. The type of BGP peer (iBGP or eBGP) that advertised this route to R1
   b. R1's ASN
   c. The next-hop router's ASN
   d. The AS_Path length

7. The following line of output was extracted from the output of the **show ip bgp** command on Router R1. Which of the following can be determined from this output?

```
   Network          Next Hop            Metric LocPrf Weight Path
*  130.1.1.0/28     1.1.1.1                                  0 1 2 3 4 i
```

   a. The route is learned from an eBGP peer.
   b. The route has no more than three ASNs in the AS_Path.
   c. The route is the best route for this prefix.
   d. None of these facts can be positively determined by this output.

8. Router R1 has eBGP connections to I1 and I2, routers at the same ISP. The company that owns R1 can use public address range 130.1.16.0/20. The following output lists all the IP routes in R1's routing table within this range. Which of the following answers would cause R1 to advertise the 130.1.16.0/20 prefix to its eBGP peers? (You should assume default settings for any parameters not mentioned in this question.)

```
R1# show ip route 130.1.16.0 255.255.240.0 longer-prefixes
! lines omitted...
O       130.1.16.0/24 [110/3] via 10.5.1.1, 00:14:36, FastEthernet0/1
O       130.1.17.0/24 [110/3] via 10.5.1.1, 00:14:36, FastEthernet0/1
O       130.1.18.0/24 [110/3] via 10.5.1.1, 00:14:36, FastEthernet0/1
```

   a. Configure R1 with the **network 130.1.16.0 mask 255.255.240.0** command.
   b. Configure R1 with the **network 130.1.16.0 mask 255.255.240.0 summary-only** command.
   c. Redistribute from OSPF into BGP, filtering so that only routes in the 130.1.16.0/20 range are redistributed.
   d. Redistribute from OSPF into BGP, filtering so that only routes in the 130.1.16.0/20 range are redistributed, and create a BGP summary for 130.1.16.0/20.

Foundation Topics

External BGP for Enterprises

Some of the core operational concepts of BGP mirror those of EIGRP and OSPF. BGP first forms a neighbor relationship with peers. BGP then learns information from its neighbors, placing that information in a table—the BGP table. Finally, BGP analyzes the BGP table to choose the best working route for each prefix in the BGP table, placing those routes into the IP routing table.

This section discusses external BGP (eBGP), focusing on two of the three aspects of how a routing protocol learns routes: forming neighborships and exchanging the reachability or topology information that is stored in the BGP table. First, this section examines the baseline configuration of eBGP peers (also called neighbors), along with several optional settings that may be needed specifically for eBGP connections. This configuration should result in working BGP neighborships. Then, this section examines the BGP table, listing the prefix/length and path attributes (PA) learned from the Internet, and the IP routing table.

eBGP Neighbor Configuration

At a minimum, a router participating in BGP needs to configure the following settings:

- The router's own ASN (**router bgp** *asn* global command)
- The IP address of each neighbor and that neighbor's ASN (**neighbor** *ip-address* **remote-as** *remote-asn* BGP subcommand)

For example, consider a typical multihomed Enterprise Internet design, as shown in Figure 13-1. In this case, the following design requirements have already been decided, but you must then determine the configuration, knowing the information in the following list and the figure:

- The Enterprise uses ASN 11.
- The connection to ISP1 (two T-1's) is considered the primary connection, with the connection to ISP3 (one T-1) being secondary.
- ISP1 advertises a default route, plus full updates.
- ISP1 uses ASN 1.
- ISP3 advertises a default route, plus partial updates that includes only ISP3's local customers.
- ISP3 uses ASN 3.
- Each ISP uses the IP address of its lowest-numbered interface for its peer relationships.

For Router E1, the BGP configuration requires only three commands, at least to configure BGP to the point where E1 will form neighborships with the two other routers. (Note that this chapter continues to change and add to this configuration when introducing new concepts.) The example also shows the configuration on Routers I1-1 and I3-1 added

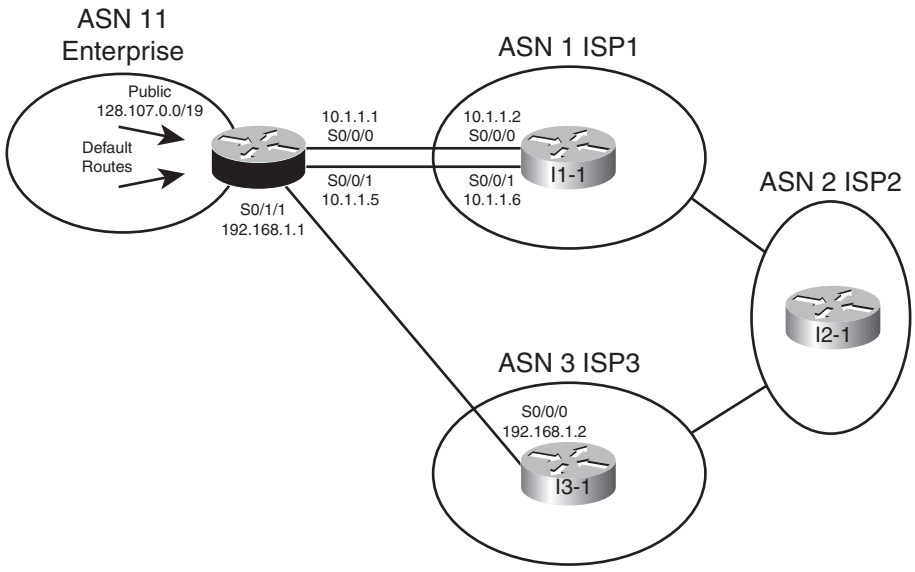


Figure 13-1 *Sample Single Multihomed Design*

solely to support the neighbor connections to E1; other BGP configuration on these routers is not shown.

Example 13-1 *BGP Configuration on E1: Neighborships Configured*

```
! Configuration on router E1
router bgp 11
 neighbor 10.1.1.2 remote-as 1
 neighbor 192.168.1.2 remote-as 3
! Next commands are on I1-1
router bgp 1
 neighbor 10.1.1.1 remote-as 11
! Next commands are on I3-1
router bgp 3
 neighbor 192.168.1.1 remote-as 11
```

The gray portions of the output highlight the configuration of the local ASN and the neighbors' ASNs – parameters that must match for the neighborships to form. First, E1 configures its own ASN as 11 by using the **router bgp 11** command. The other routers must refer to ASN 11 on the **neighbor** commands that refer to E1; in this case I1-1 refers to ASN 11 with its **neighbor 10.1.1.1 remote-as 11** command. Conversely, I1-1's local ASN (1) on its **router bgp 1** global command must match what E1 configures in a **neighbor** command, in this case with E1's **neighbor 10.1.1.2 remote-as 1** command.

Requirements for Forming eBGP Neighborships

Routers must meet several requirements to become BGP neighbors:

- A local router's ASN (on the **router bgp** *asn* command) must match the neighboring router's reference to that ASN with its **neighbor remote-as** *asn* command.
- The BGP router IDs of the two routers must not be the same.
- If configured, MD5 authentication must pass.
- Each router must be part of a TCP connection with the other router, with the remote router's IP address used in that TCP connection matching what the local router configures in a BGP **neighbor remote-as** command.

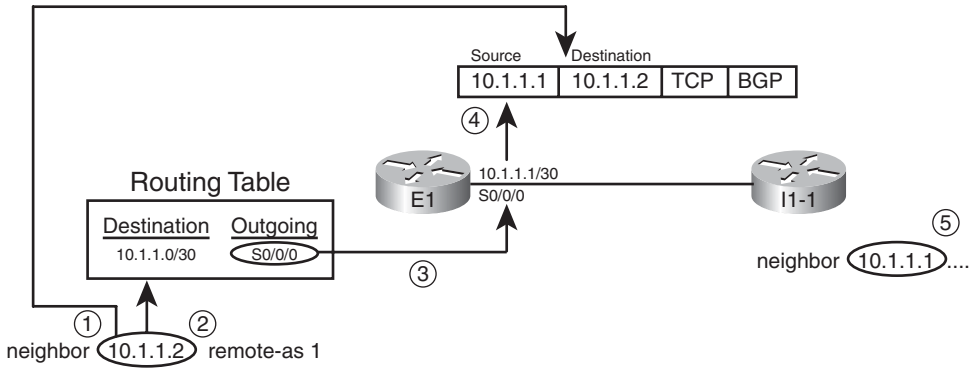
Consider the first two items in this list. First, the highlights in Example 13-1 demonstrate the first of the four requirements. Next, the second requirement in the list requires only a little thought if you recall the similar details about router IDs (RID) with EIGRP and OSPF. Like EIGRP and OSPF, BGP defines a 32-bit router ID, written in dotted-decimal notation. And like EIGRP and OSPF, BGP on a router chooses its RID the same general way, by using the following steps, in order, until a BGP RID has been chosen:

- **Configured:** Use the setting of the **bgp router-id** *rid* router subcommand.
- **Highest Loopback:** Choose the highest numeric IP address of any up/up loopback interface, at the time the BGP process initializes.
- **Highest other interface:** Choose the highest numeric IP address of any up/up non-loopback interface, at the time the BGP process initializes.

The third requirement in the list, the MD5 authentication check, occurs only if authentication has been configured on at least one of the two routers, using the **neighbor** *neighbor-ip* **password** *key* command. If two BGP neighbors configure this command, referring to the other routers' IP address, while configuring a matching MD5 authentication key value, the authentication passes. If both omit this command, then no authentication occurs, and the neighbor can still work. However, if the keys do not match, or if only one router configures authentication, then authentication fails.

The fourth neighbor requirement–that the IP addresses used for the neighbor TCP connection match–requires a more detailed discussion. BGP neighbors first form a TCP connection; later, BGP messages flow over that connection, which allows BGP routers to know when the messages arrived at the neighbor, and when they did not.

A BGP router creates the TCP connection by trying to establish a TCP connection to the address configured in the **neighbor** *neighbor-ip* **remote-as** command. However, IOS does not require the BGP configuration to explicitly state the source address that router uses when establishing this TCP connection, and if not explicitly configured, IOS picks an IP address on the local router. By default, IOS chooses its BGP source IP address for a given neighbor as the interface IP address of the outgoing interface of the route used to forward packets to that neighbor. That's a lot of words to fight through, and much more easily seen with a figure, such as Figure 13-2, which focuses on the eBGP connection between E1 and I1-1 shown in Figure 13-1.



Key Topic **Figure 13-2** *A Default Choice for Update Source*

A description of the steps in logic shown in the figure follows:

Step 1. E1 finds the **neighbor 10.1.1.2** command, so E1 sends the BGP messages for this neighbor inside packets with destination IP address 10.1.1.2.

Step 2. E1 looks in the IP routing table for the route that matches destination 10.1.1.2.

Step 3. The route matched in Step 2 lists S0/0/0 as outgoing interface.

Step 4. E1's interface IP address for S0/0/0 is 10.1.1.1, so E1 uses 10.1.1.1 as its source IP address for this BGP peer.

Step 5. The **neighbor** command on the other router, I1-1, must refer to E1's source IP address (10.1.1.1 in this case).

Now, consider again the last of the four requirements to become neighbors. Restated, for proper operation, the BGP update source on one router must match the IP address configured on the other router's **neighbor** command, and vice versa. As shown in Figure 13-2, E1 uses update source 10.1.1.1, with I1-1 configuring the **neighbor 10.1.1.1...** command. Conversely, I1-1 uses 10.1.1.2 as its update source for this neighbor relationship, with E1 configuring a matching **neighbor 10.1.1.2** command.

Note: The update source concept applies per neighbor.

Issues When Redundancy Exists Between eBGP Neighbors

In many cases, a single Layer 3 path exists between eBGP neighbors. For example, a single T1, or single T3, or maybe a single MetroE Virtual Private Wire Service (VPWS) path exists between the two routers. In such cases, the eBGP configuration can simply use the interface IP addresses on that particular link. For example, in Figure 13-1, a single serial link exists between Routers E1 and I3-1, and they can reasonably use the serial link's IP addresses, as shown in Example 13-1.

However, when redundant Layer 3 paths exist between two eBGP neighbors, the use of interface IP addresses for the underlying TCP connection can result in an outage when only one of the two links fails. BGP neighborships fail when the underlying TCP connection

fails. TCP uses a concept called a *socket*, which consists of a local TCP port number and an IP address. That IP address must be associated with a working interface (an interface whose state is line status up, line protocol up, per the **show interfaces** command). If the interface whose IP address is used by BGP were to fail, then the TCP socket would fail, closing the TCP connection. As a result, the BGP neighborship can only be up when the associated interfaces also happens to be up.

Two alternative solutions exist in this case. One option would be to configure two **neighbor** commands on each router, one for each of the neighbor's interface IP addresses. This solves the availability issue, because if one link fails, the other neighborship can remain up and working. However, in this case, both neighborships exchange BGP routes, consuming bandwidth and more memory in the BGP table.

The preferred option, which uses loopback interfaces as the TCP connection endpoints, solves the availability problem while avoiding the extra overhead. The two routers each configure a loopback interface and IP address, and use those loopback IP addresses as the source of their single BGP TCP connection. If one of the multiple links fails, the loopback interface does not fail. As long as the two routers have working routes to reach each other's loopback IP addresses, the TCP connection does not fail.

Configuring eBGP peers to use a loopback interface IP address with BGP requires several steps, as follows:

Step 1. Configure an IP address on a loopback interface on each router.

Step 2. Tell BGP on each router to use the loopback IP address as the source IP address using the **neighbor... update-source** *ip-address* command.

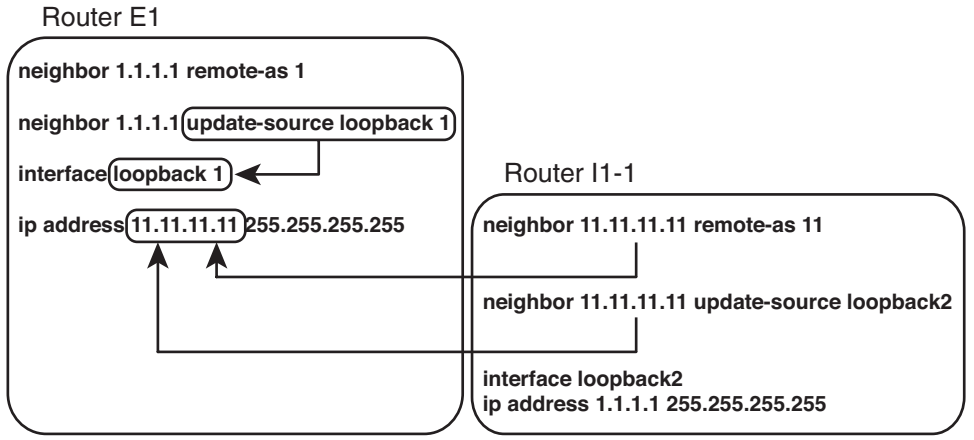
Step 3. Configure the BGP **neighbor** command on each router to refer to the other router's loopback IP address at the neighbor IP address in the **neighbor** *neighbor-ip* **remote-as** command.

Step 4. Make sure each router has IP routes so that they can forward packets to the loopback interface IP address of the other router.

Step 5. Configure eBGP multihop using the **neighbor... ebgp-multihop** *hops* command.

The first three steps in the list require configuration on both the routers. Figure 13-3 shows the details related to the first three steps, focusing on Router E1's use of its loopback1 interface (based on Figure 13-1).

The fourth step in the list is an overt reminder that for TCP to work, both routers must be able to deliver packets to the IP address listed in the **neighbor** commands. Because the **neighbor** commands now refer to loopback IP addresses, the routers cannot rely on connected routes for forwarding the packets. To give each router a route to the other router's loopback, you can run an instance of an IGP to learn the routes, or just configure static routes. If using static routes, make sure to configure the routes so that all redundant paths would be used (as seen in upcoming Example 13-2). If using an IGP, make sure the configuration allows the two routers to become IGP neighbors over all redundant links as well.



Key Topic **Figure 13-3** *Using Loopbacks with Update Source for eBGP*

eBGP Multihop Concepts

The fifth configuration step for using loopback IP addresses with eBGP peers refers to a feature called *eBGP multihop*. By default, when building packets to send to an eBGP peer, IOS sets the IP Time-To-Live (TTL) field in the IP header to a value of 1. With this default action, the eBGP neighborship fails to complete when using loopback interface IP addresses. The reason is that when the packet with TTL=1 arrives at the neighbor, the neighbor discards the packet.

The logic of discarding the BGP packets may be a bit surprising, so an example can help. For this example, assume the default action of TTL=1 is used and that eBGP multihop is not configured yet. Router E1 from Figure 13-1 is trying to establish a BGP connection to I1-1, using I1-1's loopback IP address 1.1.1.1, as shown in Figure 13-3. The following occurs with the IP packets sent by E1 when attempting to form a TCP connection for BGP to use:

Step 1. E1 sends a packet to destination address 1.1.1.1, TTL=1.

Step 2. I1-1 receives the packet. Seeing that the packet is not destined for the receiving interface's IP address, I1-1 passes the packet off to its own IP forwarding (IP routing) logic.

Step 3. I1-1's IP routing logic matches the destination (1.1.1.1) with the routing table and finds interface loopback 2 as the outgoing interface.

Step 4. I1-1's IP forwarding logic decrements TTL by 1, decreasing TTL to 0, and as a result, I1-1 discards the packet.

In short, the internal IOS packet forwarding logic decrements the TTL before giving the packet to the loopback interface, meaning that the normal IP forwarding logic discards the packet.

Configuring the routers with the **neighbor ebgp-multihop 2** command, as seen in upcoming Example 13-2, solves the problem. This command defines the TTL that the router will use when creating the BGP packets (2 in this case). As a result, the receiving router will decrement the TTL to 1, so the packet will not be discarded.

Configuring for eBGP Redundancy and Authentication

To pull these optional configuration steps together, Example 13-2 shows a new configuration for E1 and I1-1 (as compared with Example 13-1). In this case:

- Both routers use BGP authentication, with authentication key **fred**.
- Two Layer 3 paths exist between E1 and I1-1, so the configuration uses all the steps required to make both routers use the loopback interfaces listed in Figure 13-3.
- Both routers use eBGP multihop. (Otherwise, the neighborships would fail.)
- Both routers use static routes to provide reachability to the loopback interfaces, with two routes each, one using each redundant path.

Example 13-2 *Broader eBGP Configuration Example*

```
! Configuration on router E1
interface loopback 1
 ip address 11.11.11.11 255.255.255.255
!
ip route 1.1.1.1 255.255.255.255 s0/0/0
ip route 1.1.1.1 255.255.255.255 s0/0/1
!
router bgp 11
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 update-source loopback 1
 neighbor 1.1.1.1 ebgp-multihop 2
 neighbor 1.1.1.1 password fred
```

```
! Next commands are on I1-1
interface loopback 2
 ip address 1.1.1.1 255.255.255.255
!
ip route 11.11.11.11 255.255.255.255 s0/0/0
ip route 11.11.11.11 255.255.255.255 s0/0/1
!
router bgp 1
 neighbor 11.11.11.11 remote-as 11
 neighbor 11.11.11.11 update-source loopback 2
 neighbor 11.11.11.11 ebgp-multihop 2
 neighbor 11.11.11.11 password fred
```

Besides listing the various configuration commands, this example is also the first in this chapter that demonstrates how to configure multiple settings for a single BGP neighbor. The IOS BGP **neighbor** command has many options. To set those options for a single neighbor, the command begins with **neighbor** *neighbor-ip-address*, followed by the parameters. So, instead of one **neighbor** command with a large number of parameters, the configuration includes several **neighbor** commands, each listing the IP address of the same neighbor, but each listing a different set of parameters. In this case, each router now uses

four **neighbor** commands: one to define the remote ASN, one for the update source, one to enable eBGP multihop, and one for the authentication key.

BGP Internals and Verifying eBGP Neighbors

Similar to OSPF, the BGP neighbor relationship goes through a series of states over time. Although the Finite State Machine (FSM) for BGP neighbor states has many twists and turns, particularly for handling exceptions, retries, and failures, the overall process works as follows:

Step 1. A router tries to establish a TCP connection with the IP address listed on a **neighbor** command, using well-known destination port 179.

Step 2. When the three-way TCP connection completes, the router sends its first BGP message, the BGP *Open* message, which generally performs the same function as the EIGRP and OSPF Hello messages. The Open message contains several BGP parameters, including those that must be verified before allowing the routers to become neighbors.

Step 3. After an Open message has been sent and received and the neighbor parameters match, the neighbor relationship is formed, and the neighbors reach *established* state.

Table 13-2 lists the various BGP states. If all works well, the neighborship reaches the final state: *established*. When the neighbor relationship (also called a BGP peer or BGP peer connection) reaches the established state, the neighbors can send BGP Update messages, which list PAs and prefixes. However, if neighbor relationship fails for any reason, then the neighbor relationship can cycle through all the states listed in Table 13-2 while the routers periodically attempt to bring up the neighborship.

Key Topic

Table 13-2 *BGP Neighbor States*

State	Typical Reasons
Idle	The BGP process is either administratively down or awaiting the next retry attempt.
Connect	The BGP process is waiting for the TCP connection to be completed. You cannot determine from this state information whether the TCP connection can complete.
Active	The TCP connection has been completed, but no BGP messages have been sent to the peer yet.
Opensent	The TCP connection exists, and a BGP Open message has been sent to the peer, but the matching Open message has not yet been received from the other router.
Openconfirm	An Open message has been both sent to and received from the other router. The next step is to receive a BGP Keepalive message (to confirm all neighbor-related parameters matched) or BGP Notification message (to learn there is some mismatch in neighbor parameters).

Table 13-2 *BGP Neighbor States*

State	Typical Reasons
Established	All neighbor parameters match, the neighbor relationship works, and the peers can now exchange Update messages.

Verifying eBGP Neighbor Status

The two most common commands to display a BGP neighbor's status are **show ip bgp summary** and **show ip bgp neighbors** [*neighbor-id*]. Interestingly, most people use the first of these commands because it supplies a similar amount of information, 1 line per neighbor, as do the familiar **show ip eigrp neighbors** and **show ip ospf neighbor** commands. The **show ip bgp neighbors** command lists a large volume of output per neighbor, which, although useful, usually contains far too much information for the verification of the current neighbor state. Examples 13-3 and 13-4 show samples of the output of each of these two commands on Router E1, respectively, based on the configuration shown in Example 13-2, with some description following each example.

Example 13-3 *Summary Information with the* **show ip bgp summary** *Command*

```
E1# show ip bgp summary
BGP router identifier 11.11.11.11, local AS number 11
BGP table version is 26, main routing table version 26
6 network entries using 792 bytes of memory
7 path entries using 364 bytes of memory
6/4 BGP path/bestpath attribute entries using 888 bytes of memory
5 BGP AS-PATH entries using 120 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 2) using 32 bytes of memory
BGP using 2196 total bytes of memory
BGP activity 12/6 prefixes, 38/31 paths, scan interval 60 secs

Neighbor        V    AS  MsgRcvd  MsgSent  TblVer  InQ OutQ  Up/Down  State/PfxRcd
1.1.1.1         4     1       60       61      26    0    0  00:45:01        6
192.168.1.2     4     3      153      159      26    0    0  00:38:13        1
```

The first line in the summary lists the local router's BGP RID (11.11.11.11), along with the local router's ASN (11). The rest of the summary focuses on statistics for the BGP table entries. The bottom of the output lists a heading line (highlighted in the output), plus one line per neighbor, with two neighbors in this case. The Neighbor column lists the IP address as defined on the local router's **neighbor** command and not the neighbor's BGP RID. Other notable information includes the neighbor's ASN (as configured on the local router's **neighbor remote-as** command), the time spent in the current state, and an interesting heading: State/PfxRcd.

This final heading, State/PfxRcd, either lists the BGP neighbor state, as summarized in Table 13-2, or the number of prefixes received (PfxRcd) from that neighbor. A numeric value under this heading implies a neighbor state of established, because the peers must be in established state before Updates can be sent. If the peer is not in an established state, the value in this heading lists the text name of the current BGP state.

Example 13-4 shows a sample of the **show ip bgp neighbors 1.1.1.1** command on router E1, which displays information about the connection to Router I1-1 in Figure 13-1. This command lists several facts not seen in the shorter **show ip bgp summary** command output in Example 13-3. The example highlights some of those key items, with the following comments referring to those highlighted items, in order:

- The neighbor is an eBGP neighbor (external link).
- The neighbor's BGP RID (1.1.1.1).
- The current state (Established) is explicitly listed.
- Route refresh is enabled (as referenced in Chapter 14's section titled "Clearing BGP Neighbors").
- The eBGP multihop setting (2 hops).
- Local and remote TCP socket information (IP addresses and port numbers).

Example 13-4 *Detailed Information with the* **show ip bgp neighbors** *Command*

```
E1# show ip bgp neighbors 1.1.1.1
BGP neighbor is 1.1.1.1,  remote AS 1, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:45:08
  Last read 00:00:02, last write 00:00:38, hold time is 180, keepalive interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  2          2
    Notifications:          0          0
    Updates:               16         12
    Keepalives:            43         47
    Route Refresh:          0          0
    Total:                 61         61
  Default minimum time between advertisement runs is 30 seconds
```

```
 For address family: IPv4 Unicast
  BGP table version 26, neighbor version 26/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                                  Sent       Rcvd
  Prefix activity:                ----       ----
    Prefixes Current:               6          6 (Consumes 312 bytes)
    Prefixes Total:                19          7
    Implicit Withdraw:             11          0
    Explicit Withdraw:              2          1
    Used as bestpath:             n/a          5
    Used as multipath:            n/a          0

                                  Outbound    Inbound
  Local Policy Denied Prefixes:   --------    -------
    AS_PATH loop:                    n/a          2
    Total:                             0          2
  Number of NLRIs in the update sent: max 3, min 1

  Address tracking is enabled, the RIB does have a route to 1.1.1.1
  Connections established 2; dropped 1
  Last reset 00:45:10, due to Peer closed the session
  External BGP neighbor may be up to 2 hops away.
  Transport(tcp) path-mtu-discovery is enabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 2
Local host: 11.11.11.11, Local port: 179
Foreign host: 1.1.1.1, Foreign port: 28995
Connection tableid (VRF): 0

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x8217A0):
Timer           Starts     Wakeups             Next
Retrans             49           0              0x0
TimeWait             0           0              0x0
AckHold             49          46              0x0
SendWnd              0           0              0x0
KeepAlive            0           0              0x0
GiveUp               0           0              0x0
PmtuAger             0           0              0x0
DeadWait             0           0              0x0
Linger               0           0              0x0
ProcessQ             0           0              0x0
```

```
iss: 2070882650  snduna: 2070884280  sndnxt: 2070884280      sndwnd:  15890
irs: 3327995414  rcvnxt: 3327996693  rcvwnd:       16156  delrcvwnd:    228

SRTT: 300 ms, RTTO: 306 ms, RTV: 6 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 300 ms, ACK hold: 200 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable, md5
IP Precedence value : 6

Datagrams (max data segment is 516 bytes):
Rcvd: 98 (out of order: 0), with data: 50, total data bytes: 1278
Sent: 99 (retransmit: 0, fastretransmit: 0, partialack: 0, Second Congestion: 0),
with data: 50, tot
al data bytes: 1629
 Packets received in fast path: 0, fast processed: 0, slow path: 0
 fast lock acquisition failures: 0, slow path: 0
E1# show tcp brief
TCB           Local Address               Foreign Address          (state)
66D27FE0      192.168.1.1.179             192.168.1.2.16489        ESTAB
66D27378      11.11.11.11.179             1.1.1.1.28995            ESTAB
```

Note that the end of the example shows another command that you can use to confirm the TCP socket details of the underlying TCP connection: **show tcp brief**.

Administratively Controlling Neighbor Status

Interestingly, Cisco IOS provides a means by which network operations personnel can administratively disable any BGP neighbor. To do so, the operator would enter BGP configuration mode and issue the **neighbor** *neighbor-ip* **shutdown** command. This command brings down the current neighbor to an idle state. Later, when the BGP connection should be brought up, the operator should repeat the process, but with the **no** version of the command (**no neighbor** *neighbor-ip* **shutdown**).

These commands can be particularly useful to try in lab when learning BGP. Teamed with the **debug ip bgp** command, you can bring down neighbors and see the somewhat-readable BGP messages. These messages list the BGP states from Table 13-2. It also shows the information inside the Open messages. Example 13-5 shows a sample, with the debug messages that note a state transition highlighted. The output also lists the **show ip bgp summary** command, with the administratively idle state created by the **neighbor 1.1.1.1 shutdown** BGP configuration command on Router E1.

Example 13-5 *BGP Shutdown and BGP Neighbor State Transitions*

```
E1# debug ip bgp
BGP debugging is on for address family: IPv4 Unicast
E1# conf t
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
E1(config)# router bgp 11
E1(config-router)# neighbor 1.1.1.1 shutdown
E1(config-router)#

*Aug 11 20:23:01.335: BGPNSF state: 1.1.1.1 went from nsf_not_active to
nsf_not_active
*Aug 11 20:23:01.335: BGP: 1.1.1.1 went from Established to Idle
*Aug 11 20:23:01.335: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down Admin. Shutdown

E1(config-router)# do show ip bgp summary
! lines omitted for brevity

Neighbor         V     AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1          4      1      87      87        0    0    0 00:00:06 Idle (Admin)
192.168.1.2      4      3     173     183       41    0    0 00:58:47         1

E1(config-router)# no neighbor 1.1.1.1 shutdown
E1(config-router)#
*Aug 11 20:23:26.571: BGP: 1.1.1.1 went from Idle to Active
*Aug 11 20:23:26.571: BGP: 1.1.1.1 open active, local address 11.11.11.11
*Aug 11 20:23:26.575: BGP: 1.1.1.1 read request no-op
*Aug 11 20:23:26.575: BGP: 1.1.1.1 went from Active to OpenSent
*Aug 11 20:23:26.575: BGP: 1.1.1.1 sending OPEN, version 4, my as: 11, holdtime
180 seconds
*Aug 11 20:23:26.579: BGP: 1.1.1.1 send message type 1, length (incl. header) 45
*Aug 11 20:23:26.583: BGP: 1.1.1.1 rcv message type 1, length (excl. header) 26
*Aug 11 20:23:26.587: BGP: 1.1.1.1 rcv OPEN, version 4, holdtime 180 seconds
*Aug 11 20:23:26.587: BGP: 1.1.1.1 rcv OPEN w/ OPTION parameter len: 16
*Aug 11 20:23:26.587: BGP: 1.1.1.1 rcvd OPEN w/ optional parameter type 2
(Capability) len 6
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has CAPABILITY code: 1, length 4
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has MP_EXT CAP for afi/safi: 1/1
*Aug 11 20:23:26.587: BGP: 1.1.1.1 rcvd OPEN w/ optional parameter type 2
(Capability) len 2
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has CAPABILITY code: 128, length 0
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has ROUTE-REFRESH capability(old) for all
address-families
*Aug 11 20:23:26.587: BGP: 1.1.1.1 rcvd OPEN w/ optional parameter type 2
(Capability) len 2
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has CAPABILITY code: 2, length 0
*Aug 11 20:23:26.587: BGP: 1.1.1.1 OPEN has ROUTE-REFRESH capability(new) for all
address-families
BGP: 1.1.1.1 rcvd OPEN w/ remote AS 1
*Aug 11 20:23:26.587: BGP: 1.1.1.1 went from OpenSent to OpenConfirm
*Aug 11 20:23:26.591: BGP: 1.1.1.1 went from OpenConfirm to Established
*Aug 11 20:23:26.591: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
*Aug 11 20:23:26.603: BGP_Router: unhandled major event code 128, minor 0
```

BGP Message Summary

So far, this chapter has mentioned three of the four BGP messages. For reference, Table 13-3 lists the four BGP messages, with comparisons to EIGRP messages for perspective.

Key Topic

Table 13-3 *BGP Message Types*

Message	Purpose	Similarity with EIGRP
Open	Used to establish a neighbor relationship and exchange basic parameters, including ASN and MD5 authentication values.	Hello
Keepalive	Sent on a periodic basis to maintain the neighbor relationship. The lack of receipt of a Keepalive message within the negotiated Hold timer causes BGP to bring down the neighbor connection.	Hello
Update	Used to exchange PAs and the associated prefix/length (NLRI) that use those attributes.	Update
Notification	Used to signal a BGP error; typically results in a reset to the neighbor relationship.	No direct equivalent

Verifying the BGP Table

When an Enterprise router has established its eBGP neighbor relationships, that router can advertise and learn routes using BGP. To learn routes, an Enterprise BGP router does not need additional configuration beyond the configuration of eBGP neighbors as discussed in the first section of this chapter. To advertise routes to eBGP peers, particularly the public IP address prefix(es) used by that Enterprise, the Enterprise BGP router needs some additional configuration, as discussed in the upcoming section "Injecting Routes into BGP for Advertisement to the ISPs."

The BGP table plays a key role in the process of learning and using routing information with BGP. A router stores all learned BGP prefixes and PAs in its BGP table. The router will later choose which route for each prefix is the best BGP route. The router can then advertise its BGP table to its neighbors, advertising only the best route for each prefix.

This section begins with a brief examination of the BGP Update process by which BGP neighbors exchange routing information. Next, the text looks at the various **show** commands that can be used to examine and confirm the contents of the BGP table.

The BGP Update Message

When a BGP neighborship reaches the established state, those neighbors begin sending BGP Update messages to each other. The router receiving an Update places those learned

prefixes into its BGP table, regardless of whether the route appears to be the best route. Like EIGRP and OSPF, BGP puts all learned routing information into its table, and then BGP processes all such potential routes to choose the best route for each prefix.

The BGP Update message itself can be revealing about the motivations behind BGP. Figure 13-4 shows the format of the Update message.

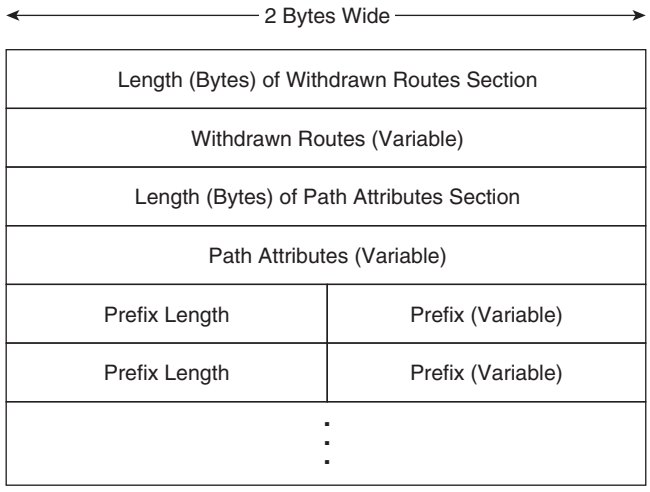


Figure 13-4 *Format of the BGP Update Message*

Interestingly, the format of the Update message tells us something about the nature of BGP as a *Path Vector* algorithm. The message lists a set of PAs and then a potentially long list of prefixes that use that set of PAs. So, you might view the BGP Update message as focusing on advertising paths, or a set of PAs, along with the associated list of prefixes that use the advertised path. (Both are important, of course.) Then, because BGP uses the information in the combined set of PAs to make a decision of which path is best, its underlying logic is called *path vector*.

Note: BGP also uses the term Network Layer Reachability Information (NLRI) to describe the IP prefix and length. This book uses the more familiar term *prefix*.

BGP uses the Update message to both announce and withdraw routes. For example, when a router realizes that a route in the router's BGP table has failed, that router withdraws that route by sending a BGP Update to its neighbors, listing the prefix in the list of withdrawn routes. When a router receives an Update that lists a prefix as withdrawn, that router knows that the route has failed. (Note the field near the top of the Update message that lists withdrawn routes.) That same Update message may contain other announced prefixes later in the Update message.

Examining the BGP Table

One of the key tasks in a BGP verification plan should be to examine the prefixes in the BGP table and confirm that the right prefixes have been learned from the expected neighbors. The BGP table should hold all learned prefixes, from each neighbor, except for any prefixes filtered by an inbound BGP filter. For example, in a router configured a **neighbor route-map in** command, the local router would first filter the routes and then add the allowed routes into the BGP table. (Chapter 14's section "Route Filtering and Clearing BGP Peers" discusses the filtering and its impact on the BGP table.)

As an example, consider Figure 13-5, which shows the same basic topology as Figure 13-1 but with only the information pertinent to the upcoming discussions listed in the figure. In this case, five prefixes exist somewhere in the Internet, with ISP1 and ISP3 learning these prefixes from ISP2. An additional prefix exists at the site of a customer of ISP3. The design calls for the following actions by ISP1 and ISP3 in their eBGP advertisements to the Enterprise:

- ISP1 should supply a default route plus full BGP updates.
- ISP3 should supply a default route plus partial BGP updates that include only ISP3's customers' prefixes (for example, 192.135.250.0/28).

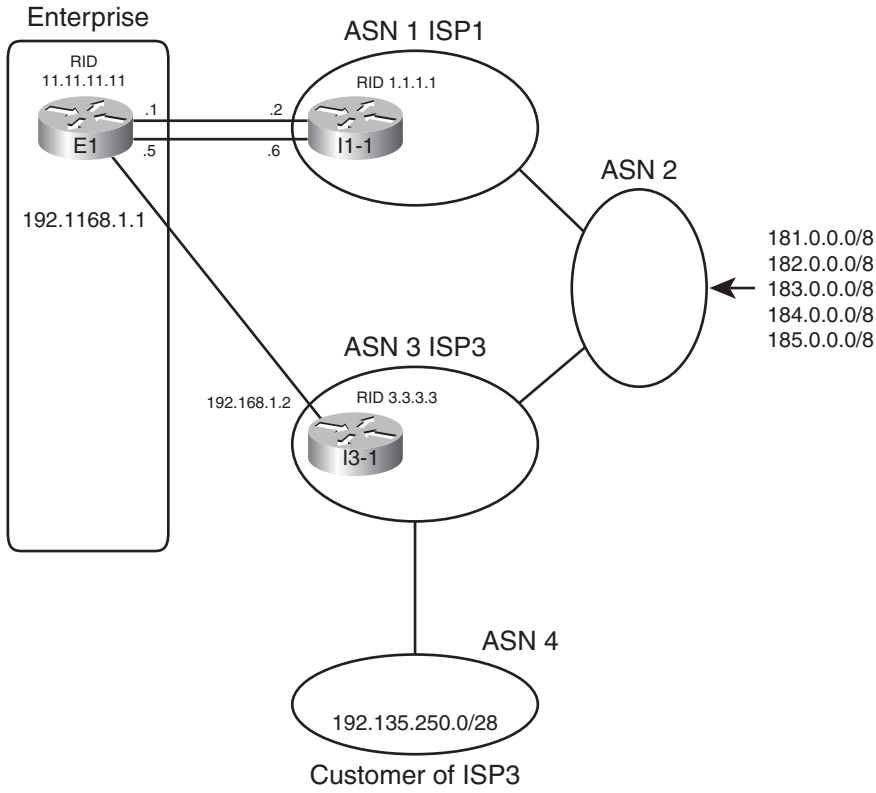


Figure 13-5 *Three Prefixes to be Advertised to E1*

The **show ip bgp** lists the entirety of the BGP routing table. Example 13-6 shows a sample from Router E1. Note that the configuration of this network is based on Example 13-2, with Routers E1 and I1-1 still using their loopback interfaces in their **neighbor** commands.

Example 13-6 *E1's BGP Table with Routes Learned from the ISPs*

```
E1# show ip bgp
BGP table version is 78, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  0.0.0.0          192.168.1.2              0             0 3 i
*>                  1.1.1.1                  0             0 1 i
*> 181.0.0.0/8      1.1.1.1                                0 1 2 111 111 i
*> 182.0.0.0/8      1.1.1.1                                0 1 2 222 i
*> 183.0.0.0/8      1.1.1.1                                0 1 2 i
*> 184.0.0.0/8      1.1.1.1                                0 1 2 i
*> 185.0.0.0/8      1.1.1.1                                0 1 2 i
*  192.135.250.0/28 1.1.1.1                                0 1 2 3 4 i
*>                  192.168.1.2                            0 3 4 i
```

First, examine the overall format and the headings in the output of the **show ip bgp** command. The Network column lists the prefix/length (NLRI). The Next Hop heading lists the next-hop IP address that would be used for the route. Then, skipping over to the far right, the Path heading lists the AS_Path PA. (Note that it is difficult to see the beginning of the AS_Path, but the weight [another PA] for each route is 0 in this case, so the next number after the 0, in this case, is the beginning of the AS_Path.)

Next, focus on the last two lines of output from the **show ip bgp** command. Each of the last two lines describes a different route to reach 192.135.250.0/28—one with next-hop 1.1.1.1 (router I1-1) and one with next-hop 192.168.1.2 (router I3-1). Because the second of these two lines does not list a prefix (under the heading "Network"), the output implies that this line is just another route for the prefix listed on the previous line. Next, examine the highlighted AS_Path values at the end of each of these lines. For the route through I1-1 (1.1.1.1), the AS_Path lists ASNs 1, 2, 3, and 4. Similarly, the AS_Path for the other route lists only ASNs 3 and 4.

Note: BGP **show** commands list the AS_Path with the first-added ASN on the right and the last-added ASN on the left. BGP uses this convention because when BGP adds an ASN to the AS_Path, BGP prepends the ASN to the list, causing the new ASN to show up as the leftmost ASN in the AS_Path.

Continuing to focus on the final two lines of the **show ip bgp** output, examine the far left part of the output, and note that the second of these two lines has a > highlighted. Per the legend at the top of the command output, the > denotes the chosen best route. In this

case, none of the routers inside the various ISPs set PAs for the purpose of influencing the best path choice, so the first used BGP best path decision is the shortest AS_Path. As a result, the path through ISP3, ASN 3, is best, having only 2 ASNs, compared to the path through ISP1, ASN 1, with four ASNs.

You can confirm that all E1's BGP table entries were learned using eBGP, rather than iBGP, by the absence of the letter "i" in the third column. Immediately after the *>, a space appears in the output. If a route was learned with iBGP, an "i" would appear in this third character position. By implication, all the routes in Example 13-6 are eBGP routes due to the absence of the letter i in the third character of possible output.

Finally, taking a broader view of the output of the **show ip bgp** command, consider which prefixes have two known routes and which have only one. Then, consider the design requirements listed before Example 13-6: I1-1 would advertise all prefixes, plus a default, but I3-1 would advertise only partial updates plus a default. As such, I3-1 did not advertise the prefixes that begin 181 through 185, by design, resulting in Router E1 only learning one route for each of these prefixes.

E1 chose the route through I3-1 as the best route for prefix 192.135.250.0/28. Example 13-7 shows the details of the IP routing table entry for this route.

Example 13-7 *E1's IP Route for 192.135.250.0/28*

```
E1# show ip route 192.135.250.0 255.255.255.240
Routing entry for 192.135.250.0/28
  Known via "bgp 11", distance 20, metric 0
  Tag 3, type external
  Last update from 192.168.1.2 00:10:27 ago
  Routing Descriptor Blocks:
  * 192.168.1.2, from 192.168.1.2, 00:10:27 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2
      Route tag 3
```

The output of the **show ip route 192.135.250.0 255.255.255.240** command lists the source of the route (BGP process 11), the next-hop router (192.168.1.2), and the AS Path length (AS Hops 2). The output also confirms that the route is an external (eBGP) route.

Most of the remaining details in the BGP table relate to BGP PAs, which are discussed in more detail in Chapter 15, "BGP Path Control."

Viewing Subsets of the BGP Table

When accepting full or partial BGP updates, the sheer number of BGP table entries can be much too large for the **show ip bgp** command to be useful. The command could list thousands, or even hundreds of thousands, of prefixes. In practice, you need to be comfortable with a variety of options on the **show ip bgp** command, each listing a different part of the BGP table.

For example, you will likely want to look at BGP table entries for specific prefixes, including the default route prefix of 0.0.0.0/0. Additionally, you may want to see routes per neighbor, and see which routes were heard from that neighbor–and which of those routes passed through any inbound route filters to make it into the BGP table. Finally, to verify whether neighboring ISPs sent full or partial updates, you can look at counters for the number of prefixes learned from each neighbor. Although you probably will never know the exact number of prefixes to expect, you should see a significant difference in the number of prefixes learned from a neighbor sending full updates as compared to a neighbor sending partial updates.

Table 13-4 summarizes some of the key command options that can supply these subsets of information.

Table 13-4 *Verification Commands for eBGP-Learned Routes*

Verification Step	Command
List possible default routes.	**show ip bgp 0.0.0.0 0.0.0.0**
List possible routes, per prefix.	**show ip bgp** *prefix [subnet-mask]*
List routes learned from one neighbor, before any inbound filtering is applied.	**show ip bgp neighbors** *ip-address* **received-routes**
List routes learned from a specific neighbor that passed any inbound filters.	**show ip bgp neighbors** *ip-address* **routes**
Lists routes advertised to a neighbor after applying outbound filtering.	**show ip bgp neighbors** *ip-address* **advertised-routes**
List the number of prefixes learned per neighbor.	**show ip bgp summary**

Example 13-8 shows a few samples of these commands on Router E1 from Figures 13-1 and 13-5. The configuration remains unchanged since Example 13-2.

Example 13-8 *Command Samples from Table 13-4*

```
E1# show ip bgp 0.0.0.0 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 75
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  3
    192.168.1.2 from 192.168.1.2 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, external
```

```
  1
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best

E1# show ip bgp 192.135.250.0
BGP routing table entry for 192.135.250.0/28, version 78
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  1 2 3 4
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, external
  3 4
    192.168.1.2 from 192.168.1.2 (3.3.3.3)
      Origin IGP, localpref 100, valid, external, best

E1# show ip bgp summary
BGP router identifier 11.11.11.11, local AS number 11
BGP table version is 78, main routing table version 78
7 network entries using 924 bytes of memory
9 path entries using 468 bytes of memory
8/5 BGP path/bestpath attribute entries using 1184 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 2) using 32 bytes of memory
BGP using 2776 total bytes of memory
BGP activity 7/0 prefixes, 53/44 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4     1     186     189       78    0    0 00:53:33        7
192.168.1.2     4     3     161     199       78    0    0 00:51:48        2
```

The first command, **show ip bgp 0.0.0.0 0.0.0.0**, displays details about the default routes in the BGP table. The output lists three lines per route, with the AS_Path on the first line. Working through the highlighted portions of the output, in this case, the AS_Path is either 3 or 1, because the ISP routers each originated the route, and those neighboring ASNs are ASN 1 and ASN 3. The output also lists the next-hop address of the route (192.168.1.2 and 1.1.1.1) and the neighbor's BGP RID (I1-1's is 1.1.1.1 and I3-1's is 3.3.3.3). Finally, instead of the > seen in the output of **show ip bgp**, this command simply lists the term "best" for the best route.

The next command, **show ip bgp 192.135.250.0**, looks much like the first. In this case, with no subnet mask listed in the command, IOS displays information for any prefix 192.135.250.0 regardless of prefix length. The output again lists three lines per route beginning with the AS_Path values (as highlighted).

The final command listed earlier in Table 13-4, **show ip bgp summary**, lists the number of prefixes received from each neighbor on the far right side. Also, you can see the amount of memory used for the prefixes (listed as network entries) and for different PAs.

The rest of the commands from Table 13-4 focus on displaying information relative to whether BGP filtering has yet occurred. The first, **show ip bgp neighbors** *neighbor-ip* **received-routes**, lists routes received from the neighbor before inbound BGP filtering. The second, **show ip bgp neighbors** *neighbor-ip* **routes**, lists routes received from that neighbor that passed through any inbound filtering. These commands are particularly useful when verifying the results of any configured BGP filters or route maps. The section "Displaying the Results of BGP Filtering" in Chapter 14 discusses the information in these commands and an extra configuration requirement to use the **received-routes** option.

Injecting Routes into BGP for Advertisement to the ISPs

So far, this chapter has focused on configuring eBGP peers and the routes learned by Enterprise routers from eBGP peers at ISPs. These outbound routes let the Enterprise routers forward packets toward the Internet.

At the same time, the ISPs need to learn routes for the Enterprise's public IP address space. This chapter assumes that the choice to use BGP has already been made, so using BGP to advertise the Enterprise's public IP address range makes good sense. This short final major section of this chapter examines the options for advertising these routes. In particular, this section looks at two options:

- BGP **network** command
- Redistribution from an IGP

Injecting Routes Using the network Command

The BGP **network** router subcommand differs significantly from the **network** command used by IGPs. For OSPF and EIGRP, the **network** command lists parameters that the router then compares to all its interface IP addresses. If matched, the router enables the IGP routing protocol on those interfaces. BGP does not use the **network** command to enable BGP on interfaces–in fact, BGP has no concept of being enabled on interfaces at all. For a point of comparison, note that the **show ip ospf interface** and **show ip eigrp interfaces** commands identify the enabled interfaces for OSPF and EIGRP, respectively, but no such equivalent BGP command even exists.

The BGP **network** command does cause a comparison to occur, but the comparison occurs between the **network** command's parameters and the *contents of that router's IP routing table*, as follows:

> Look for a route in the router's current IP routing table that exactly matches the parameters of the **network** command; if a route for that exact prefix/length exists, put the equivalent prefix/length into the local BGP table.

Note: The preceding statement, and the remaining logic in this section, assumes a BGP default setting of **no auto-summary**. The effect of reversing this setting to **auto-summary** is described in the upcoming section "The Effect of **auto-summary** on the BGP **network** Command."

For example, the Enterprise shown earlier on the left side of both Figures 13-1 and 13-5 might use a private address range and use NAT to translate to use public addresses. For example, the Enterprise might use private Class A network 10.0.0.0 for all private address needs and public address block 128.107.0.0/19 for public addresses. Enterprise Router E1 would then need to advertise the public prefix (128.107.0.0/19) to its ISPs, but not the private address range. Example 13-9 shows an example.

Example 13-9 *E1's Configuration of a* **network** *Command to Advertise Prefixes with eBGP*

```
router bgp 11
 network 128.107.0.0 mask 255.255.224.0
E1# sh ip bgp
BGP table version is 9, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  0.0.0.0          192.168.1.2              0             0 3 i
*>                   1.1.1.1                  0             0 1 i
*> 128.107.0.0/19   10.1.1.66                3         32768 i

*> 181.0.0.0/8      1.1.1.1                                  0 1 2 111 111 i
*> 182.0.0.0/8      1.1.1.1                                  0 1 2 222 i
*> 183.0.0.0/8      1.1.1.1                                  0 1 2 i
*> 184.0.0.0/8      1.1.1.1                                  0 1 2 i
*> 185.0.0.0/8      1.1.1.1                                  0 1 2 i
*  192.135.250.0/28 1.1.1.1                                  0 1 2 3 4 i
*>                  192.168.1.2                            0 3 4 i
```

The **network 128.107.0.0 mask 255.255.224.0** command lists both the subnet number and mask. It adds this prefix to the BGP table only if the exact prefix with that same mask exists in Router E1's routing table. In this case, such a route existed, so the **show ip bgp** command output that follows now lists 128.107.0.0/19 in the BGP table.

In some cases, the Internet-connected router may not have a single route for the entire public prefix. For example, with such a large range of public addresses as 128.107.0.0/19, the Enterprise will most likely have broken that range into subnets, and the Enterprise router may not have a route for the entire range. For example, Router E1 might see routes for 128.107.1.0/24, 128.107.2.0/24, and so on but no route for 128.107.0.0/19.

When a router knows routes only for subsets of the prefix that needs to be advertised, an additional step is needed when using the **network** command. For instance, the **network 128.107.0.0 mask 255.255.224.0** command will not add this prefix to the BGP table even if routes for subsets of this range exist, such as 128.107.1.0/24. So, either configure a static route for the entire range, with outgoing interface null0, on the Internet facing router, or use IGP route summarization to create a summary route for the entire prefix with IGP.

Note: The static route for 128.107.0.0/19 to null0—a discard route—is not meant to be advertised to other routers. It's only purpose is to enable the operation of the **network** command. This discard route should not cause routing problems on the local router, because of the more specific routes for subnets inside the same range of addresses.

Finally, the **network** command examples in this section use the **mask** parameter, but if omitted, IOS assumes a classful network mask. For example, a **network 9.0.0.0** command assumes a Class A default mask of 255.0.0.0, and the **network 128.1.0.0** command assumes a Class B default mask of 255.255.0.0.

The Effect of auto-summary on the BGP network Command

As of Cisco IOS version 12.3 mainline, BGP defaults to a setting of **no auto-summary**, and the previous section's discussion of the **network** command assumed this default setting. However, if the configuration is changed to **auto-summary**, then IOS makes a small change in how it interprets the **network** command.

The change in logic occurs only when the **network** command omits its **mask** parameter; there is no difference in logic if the mask parameter is explicitly configured. When the **network** command refers to a Class A, B, or C network, with no **mask** parameter configured, and with **auto-summary** configured, the router adds a route for that classful network to the BGP table:

- If the exact classful route is in the IP routing table

or

- If any subset routes of that classful network are in the routing table

In summary, of the two actions in the list, the first occurs regardless of the **auto-summary** setting, and the second occurs only if **auto-summary** is configured.

For example, with **network 9.0.0.0** configured, regardless of the **auto-summary** setting, if a route to 9.0.0.0/8 exists, the router adds 9.0.0.0/8 to the BGP table. However, if the **network 9.0.0.0** (without the mask parameter) and the **auto-summary** commands were both configured, and if only a subset route exists (for example, 9.1.1.0/24), but no route for exactly 9.0.0.0/8 exists, then the router still adds a route for the classful network (9.0.0.0/8) to the BGP table. This second example demonstrates the additional logic that occurs with the **auto-summary** command configured.

Injecting Routes Using Redistribution

Instead of using a BGP **network** command to add routes to the BGP table, the Enterprise BGP routers can instead redistribute routes from an IGP into BGP. The end goals are the same:

- Inject the public address range, but not the private IP address range, into the BGP table.
- Advertise one route for the public address range, instead of any individual subnets of the range.

The Enterprise routers that run BGP often already run the IGP as well and have learned routes for either the entire public range as one route or with subset routes. If a single route exists for the entire public range, for example the 128.107.0.0/19 range used in the last several examples, then the engineer simply needs to add a **redistribute** command to the BGP configuration to redistribute that route, and only that route, into BGP. If only subset routes exist, one of several additional steps need to be taken to meet the design goal to inject one route for the entire public address range.

Example 13-10 shows the majority of the work in a case for which Router E1 has three subset routes in the 128.107.0.0/19 range: 128.107.1.0/24, 128.107.2.0/24, and 128.107.3.0/24. However, E1 does not have a single route for the entire 128.107.0.0/19 public prefix. The example shows the redistribution configuration, all of which uses the same familiar redistribution commands shown in Chapters 9 and 10. The configuration matches prefixes in the public range and redistributes them into BGP.

Example 13-10 *Redistributing OSPF into BGP, but for Public Range Only*

```
router bgp 11
 redistribute ospf 1 route-map only-128-107
!
route-map only-128-107 permit
 match ip address prefix 128-107
!
ip prefix-list 128-107 permit 128.107.0.0/19 le 32

E1# show ip route 128.107.0.0 255.255.224.0 longer-prefixes
! Legend omitted for brevity

Gateway of last resort is 1.1.1.1 to network 0.0.0.0

     128.107.0.0/24 is subnetted, 3 subnets
O       128.107.3.0 [110/3] via 10.1.1.66, 00:05:26, FastEthernet0/0
O       128.107.2.0 [110/3] via 10.1.1.66, 00:05:26, FastEthernet0/0
O       128.107.1.0 [110/3] via 10.1.1.66, 00:05:36, FastEthernet0/0

E1# show ip bgp 128.107.0.0/19 longer-prefixes
BGP table version is 11, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.1.0/24   10.1.1.66                3          32768 ?
*> 128.107.2.0/24   10.1.1.66                3          32768 ?
*> 128.107.3.0/24   10.1.1.66                3          32768 ?
```

The two **show** commands following the configuration list the IP routes that should match the redistribution configuration, and the resulting BGP table entries. The **show ip route 128.107.0.0 255.255.224.0 longer-prefixes** command lists all three IP routes in the public address range in this case. The **show ip bgp 128.107.0.0/19 longer-prefixes** command shows the same range, listing the three BGP table entries created by the **redistribute ospf** command. These BGP table entries list the same next-hop IP addresses listed in the OSPF routes in the IP routing table, with the same metrics.

Left as is, this configuration results in Router E1 advertising all three BGP routes to the ISPs. However, to reach the goal of advertising only a single route for the entire public prefix 128.107.0.0/19, another step must be taken, typically one of the following:

- Use IGP route summarization to create the route for the entire prefix.
- Configure a null static route (a discard route) for the entire prefix on the Internet-connected router.
- Configure BGP route summarization to make BGP advertise only the entire prefix.

The first two would cause E1 to list a route for the entire public prefix–128.107.0.0/19 in this case–in its IP routing table. The redistribution configuration could then be changed so that only that exact prefix would be redistributed. (For example, removing the **le 32** parameter from the **ip prefix-list 128-107 permit 128.107.0.0/19 le 32** command would make this command match only the exact route.)

The third option would be to use BGP route summarization, telling Router E1 that when any subnet routes of 128.107.0.0/19 exists in the BGP table, advertise only 128.107.0.0/19 but none of the subset routes. Example 13-11 shows this last option.

Example 13-11 *The BGP* **aggregate-address** *Command to Advertise the Entire Public IP Address Prefix*

```
E1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
E1(config)# router bgp 11
E1(config-router)#aggregate-address 128.107.0.0 255.255.224.0 summary-only
E1(config-router)#^Z

E1# show ip bgp 128.107.0.0/19 longer-prefixes
BGP table version is 15, local router ID is 11.11.11.11
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.0.0/19   0.0.0.0                              32768 i
s> 128.107.1.0/24   10.1.1.66                3           32768 ?

s> 128.107.2.0/24   10.1.1.66                3           32768 ?
s> 128.107.3.0/24   10.1.1.66                3           32768 ?
```

Note that with the addition of the **aggregate-address** command, the BGP table now also has a route for 128.107.0.0/19, which will be advertised to E1's neighbors at the two ISPs. Also, the **summary-only** keyword in the **aggregate-address** command tells IOS to suppress the advertisement the subset routes, as noted by the code "s" beside the other three routes listed at the end of the example.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 13-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 13-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The plan shows the use of public prefix 200.1.1.0/26 by an Enterprise. What methods should I consider adding to my implementation plan for advertising that prefix to my ISPs using BGP? (2)	

Implementation Plan Peer Review Table

Table 13-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 13-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows Enterprise Router R1, with two parallel Layer 3 paths to ISP Router R2, with a need for BGP. What options exist for high availability eBGP peering? (2) Which is better?	

Table 13-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The implementation plan shows an Enterprise router with an eBGP connection to an ISP router, using a loopback interface as Update source. What other feature must be configured to make the eBGP connection work?	
Router R1 connects via eBGP to Router I1 at ISP1. R1 has routes for 130.1.1.0/24 and 130.1.2.0/24 in its routing table. The design claims the company uses 130.1.0.0/21 as its public range. What methods can be used to advertise one route for the entire range to the eBGP peer? (2)	

Create an Implementation Plan Table

This chapter does not focus on implementation or verification, but it did review one concept about static routes, as listed in Table 13-7.

Table 13-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an eBGP connection as follows: local AS 1, remote AS 2, remote router uses 1.1.1.1 for BGP peering, with 1.1.1.1 being an IP address on a common link between the routers.	
Configure an eBGP connection as follows: local AS 1, remote AS 2, local uses loopback1 (1.1.1.1), remote uses loopback2 (2.2.2.2).	
Add to the previous row the configuration to use MD5 authentication, key "barney."	
Administratively disable the neighbor configured in the previous two items in this table	
Re-enable the neighbor that was disabled in the previous row of this table.	
Cause the advertisement of IGP-learned prefix 130.1.1.0/24 to the neighbor configured in this table, without redistribution.	
Repeat the task in the previous row of this table, but this time with route redistribution, assuming OSPF process 1 is used for the IGP.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 13-8 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 13-8 *Verification Plan Memory Drill*

Information Needed	Commands
Display a single-line neighbor status for each iBGP neighbors.	
Display the number of prefixes learned from a neighbor. (List where the information is located.)	
Display the number of prefixes advertised to a neighbor. (List where the information is located.)	
Display the local and neighbor ASN.	
Display the number of eBGP hops allowed.	
List the current TCP ports used for BGP connections.	
List all prefixes in the BGP table.	
List all the best routes in the BGP table.	
Find the AS_Path for each BGP Table entry. (Describe how.)	
Determine if a particular BGP table entry is iBGP-learned. (Describe how.)	
Display one-line entries for all BGP table entries with a given prefix/length, plus any subnets inside that ranges.	
List possible default routes.	
List possible routes per prefix.	
List routes learned from one neighbor, which passed any inbound filters.	
List routes learned from one neighbor before any inbound filtering is applied.	
Display routes suppressed and added to the BGP table due to BGP route summarization (aggregation).	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 13-9 lists a reference of these key topics and the page numbers on which each is found.

Key Topic

Table 13-9 *Key Topics for Chapter 13*

Key Topic Element	Description	Page Number
List	Minimal eBGP configuration checklist	423
List	Required plus commonly used optional eBGP configuration command checklist	425
List	Rules for how a router chooses its BGP Router ID	425
Figure 13-2	Diagram of how the **router bgp** and **neighbor remote-as** commands on neighboring routers refer to ASNs	426
List	eBGP Configuration checklist including use of loopbacks as Update source and eBGP Multihop	427
Figure 13-3	Diagram of how the configuration matches on two BGP neighbors when using loopbacks	428
Table 13-2	BGP neighbor states	430
Table 13-3	BGP Messages	436
Table 13-4	Commonly used **show** commands to display eBGP-learned routes	441

Complete the Tables and Lists from Memory

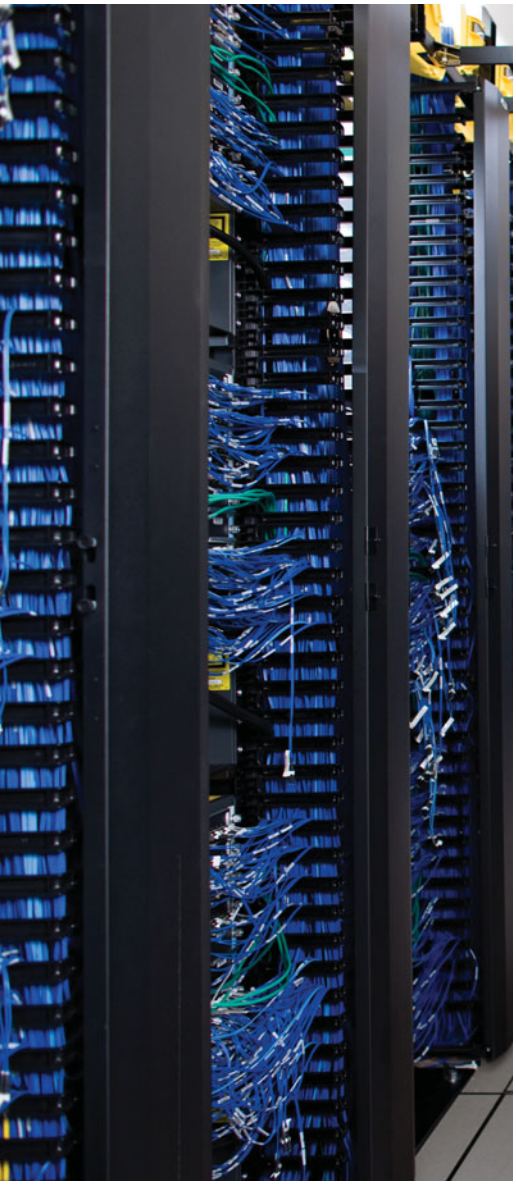
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

eBGP multihop, Update Source (BGP), Established (BGP State), Open, Update, Active (BGP state), BGP Table

This page intentionally left blank



This chapter covers the following subjects:

Internal BGP Between Internet-Connected Routers: This section examines the need for iBGP peering inside an Enterprise, and the required commonly used optional configuration settings.

Avoiding Routing Loops When Forwarding Toward the Internet: This section discusses the issues that occur when the Internet-connected routers may forward packets to each other through routers that do not use BGP, and how such a design requires some means to supply BGP-learned routes to the internal Enterprise routers.

Route Filtering and Clearing BGP Peers: This section gives a brief description of the options for filtering the contents of BGP Updates, along with explaining some operational issue related to the BGP **clear** command.

CHAPTER 14

Internal BGP and BGP Route Filtering

Outbound routing is simple with a single Internet-connected router. The Enterprise IGP could flood a default route throughout the Enterprise funneling all Internet traffic toward the one Internet-connected router. That router could then choose the best route to any and all Internet destinations it learned with eBGP.

With two (or more) Internet-connected routers in a single Enterprise, additional issues arise, in particular, issues related to outbound routing. These issues require the use of BGP between Enterprise routers; in some cases, the design may require BGP even on Enterprise routers that do not peer with routers at the various ISPs. This chapter examines the scenarios in which using iBGP makes sense, and shows the related configuration and verification commands.

Chapter 14 begins focusing on the issues that occur when an Enterprise uses a pair of Internet-connected routers. In particular, the examples use the sample network shown in Figure 14-1. This design uses the same ISPs and ISP routers as in Chapter 13, "External BGP," with familiar IP address ranges but with a few different links. The design now also shows two of the core routers (actually Layer 3 switches) inside the Enterprise–routers that do not directly connect to any ISP. Figure 14-1 shows the design that will be referenced throughout the chapter.

Note: Figure 14-1 shows the IP addresses as just the last octet of the address; in these cases, the first three octets are 10.1.1.

The first section of this chapter focuses on concepts, configuration, and verification of the iBGP connection between E1 and E2 in the figure. The second major section of this chapter, "Avoiding Routing Loops when Forwarding Toward the Internet," examines the need for iBGP on routers internal to the Enterprise, such as Routers Core1 and Core2 in the figure. The final section of this chapter examines the process of filtering both iBGP and eBGP routing updates.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 14-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in

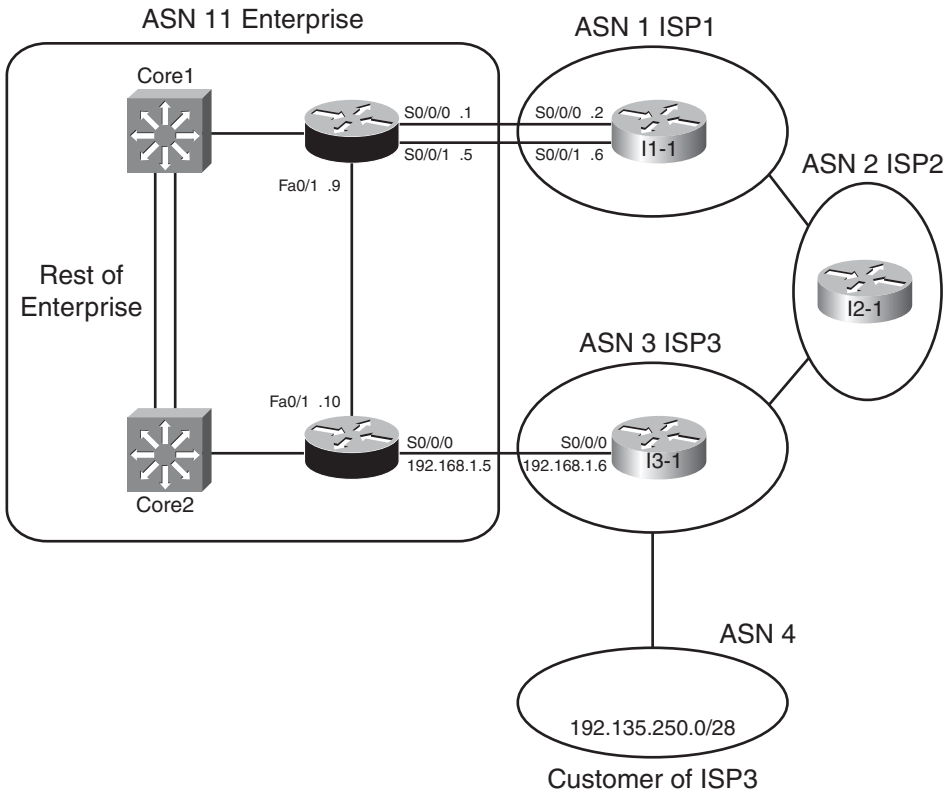


Figure 14-1 *Dual Internet Router Design Used Throughout Chapter 14*

those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 14-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Internal BGP Between Internet-Connected Routers	1–4
BGP Synchronization and iBGP Meshes	5
Route Filtering and Clearing BGP Peers	6–8

1. R1 in ASN 1 with loopback1 address 1.1.1.1 needs to be configured with an iBGP connection to R2 with loopback2 IP address 2.2.2.2. The connection should use the loopbacks. Which of the following commands is required on R1?

   a. **neighbor 1.1.1.1 remote-as 1**

   b. **neighbor 2.2.2.2 remote-as 2**

c. **neighbor 2.2.2.2 update-source loopback1**

d. **neighbor 2.2.2.2 ibgp-multihop 2**

e. **neighbor 2.2.2.2 ibgp-mode**

2. The following output occurred as a result of the **show ip bgp** command on Router R1. The output shows all BGP table entries on R1. How many iBGP-learned routes exist on this router?

```
*>i181.0.0.0/8       10.100.1.1               0    100      0 1 2 111 112 i
*>i182.0.0.0/8       10.100.1.1               0    100      0 1 2 222 i
*>i183.0.0.0/8       10.100.1.1               0    100      0 1 2 i
*>i184.0.0.0/8       10.100.1.1               0    100      0 1 2 i
*> 192.135.250.0/28 192.168.1.6                            0 3 4 i
```

a. 1

b. 2

c. 3

d. 4

e. 5

3. The following output on Router R1 lists details of a BGP route for 190.1.0.0/16. Which of the following is true based on this output? (Choose 2)

```
R1# show ip bgp 190.1.0.0/16
BGP routing table entry for 190.1.0.0/16, version 121
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  1 2 3 4
    1.1.1.1 from 2.2.2.2 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

a. R1 has a **neighbor 1.1.1.1** command configured.

b. R1 has a **neighbor 2.2.2.2** command configured.

c. The **show ip bgp** command lists a line for 190.1.0.0/16 with both an > and an i on the left.

d. R1 is in ASN 1.

4. A company uses Routers R1 and R2 to connect to ISP1 and ISP2, respectively, with Routers I1 and I2 used at the ISPs. R1 peers with I1 and R2; R2 peers with I2 and R1. Assuming as many default settings as possible are used on all four routers, which of the following is true about the next-hop IP address for routes R1 learns over its iBGP connection to R2?

a. The next hop is I2's BGP RID.

b. The next hop is I2's IP address used on the R2-I2 neighbor relationship.

c. The next hop is R2's BGP RID.

d. The next hop is R2's IP address used on the R1–R2 neighbor relationship.

5. A company uses Routers R1 and R2 to connect to ISP1 and ISP2, respectively, with Routers I1 and I2 used at the ISPs. R1 peers with I1 and R2; R2 peers with I2 and R1. R1 and R2 do not share a common subnet, relying on other routers internal to the Enterprise for IP connectivity between the two routers. Which of the following could be used to prevent potential routing loops in this design? (Choose 2)

   a. Using an iBGP mesh inside the Enterprise core
   b. Configuring default routes in the Enterprise pointing to both R1 and R2
   c. Redistributing BGP routes into the Enterprise IGP
   d. Tunneling the packets for the iBGP connection between R1 and R2

6. R1 is currently advertising prefixes 1.0.0.0/8, 2.0.0.0/8, and 3.0.0.0/8 over its eBGP connection to neighbor 2.2.2.2 (R2). An engineer configures a prefix list (fred) on R1 that permits only 2.0.0.0/8 and then enables the filter with the **neighbor R2 prefix-list fred out** command. Upon exiting configuration mode, the engineer uses some **show** commands on R1, but no other commands. Which of the following is true in this case?

   a. The **show ip bgp neighbor 2.2.2.2 received-routes** command lists the three original prefixes.
   b. The **show ip bgp neighbor 2.2.2.2 advertised-routes** command lists the three original prefixes.
   c. The **show ip bgp neighbor 2.2.2.2 routes** command lists the three original prefixes.
   d. The **show ip bgp neighbor 2.2.2.2 routes** command lists only 2.0.0.0/8.
   e. The **show ip bgp neighbor 2.2.2.2 advertised-routes** command lists only 2.0.0.0/8.

7. Which of the following three BGP filtering methods enabled with the **neighbor** command will filter BGP prefixes based on the prefix and prefix length? (Choose 3)

   a. A **neighbor distribute-list out** command, referencing a standard ACL
   b. A **neighbor prefix-list out** command
   c. A **neighbor filter-list out** command
   d. A **neighbor distribute-list out** command, referencing an extended ACL
   e. A **neighbor route-map out** command

8. Which of the following commands causes a router to bring down BGP neighbor relationships?

   a. **clear ip bgp** *
   b. **clear ip bgp 1.1.1.1**
   c. **clear ip bgp** * **soft**
   d. **clear ip bgp 1.1.1.1 out**

Foundation Topics

Internal BGP Between Internet-Connected Routers

When an Enterprise uses more than one router to connect to the Internet, and those routers use BGP to exchange routing information with their ISPs, those same routers need to exchange BGP routes with each other as well. The BGP neighbor relationships occur inside that Enterprise–inside a single AS–making these routers iBGP peers.

This first major section of Chapter 14 begins with a look at why two Internet-connected routers need to have an iBGP neighbor relationship. Then, the text looks at various iBGP configuration and verification commands. Finally, discussion turns to a common issue that occurs with next-hop reachability between iBGP peers, with an examination of the options to overcome the problem.

Establishing the Need for iBGP with Two Internet-Connected Routers

Two Internet-connected routers in an Enterprise need to communicate BGP routes to each other because these routers may want to forward IP packets to the other Internet-connected router, which in turn would forward the packet into the Internet. With an iBGP peer connection, each Internet-connected router can learn routes from the other router and decide if that other router has a better route to reach some destinations in the Internet. Without that iBGP connection, the routers have no way to know if the other router has a better BGP path.

For example, consider Figure 14-2, which shows two such cases. The figure shows a topology that uses the following design options, as agreed upon with ISP1 and ISP3:

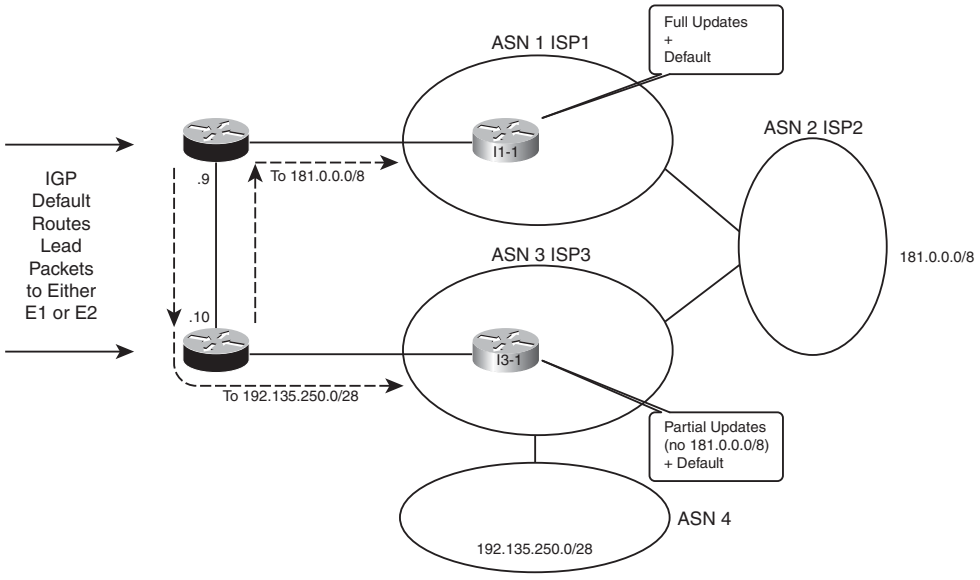


Figure 14-2 *Choosing the Best Routes from ASN 11 to 181.0.0.0/8 and 192.135.250.0/28*

- ISP1 sends full routing updates and a default route.
- ISP3 sends partial updates and a default route.

First, consider the eBGP routing updates, particularly for the two prefixes highlighted in the figure. Both ISP1 and ISP3 know routes for 181.0.0.0/8, but ISP3's agreement with the Enterprise is that ISP3 sends partial updates. This usually means ISP3 sends updates for prefixes in its own ASN plus prefixes for customers attached to its ASN, such as 192.135.250.0/28 in this case. ISP1, however, sends full updates, so E1 learns an eBGP route for both 181.0.0.0/8 and 192.135.250.0/28, but Router E2 only learns an eBGP route for 192.135.250.0/28.

Next, take a closer look at the routes for 181.0.0.0/8, both on E1 and E2. Only E1 learns an eBGP route for 181.0.0.0/8; E2 does not, because of ISP3's partial updates. If E1 and E2 did not use iBGP between each other, E2 would never know that E1 had a good route for 181.0.0.0/8. So, without an iBGP connection, packets destined to hosts in 182.0.0.0/8 if they arrived at E2 would be sent to ISP3 because of E2's default route learned from ISP3. However, if E1 and E2 form an iBGP neighbor relationship, E2 would know a route for 181.0.0.0/8 through E1 and would choose this route as its best route and would forward such packets to E1. Then E1 would forward the packets to ISP1, as shown in the figure.

Finally, take a closer look at the routes for 192.135.250.0/28 on both E1 and E2. If none of the ISPs changed the default PA settings for these routes, both E1 and E2 would choose the route through E2 as the better route, due to the shorter AS_Path length (two ASNs away through ISP3 versus four ASNs away through ISP1). Without iBGP between E1 and E2, E1 would not learn of this better route through E2, so any packets destined to 192.135.250.0/28 that reach E1 would be forwarded to ISP1. With iBGP, E1 would know of E2's better route and forward the packets toward E2, as shown in the figure.

For both prefixes, iBGP allowed both routers in the same ASN to reach the same conclusion about the better router through which to send packets for each Internet destination.

Configuring iBGP

The most basic iBGP configuration differs only slightly compared to eBGP configuration. The configuration does not explicitly identify an eBGP versus an iBGP peer. Instead, for iBGP, the neighbor's ASN listed on the **neighbor... remote-as** *neighbor-asn* command lists the same ASN as the local router's **router bgp** command. eBGP **neighbor remote-as** commands list a different ASN.

When two iBGP peers share a common physical link, such as E1 and E2 in Figure 14-1, the iBGP configuration simply requires a single **neighbor remote-as** command on each router. Example 14-1 shows the BGP configuration on both Router E1 and E2 with this single **neighbor** command highlighted. The rest of the configuration lists the commands used to configure other BGP settings (as described after the example). Note that Figure 14-1 in the introduction to this chapter shows more detail about the eBGP peers as well.

Example 14-1 *BGP Configuration on E1: Neighborships Configured*

```
! Configuration on router E1
router bgp 11
 no synchronization
 bgp log-neighbor-changes
 aggregate-address 128.107.0.0 255.255.224.0 summary-only
 redistribute ospf 1 route-map only-128-107
 neighbor 1.1.1.1 remote-as 1
 neighbor 1.1.1.1 password fred
 neighbor 1.1.1.1 ebgp-multihop 2
 neighbor 1.1.1.1 update-source Loopback1
 neighbor 10.1.1.10 remote-as 11
 no auto-summary
!
! Next, static routes so that the eBGP neighbor packets can reach
! I1-1's loopback interface address 1.1.1.1
ip route 1.1.1.1 255.255.255.255 Serial0/0/0
ip route 1.1.1.1 255.255.255.255 Serial0/0/1
!
ip prefix-list 128-107 seq 5 permit 128.107.0.0/19 le 32
!
route-map only-128-107 permit 10
 match ip address prefix-list 128-107 Neighbor 192.168.1.2 remote-as 3
```

```
! Now, on router E2
router bgp 11
 no synchronization
 bgp log-neighbor-changes
 network 128.107.32.0
 aggregate-address 128.107.0.0 255.255.224.0 summary-only
 redistribute ospf 1 route-map only-128-107
 neighbor 10.1.1.9 remote-as 11
 neighbor 192.168.1.6 remote-as 3
 neighbor 192.168.1.6 password barney
 no auto-summary
!
ip prefix-list 128-107 seq 5 permit 128.107.0.0/19 le 32
!
route-map only-128-107 permit 10
 match ip address prefix-list 128-107 Neighbor 192.168.1.2 remote-as 3
```

Only the four highlighted configuration commands are required for the E1-E2 iBGP peering. Both refer to the other router's IP address on the FastEthernet link between the two routers, and both refer to ASN 11. The two routers then realize the neighbor is an iBGP neighbor because the neighbor's ASN (11) matches the local router's ASN, as seen on the **router bgp 11** command.

The example also lists the rest of the BGP configuration. Focusing on Router E1, the configuration basically matches the configuration of Router E1 from the end of Chapter 13, except that E1 has only one eBGP peer (I1-1) in this case instead of two eBGP peers. The configuration includes the eBGP peer connection to I1-1, using loopback interfaces (1.1.1.1 on I1-1 and 11.11.11.11 on E1). The eBGP peers need to use eBGP multihop because of the use of the loopbacks, and they use MD5 authentication as well. Finally, the configuration shows the redistribution of the Enterprise's public address range of 128.107.0.0/19 by redistributing from OSPF and summarizing with the **aggregate-address** BGP subcommand.

E2's configuration lists the same basic parameters, but with a few differences. E2 does not use a loopback for its peer connection to I3-1, because only a single link exists between the two routers. As a result, E2 also does not need to use eBGP multihop.

Refocusing on the iBGP configuration, Example 14-1 uses the interface IP addresses of the links between Routers E1 and E2. However, often the Internet-connected routers in an Enterprise do not share a common subnet. For example, the two routers may be in separate buildings in a campus for the sake of redundancy. The two routers may actually be in different cities, or even different continents. In such cases, it makes sense to configure the iBGP peers using a loopback IP address for the TCP connection so that a single link failure does not cause the iBGP peer connection to fail. For example, in Figure 14-1, if the FastEthernet link between E1 and E2 fail, the iBGP connection defined in Example 14-1, which uses the interface IP addresses of that link, would fail even though a redundant IP path exists between E1 and E2.

The configuration to use loopback interfaces as the update source mirrors that same configuration for eBGP peers, except that iBGP peers do not need to configure the **neighbor... ebgp-multihop** command. One difference between iBGP and eBGP is that IOS uses the low TTL of 1 for eBGP connections by default but does not for iBGP connections. So, for iBGP connections, only the following steps are required to make two iBGP peers use a loopback interface:

Key Topic

Step 1. Configure an IP address on a loopback interface on each router.

Step 2. Configure each router to use the loopback IP address as the source IP address, for the neighborship with the other router, using the **neighbor... update-source** *ip-address* command.

Step 3. Configure the BGP **neighbor** command on each router to refer to the other router's loopback IP address as the neighbor IP address in the **neighbor** *neighbor-ip* **remote-as** command.

Step 4. Make sure each router has IP routes so that they can forward packets to the loopback interface IP address of the other router.

Example 14-2 shows an updated iBGP configuration for Routers E1 and E2 to migrate to use a loopback interface. In this case, E1 uses loopback IP address 10.100.1.1/32 and E2 uses 10.100.1.2/32. OSPF on each router has already been configured with a **network 10.0.0.0 0.255.255.255 area 0** command (not shown), which then causes OSPF to advertise routes to reach the respective loopback interface IP addresses.

Example 14-2 *iBGP Configuration to Use Loopbacks as the Update Source*

```
! Configuration on router E1
interface loopback 0
 ip address 10.100.1.1 255.255.255.255
router bgp 11
 neighbor 10.100.1.2 remote-as 11
 neighbor 10.100.1.2 update-source loopback0
```

```
! Configuration on router E2
interface loopback 1
 ip address 10.100.1.2 255.255.255.255
router bgp 11
 neighbor 10.100.1.1 remote-as 11
 neighbor 10.100.1.1 update-source loopback1
```

The highlighted portions of the output link the key values together for the E1's definition of its loopback as the update source and E2's reference of that same IP address on its **neighbor** command. The **neighbor 10.100.1.2 update-source loopback0** command on E1 tells E1 to look to interface loopback0 for its update source IP address. Loopback0's IP address on E1 has IP address 10.100.1.1. Then, E2's **neighbor** commands for Router E1 all refer to that same 10.100.1.1 IP address, meeting the requirement that the update source on one router matches the IP address listed on the other router's **neighbor** command.

Verifying iBGP

iBGP neighbors use the same messages and neighbor states as eBGP peers. As a result, the same commands in Chapter 13 for BGP neighbor verification can be used for iBGP peers. Example 14-3 shows a couple of examples, using Router E1's iBGP neighbor relationship with E2 (10.100.1.2) based on the configuration in Example 14-2.

Example 14-3 *Verifying iBGP Neighbors*

```
E1# show ip bgp summary
BGP router identifier 11.11.11.11, local AS number 11
BGP table version is 190, main routing table version 190
11 network entries using 1452 bytes of memory
14 path entries using 728 bytes of memory
11/7 BGP path/bestpath attribute entries using 1628 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 3 (at peak 4) using 96 bytes of memory
BGP using 4072 total bytes of memory
BGP activity 31/20 prefixes, 100/86 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4     1     339     344      190    0    0 00:28:41        7
10.100.1.2      4    11      92     132      190    0    0 01:02:04        3
```

```
E1# show ip bgp neighbors 10.100.1.2
BGP neighbor is 10.100.1.2,  remote AS 11, internal link
  BGP version 4, remote router ID 10.100.1.2
  BGP state = Established, up for 01:02:10
  Last read 00:00:37, last write 00:00:59, hold time is 180, keepalive interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
! lines omitted for brevity
```

Key Topic

The **show ip bgp summary** command lists E1's two neighbors. As with eBGP peers, if the last column (the State/PfxRcd column) lists a number, then the neighbor has reached established state, and BGP Update messages can be sent. The output can distinguish between an iBGP or eBGP neighbor but only by comparing the local router's ASN (in the first line of output) to the ASN listed in each line at the bottom of the output.

The **show ip bgp neighbors 10.100.1.2** command lists many details specifically for the neighbor. In particular, it states that the neighbor is an iBGP neighbor with the phrase "internal link," as highlighted in the output.

Examining iBGP BGP Table Entries

To better understand the BGP table with two (or more) Internet-connected routers inside the same company, start with one prefix, and compare the BGP table entries on the two routers for that one prefix. By examining several such examples, you can appreciate more about the benefits and effects of these iBGP neighborships.

This section examines the BGP tables on Routers E1 and E2, focusing on the prefixes highlighted in Figure 14-2–namely, prefixes 181.0.0.0/8 and 192.135.250.0/28. To make reading the output of the **show** commands a little more obvious, Figure 14-3 collects some key pieces of information into a single figure. This figure shows the two BGP neighbor relationships on each router, showing the update source and neighbor IP address of each BGP neighbor relationship. It also lists the BGP RID of the routers.

Examples 14-4 and 14-5 compare the output on Routers E2 and E1 for prefix 181.0.0.0/8. Example 14-4 lists output on Router E2, listing the BGP table entries for prefix 181.0.0.0/8. Remember, the design calls for ISP3 to only send partial updates, so E2 has not received an eBGP route for 181.0.0.0/8 from I3-1. However, E1 has indeed learned of that prefix from I1-1 (ISP1), and E1 has already advertised prefix 181.0.0.0/8 to E2.

Note: Several BGP routes seen in the examples in this chapter originate in ASNs not shown in the figure. The figure shows enough of the topology to understand the first few ASNs in the AS_Path for these routes.

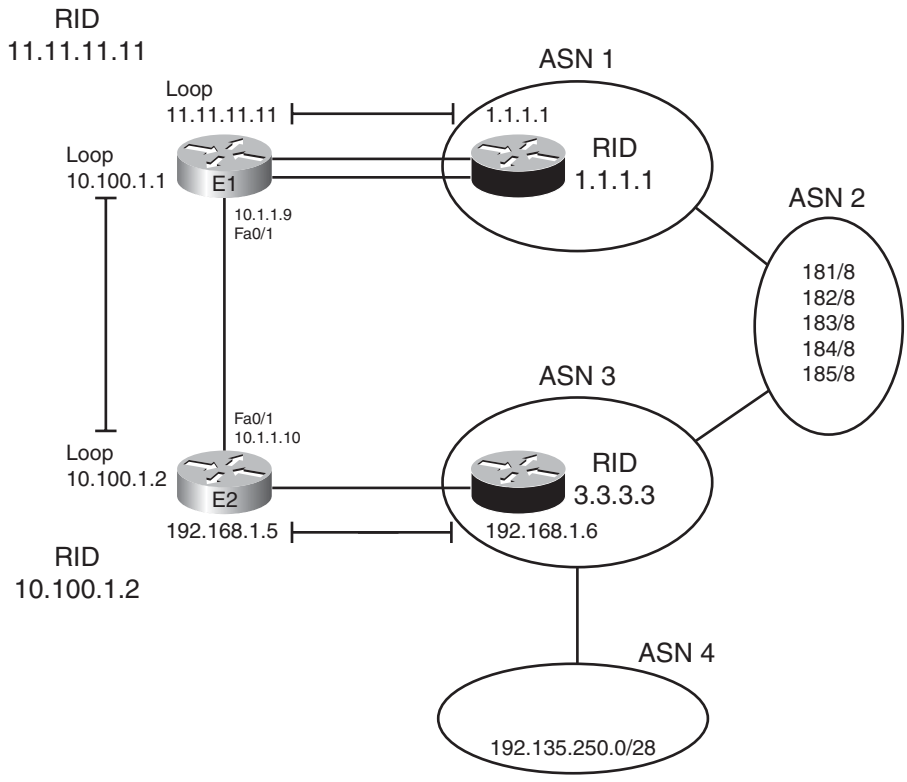


Figure 14-3 *Reference Information for BGP Table Verification*

Example 14-4 *Notations of iBGP Learned Routes in the* **show ip bgp** *Command*

```
E2# show ip bgp 181.0.0.0/8 longer-prefixes
BGP table version is 125, local router ID is 10.100.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i181.0.0.0/8      1.1.1.1                  0    100      0 1 2 111 112 i

E2# show ip bgp 181.0.0.0/8
BGP routing table entry for 181.0.0.0/8, version 121
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  1 2 111 111
    1.1.1.1 from 10.100.1.1 (11.11.11.11)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

The first command, **show ip bgp 181.0.0.0/8 longer-prefixes**, lists output with the same general format as the **show ip bgp** command, but it limits the output to the prefixes in the listed range. Only one such route exists in this case. The legend information at the top of the output, plus the headings and meanings of the different fields, is the same as with the **show ip bgp** command.

Next, the first command's output denotes this route as an iBGP-learned route with code "i" in the third character. The second command in the example, **show ip bgp 181.0.0.0/8**, which displays a more detailed view of the BGP table entry and denotes this route as iBGP-learned with the word "internal." Similarly, the briefer **show ip bgp 181.0.0.0/8** command output lists this one route as E2's best route by displaying a > in the second column, whereas the more verbose output in the second command simply lists this route as "best."

Next, consider these same commands on Router E1, as shown in Example 14-5. Comparing the highlighted fields as matched in each of the examples:

- Both list the same AS_Path (1, 2, 111, 112) because iBGP peers do not add ASNs to the AS_Path when advertising to each other. So, both E1 and E2 have the same perspective on the AS_Path and AS_Path length.
- Both list the one route for 181.0.0.0/8 as the best path in part because each has learned only one such path.
- Both list a next_hop (a BGP PA) as 1.1.1.1, which is I1-1's loopback interface used in the E1 to I1-1 BGP neighbor relationship (also called the BGP neighbor ID).
- E2 lists the route as an internal (iBGP-learned) route, whereas E1 lists it as an external route.

Example 14-5 *Router E1's* **show** *Commands for BGP Routes for 181.0.0.0/8*

```
E1# show ip bgp 181.0.0.0/8 longer-prefixes
BGP table version is 190, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 181.0.0.0/8      1.1.1.1                                0 1 2 111 112 i

E1# show ip bgp 181.0.0.0/8
BGP routing table entry for 181.0.0.0/8, version 181
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
        2
  1 2 111 111, (received & used)
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, localpref 100, valid, external, best
```

The output from these examples confirms that E1 learned the eBGP route for 181.0.0.0/8, advertised it to E2, and E2 chose to use that iBGP learned route as its best route to reach 181.0.0.0/8.

Next, consider the route for 192.135.250.0/28, a route learned in the full BGP updates from ISP1's Router I1-1 and in the partial BGP updates from ISP3's Router I3-1. After exchanging this route using their iBGP peering, both E1 and E2 should see two possible routes: an eBGP route learned from their one connected ISP and the iBGP route learned from each other. Again assuming that the ISPs have not made any attempt to set PA values to influence the best path choice, and knowing the neither E1 nor E2 have configured BGP to influence the best path choice, the route through E2 should be best because of the shorter AS_Path.

Example 14-6 shows the output of the **show ip bgp** command on both E1 and E2, again for comparison. Note that the command used in the examples, **show ip bgp 192.135.250.0/28 longer-prefixes**, is used because it lists only the routes for that prefix, rather than the full BGP table displayed by **show ip bgp**, but the format of the output is almost identical.

Example 14-6 *Comparing BGP Routes for 192.135.250.0/28 on E1 and E2*

```
! First, on E1:
E1# show ip bgp 192.135.250.0/28 longer-prefixes
BGP table version is 26, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  192.135.250.0/28 1.1.1.1                                 0 1 2 3 4 i
*>i                 192.168.1.6              0     100      0 3 4 i
```

```
! Next, on E2:
E2# show ip bgp 192.135.250.0/28 longer-prefixes
BGP table version is 25, local router ID is 10.100.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.135.250.0/28 192.168.1.6                          0 3 4 i
```

First, E1 lists two routes for this prefix, one external and one internal. The output identifies external routes by the absence of an "i" in the third character, whereas the output lists an "i" in the third character for internal routes. In this case, E1's internal route, with next_hop 192.168.1.6, is E1's best route, as was shown back in Figure 14-2. E1 chose this iBGP route because of the shorter AS_Path length; the AS_Path is highlighted at the end of each line.

E2's output in the second half of Example 14-6 lists only a single route–its eBGP route for 192.135.250.0/28. That only one route appears, rather than two, is a good example of the effect of two rules about how BGP operates:

- Only advertise the best route in any BGP Update.
- Do not advertise iBGP-learned routes to iBGP peers.

E2's output lists a single route for 192.135.250.0/28—its external route learned from ISP3—because E1 chooses not to advertise a route for 192.135.250.0/28 over the iBGP connection. If you look back at E1's output, E1's best route for this prefix is its internal route; so, if E1 were to advertise any route for this prefix to E2, E1 would advertise this internal route, because it is E1's best BGP route for that prefix. However, the second rule–do not advertise iBGP-learned routes to iBGP peers–prevents E1 from advertising this route back to E2. (Logically speaking, it makes no sense for E1 to tell E2 about a route when E2 is the router that originally advertised the route to E1 in the first place—a concept much like Split Horizon, although technically the term does not apply to BGP.) As a result, E2 lists a single route for 192.135.250.0/28.

Note that if the route for 192.135.250.0/28 through ISP3 failed, then E1 would start using the route through ISP1 as its best route. E1 would then advertise that best route to E2 that could then forward traffic through E1 for destinations in 192.135.250.0/28.

Understanding Next-Hop Reachability Issues with iBGP

With IGPs, the IP routes added to the IP routing table list a next-hop IP address. With few exceptions, the next-hop IP address routes exist in a connected subnet. For example, the E1-E2 iBGP connection uses loopback interfaces 10.100.1.1 (E1) and 10.100.1.2 (E2). E1's OSPF-learned route to reach 10.100.1.2 lists outgoing interface Fa0/1, next-hop 10.1.1.10–an address in the LAN subnet that connects E1 and E2. (See Figure 14-3 a few pages back for reference.)

Examples 14-5 and 14-6 also happened to show two examples of iBGP-learned routes and their next-hop addresses. The next-hop addresses were not in connected subnets; the next-hop addresses were not even IP addresses on a neighboring router. The two examples were as follows; again, it may be helpful to refer to the notations in Figure 14-3:

- Example 14-4: E2's route for 181.0.0.0/8 lists next-hop address 1.1.1.1, a loopback interface IP address on I1-1.
- Example 14-6: E1's route for 192.135.250.0/28 lists next-hop address 192.168.1.6, which is I3-1's interface IP address on the link between E2 and I3-1.

In fact, in the case of Example 14-4, the output of the **show ip bgp 181.0.0.0/8** command on E2 listed the phrase "1.1.1.1 from 10.100.1.1 (11.11.11.11)". This phrase lists the next hop (1.1.1.1) of the route, the neighbor from which the route was learned (10.100.1.1 or E1) and the neighbor's BGP RID (11.11.11.11, as listed in Figure 14-3).

BGP advertises these particular IP addresses as the next-hop IP addresses because of a default behavior for BGP. By default, when a router advertises a route using eBGP, the advertising router lists its own update-source IP address as the next-hop address of the route. In other words, the next-hop IP address is the IP address of the eBGP neighbor, as listed on

the **neighbor remote-as** command. However, when advertising a route to an iBGP peer, the advertising router (by default) does not change the next-hop address. For example, when I1-1 advertises 181.0.0.0/8 to E1, because it is an eBGP connection, I1-1 sets its own IP address (1.1.1.1)–specifically the IP address I1-1 uses on its eBGP peer connection to E1—as the next-hop. When E1 advertises that same route to iBGP peer E2, E1 does not change the next-hop address of 1.1.1.1, so router E2's iBGP-learned route lists 1.1.1.1 as the next-hop address.

The IP routing process can use routes whose next-hop addresses are not in connected subnets as long as each router has an IP route that matches the next-hop IP address. So, engineers must understand these rules about how BGP sets the next-hop address and ensure each router can reach the next-hop address listed in the BGP routes. Two main options exist to ensure reachability to these next-hop addresses:

- Create IP routes so that each router can reach these next-hop addresses that exist in other ASNs.
- Change the default iBGP behavior with the **neighbor next-hop-self** command.

The text now examines each of these two options in more detail.

Ensuring Routes Exist to the Next-Hop Address

Routers can still forward packets using routes whose next-hop addresses are not in connected subnets. To do so, when forwarding packets, the router performs a recursive route table lookup. For example, for packets arriving at E2 with a destination of 181.0.0.1, the following would occur:

Step 1. E2 would match the routing table for destination address 181.0.0.1, matching the route for 181.0.0.0/8, with next hop 1.1.1.1.

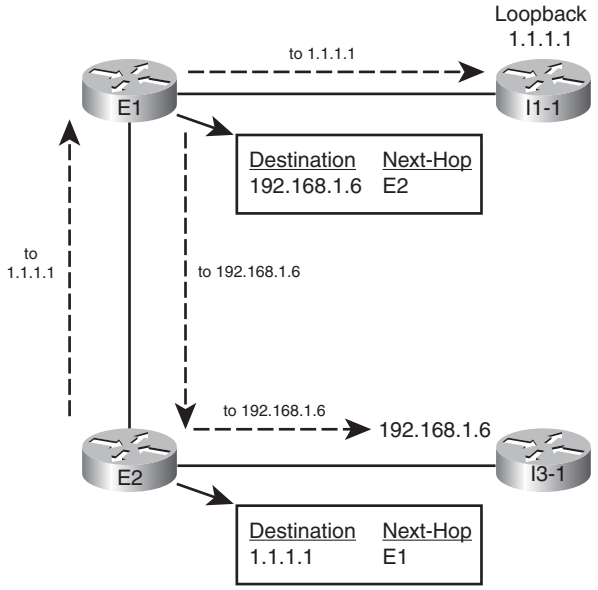
Step 2. E2 would next look for its route matching destination 1.1.1.1–the next-hop of the first route–and forward the packet based on that route.

So, no matter the next-hop IP address listed in the routing table, as long as a working route exists to reach that next-hop IP address, the packet can be forwarded. Figure 14-4 shows the necessary routes in diagram form using two examples. E1 has a route to 192.135.250.0/28 with next hop 192.168.1.6; two arrowed lines show the required routes on Routers E1 and E2 for forwarding packets to this next-hop address. Similarly, the dashed lines also show the necessary routes on E2 and E1 for next-hop address 1.1.1.1, the next-hop IP address for their routes to reach 181.0.0.0/8.

Two easily implemented solutions exist to add routes for these nonconnected next-hop IP addresses: Either add static routes or use an IGP between the Enterprise and the ISPs for the sole purpose of advertising these next-hop addresses.

Using neighbor... next-hop-self to Change the Next-Hop Address

The second option for dealing with these nonconnected next-hop IP addresses changes the iBGP configuration so that a router changes the next-hop IP address on iBGP-advertised routes. This option simply requires the **neighbor** *neighbor-ip* **next-hop-self** com-



Key Topic **Figure 14-4** *Ensuring Routes Exist for Next-Hop Addresses in Other ASNs*

mand to be configured for the iBGP neighbor relationship. A router with this command configured advertises iBGP routes with its own update-source IP address as the next-hop IP address. And because the iBGP neighborship already relies on a working route for these update source IP addresses, if the neighborship is up, then IP routes already exist for these next-hop addresses.

For example, on the iBGP connection from E1 to E2, E1 would add the **neighbor 10.100.1.2 next-hop-self** command, and E2 would add the **neighbor 10.100.1.1 next-hop-self** command. When configured, E1 advertises iBGP routes with its update source IP address (10.100.1.1) as the next-hop address. E2 likewise advertises routes with a next-hop address of 10.100.1.2. Example 14-7 shows E2's BGP table, with a few such examples highlighted, after the addition of these two configuration commands on the respective routers.

Example 14-7 *Seeing the Effects of* **next-hop-self** *from Router E2*

```
E2#show ip bgp
BGP table version is 76, local router ID is 10.100.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          192.168.1.6              0             0 3 i
* i                  10.100.1.1               0    100      0 1 i
* i128.107.0.0/19   10.100.1.1               0    100      0 i
*>                   0.0.0.0                             32768 i
```

```
s> 128.107.1.0/24    10.1.1.77                2          32768 ?
s> 128.107.2.0/24    10.1.1.77                2          32768 ?
s> 128.107.3.0/24    10.1.1.77                2          32768 ?
*>i181.0.0.0/8       10.100.1.1               0    100       0 1 2 111 112 i
*>i182.0.0.0/8       10.100.1.1               0    100       0 1 2 222 i
*>i183.0.0.0/8       10.100.1.1               0    100       0 1 2 i
*>i184.0.0.0/8       10.100.1.1               0    100       0 1 2 i
*>i185.0.0.0/8       10.100.1.1               0    100       0 1 2 i
*> 192.135.250.0/28 192.168.1.6                            0 3 4 i
```

This completes the discussion of iBGP configuration and operation as related to the routers actually connected to the Internet. The next section continues the discussion of iBGP but with a focus on some particular issues with routing that may require iBGP on routers other than the Internet-connected routers.

Avoiding Routing Loops when Forwarding Toward the Internet

A typical Enterprise network design uses default routes inside the Enterprise, as advertised by the IGP, to draw all Internet traffic toward one or more Internet-connected routers. The Internet-connected routers then forward the traffic into the Internet.

However, as discussed in Chapter 12, "Internet Connectivity and BGP," in the section "Choosing One Path over Another Using BGP," routing loops can occur when the Internet-connected routers do not have a direct connection to each other. For example, if the Internet-connected routers sit on opposite sides of the country, the two routers may be separated by several other internal routers in the Enterprise because they do not have a direct link.

To show a simple example, the same Enterprise network design shown in all previous figures in this chapter can be changed slightly by just disabling the FastEthernet link between the two routers, as shown in Figure 14-5.

Figure 14-5 shows an example of the looping problem. The figure uses the same general criteria as the other examples in this chapter, so that E1's best route for 192.135.250.0/28 points to Router E2 as the next hop. E1's best route for the next-hop IP address for its route to 192.135.250.0/28–regardless of whether using **next-hop self** or not–sends the packet back toward the Enterprise core. However, some (or possibly all) of the Enterprise routers internal to the Enterprise, such as WAN1 and Core1, use a default that sends all packets toward Router E1. Per the steps in the figure, the following happens for a packet destined to 192.135.250.1:

Step 1. WAN1 sends the packet using its default route to Core1.

Step 2. Core1 sends the packet using its default route to E1.

Step 3. E1 matches its BGP route for 192.135.250.0/28, with next hop E2 (10.100.1.2); the recursive lookup on E1 matches a route for 10.100.1.2 with next hop Core1, so E1 sends the packet back to Core1.

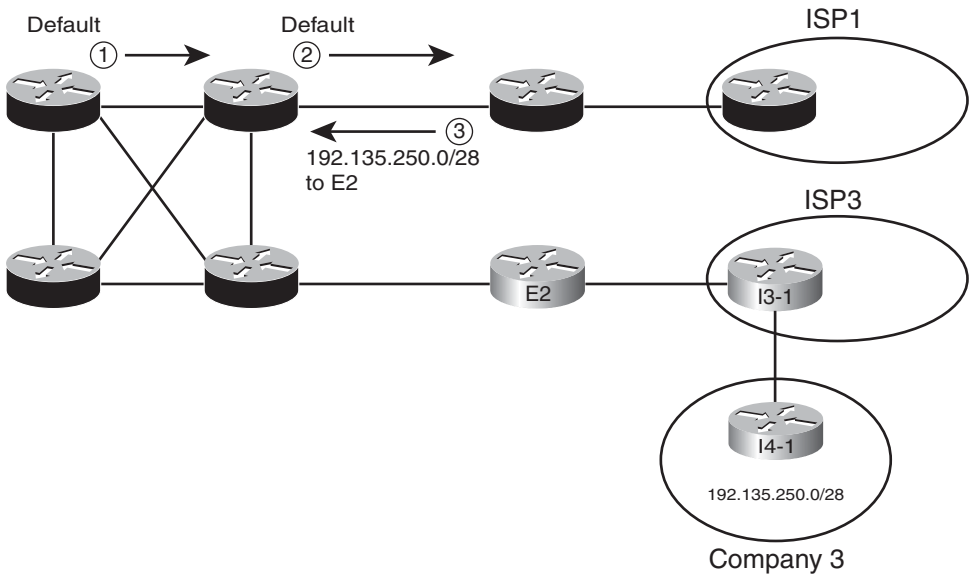


Figure 14-5 *Routing Loop for Packets Destined to 192.135.250.1*

At this point, Steps 2 and 3 repeat until the packet's TTL mechanism causes one of the routers to discard the packet.

The lack of knowledge about the best route for subnet 192.135.250.0/28, particularly on the routers internal to the Enterprise, causes this routing loop. To avoid this problem, internal routers, such as Core1 and Core2, need to know the best BGP routes. Two solutions exist to help these internal routers learn the routes:

- Run BGP on at least some of the routers internal to the Enterprise (such as Core1 and Core2 in Figure 14-5).
- Redistribute BGP routes into the IGP (not recommended).

Both solutions solve the problem by giving some of the internal routers the same best-path information already known to the Internet-connected routers–for example, if Core1 knew a route for 192.135.250.0/28, and that route caused the packets to go to Core2 next and then on to Router E2, the loop could be avoided. This section examines both solutions briefly.

Note: Other BGP features, namely BGP Confederations and BGP Route Reflectors, exist as well. However, these more advanced features are outside the scope of this book.

Using an iBGP Mesh

To let the internal routers in the Enterprise learn the best BGP routes, one obvious solution is to just run BGP on these routers as well. The not so obvious part relates to the implementation choice of what routers need to be iBGP peers with each other. Based on the topology shown in Figure 14-5, at first glance, the temptation might be to run BGP on E1, E2, Core1, and Core2, but use iBGP peers as shown in Figure 14-6.

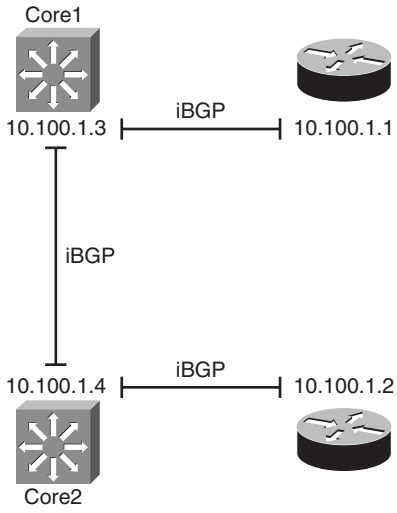


Figure 14-6 *Partial Mesh of iBGP Peers*

The iBGP peers shown in the figure actually match the kinds of IGP neighbor relationships you might expect to see with a similar design. With an IGP routing protocol, each router would learn routes and tell its neighbor so that all routers would learn all routes. Unfortunately, with this design, not all the routers learn all the routes because of the following feature of iBGP:

> When a router learns routes from an iBGP peer, that router does not advertise the same routes to another iBGP peer.

Key Topic

Note: This particular iBGP behavior helps prevent BGP routing loops.

Because of this feature, to ensure that all four routers in ASN 11 learn the same BGP routes, a full mesh of iBGP peers must be created. By creating an iBGP peering between all routers inside ASN 11, they can all exchange routes directly and overcome the restriction. In this case, six such neighborships exist: one between each pair of routers.

The configuration itself does not require any new commands that have not already been explained in this book. However, for completeness, Example 14-8 shows the configuration on both E1 and Core1. Note that all configuration related to iBGP has been included, and the routers use the loopback interfaces shown in Figure 14-6.

Example 14-8 *iBGP Configuration for the Full Mesh Between E1, E2, Core, and Core2–E1 and Core1 Only*

```
! First, E1's configuration
router bgp 11
 neighbor 10.100.1.2 remote-as 11
 neighbor 10.100.1.2 update-source loopback0
```

```
 neighbor 10.100.1.2 next-hop-self
!
 neighbor 10.100.1.3 remote-as 11
 neighbor 10.100.1.3 update-source loopback0
 neighbor 10.100.1.3 next-hop-self
!
 neighbor 10.100.1.4 remote-as 11
 neighbor 10.100.1.4 update-source loopback0
 neighbor 10.100.1.4 next-hop-self
```

```
! Next, Core1's configuration
interface loopback0
 ip address 10.100.1.3 255.255.255.255
!
router bgp 11
 neighbor 10.100.1.1 remote-as 11
 neighbor 10.100.1.1 update-source loopback0
!
 neighbor 10.100.1.2 remote-as 11
 neighbor 10.100.1.2 update-source loopback0
!
 neighbor 10.100.1.4 remote-as 11
 neighbor 10.100.1.4 update-source loopback0
```

The configurations on E1 and Core1 mostly match. The commonly used commands simply define the neighbor's ASN (**neighbor... remote-as**) and list the local router's BGP update source interface (**neighbor... update-source**). However, note that the engineer also configured E1–the Internet-connected router–with the **neighbor... next-hop-self** command. In this case, the Internet-connected routers want to set their own update-source IP addresses as the next hop for any routes. However, the engineer purposefully chose not to use this command on the two internal routers (Core1 and Core2) because the eventual destination of these packets will be to make it to either E1 or E2 and then out to the Internet. By making the next-hop router for all iBGP-learned routes an address on one of the Internet-connected routers, the packets will be correctly forwarded.

For perspective, Example 14-9 shows Core1's BGP table after adding the configuration shown in Example 14-8, plus the equivalent configuration in E2 and Core2. Focusing on the routes for 181.0.0.0/8 and 192.135.250.0/28 again, note that E1 and E2 had already agreed that E1's route for 181.0.0.0/8 was best and that E2's route for 192.135.250.0/28 was best. As a result, Core1 knows only one route for each of these destinations, as shown in the example. Also, the next-hop addresses for each route refer to the correct of the two Internet-connected routers: 10.100.1.1 (E1) for the route to 181.0.0.0/8 and 10.100.1.2 (E2) for the route to 192.135.250.0/28.

Example 14-9 *BGP Table on Router Core1*

```
Core-1# show ip bgp
BGP table version is 10, local router ID is 10.100.1.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
r i0.0.0.0          10.100.1.2               0    100      0 3 i
r>i                 10.100.1.1               0    100      0 1 i
* i128.107.0.0/19   10.100.1.2               0    100      0 i
*>i                 10.100.1.1               0    100      0 i
*>i181.0.0.0/8      10.100.1.1               0    100      0 1 2 111 112 i
*>i182.0.0.0/8      10.100.1.1               0    100      0 1 2 222 i
*>i183.0.0.0/8      10.100.1.1               0    100      0 1 2 i
*>i184.0.0.0/8      10.100.1.1               0    100      0 1 2 i
*>i185.0.0.0/8      10.100.1.1               0    100      0 1 2 i
*>i192.135.250.0/28 10.100.1.2               0    100      0 3 4 i
```

IGP Redistribution and BGP Synchronization

You can also redistribute BGP routes into the IGP to solve the routing loop problem. This solution prevents the routing loop by giving the internal Enterprise routers knowledge of the best exit point for each known Internet destination.

Although it solves the problem, particularly when just learning with lab gear at home, redistribution of BGP routes into an IGP is generally not recommended. This redistribution requires a relatively large amount of memory and a relatively large amount of processing by the IGP with the much larger number of routes to process. Redistributing the number of routes in the full Internet BGP table could crash the IGP routing protocols.

Note: BGP consumes less memory and uses less CPU for a large number of routes as compared to the equivalent number of routes advertised by an IGP, particularly when compared to OSPF. So using the iBGP mesh may cause internal routers to learn all the same routes but without risk to the IGP.

Although not recommended, the idea of redistributing eBGP-learned Internet routes into the Enterprise IGP needs to be discussed as a backdrop to discuss a related BGP feature called *synchronization* or *sync*. The term refers to the idea that the iBGP-learned routes must be synchronized with IGP-learned routes for the same prefix before they can be used. In other words, if an iBGP-learned route is to be considered to be a usable route, then that same prefix must be in the IP routing table and learned using some IGP protocol such as EIGRP or OSPF. More formally, the synchronization features tells a BGP router the following:

> Do not consider an iBGP-learned route as "best" unless the exact prefix was learned via an IGP and is currently in the IP routing table.

For companies, such as the Enterprise shown in Figure 14-5, the combination of redistributing eBGP routes into an IGP, and configuring synchronization on the two routers that run BGP (E1 and E2), prevents the routing loop shown in that figure. Again using prefix 192.135.250.0/28 as an example (see Figure 14-5), E2 again learned this prefix with eBGP. E1 learns this same prefix through its iBGP neighborship with E2, and both agree that E2's BGP route is best.

When E2 has successfully redistributed prefix 192.135.250.0/28 into the Enterprise's IGP (OSPF in the examples in this chapter), E1, with sync enabled, thinks like this:

> I see an IGP route for 192.135.250.0/28 in my IP routing table, so my iBGP route for that same prefix is safe to use.

However, if for some reason the redistribution does not result in an IGP route for 192.135.250.0/28, then E1 thinks as follows:

> I do not see an IGP-learned route for 192.135.250.0/28 in my IP routing table, so I will not consider the iBGP route through E2 to be usable.

In this second case, E1 uses its eBGP-learned route through I1-1, which defeats the routing loop caused at Step 3 of Figure 14-5.

Later IOS versions default to disable synchronization because most sites avoid redistributing routes from BGP into an IGP when using BGP for Internet routes, instead preferring iBGP meshes (or alternatives) to avoid these routing black holes. The setting is applied to the entire BGP process, with the **synchronization** command enabling synchronization and the **no synchronization** command (default) disabling it.

Note: The suggestion to avoid redistribution from BGP into an IGP generally applies to cases in which BGP is used to exchange Internet routes. However, BGP can be used for other purposes as well, including the implementation of Multiprotocol Label Switching (MPLS). Redistribution from BGP into an IGP when using BGP for MPLS is reasonable and commonly done.

Route Filtering and Clearing BGP Peers

BGP allows the filtering of BGP Update messages on any BGP router. The router can filter updates per neighbor for both inbound and outbound Updates on any BGP router.

After adding a new BGP filter to a router's configuration, the BGP neighbor relationships must be reset or cleared to cause the filter to take effect. The IOS BGP **clear** command tells the router specifically how to reset the neighborship. This section also examines the variations on the BGP **clear** command, including the more disruptive hard reset options and the less disruptive soft reset options.

BGP Filtering Overview

BGP filtering works generally like IGP filtering, particularly like EIGRP. Similar to EIGRP, BGP Updates can be filtered on any router, without the restrictions that exist for OSPF with various area design issues. The filtering can examine the prefix information about

each router and both the prefix and prefix length information, in either direction (in or out), on any BGP router.

The biggest conceptual differences between BGP and IGP filtering relate to what BGP can match about a prefix to make a choice of whether to filter the route. EIGRP focuses on matching the prefix/length. BGP can also match the prefix/length but can also match the large set of BGP Path Attributes (PA). For example, a filter could compare a BGP route's AS_Path PA and check to see if the first ASN is 4, that at least three ASNs exist, and that the AS_Path does not end with 567. The matching of routes based on their PA settings has no equivalent with any of the IGPs.

The biggest configuration difference between BGP and IGP filtering, beside the details of matching BGP PAs, has to do with the fact that the filters must apply to specific neighbors with BGP. With EIGRP, the filters can be applied to all outbound updates from EIGRP, or all inbound updates into EIGRP, using a single EIGRP **distribute-list** command. BGP configuration does not allow filtering of all inbound or outbound updates. Instead, the BGP filtering configuration enables filters per neighbor (using a **neighbor** command), referencing the type of BGP filter, the filter number or name, and the direction (in or out). So, a router could literally use the same filter for all BGP Updates sent by a router, but the configuration would require a **neighbor** command for each neighbor that enabled the same filter.

The ROUTE course and exam focus on Enterprise routing topics, whereas BGP filtering—especially the more detailed filtering with BGP PAs—is used most frequently by ISP network engineers. As a result, CCNP ROUTE appears to cover BGP filtering very lightly, at least compared to IGP filtering.

This section does briefly describe the BGP filtering commands, showing a few samples just for perspective. Table 14-2 summarizes the BGP filtering options and commands, along with the fields in the BGP Update message that can be matched with each type. Following the table, the text shows an example of how an Enterprise might apply an outbound and inbound filter based on prefix/length.

Table 14-2 *BGP Filtering Tools*

BGP Subcommand	Commands Referenced by neighbor Command	What Can Be Matched
neighbor distribute-list (standard ACL)	**access-list**, **ip access-list**	Prefix, with WC mask
neighbor distribute-list (extended ACL)	**access-list**, **ip access-list**	Prefix and prefix length, with WC mask for each
neighbor prefix-list	**ip prefix-list**	Exact or "first *N*" bits of prefix, plus range of prefix lengths

neighbor filter-list	**ip as-path access-list**	AS_PATH contents; all NLRI whose AS_PATHs are matched considered to be a match
neighbor route-map	**route-map**	Prefix, prefix length, AS_PATH, and/or any other PA matchable within a BGP route map

Inbound and Outbound BGP Filtering on Prefix/Length

Enterprises that choose to use BGP benefit from both learning routes from the connected ISPs and advertising the Enterprise's public prefix to the same ISPs. However, when the eBGP connections to the various ISPs come up, the Enterprise BGP routers advertise all the best routes in each router's BGP table over the eBGP connection. As a result, the ISPs could learn a best route that causes one ISP to send packets to the Enterprise, with the Enterprise then forwarding the packet out to another ISP. In such a case, the Enterprise AS would be acting as a *transit AS*; in other words, an AS through which packets go through, rather than being the destination or source of the packet.

The Enterprise engineers can, and probably should, make an effort to filter inappropriate routes sent to the ISP over the eBGP peer connections with the goal of preventing the Enterprise AS from becoming a transit AS. Additionally, the Enterprise can filter all private IP address ranges, in case any such address ranges get into the Enterprise BGP router's BGP table.

As an example, consider Figure 14-7, with the now-familiar prefix 192.135.250.0/28. As seen in earlier examples, both E1 and E2 learn this prefix, and both agree that the best route from ASN 11 (the Enterprise) toward this prefix is through E2. The figure shows the BGP routing updates as dashed lines.

E1's best route for 192.135.250.0/28 lists E2 as the next-hop router, so without any filtering in place, E1 then advertises prefix 192.135.250.0/28 to Router I1-1 in ISP1. I1-1 may be configured to filter this prefix. (In the examples throughout Chapters 13 and 14, router I1-1 was indeed configured to filter such prefixes.) However, if the Enterprise did not filter this prefix when advertising to ISP1, and ISP1 did not filter it, then ISP1 might choose the route through ASN 11 as its best route, making ASN 11 a transit AS for this prefix and consuming the Enterprise's Internet bandwidth.

Typically, an Enterprise would use outbound filtering on its eBGP neighborships, filtering all routes except for the known public prefixes that need to be advertised into the Internet. Example 14-10 shows just such a case, using the **neighbor prefix-list** command. The example also highlights a particularly useful command, **show ip bgp neighbor** *neighbor-id* **advertised-routes**, which shows the post-filter BGP update sent to the listed neighbor. The example shows the BGP Update before adding the filter, after adding the filter, and then after clearing the peer connection to router I1-1.

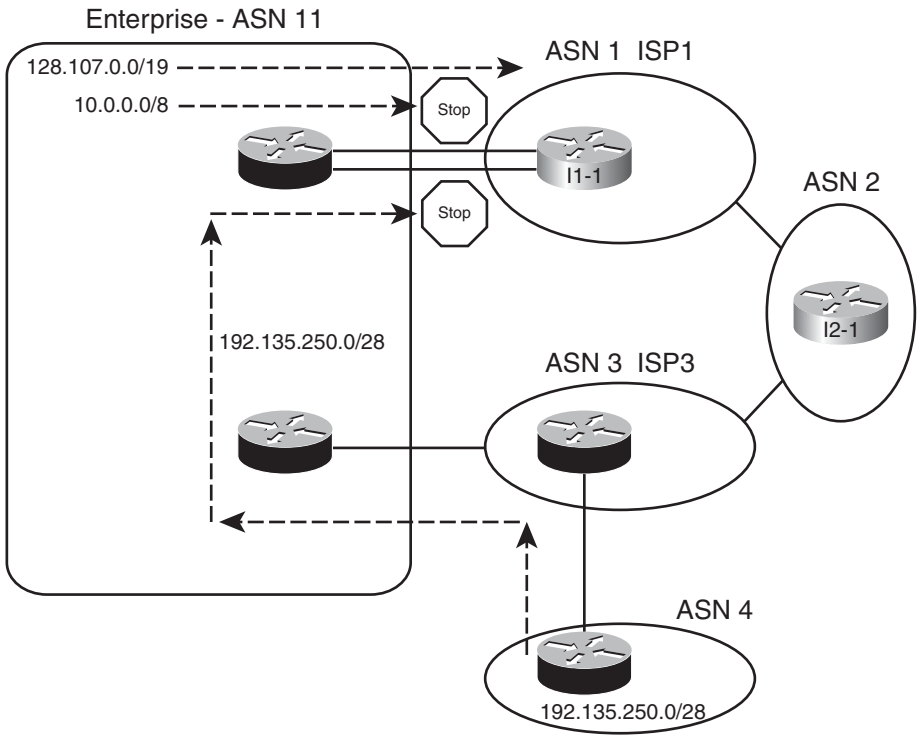


Figure 14-7 *The Need for Enterprise BGP Filtering*

Key Topic

Example 14-10 *Filtering to Allow Only Public Prefix 128.107.0.0/19 Outbound*

```
! The next command occurs before filtering is added.
E1# show ip bgp neighbor 1.1.1.1 advertised-routes
BGP table version is 16, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.0.0/19   0.0.0.0                                  32768 i
*>i192.135.250.0/28 10.100.1.2               0    100      0 3 4 i

Total number of prefixes 2

! Next, the filtering is configured.
E1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

E1(config)#ip prefix-list only-public permit 128.107.0.0/19
E1(config)#router bgp 11
```

```
E1(config-router)#neighbor 1.1.1.1 prefix-list only-public out
E1(config-router)#^Z
E1#

! Next, the Update sent to I1-1 is displayed.
E1# show ip bgp neighbor 1.1.1.1 advertised-routes
BGP table version is 16, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.0.0/19   0.0.0.0                              32768 i
*>i192.135.250.0/28 10.100.1.2               0    100      0 3 4 i

Total number of prefixes 2

! Next, the peer connection is cleared, causing the filter to take effect.
E1# clear ip bgp 1.1.1.1
E1#
*Aug 17 20:19:51.763: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down User reset
*Aug 17 20:19:52.763: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up

! Finally, the Update is displayed with the filter now working.
E1# show ip bgp neighbor 1.1.1.1 advertised-routes
BGP table version is 31, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.0.0/19   0.0.0.0                              32768 i

Total number of prefixes 1
```

Example 14-10 shows an interesting progression if you just read through the example start to finish. To begin, the **show ip bgp 1.1.1.1 advertised-routes** command lists the routes that E1 has advertised to neighbor 1.1.1.1 (Router I1-1) in the past. Then, the configuration shows a prefix-list that matches only 128.107.0.0/19, with a permit action; all other prefixes will be denied by the implied deny all at the end of each prefix list. Then, the **neighbor 1.1.1.1 prefix-list only-public out** BGP subcommand tells BGP to apply the prefix list to filter outbound routes sent to I1-1.

The second part of the output shows an example of how BGP operates on a Cisco router, particularly how BGP requires that the neighbor be cleared before the newly configured filter takes effect. Router E1 has already advertised two prefixes to this neighbor:

128.107.0.0/19 and 192.135.250.0/28, as seen at the beginning of the example. To make the filtering action take effect, the router must be told to clear the neighborship with router I1-1. The **clear ip bgp 1.1.1.1** command tells E1 to perform a hard reset of that neighbor connection, which brings down the TCP connection, and removes all BGP table entries associated with that neighbor. The neighbor (I1-1, using address 1.1.1.1) also removes its BGP table entries associated with Router E1. After the neighborship recovers, E1 resends its BGP Update to Router I1-1–but this time with one less prefix, as noted at the end of the example with the output of the **show ip bgp neighbor 1.1.1.1 advertised-routes** command.

This same filtering action could have been performed with several other configuration options: using the **neighbor distribute-list** or **neighbor route-map** commands. The **neighbor distribute-list** command refers to an IP ACL, which tells IOS to filter routes based on matching the prefix (standard ACL) or prefix/length (extended ACL). The **neighbor route-map** command refers to a route-map that can use several matching options to filter routes, keeping routes matched with a route-map permit clause and filtering routes matched with a route-map deny clause. Example 14-11 shows two such options just for comparison's sake.

Example 14-11 *Alternatives to the Configuration in Example 14-10*

```
! First option – ACL 101 as a distribute-list
access-list 101 permit ip host 128.107.0.0 host 255.255.224.0
router bgp 11
 neighbor 1.1.1.1 distribute-list 101 out

! Second option: Same prefix list as Example 12-10, referenced by a route map
ip prefix-list only-public seq 5 permit 128.107.0.0/19
!
route-map only-public-rmap permit 10
 match ip address prefix-list only-public
!
router bgp 11
 neighbor 1.1.1.1 route-map only-public-rmap out
```

Clearing BGP Neighbors

As noted in Example 14-10 and the related explanations, IOS does not cause a newly configured BGP filter to take effect until the neighbor relationship is cleared. The neighborship can be cleared in several ways, including reloading the router and by administratively disabling and re-enabling the BGP neighborship using the **neighbor shutdown** and **no neighbor shutdown** configuration commands. However, IOS supports several options on the **clear ip bgp** exec command for the specific purpose of resetting BGP connections. This section examines the differences in these options.

Each variation on the **clear ip bgp...** command either performs a hard reset or soft reset of one or more BGP neighborships. When a hard reset occurs, the local router brings down the neighborship, brings down the underlying TCP connection, and removes all BGP table entries learned from that neighbor. Both the local and neighboring router reacts just like it

does for any failed BGP neighborship by removing its BGP table entries learned over that neighborship. With a soft reset, the router does not bring down the BGP neighborship or the underlying TCP connection. However, the local router re-sends outgoing Updates, adjusted per the outbound filter and reprocesses incoming Updates per the inbound filter, which adjusts the BGP tables based on the then-current configuration.

Table 14-3 lists many of the variations on the **clear ip bgp** command, with a reference as to whether it uses hard or soft reset.

Key Topic

Table 14-3 *BGP* **clear** *Command Options*

Command	Hard or Soft	One or All Neighbors	Direction (in or out)
clear ip bgp *	Hard	all	both
clear ip bgp *neighbor-id*	Hard	one	both
clear ip bgp *neighbor-id* **out**	Soft	one	out
clear ip bgp *neighbor-id* **soft out**	Soft	one	out
clear ip bgp *neighbor-id* **in**	Soft	one	in
clear ip bgp *neighbor-id* **soft in**	Soft	one	in
clear ip bgp * **soft**	Soft	all	both
clear ip bgp *neighbor-id* **soft**	Soft	one	both

The commands listed in the table should be considered as pairs. In the first pair, both commands perform a hard reset. The first command uses a * instead of the neighbor IP address, causing a hard reset of all BGP neighbors, while the second command resets that particular neighbor.

The second pair of commands performs soft resets for a particular neighbor but only for outgoing updates, making these commands useful when a router changes its outbound BGP filters. Both commands do the same function; two such commands exist in part because of the history of the BGP implementation in the IOS. When issued, these two commands cause the router to reevaluate its existing BGP table and create a new BGP Update for that neighbor. The router builds that new Update based on the existing configuration, so any new or changed outbound filters affect the contents of the Update. The router sends the new BGP Update, and the neighboring router receives the new Update and adjusts its BGP table as a result.

The third pair of commands performs soft resets for a particular neighbor, but only for incoming updates, making these commands useful when a router changes its inbound BGP filters. However, unlike the two previous commands in the table, these two commands do have slightly different behavior and need a little more description.

The **clear ip bgp** *neighbor-id* **soft in** command, the older command of the two, works only if the configuration includes the **neighbor** *neighbor-id* **soft-reconfiguration inbound** BGP configuration command for this same neighbor. This configuration command causes the router to retain the received BGP Updates from that neighbor. This consumes extra memory on the router, but it gives the router a copy of the original pre-filter Update received from that neighbor. Using that information, the **clear ip bgp** *neighbor-id* **soft in** tells IOS to reapply the inbound filter to the cached received Update, updating the local router's BGP table.

The newer version of the **clear ip bgp** command, namely the **clear ip bgp** *neighbor-id* **in** command (without the **soft** keyword), removes the requirement for the **neighbor** *neighbor-id* **soft-reconfiguration inbound** configuration command. Instead, the router uses a newer BGP feature, the *route refresh* feature, which essentially allows a BGP router to ask its neighbor to re-send its full BGP Update. The **clear ip bgp** *neighbor-id* **in** command tells the local router to use route refresh feature to ask the neighbor to re-send its BGP Update, and then the local router can apply its current inbound BGP filters, updating its BGP table.

Example 14-12 shows a sample of how to confirm whether a router has the route refresh capability. In this case, both the local router (E1 from Figure 14-5) and the neighbor (I1-1 from Figure 14-5) have route refresh capability. As a result, E1 can perform a soft reset inbound without the need to consume the extra memory with the **neighbor soft-reconfiguration inbound** configuration command.

Example 14-12 *Alternatives to the Configuration in Example 14-10*

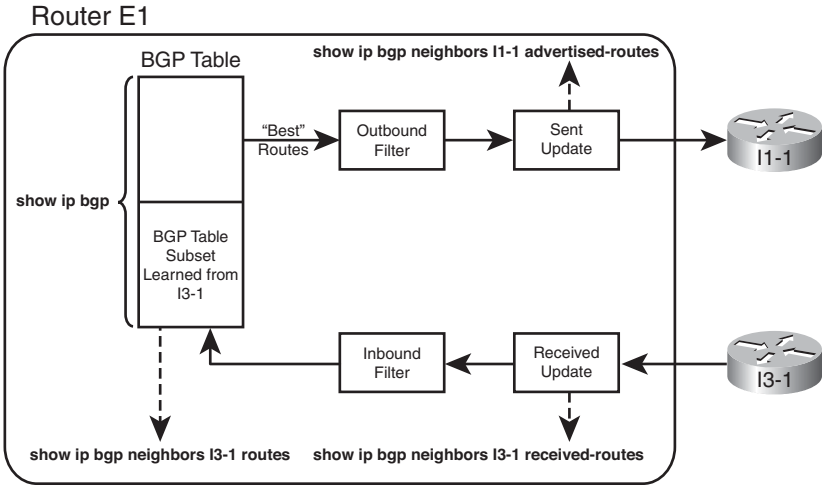
```
E1# show ip bgp neighbor 1.1.1.1
BGP neighbor is 1.1.1.1,  remote AS 1, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:04:21
  Last read 00:00:20, last write 00:00:48, hold time is 180, keepalive interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
! Lines omitted for brevity
```

The last pair of commands in Table 14-3 do a soft reset both inbound and outbound at the same time, either for all neighbors (the * option) or for the single neighbor listed in the **clear** command.

Displaying the Results of BGP Filtering

To verify and troubleshoot filtering configurations, you need to see both the before and after results of the filter. IOS provides several **show** commands that allow you to do exactly that. For instance, Example 14-10 shows several cases of the **show ip bgp neighbor advertised-routes** command that shows the post-filter BGP Updates sent by a router. Figure 14-8 summarizes these commands, showing how they can be used to display the pre- and post-filter BGP table contents. The figure shows Router E1, with inbound filtering

for Updates received from Router I3-1 and outbound filtering of BGP Updates sent to Router I1-1.



Key Topic

Figure 14-8 show *Commands Related to BGP Filtering*

The commands for displaying inbound updates, at the bottom of the figure, display output in the same format as the **show ip bgp** command. These commands restrict the contents to either exactly what has been received from that one neighbor (the **show ip bgp neighbor received-routes** command) or what has been received and passed through any inbound filter (the **show ip bgp neighbor routes** command).

One of the two commands helpful for the inbound direction, namely the **show ip bgp neighbor received-routes** command, requires the configuration of the BGP subcommand **neighbor soft-reconfiguration inbound**. As a result, to see the pre-filter BGP Update received from a neighbor, a router must configure this extra command, which causes the router to use more memory to store the inbound Update. (A router cannot use the BGP route refresh option just to get another copy of the incoming Update to list in this command.) However, when learning in a lab, the extra memory should not pose a problem.

Of the two commands for outbound filtering, the post-filter command is somewhat obvious, but there is no command to specifically display a pre-filter view of the BGP Update sent to a neighbor. However, BGP advertises the best route for each prefix in the BGP table, within certain restrictions. Those restrictions include that BGP will not advertise iBGP-learned routes to an iBGP peer, and a router will not advertise the best route back to the same neighbor that advertised that route. So, to see the pre-filter BGP table entries, use the **show ip bgp** command, look for all the best routes, and then consider the additional rules. Use the **show ip bgp neighbor advertised-routes** to display the post-filter BGP Update for a given neighbor.

Example 14-13 shows the output of these commands on E1. In this case, E1 has been already been configured with an inbound filter that filters inbound prefixes 184.0.0.0/8 and

185.0.0.0/8. (The filter configuration is not shown.) As a result, the post-filter output lists five prefixes, and the pre-filter output lists seven prefixes. The example also shows the error message when soft-reconfiguration is not configured.

Example 14-13 *Displaying the BGP Table Pre- and Post-Inbound Filter*

```
E1# show ip bgp neighbors 1.1.1.1 routes
BGP table version is 78, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          1.1.1.1                  0             0 1 i
*> 181.0.0.0/8      1.1.1.1                                0 1 2 111 111 i
*> 182.0.0.0/8      1.1.1.1                                0 1 2 222 i
*> 183.0.0.0/8      1.1.1.1                                0 1 2 i
*  192.135.250.0/28 1.1.1.1                                0 1 2 3 4 i

Total number of prefixes 5
E1# show ip bgp neighbors 1.1.1.1 received-routes
% Inbound soft reconfiguration not enabled on 1.1.1.1

E1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
E1(config)#router bgp 11
E1(config-router)#neighbor 1.1.1.1 soft-reconfiguration inbound
E1(config-router)#^Z
E1#
E1# show ip bgp neighbors 1.1.1.1 received-routes
BGP table version is 78, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          1.1.1.1                  0             0 1 i
*> 181.0.0.0/8      1.1.1.1                                0 1 2 111 111 i
*> 182.0.0.0/8      1.1.1.1                                0 1 2 222 i
*> 183.0.0.0/8      1.1.1.1                                0 1 2 i
*> 184.0.0.0/8      1.1.1.1                                0 1 2 i
*> 185.0.0.0/8      1.1.1.1                                0 1 2 i
*  192.135.250.0/28 1.1.1.1                                0 1 2 3 4 i

Total number of prefixes 7
```

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Planning Practice Tables."

Design Review Table

Table 14-4 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 14-4 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The plan shows a typical single-multihomed design with two routers connected to 2 ISPs. How will you ensure next-hop reachability? (2)	
The plan shows the same design as the last item. The two Enterprise Internet-connected routers do not have a direct link between each other. What methods discussed in this chapter can be used to prevent packet loops in the Enterprise core? (2)	
The plan shows the same design as the previous items but with public range 200.1.1.0/24 being the only public address range used by the Enterprise. How can the Enterprise avoid becoming a transit AS?	

Implementation Plan Peer Review Table

Table 14-5 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 14-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a typical single-multihomed design with two routers (R1 and R2) connected to 2 ISPs. Will R1 and R2 be BGP neighbors? Why?	
The plan shows the same design as the previous item. What configuration setting must be used to ensure the routers are iBGP rather than eBGP peers?	
The plan calls for filtering all prefixes except the 200.1.1.0/24 public address range when advertising the any eBGP peers. Which **neighbor** command options exist for filtering based on the prefix/length? (3)	

Create an Implementation Plan Table

This chapter does not focus on implementation or verification, but it did review one concept about static routes, as listed in Table 14-6.

Table 14-6 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an iBGP peer.	
Advertise the local router's Update source IP address as the next-hop address to iBGP peers.	
Configure an iBGP mesh with peers 1.1.1.1, 2.2.2.2, 3.3.3.3.	
Enable BGP synchronization.	
Configure filtering of routes sent to eBGP peer 9.9.9.9, using a prefix list to allow only 200.1.1.0/24.	
Configure filtering of routes sent to eBGP peer 9.9.9.9, using an ACL to allow only 200.1.1.0/24.	

Choosing Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 14-7 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 14-7 *Verification Plan Memory Drill*

Information Needed	Commands
Display a single-line neighbor status for all iBGP neighbors.	
Determine if a particular BGP table entry is iBGP-learned.	
Determine the next-hop IP address of an iBGP learned route.	
Identify the neighbor from which a BGP route was learned.	
Display one-line entries for all BGP table entries with a given prefix/length, plus any subnets inside that ranges.	
Display BGP routes learned from a neighbor, before being processed by an inbound filter.	
The same as the previous item, but after applying the inbound filter.	
Display BGP routes sent to a neighbor but after applying the outbound filter.	
Display whether a neighbor can perform BGP route refresh.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 14-8 lists a reference of these key topics and the page numbers on which each is found.

Table 14-8 *Key Topics for Chapter 14*

Key Topic Element	Description	Page Number
List	Configuration steps for iBGP peer using a loopback as Update source.	462
Text	Paragraph about the default behavior of eBGP and iBGP peers regarding the setting of the next-hop IP address.	464
Figure 14-4	eBGP next-hop default behavior.	470
Text	iBGP behavior regarding not forwarding iBGP learned routes.	473
Figure 14-7	Preventing an Enterprise from becoming a Transit AS.	479
Table 14-3	Options on the BGP **clear** command.	482
Figure 14-8	Reference for BGP filtering verification commands.	484

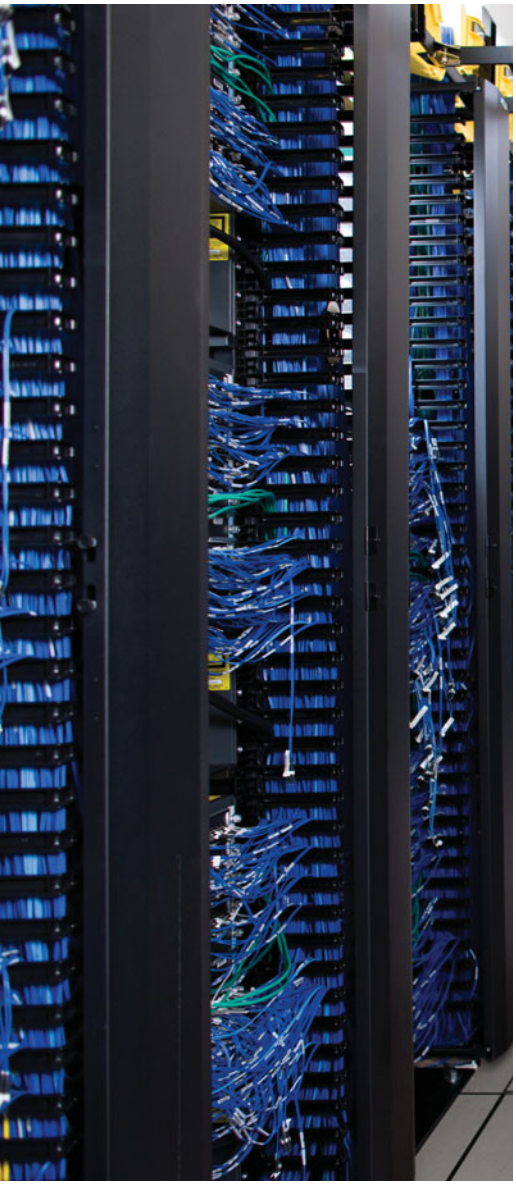
Complete the Tables and Lists from Memory

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

Definitions of Key Terms

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

BGP synchronization, iBGP Mesh, Next-hop self, BGP soft reset, BGP hard reset



This chapter covers the following subjects:

BGP Path Attributes and Best Path Algorithm: This section describes the BGP Path Attributes that have an impact on the BGP best path algorithm–the algorithm BGP uses to choose the best BGP route for each destination prefix.

Influencing an Enterprise's Outbound Routes: This section shows how to use the BGP features that influence the BGP best path algorithm.

Influencing an Enterprise's Inbound Routes with MED: This section shows how to use the MultiExit Discriminator BGP feature that influences the BGP best path algorithm for inbound routes.

CHAPTER 15

BGP Path Control

IGPs choose the best route based on some very straightforward concepts. RIP uses the least number of router hops between a router and the destination subnet. EIGRP uses a formula based on a combination of the fastest bandwidth and least delay. And OSPF uses lowest cost but with that cost typically calculated based on bandwidth.

BGP uses a much more detailed process to choose the best BGP route. BGP does not consider router hops, bandwidth, or delay when choosing the best route to reach each subnet. Instead, BGP defines several items to compare about the competing routes, in a particular order. Some of these comparisons use BGP features that can be set based on the router configuration, allowing network engineers to then influence which path BGP chooses as the best path.

BGP's broader set of tools allow much more flexibility when influencing the choice of best route. This BGP best path process also requires only simple comparisons by the router to choose the best route for a prefix. Although the detail of the BGP best path selection process requires more work to understand, that complexity gives engineers additional design and implementation options, and gives engineers many options to achieve their goals when working with the large interconnected networks that comprise the Internet.

Chapter 15 completes the BGP coverage in this book by examining the topic of BGP Path Control, including BGP Path Attributes (PA), the BGP Best Path selection process, along with a discussion of how to use four different features to influence the choice of best path by setting BGP PA values.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 15-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 15-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
BGP Path Attributes and Best Path Algorithm	1–3

Influencing an Enterprise's Outbound Routes	4–6
Influencing an Enterprise's Inbound Routes with MED	7

1. An engineer is preparing an implementation plan in which the configuration needs to influence BGP's choice of best path. Which of the following is least likely to be used by the configuration in this implementation plan?

    a. Weight
    b. Origin code
    c. AS_Path
    d. Local_Pref

2. Router R1 learns two routes with BGP for prefix 200.1.0.0/16. Comparing the two routes, route 1 has a longer AS_Path Length, bigger MED, bigger Weight, and smaller Local Preference. Which of the following is true about Router R1's choice of best path for this prefix?

    a. Route 1 is the best route.
    b. Route 2 is the best route.
    c. The routes tie as best, but one will be picked to be placed in the routing table based on tiebreakers.
    d. Neither route is considered best.

3. Router R1 learns two routes with BGP for prefix 200.1.0.0/16. Comparing the two routes, route 1 has a shorter AS_Path Length, smaller MED, the same Weight, and smaller Local Preference. Which of the following is true about Router R1's choice of best path for this prefix?

    a. Route 1 is the best route.
    b. Route 2 is the best route.
    c. The routes tie as best, but one will be picked to be placed in the routing table based on tiebreakers.
    d. Neither route is considered best.

4. An engineer has been told to create an implementation plan to influence the choice of best BGP route on a single router using the Weight feature. The sole Enterprise Internet-connected router, Ent1, has neighbor relationships with Routers ISP1 and ISP2, which reside inside two different ISPs. The goal is to prefer all routes learned from ISP1 over ISP2 using Weight. Which of the following answers lists a configuration step that would not be useful for achieving these goals? (Choose two.)

    a. Configuring the **neighbor weight** command on Ent1.
    b. Having the ISPs configure the **neighbor route-map out** command on ISP1 and ISP2, with the route map setting weight.
    c. Configuring the **set weight** command inside a route map on Router Ent1.

d. Configuring a prefix list to match all class C networks.

5. The following output on Router R1 lists details of a BGP route for 190.1.0.0/16. Which of the following is true based on this output? (Choose two.)

```
R1# show ip bgp 190.1.0.0/16
BGP routing table entry for 190.1.0.0/16, version 121
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  1 2 3 4
    1.1.1.1 from 2.2.2.2 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

a. R1 has a **neighbor 1.1.1.1** command configured.

b. R1 has a **neighbor 2.2.2.2** command configured.

c. The **show ip bgp** command lists a line for 190.1.0.0/16 with both an ">" and an "i" on the left.

d. R1 is in ASN 1.

6. An Enterprise router, Ent1, displays the following excerpt from the **show ip bgp** command. ENT1 has an eBGP connection to an ISP router with address 3.3.3.3 and an iBGP connection to a router with address 4.4.4.4. Which of the following is most likely to be true?

```
   Network          Next Hop            Metric LocPrf Weight Path
*>                   3.3.3.3                  0             0 1 1 1 1 2
18 i
```

a. The Enterprise likely uses ASN 1.

b. The neighboring ISP likely uses ASN 1.

c. The route has been advertised through ASN 1 multiple times.

d. Router Ent1 will add another ASN to the AS_Path before advertising this route to its iBGP peer (4.4.4.4).

7. The following line of output was gathered on Enterprise Router Ent1 using the command **show ip route**. Which of the following answers is most likely to be true, based on this output?

```
B        128.107.0.0 [20/10] via 11.11.11.11, 00:02:18
```

a. This router has set the Weight of this route to 10.

b. This router's BGP table lists this route as an iBGP route.

c. This router's MED has been set to 10.

d. This router's BGP table lists an AS_Path length of 10 for this route.

Foundation Topics

BGP Path Attributes and Best Path Algorithm

BGP supports a wide variety of Path Attributes. Some of the PAs exist solely to be used as part of the litany of options in the BGP best path algorithm, some have nothing to do with the BGP best path algorithm, and some impact the best path algorithm as well as being used for other purposes. For example, the Local Preference PA exists to give control to a single AS regarding their outbound routes from an AS-wide perspective. Conversely, the BGP Next_Hop PA provides BGP a place to list the next-hop IP address for a path, but it does not provide a useful means for engineers to set different values for the purpose of influencing the best path choice.

The term *BGP best path algorithm* refers to the process by which BGP on a single router examines the competing BGP paths (routes) in its BGP table, for a single prefix, choosing one route as the best route. The best path algorithm has many steps, but it eventually results in the choice of a single route for each prefix as that router's best BGP path.

The initial major section of this chapter examines the BGP PAs used by the BGP best path algorithm, the BGP best path algorithm itself, and some related topics.

BGP Path Attributes

BGP Path Attributes define facts about a particular route or path through a network. Each PA defines something different about the path, so to truly understand BGP PAs, you need to examine each PA. This section begins by reviewing a few PAs that should now be familiar to you if you have read the preceding BGP chapters, and then this section introduces a few new PAs.

BGP uses the *Autonomous System Path* (AS_Path) PA for several purposes, as already seen in Chapters 12, 13, and 14. This particular PA lists the ASNs in the end-to-end path. BGP uses the AS_Path PA as its primary loop-prevention tool: When an eBGP peer receives an Update, if its own ASN is already in the received AS_Path, then that route has already been advertised into the local ASN and should be ignored. In addition to loop prevention, the BGP best path algorithm uses the AS_Path PA to calculate the AS_Path length, which the algorithm considers as one of its many steps.

BGP also defines the *next-hop IP address* (Next_Hop) of a route as a PA. BGP may advertise one of several different IP addresses as a route's Next_Hop, depending on several factors. To support such features, BGP needs to list the Next_Hop IP address for each path (route), and BGP defines this concept in the Next_Hop PA. The best path algorithm includes a check related to the Next_Hop IP address of the route.

Table 15-2 lists these two PAs, plus a few more PAs, and a related BGP feature (Weight) that is not a PA but is used by Cisco BGP best path implementation. The table lists the PAs in the same order that they will be considered by the BGP best path algorithm. The table also describes each feature listed in the table, relative to whether it is most useful to influence outbound routes (away from the Enterprise) and inbound routes (toward the Enterprise).

Table 15-2 *BGP Path Attributes That Affect the BGP Best Path Algorithm*

Key Topic

PA	Description	Enterprise Route Direction (Typical)
NEXT_HOP	Lists the next-hop IP address used to reach a prefix.	N/A
Weight[1]	A numeric value, range 0 through $2^{16} - 1$, set by a router when receiving Updates, influencing that one router's route for a prefix. Not advertised to any BGP peers.	Outbound
Local Preference (LOCAL_PREF)	A numeric value, range 0 through $2^{32} - 1$, set and communicated throughout a single AS for the purpose of influencing the choice of best route for all routers in that AS.	Outbound
AS_PATH (length)	The number of ASNs in the AS_Path PA.	Outbound, Inbound
ORIGIN	Value implying the route was injected into BGP; I (IGP), E (EGP), or ? (incomplete information).	Outbound
Multi Exit Discriminator (MED)	Set and advertised by routers in one AS, impacting the BGP decision of routers in the other AS. Smaller is better.	Inbound

[1]Weight is not a BGP PA; it is a Cisco-proprietary feature that acts somewhat like a PA.

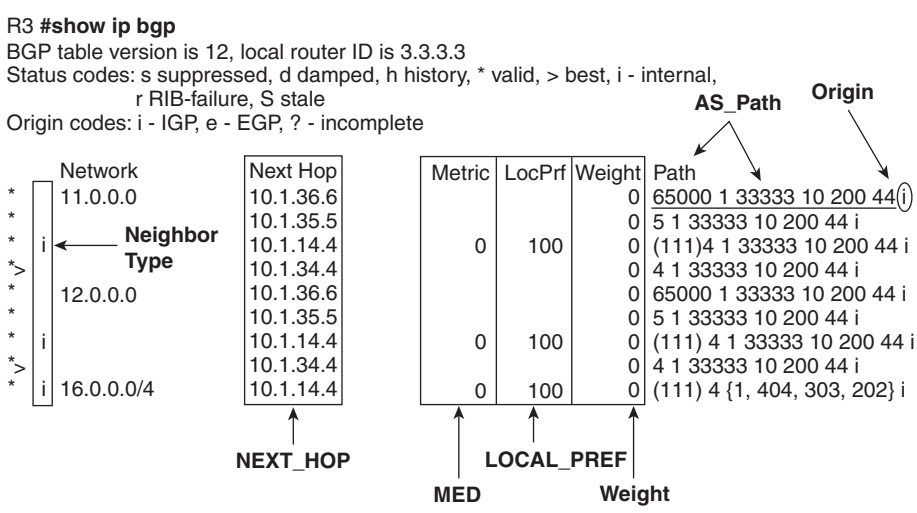
The short descriptions in the table can be helpful for review when doing your final preparation study, but the table does not hold enough information to truly appreciate how an engineer might use these PAs effectively. The second and third major sections of this chapter examine most of these PAs and how to influence the best path choice with each.

To find the current settings of the features in Table 15-2, you can use commands like **show ip bgp** and **show ip bgp** *prefix/length*. However, picking the values out of the clutter in the output of the **show ip bgp** command can be a challenge. Figure 15-1 shows a sample of this command's output and some notations on where to find the various PA settings.

The examples throughout this chapter include examples of these commands, along with the PA settings as changed by various route maps.

Overview of the BGP Best Path Algorithm

The BGP best path algorithm follows the steps shown in shorthand form in Table 15-3. The table lists steps 0 through 8, a short descriptive phrase, and a notation about the criteria for one value to be better than another.



Key Topic

Figure 15-1 *Finding PA Settings in the Output of the* **show ip bgp** *Command*

Key Topic

Table 15-3 *BGP Decision Process Plus Mnemonic: N WLLA OMNI*

Step	Mnemonic Letter	Short Phrase	Which Is Better?
0	N	Next hop: reachable?	If no route to reach Next_Hop, router cannot use this route.
1	W	Weight	Bigger.
2	L	LOCAL_PREF	Bigger.
3	L	Locally injected routes	Locally injected is better than iBGP/eBGP learned.
4	A	AS_PATH length	Smaller.
5	O	ORIGIN	Prefer I over E. Prefer E over ?
6	M	MED	Smaller.
7	N	Neighbor Type	Prefer eBGP over iBGP.
8	I	IGP metric to Next_Hop	Smaller.

Note: The step numbering of the BGP best path steps does not exist in the BGP RFCs. The steps are numbered in this book for easier reference. Because the RFCs do not dictate

a particular step numbering, other references likely use different step numbers; do not be concerned about memorizing the step numbers.

Starting with a Step 0 may seem odd, but it helps make an important point about the logic listed at this step. Some BGP best path references include the logic in this step as a best path step, and some just list this same information as a side note. Regardless, the step 0 concept is important. For step 0, a router looks at the BGP route and compares the Next_Hop IP address to the IP routing table.

If that router does not have a matching IP route for the BGP route's Next_Hop IP address, then that router will not know how to forward packets for that particular prefix, using that particular route. To avoid using such a route, at Step 0, the BGP best path algorithm removes such routes from consideration.BGP then uses the following eight steps, in order, until one best route is chosen for a given prefix.

If a router still did not determine a best route when finishing Step 8, the router takes several other small steps to break the tie. At this point, the competing routes are considered to be just as good as each other. However, unlike IGPs, BGP needs to choose one and only one route as best, in part because BGP advertises only the best routes to its neighbors. In such cases, BGP breaks the tie with these additional steps, which would be considered Steps 9-11:

Step 9. Oldest (longest-known) eBGP route

Step 10. Lowest neighbor BGP RID

Step 11. Lowest neighbor IP address

Taking a more detailed view of the entire best path algorithm, BGP begins by choosing the oldest known route for a given prefix as the best route. It then takes the next longest-known route for that same prefix and compares the two routes using the best path algorithm. The router eventually chooses one of the two BGP routes as the best path (route). If another route exists for the same prefix, the router repeats the process, using the winner of the previous comparisons and the new route, choosing one of those as the better route. The process continues until all routes have been considered, with one route being listed as best in the BGP table.

For example, if Router R1 were considering two routes for prefix 181.0.0.0/8, it would first make sure that both routes had reachable Next_Hop IP addresses. The router would then compare the weight settings, choosing the route with the bigger weight. If they tied on weight, the router would prefer the route with a bigger Local_Pref. If again a tie, the router would prefer the one route that was injected into BGP locally (using the **network** command or using route redistribution). If neither or both routes were locally injected, the router moves on to AS_Path length, and so on, until the router chooses one of the two as the better route.

As soon as one of the steps determines a best route, the comparison of those two routes stops.

Perspectives on the Core 8 Best Path Steps

Some of the BGP best path steps purposefully give the engineer a tool for influencing the choice of best path, whereas other steps have a different purpose, often simply being a side effect of some BGP feature. So, when an engineer starts building a BGP implementation plan, only a subset of the core 8 BGP best path steps need be considered, as follows:

- Weight (Step 1)
- Local_Pref (Step 2)
- AS_Path Length (Step 4)
- MED (often called metric) (Step 6)

Because the ROUTE exam focuses on the more practical aspects of BGP for Enterprises, it gives much more attention to these four features and less attention to the other BGP best path steps. This chapter describes each of these four features to some depth in the context of best path selection. However, before focusing on these four items, it can be helpful to see a small glimpse into the meaning of the other steps, which can be helpful as you work to memorize the steps in the BGP best path algorithm.

Step 3 compares the source from which the routes were added to the BGP table. When the BGP best path algorithm compares two routes at this step, if one were injected into BGP locally, and the other were not (it was learned from a BGP peer), the router chooses the route that was injected locally. Chapter 13, "External BGP," section "Injecting Routes into BGP for Advertisement to the ISPs," describes the two ways to locally inject these routes, the **network** command and redistribution from an IGP.

Step 5 refers to the BGP Origin PA. The Origin PA attempts to identify the source from *outside BGP* from which the route was injected into BGP. The three Origin code values are

- **I:** Injected from an IGP (for example, redistribution from EIGRP)
- **E:** Injected from Exterior Gateway Protocol (EGP)
- **?:** Undetermined

Although the original intent of the Origin PA is to identify the source from which BGP learned the route, routers can also set the Origin PA as part of a strategy to influence the BGP best path.

Step 7 refers to the Neighbor type: iBGP or eBGP. Remembering that BGP compares two routes at a time, if one is learned with eBGP, and the other with iBGP, the router chooses the eBGP route as best. Using this feature to influence the best path choice would be difficult, because the ASN in which a router resides is fixed by the BGP design.

Finally, Step 8 refers to the IGP metric to the Next_Hop address. At this step, the router compares the metrics of the IP routes for each Next_Hop IP address and chooses the BGP route with the lower IGP metric to its Next_Hop. (If an IGP-learned route is not used, for example, if both use connected routes, BGP considers the metrics to tie.) It is conceivable that an engineer might tune the IGP to manipulate BGP's best path choice, but this step is so far into the algorithm that the earlier and more flexible settings would be much better options.

Memorization Tips for BGP Best Path

This short section suggests a mnemonic tool to help you memorize Steps 0 through 8 of the BGP best path algorithm. Feel free to skim this section for now, or ignore it entirely–there is no requirement that you memorize the best path algorithm using the mnemonics in this section. (However, you may want to at least review upcoming Figure 15-2, which gives a good visual reference for some of the information summarized in Table 15-3.) But you should plan on memorizing the list at some point before the exam, even if you ignore the mnemonic device.

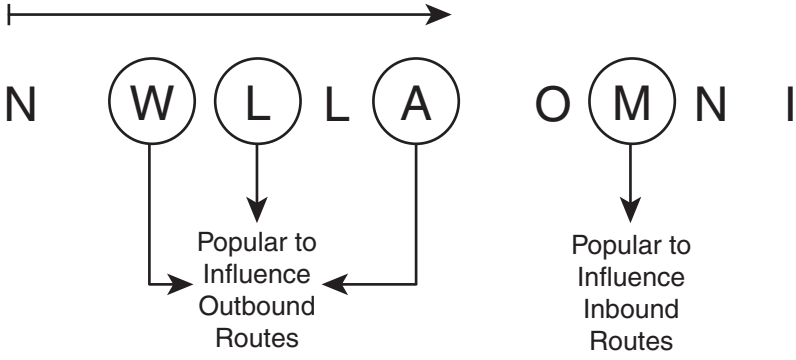


Figure 15-2 *BGP Best Path Mnemonics*

First, if you refer back to the BGP best path algorithm as listed in table 15-3, you see that the second column lists a single-letter mnemonic letter. These letters match the first letter of the description in the third column of that table. Then, take these initial letters and group them as follows:

- N
- WLLA
- OMNI

The N is listed separately because it represents the "is the next-hop reachable" logic of Step 0 and is somewhat separate from the other steps.

The mnemonic groups the eight main steps as two sets of four letters for a couple of reasons. Both sets can be pronounced, even if they don't spell words. It should be easier to memorize as two sets of four. And maybe most important, the first set of four letters, representing Steps 1 through 4, include all the features that engineers typically use to influence outbound routes from the Enterprise:

- **WLLA:** Refers to the three steps that an engineer might use to influence outbound routes: Weight, Local_Pref, and AS_Path length. (Additionally, the second L, in WLLA for Step 3, represents the "Locally injected routes" choice.)

- **OMNI:** As listed in Table 15-3, the letters represent Origin (I or ?), MED, neighbor type (eBGP over iBGP), and IGP metric to Next-hop.

So, if you can memorize N WLLA OMNI, by the time you've read this chapter you can probably pick out which of those correlate to the four bigger topics later in this chapter: Weight, Local_Pref, AS_Path length, and MED. Hopefully with a little more study, you can memorize the rest of the list.

Figure 15-2 shows the mnemonic letters in graphical form just as another aid in memorizing the steps. It also shows a reminder of which features are most likely to be used to influence outbound routes from the Enterprise, and the one setting (MED) most likely to be used to influence inbound routes into the Enterprise.

The rest of this chapter focuses on a deeper explanation of the four best path steps that engineers typically use to influence the choice of best path.

Influencing an Enterprise's Outbound Routes

This section examines three different features that can influence the outbound routes from an Enterprise: Weight, the Local_Pref PA, and AS_Path length. The topics are listed in the order used by the BGP best path algorithm (Steps 1, 2, and 4). It also introduces the concept of a Routing Table Manager (RTM) function on a router.

Influencing BGP Weight

A Cisco router can use the BGP Weight, on that single router, to influence that one router's choice of outbound route. To do so, when a router receives a BGP Update, that router can set the Weight either selectively, per route, using a route map, or for all routes learned from a single neighbor. The router's best path algorithm then examines the Weight of competing routes, choosing the route with the bigger Weight.

The Cisco-proprietary Weight settings configured on a single router can influence only that one router because the Weight cannot be communicated to other neighboring BGP routers. So, to use the Weight, a router must be configured to examine incoming Updates to set the Weight. The Weight cannot simply be learned in a received Update because that Update message does not support a field in which to communicate the Weight setting.

Table 15-4 summarizes some of the key facts about BGP administrative Weight. Following the table, the text first explains a sample internetwork and its existing configuration, a configuration that begins with configurations that do not set any values that influence the choice of best paths. The next section shows how to set the Weight using the **neighbor route-map in** command, which allows a router to set different Weights for different routes. The second example shows how to set the Weight for all routes learned from a neighbor, using the **neighbor weight** command.

Key Topic

Table 15-4 *Key Features of Administrative Weight*

Feature	Description
Is it a PA?	No; Cisco proprietary feature

Table 15-4 *Key Features of Administrative Weight*

Feature	Description
Purpose	Identifies a single router's best route
Scope	Set on inbound route Updates; influences only that one router's choice
Range	0 through 65,535 ($2^{16} - 1$)
Which is best?	Bigger values are better
Default	0 for learned routes, 32,768 for locally injected routes
Defining a new default	Not supported
Configuration	**neighbor route-map** (per prefix) **neighbor weight** (all routes learned from this neighbor)

Note: For those of you memorizing using the N WLLA OMNI mnemonic, Weight is the W in WLLA.

Sample Internetwork Used in the Weight Examples

Figure 15-3 shows a sample Internetwork used to demonstrate setting the Weight. The figure shows a single Enterprise and a single Enterprise router. The following design requirements have already been met by the configuration in Router E1 and in the ISP routers:

- E1 and I1-1 uses loopback IP addresses (11.11.11.11 and 1.1.1.1) for their neighborship.
- E1 and I3-1 use interface IP addresses for their neighborship.
- None of the routers have attempted to change any settings that can impact the choice of best path.

Next, to have some routes to manipulate with the upcoming examples, the ISP routers each advertise BGP routes for the same five prefixes. Figure 15-4 shows five such prefixes that both ISPs advertise to E1.

Just to get a little deeper understanding of the best path algorithm before getting into the Weight configuration, consider the original configuration state of the sample internetwork, with no attempt to influence E1's choice of best path. The best path for four of the five prefixes will be obvious. Prefixes 181.0.0.0/8 and 182.0.0.0/8 have a shorter AS_Path through ISP1, and 184.0.0.0/8 and 185.0.0.08 have a shorter AS_Path through ISP3. Only 183.0.0.0/8 is in question because its AS_Path length for the competing routes is equal. Example 15-1 shows the output of the **show ip bgp 176.0.0.0/4 longer-prefixes** command, which lists all five of the BGP prefixes listed in Figure 15-4, confirming the results. (Prefix 176.0.0.0/4 implies a range of values whose first octets are in the range 176 through 191, which includes the routes listed in Example 15-1.)

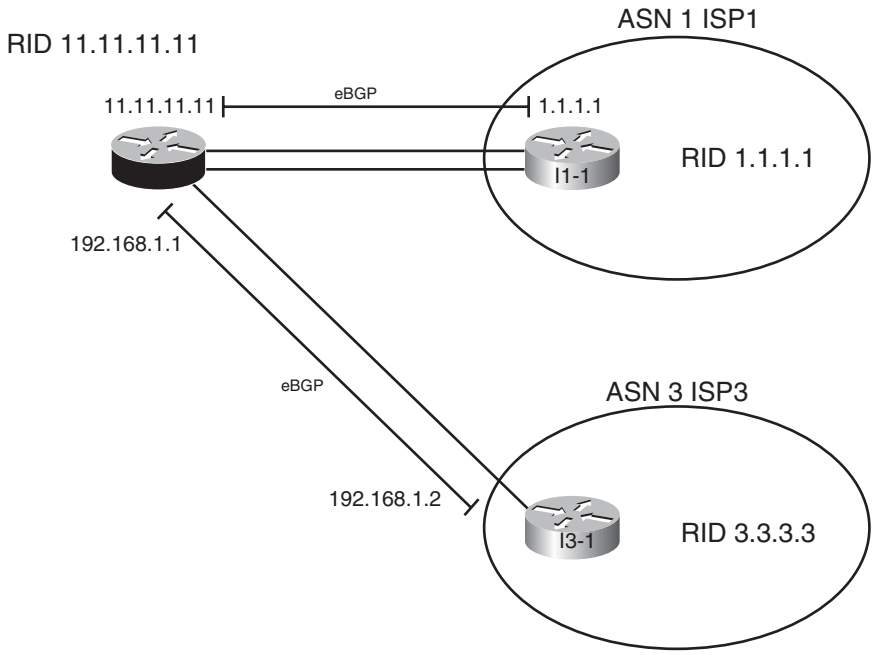


Figure 15-3 *Sample Internetwork for BGP Weight Examples*

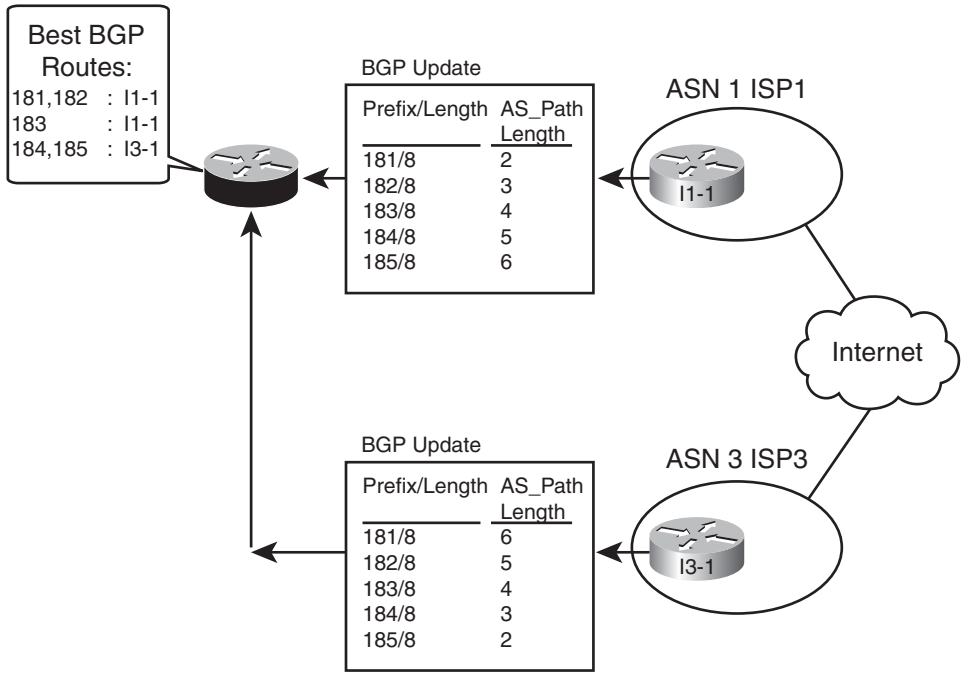


Figure 15-4 *Prefixes and AS_Path Lengths Used in Upcoming Examples*

Example 15-1 *BGP Configuration on E1: Neighborships Configured*

```
E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 41, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  181.0.0.0/8      192.168.1.2              0             0 3 2 50 51 52 1811 i
*>                  1.1.1.1                  0             0 1 1811 i
*  182.0.0.0/8      192.168.1.2              0             0 3 2 50 51 1822 i
*>                  1.1.1.1                  0             0 1 2 1822 i
*  183.0.0.0/8      192.168.1.2              0             0 3 2 50 1833 i
*>                  1.1.1.1                  0             0 1 2 50 1833 i
*> 184.0.0.0/8      192.168.1.2              0             0 3 2 1844 i
*                   1.1.1.1                  0             0 1 2 50 51 1844 i
*> 185.0.0.0/8      192.168.1.2              0             0 3 1855 i
*                   1.1.1.1                  0             0 1 2 50 51 52 1855 i
```

First, consider the best path algorithm on Router E1 for 181.0.0.0/8. E1 knows two BGP routes for 181.0.0.0/8, as expected. The one listed as the best path has 1.1.1.1 (I1-1) as Next_Hop. The following list outlines the best path logic:

Step 0. The Next_Hop of each is reachable. (Otherwise the neighbors would not be up.)

Step 1. The Weight ties (both 0).

Step 2. The Local_Pref ties (unset, so no value is listed; defaults to 100).

Step 3. Neither route is locally injected; both are learned using BGP, so neither is better at this step.

Step 4. AS_Path length is shorter for the route through I1-1 (1.1.1.1).

Next, consider the example of the route to 183.0.0.0/8. E1 currently lists the path through I1-1 (1.1.1.1) as best, but the best path decision actually falls all the way to Step 9. For completeness' sake, E1's best path logic runs as follows:

Step 0. The Next_Hop of each is reachable (otherwise the neighbors would not be up)

Step 1. The Weight ties (both 0).

Step 2. The Local_Pref ties (unset, defaults to 100).

Step 3. Neither route is locally injected.

Step 4. AS_Path length is 4 in both cases.

Step 5. Both Origin codes are i.

Step 6. MED, listed under the Metric column, ties (0).

Step 7. Neighbor type for each neighbor is eBGP.

Step 8. IGP metric does not apply, because neither uses IGP routes. (The routes from E1 to 1.1.1.1 are static routes.)

Step 9. The route learned from 1.1.1.1 is the oldest route.

Although you may believe the claims at Step 9, the output in Example 15-1 does not explicitly state that fact. However, when IOS lists output in the variations of the **show ip bgp** command, the oldest route for each prefix is listed last, and the newest (most recently learned) is listed first. Example 15-2 confirms this logic, and confirming how Step 9 works in this case. Example 15-2 clears peer 1.1.1.1 (I1-1), making E1's route through 192.168.1.2 (I3-1) become the oldest known route for 183.0.0.0/8:

Example 15-2 *Clearing Neighbors to Force a* **New Route**

```
E1# clear ip bgp 1.1.1.1
E1#
*Aug 24 11:30:41.775: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down User reset
*Aug 24 11:30:43.231: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 47, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 181.0.0.0/8      1.1.1.1                  0             0 1 1811 i
*                   192.168.1.2              0             0 3 2 50 51 52 1811 i
*> 182.0.0.0/8      1.1.1.1                  0             0 1 2 1822 i
*                   192.168.1.2              0             0 3 2 50 51 1822 i
*  183.0.0.0/8      1.1.1.1                  0             0 1 2 50 1833 i
*>                  192.168.1.2              0             0 3 2 50 1833 i
*  184.0.0.0/8      1.1.1.1                  0             0 1 2 50 51 1844 i
*>                  192.168.1.2              0             0 3 2 1844 i
*  185.0.0.0/8      1.1.1.1                  0             0 1 2 50 51 52 1855 i
*>                  192.168.1.2              0             0 3 1855 i
```

After the hard reset of peer 1.1.1.1, E1's oldest-known route for 183.0.0.0/8 is the route through 192.168.1.2, listed second (last). That E1 now chooses this route as best is another confirmation that E1's best path decision fell to Step 9.

Setting the BGP Administrative Weight Using a Route Map

The **neighbor** *neighbor-ip* **route-map in** BGP subcommand tells a router to apply the route map to all BGP Updates received from the listed neighbor. Such route maps *always attempt to filter routes*. The router allows routes first matched in a permit clause and filters (discards) routes first matched with a deny clause.

BGP route maps can also be used to change the PAs of routes by using the **set** command. For example, a router could use a **neighbor 1.1.1.1 route-map fred in** command. The route

map could contain permit clauses that cause some routes to not be filtered. In those same route map clauses, the inclusion of commands such as **set weight 100** and **set local-preference 200** can be used to set items such as the Weight or Local_Pref of a route. (Although you can configure a **set** command in a route map deny clause, the set has no effect because the deny clause filters the route.)

Example 15-3 shows a sample configuration that sets the Weight for prefix 181.0.0.0/8 as learned from I3-1 (neighbor ID 192.168.1.2). As shown in Examples 15-1, E1's original best route for this prefix is through I1-1 (1.1.1.1), due to the shorter AS_Path length at Step 4 of the best path algorithm. By setting the Weight higher on the route learned from I3-1, E1 now chooses the route through I3-1.

Example 15-3 *Setting the Weight to 50 for 181/8, as Learned from I3-1*

```
E1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
E1(config)# ip prefix-list match-181 permit 181.0.0.0/8
E1(config)# route-map set-weight-50 permit 10
E1(config-route-map)# match ip address prefix-list match-181
E1(config-route-map)# set weight 50
E1(config-route-map)# route-map set-weight-50 permit 20
E1(config-route-map)# router bgp 11
E1(config-router)# neighbor 192.168.1.2 route-map set-weight-50 in
E1(config-router)# ^Z
E1#
E1# clear ip bgp 192.168.1.2 soft
E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 48, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  181.0.0.0/8      1.1.1.1                  0             0 1 1811 i
*>                  192.168.1.2              0            50 3 2 50 51 52 1811 i
*> 182.0.0.0/8      1.1.1.1                  0             0 1 2 1822 i
*                   192.168.1.2              0             0 3 2 50 51 1822 i
*  183.0.0.0/8      1.1.1.1                  0             0 1 2 50 1833 i
*>                  192.168.1.2              0             0 3 2 50 1833 i
*  184.0.0.0/8      1.1.1.1                  0             0 1 2 50 51 1844 i
*>                  192.168.1.2              0             0 3 2 1844 i
*  185.0.0.0/8      1.1.1.1                  0             0 1 2 50 51 52 1855 i
*>                  192.168.1.2              0             0 3 1855 i

! The next command lists the pre-route-map received Update
E1# show ip bgp neigh 192.168.1.2 received-routes ¦ include 181
*  181.0.0.0/8      192.168.1.2              0             0 3 2 50 51 52 1811 i
```

```
! The next command shows the post-route-map received Update
E1# show ip bgp neigh 192.168.1.2 routes ¦ incl 181
*> 181.0.0.0/8      192.168.1.2              0            50 3 2 50 51 52 1811 i
```

The configuration uses a single-line IP prefix list that matches exactly prefix 181.0.0.0/8, and a 2-clause route map. The first route-map clause, a permit clause, matches 181.0.0.0/8. The permit action allows the route through the filter. The **set weight 50** command then sets the weight.

The second **route-map** clause, also with a permit action, matches the rest of the prefixes in the Update because there is no **match** command. The permit action allows these routes through the filter. Without clause 20, this route map would have matched all other routes with the route map's implied deny clause at the end of every route map, filtering all other routes learned from 192.168.1.2 except 181.0.0.0/8.

The configuration also includes a **neighbor 192.168.1.2 route-map set-weight-50 in** command to enable the route map for incoming updates from Router I3-1. The example also shows that the neighbor must be cleared, in this case with a soft reset of the **clear ip bgp 192.168.1.2 soft** command because the route map logic takes effect.

Examining the results of this change, note that E1 now thinks the better route is through I3-1 (192.168.1.2). The output lists the new weight of 50, with the route through I1-1 (1.1.1.1) using the default weight of 0. With weight, bigger is better.

Finally, the last two commands in the example show the pre-route-map received update (with the **received-routes** option) and the post-route-map results of the received update (with the **routes** option). The received Update does not include Weight because it is Cisco-proprietary, so E1 initially assigned the Weight to its default value (0). After applying the route map, E1 now lists a Weight of 50.

Setting Weight Using the **neighbor weight** Command

Alternatively, the weight can be set for all routes learned from a neighbor using the **neighbor weight** command. Example 15-4 shows this configuration added to E1, setting the Weight for all routes learned from I1-1 (1.1.1.1) to 60. As a result, E1's route for 181.0.0.0/8 switches back to using the route through 1.1.1.1 (I1-1).

Example 15-4 *Setting the Weight to 60 for All Routes Learned from I1-1*

```
E1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
E1(config)# router bgp 11
E1(config-router)# neighbor 1.1.1.1 weight 60
E1(config-router)#^Z
E1# clear ip bgp 1.1.1.1 soft
```

```
E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 54, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 181.0.0.0/8      1.1.1.1                  0             60 1 1811 i
*                   192.168.1.2              0             50 3 2 50 51 52 1811 i
*> 182.0.0.0/8      1.1.1.1                  0             60 1 2 1822 i
*                   192.168.1.2              0             0 3 2 50 51 1822 i
*> 183.0.0.0/8      1.1.1.1                  0             60 1 2 50 1833 i
*                   192.168.1.2              0             0 3 2 50 1833 i
*> 184.0.0.0/8      1.1.1.1                  0             60 1 2 50 51 1844 i
*                   192.168.1.2              0             0 3 2 1844 i
*> 185.0.0.0/8      1.1.1.1                  0             60 1 2 50 51 52 1855 i
*                   192.168.1.2              0             0 3 1855 i
```

The **neighbor weight** command does not use an in or out direction because Weight can only be set on input. The configuration results in all routes learned from 1.1.1.1 (I1-1) having a Weight of 60, as noted in the Weight column of the **show ip bgp** output.

Setting the Local Preference

The BGP Local Preference (Local_Pref) PA gives the routers inside a single AS a value that they can set per-route, advertise to all iBGP routers inside the AS, so that all routers in the AS agree about which router is the best exit point for packets destined for that prefix. By design, the Local_Pref can be set by routers as they receive eBGP routes by using an inbound route map. The routers then advertise the Local_Pref in iBGP updates. As a result, all the routers in the same AS can then make the same choice of which route is best, agreeing as to which router to use to exit the AS for each prefix.

As with the discussion of Weight, this section begins with a description of a sample scenario. Following that, a sample Local_Pref configuration is shown, using a route-map to set Local_Pref for routes advertised into an Enterprise. Table 15-5 summarizes some of the key features of Local_Pref as demonstrated in the upcoming pages.

Table 15-5 *Key Features of Local_Pref*

Feature	Description
PA?	Yes
Purpose	Identifies the best exit point from the AS to reach a given prefix
Scope	Throughout the AS in which it was set; not advertised to eBGP peers
Range	0 through 4,294,967,295 ($2^{32} - 1$)
Which is best?	Higher values are better

Default	100
Changing the default	Using the **bgp default local-preference <0-4294967295>** BGP sub-command
Configuration	Via **neighbor route-map** command; **in** option is required for updates from an **eBGP** peer

Note: For those of you memorizing using the N WLLA OMNI mnemonic, Local_Pref is the first L in WLLA.

Sample Internetwork Used in the Local_Pref and AS_Path Length Examples

Figure 15-5 shows a sample Internetwork used to demonstrate setting both Local_Pref, and later, AS_Path Length. The figure shows a single Enterprise with two Internet-connected routers. A full iBGP mesh exists with these two routers plus two routers internal to the Enterprise. Two eBGP neighborships exist, one with ISP1, and one with ISP3. (Note in particular that unlike Figure 15-3, E1 does not have a neighborship with router I3-1 in this case.) The following design requirements have already been met by the initial configuration in all routers shown in the figure:

- E1 and I1-1 uses loopback IP addresses (11.11.11.11 and 1.1.1.1) for their neighborship.
- E2 and I3-1 use interface IP addresses for their neighborship.
- None of the routers have attempted to change any settings that can impact the choice of best path, and Weight settings in the previous examples have been removed.

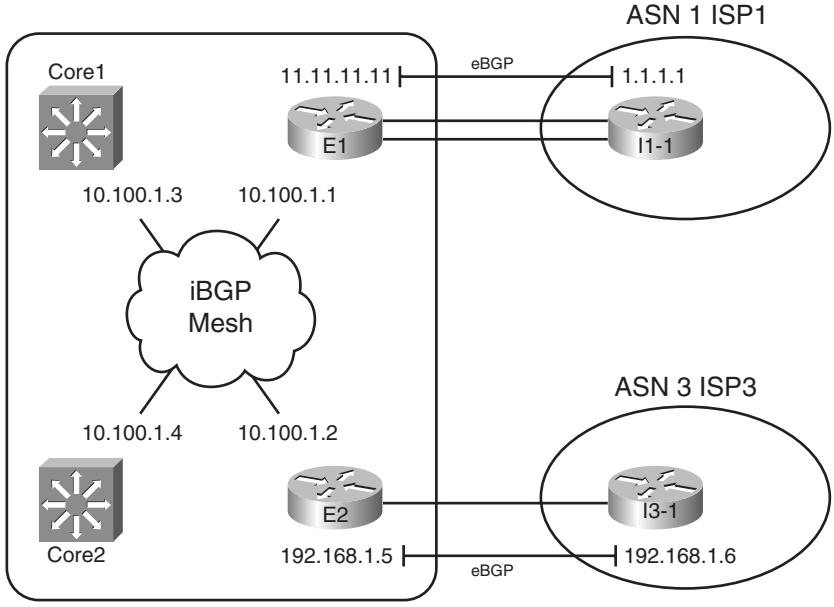


Figure 15-5 *Sample Internetwork for BGP Local_Pref and AS_Path Length Examples*

As with the Weight example, both ISPs advertise the same five prefixes, with different AS_Paths so that the routers have some prefixes to manipulate. Figure 15-6 shows five such prefixes that both ISPs advertise to E1 and E2. Note that this example network uses the same five prefixes, prefix lengths, and As_Path values as the previous Weight examples in this chapter.

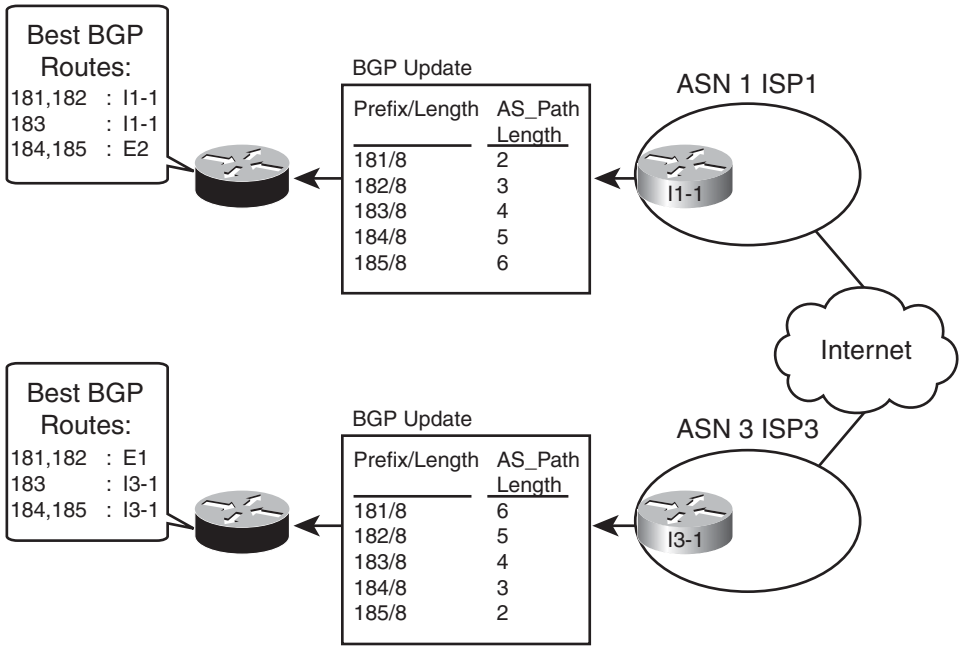


Figure 15-6 *Prefixes and AS_Path Lengths Used in Upcoming Examples*

Before showing the example of how to set the Local_Pref and how it impacts the routes, it is helpful to look at the best BGP routes on the Enterprise routers before any PAs have been changed. Example 15-6 shows the relevant BGP table entries on E1, E2, and Core1 with no attempt to influence E1's choice of best path. The best path for four of the five prefixes will be obvious, but the output listed in the commands requires some review. Prefixes 181.0.0.0/8 and 182.0.0.0/8 have a shorter AS_Path through ISP1, so E1 and E2 will agree that E1's path, through ISP1, is best. Similarly, 184.0.0.0/8 and 185.0.0.08 have a shorter AS_Path through ISP3, so both E1 and E2 agree that E2's path is best for these prefixes. Again, 183.0.0.0/8 ties on AS_Path length.

Example 15-5 *BGP Tables on E1, E2, and Core1, with No Changes to Settings That Affect Best Path*

```
! First, on router E1
E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 15, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 181.0.0.0/8      1.1.1.1                  0             0 1 1811 i
*> 182.0.0.0/8      1.1.1.1                  0             0 1 2 1822 i
* i183.0.0.0/8      10.100.1.2               0    100      0 3 2 50 1833 i
*>                  1.1.1.1                  0             0 1 2 50 1833 i
*>i184.0.0.0/8      10.100.1.2               0    100      0 3 2 1844 i
*                   1.1.1.1                  0             0 1 2 50 51 1844 i
*>i185.0.0.0/8      10.100.1.2               0    100      0 3 1855 i
*                   1.1.1.1                  0             0 1 2 50 51 52 1855 i
```

```
! Next, on router E2
E2# show ip bgp 176.0.0.0/4 longer-prefixes
! legend omitted for brevity

   Network          Next Hop            Metric LocPrf Weight Path
*>i181.0.0.0/8      10.100.1.1               0    100      0 1 1811 i
*                   192.168.1.6              0             0 3 2 50 51 52 1811 i
*>i182.0.0.0/8      10.100.1.1               0    100      0 1 2 1822 i
*                   192.168.1.6              0             0 3 2 50 51 1822 i
* i183.0.0.0/8      10.100.1.1               0    100      0 1 2 50 1833 i
*>                  192.168.1.6              0             0 3 2 50 1833 i
*> 184.0.0.0/8      192.168.1.6              0             0 3 2 1844 i
*> 185.0.0.0/8      192.168.1.6              0             0 3 1855 i
```

```
! Next, on router Core1
Core1# show ip bgp 176.0.0.0/4 longer-prefixes
! legend omitted for brevity

   Network          Next Hop            Metric LocPrf Weight Path
*>i181.0.0.0/8      10.100.1.1               0    100      0 1 1811 i
*>i182.0.0.0/8      10.100.1.1               0    100      0 1 2 1822 i
*>i183.0.0.0/8      10.100.1.1               0    100      0 1 2 50 1833 i
* i                 10.100.1.2               0    100      0 3 2 50 1833 i
*>i184.0.0.0/8      10.100.1.2               0    100      0 3 2 1844 i
*>i185.0.0.0/8      10.100.1.2               0    100      0 3 1855 i
```

First, pay close attention to the LocPrf column of output in the example. This column lists the Local_Pref settings of each route. Some list a (default) value of 100, and some list nothing. As it turns out, because Updates received from eBGP peers do not include the Local_Pref PA, IOS lists a null value for Local_Pref for eBGP-learned routes by default. However, Updates from iBGP peers do include the Local_Pref. Because this network does not have any configuration that attempts to set Local_Pref yet, the routers advertise their default Local_Pref value of 100 over the iBGP connections.

Also note that when comparing the output on both E1 and E2, the output lists a single eBGP route, but not the alternative iBGP route through the other Internet-connected router in the Enterprise. For example, E2 lists a single route for 184.0.0.0/8 and 185.0.0.0/8, through I3-1 (192.168.1.6). The reason that E2 does not list an alternative route through E1 is that E1's best route for these prefixes, as seen near the top of the example, is E1's iBGP-learned route through E2 (10.100.1.2). BGP does not allow a router to advertise iBGP-learned routes to iBGP peers, so E1 will not advertise routes for 184.0.0.0/8 or 185.0.0.0/8 to router E2.

Finally, For prefix 183.0.0.0/8, both E1 and E2 tie on the As_Path length. In this case, all best path choices tie until Step 7, which prefers eBGP routes over iBGP routes. E1 prefers its eBGP route for 183.0.0.0/8 through ISP1's Router I1-1, and E2 prefers its eBGP route through ISP3's Router I3-1.

Setting the BGP Local_Pref Using a Route Map

To set the Local_Pref, a router can use the **neighbor** *neighbor-ip* **route-map in** BGP subcommand. Typically, a router uses this command with the inbound direction for routes received from eBGP peers. Then, with no additional configuration required, the router then advertises the Local_Pref to any iBGP peers.

To show the Local_Pref configuration and results, start with the sample network shown in the previous section. The configuration will now be changed to set the Local_Pref for two different prefixes for Updates received on E1 from I1-1, as shown in Figure 15-7. Note that the figure reinforces the idea that BGP does not include the Local_Pref PA in eBGP Updates but will in iBGP Updates.

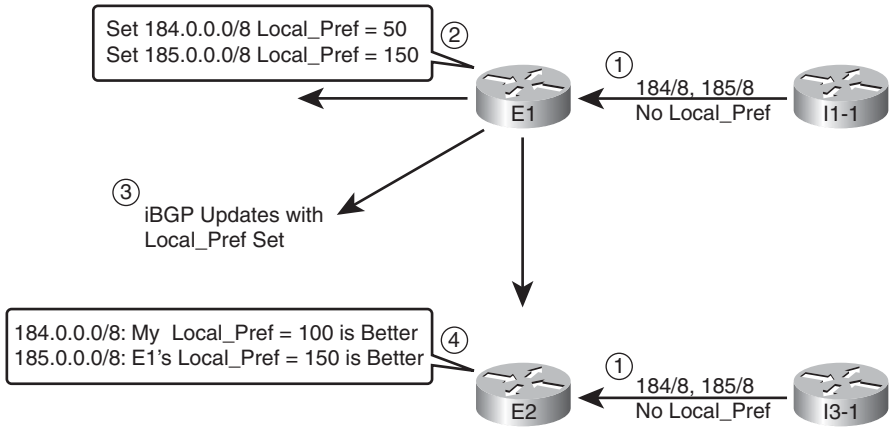


Figure 15-7 *Example Local_Pref Settings for the Upcoming Example*

Key Topic

The figure shows a series of steps, as follows:

Step 1. I1-1 and I3-1 advertise the prefixes into the Enterprise but with no Local_Pref set because the connections are eBGP peers.

Step 2. E1 sets the Local_Pref for routes learned from I1-1: 184.0.0.0/8 (50) and 185.0.0.0/8 (150).

Step 3. E1 includes the Local_Pref settings in its iBGP Updates to Core1, Core2, and E2.

Step 4. E2 realizes that E1's route for 185.0.0.0/8, Local_Pref 150, is better than E2's eBGP route for this prefix, which E2 assigned default Local_Pref 100. Conversely, E1's advertised route for 184.0.0.0/8, Local_Pref 50, is worse than E2's eBGP route for that same prefix, with assigned default Local_Pref 100.

Example 15-6 shows the configuration on router E1 to assign the Local_Pref values shown in Figure 15-7. The example also shows the results on E1 and E2. Note that the configuration differs only slightly as compared with the configuration for administrative Weight as shown in Example 15-3, the only substantive difference being the **set local-preference** route map command rather than the **set weight** command.

Example 15-6 *Configuring Local_Pref on Router E1 (Step 2 per Figure 15-7)*

```
E1# show running-config
! only pertinent portions shown
ip prefix-list match-184 seq 5 permit 184.0.0.0/8
!
ip prefix-list match-185 seq 5 permit 185.0.0.0/8
!
route-map set-LP-150 permit 10
 match ip address prefix-list match-185
 set local-preference 150
!
route-map set-LP-150 permit 15
 match ip address prefix-list match-184
 set local-preference 50
!
route-map set-LP-150 permit 20
!
router bgp 11
 neighbor 1.1.1.1 route-map set-LP-150 in
! The clearing of BGP neighbor I1-1 is done next, but not shown.
! Next, E1's Updated BGP Table

E1# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 29, local router ID is 128.107.9.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 181.0.0.0/8      1.1.1.1                  0             0 1 1811 i
*> 182.0.0.0/8      1.1.1.1                  0             0 1 2 1822 i
```

```
* i183.0.0.0/8       10.100.1.2               0    100      0 3 2 50 1833 i
*>                   1.1.1.1                  0             0 1 2 50 1833 i
*>i184.0.0.0/8       10.100.1.2               0    100      0 3 2 1844 i
*                    1.1.1.1                  0     50      0 1 2 50 51 1844 i
*> 185.0.0.0/8       1.1.1.1                  0    150      0 1 2 50 51 52 1855 i

E1# show ip bgp 185.0.0.0/8
BGP routing table entry for 185.0.0.0/8, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
        1
  1 2 50 51 52 1855, (received & used)
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 150, valid, external, best
```

```
! The next output occurs on router E2
E2# show ip bgp 185.0.0.0/8 longer-prefixes
! heading lines omitted

   Network          Next Hop            Metric LocPrf Weight Path
*>i185.0.0.0/8      10.100.1.1               0    150      0 1 2 50 51 52 1855 i
*                  192.168.1.6              0             0 3 1855 i
```

Example 15-6's output shows E1's BGP table entries, now with updated Local_Pref values as compared with Example 15-5. E1 now uses its eBGP route, Next_Hop 1.1.1.1, for prefix 185.0.0.0/8 because of the higher Local_Pref.

The end of the example shows E2 with two possible routes for 185.0.0.0/8. The following list outlines E2's BGP best path logic in this case:

Step 0. The two routes both have reachable Next_Hop IP addresses.

Step 1. Both have Weight 0 (tie).

Step 2. The iBGP route through 10.100.1.1 (E1) has a bigger (better) Local_Pref (150 versus 100) that the route through 192.168.1.6 (I3-1), so it is the better route.

Also, note that both the **show ip bgp longer-prefixes** command's briefer output, and the **show ip bgp 185.0.0.0/8** commands more verbose output, both identify the Local_Pref value. However, the longer command output does not list the Weight value.

IP Routes Based on BGP Best Paths

Some of the complexity related to BGP occurs around the BGP functions created by BGP PAs, including their use by the best path algorithm. When the BGP best path algorithm has gotten through this complexity and chosen a best route for a prefix, the router then tries to add that route to the IP routing table. However, rather than add the BGP route to

the IP routing table directly, BGP actually gives that best BGP route to another process for consideration: The IOS *Routing Table Manager (RTM)*.

The IOS RTM chooses the best route among many competing sources. For example, routes may be learned by an IGP, BGP, or even as connected or static routes. IOS collects the best such route for each prefix and feeds those into the RTM function. The RTM then chooses the best route. Figure 15-8 shows the general idea:

Among its tasks, RTM uses the concept of Administrative Distance (AD) to choose the best route among these different sources. Table 15-6 repeats the list of default AD values as shown earlier in Table 10-6 from Chapter 10, "Advanced IGP Redistribution," but with highlights for the two BGP-related default values:

For the most part, an Enterprise router should not see cases in which a prefix learned with BGP has also been learned as a connected or IGP-learned route. (Conversely, these issues

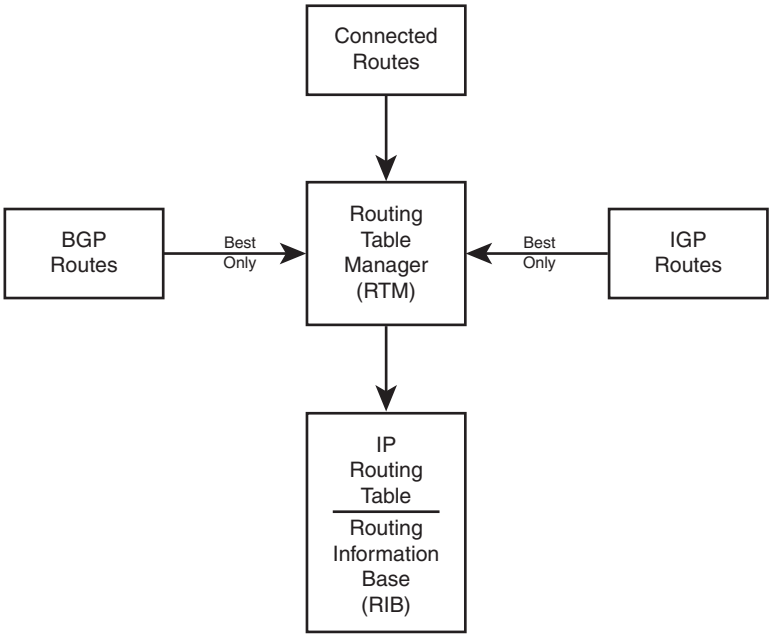


Figure 15-8 *Routing Table Manager Concept*

Table 15-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90

Table 15-6 *Default Administrative Distances*

Route Type	Administrative Distance
IGRP	100
OSPF	110
IS-IS	115
RIP	120
On-Demand Routing (ODR)	160
EIGRP (external)	170
iBGP	200
Unreachable	255

occur more often when implementing MPLS VPNs with BGP/IGP redistribution.) However, it can happen, and when it does, the **show ip bgp rib-failures** command can be helpful. This command lists routes for which BGP has chosen the route as best, but the RTM function has not placed the route into the Routing Information Base (RIB), which is simply another name for the IP routing table.

Example of a BGP RIB Failure

To show an example of a RIB failure, imagine that an Enterprise engineer needed to do some testing, so the engineer just picked an IP address range to use. The engineer tries to avoid problems by not using network 10.0.0.0, which is used throughout the Enterprise. So rather than choosing another private network, the engineer then chooses public range 185.0.0.0/8. After changing the lab configuration a few hundred times, a route for 185.0.0.0/8 leaks into the OSPF topology database.

Keep in mind that at the end of the previous example, E1 had chosen its eBGP route for 185.0.0.0/8 as its best route, and E2 had chosen its iBGP route as its best route for 185.0.0.0/8. Example 15-7 shows the results, based on RTM's comparisons of the AD values.

Example 15-7 *Example with the RTM and RIB Failures*

```
! First, E1's IP Routing table for 185.0.0.0/8
E1# show ip route 185.0.0.0 255.0.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is 1.1.1.1 to network 0.0.0.0

B    185.0.0.0/8 [20/0] via 1.1.1.1, 00:25:11
! Next, E2's IP Routing table
E2# show ip route 185.0.0.0 255.0.0.0 longer-prefixes
! Legend omitted for brevity

Gateway of last resort is 192.168.1.6 to network 0.0.0.0

O    185.0.0.0/8 [110/2] via 10.1.1.77, 00:15:44, FastEthernet0/0

E2# show ip bgp rib-failure
Network             Next Hop                        RIB-failure    RIB-NH Matches
185.0.0.0/8         10.100.1.1          Higher admin distance              n/a
```

The first command shows that E1, with an eBGP route, actually adds its route to the IP routing table. The route lists a code of B, meaning BGP. The output lists the eBGP default AD of 20, which is a better default AD than OSPF's 110. RTM added this BGP route to the IP routing table on E1 because of eBGP's better AD.

E2 currently lists its iBGP route through E1 as its current best BGP route for 185.0.0.0/8 because of the higher Local_Pref configured in Example 15-6. However, after giving this route to the RTM, RTM instead choose the lower-AD OSPF route (AD 110) rather than the higher-AD iBGP route (AD 200).

Finally, the **show ip bgp rib-failure** command lists one line for each best BGP route that the RTM does not place into the IP routing table. In this case, this command on Router E2 lists the route for 185.0.0.0/8, with the reason listed.

BGP and the maximum-paths Command

Like the IGP protocols, BGP supports the **maximum-paths** *number-of-paths* subcommand, but BGP uses significantly different logic than the IGPs. Unlike the IGP routing protocols, BGP truly needs to pick one route, and only one route, as the best path for a given prefix/length. In effect, the BGP best path algorithm already breaks the ties for "best" route for each prefix, so from BGP's perspective, one route for each prefix is always best.

BGP does allow multiple BGP routes for a prefix to be considered to tie, at least for the purpose of adding multiple routes to the IP routing table. The conditions are as follows:

> If the BGP best path algorithm does not choose a best path by Step 8 (per the numbering in this book), the routes which still tie for being best path through Step 8 will be allowed into the IP routing table, up to the number defined by the BGP **maximum-paths** *number-of-paths* router subcommand.

The section "Overview of the BGP Best Path Algorithm" earlier in this chapter lists the best path steps, including the tiebreaker steps that allow routes to be considered by the **maximum-paths** command.

Increasing the Length of the AS_Path Using AS_Path Prepend

Step 4 of the BGP best path algorithm examines the length of the AS_Path PA. The length of the AS_Path may appear to be obvious: Just add the number of ASNs listed in the AS_Path. However, some BGP features outside the scope of this book actually impact the AS_Path length calculation as well. However, for the purposes of this book, AS_Path length is simply the number of ASNs listed in the AS_Path.

The AS_Path prepend tool gives engineers a means to increase the length of an AS_Path by adding ASNs to the AS_Path, while not impacting the loop prevention role of the AS_Path PA. By increasing the length of an AS_Path, a route is less likely to become the best route. By adding ASNs that already exist inside a particular route's AS_Path, the feature does not inadvertently prevent a route from being ignored due to AS_Path loop prevention.

For example, using the design shown most recently in Figures 15-5, 15-6, and 15-7, imagine that the Enterprise considers ISP1 to be the better ISP, but they do not want to send all traffic through ISP1. So, the Enterprise network engineers could make the following type of implementation choice:

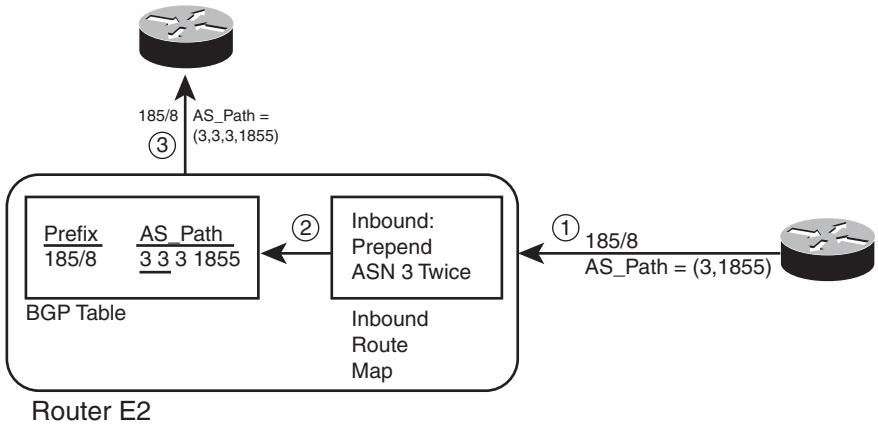
> Make the AS_Paths received from ISP3 be 2 ASNs longer.

By making such a choice, when an AS_Path through ISP1 is better, or when it's a tie on AS_Path between ISP1 and ISP3, or when the AS_Path through ISP1 is even slightly longer than through ISP3, the routers can still choose their routes through ISP1. Only when the AS_Path (before prepending) is at least 2 ASNs shorter through ISP3 can the ISP3 path be chosen.

Note: For those of you memorizing using the N WLLA OMNI mnemonic, AS_Path Length is the A in WLLA.

Figure 15-9 shows the mechanics of how an Enterprise route would prepend the AS_Path for routes received by Router E2 from ISP3, namely Router I3-1. Looking specifically at the route for 185.0.0.0/8, in this case, I3-1 has not changed the AS_Path and advertised the route with AS_Path (3, 1855). At Step 2, Router E2 prepends ASN 3–twice–making the AS_Path length 4. At Step 3, E2 advertises the route to its iBGP peers–peers that may now prefer to the other route for this prefix through Router E1.

The configuration itself requires only a little additional work compared to the other examples. As shown in Figure 15-9, Router E1 could use an inbound route map, using the **set as-path prepend 3 3** command to add the two ASNs. (The router sending the Update, ISP3's Router I3-1 in this case, could instead use an outbound route map.) Example 15-8 shows the configuration on E2 to add the ASNs at ingress into E2. (Note that all configuration for changing the Weight and Local_Pref, and the extra OSPF route for 185.0.0.0/8 shown in Example 15-6, have been removed before gathering the output in this example.)



Key Topic

Figure 15-9 *Prepending Two ASNs to an AS_Path*

Example 15-8 *Prepending Additional ASNs to the AS_Path*

```
! First, E2's new configuration
route-map add-two-asns permit 10
 set as-path prepend 3 3
router bgp 11
 neighbor 192.168.1.6 route-map add-two-asns in
!
! Next, note the AS_Path values all start with 3, 3, 3
E2# show ip bgp 176.0.0.0/4 longer-prefixes
BGP table version is 41, local router ID is 10.100.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i181.0.0.0/8      10.100.1.1               0    100      0 1 1811 i
*                   192.168.1.6              0             0 3 3 3 2 50 51 52 1811 i
*>i182.0.0.0/8      10.100.1.1               0    100        0 1 2 1822 i
*                   192.168.1.6              0             0 3 3 3 2 50 51 1822 i
*>i183.0.0.0/8      10.100.1.1               0    100      0 1 2 50 1833 i
*                   192.168.1.6              0             0 3 3 3 2 50 1833 i
* i184.0.0.0/8      10.100.1.1               0    100      0 1 2 50 51 1844 i
*>                  192.168.1.6              0             0 3 3 3 2 1844 i
*> 185.0.0.0/8      192.168.1.6              0             0 3 3 3 1855 i
```

Note: When using AS_Path prepending, do not prepend just any ASN. BGP still uses the AS_Path for loop avoidance, so using an ASN already in the AS_Path, like the ASN of the most recently added ASN (for example, ASN 3 in this case), or the local ASN (for example, ASN 11 in this case), makes the most sense.
Although presented here as a tool for influencing outbound routes, As_Path prepending can also be used to influence the inbound routes as well.

Influencing an Enterprise's Inbound Routes with MED

An Enterprise has reasonably good control over its outbound IP routes. The engineers can configure BGP to set and react to Weight, Local_Pref, and AS_Path length, manipulating each to choose a different outgoing link or different router through which to forward packets to the Internet.

An Enterprise has much less control over inbound routes: routes for packets coming back toward the Enterprise. First, these inbound routes exist on routers that the Enterprise does not own. Even if an ISP or set of ISPs can be convinced by engineers at the Enterprise to make their routes toward an Enterprise take a particular path, technical issues may prevent the design from being implemented. In particular, if the Enterprise's public IP address range is summarized, the companies that use addresses in that range may have competing goals, so no policy can be applied to influence the best route.

However, several tools exist that allow some control over the last ASN hop between an ISP and their Enterprise customer. This book examines one such tool, called Multi-Exit Discriminator (MED), originally worked for a dual-homed design–that is, with a single ISP but with multiple links to that ISP. MED was later expanded to support dual-multihomed designs (2+ ASNs, 2+ links), relying on the concept that ISPs would work together. This section examines the dual-homed case, with a single ISP.

MED Concepts

The name Multi Exit Discriminator actually describes its function to a great degree. With a dual-homed design, at least two links exist between the Enterprise and the ISP. The Enterprise can announce to the ISP a value (MED) that tells the ISP which path into the Enterprise is best. As a result, the ISP can discriminate between the multiple exit points from that ISP to the Enterprise.

Because MED lets the Enterprise ASN tell just the neighboring ASN which link into the Enterprise to use, engineers typically use MED when advertising an Enterprise's public IP address space. Those inbound routes into the Enterprise from the ISP typically consist of either one, or a few, public IP address ranges.

For example, consider a new network design as shown in Figure 15-10. In this case, the Enterprise uses the same 128.107.0.0/19 public address range used in Chapters 13 and 14. The Enterprise connects only to ASN 1 with a total of four physical links and three BGP neighbors.

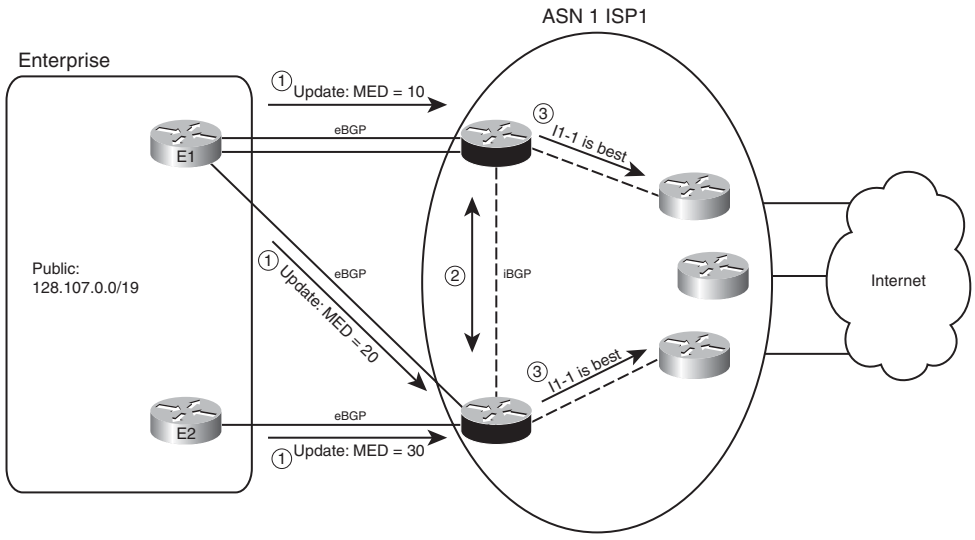


Figure 15-10 *Example of Using MED*

MED uses smallest-is-best logic. As a result, the figure shows a design in which the Enterprise engineer prefers the top BGP neighborship as the best path to use for inbound routes (MED 10), the middle link next (Med 20), and the bottom connection last (MED 30). Following the steps in the figure:

Step 1. E1 and E2 advertise 128.107.0.0/19, setting MED with an outbound route map, to various settings: MED 10 sent by E1 to I1-1, MED 20 sent by E1 to I1-4, and MED 30 sent by E2 to I1-4.

Step 2. I1-1 and I1-4 have an iBGP connection, so they learn each other's routes and agree as to which route wins based on MED.

Step 3. I1-1 and I4-1 also tell the other routers inside ISP1, causing all inbound traffic to funnel toward router I1-1.

Note that Routers I1-1 and I4-1 in this example could have chosen a better route based on all the earlier best path steps. However, a brief analysis of the steps tells us that unless someone makes an effort to override the effects of MED, these routers' best path algorithms will use MED. Assuming the Enterprise and ISP agree to rely on MED, the earlier Best Path steps should not matter. Here's why:

Step 1. **Weight:** Needs to be set locally, so if relying on MED, the ISP simply chooses to not set the Weight for received Updates from the Enterprise.

Step 2. **Local_Pref:** Again, this takes overt effort to match and set the Local_Pref, so if relying on MED, the ISP simply chooses to not set the Local_Pref.

Step 3. **Locally injected?** All these public routes from the Enterprise will be learned with eBGP and not locally injected.

Step 4. **AS_Path length:** All such routes on the ISP routers should list one ASN–the Enterprise's ASN–so all should tie on this point.

Step 5. **Origin:** Whatever the Origin is (I or ?), it should tie.

Step 6. **MED:** None of the other steps determined the best route, so now MED takes effect.

Table 15-7 summarizes the key points about MED.

Table 15-7 *Key Features of MED*

Key Topic

Feature	Description
Is it a PA?	Yes.
Purpose	Allows an AS to tell a neighboring AS the best way to forward packets into the first AS.
Scope	Advertised by one AS into another, propagated inside the AS, but not sent to any other autonomous systems.
Range	0 through 4,294,967,295 ($2^{32} - 1$).
Which is best?	Smaller is better.
Default	0
Configuration	Via **neighbor route-map out** command, using the **set metric** command inside the route map.

Note: For those of you memorizing using the N WLLA OMNI mnemonic, MED is the M in OMNI.

MED Configuration

MED configuration usually occurs on the routers in the AS that wants to control inbound routes from the neighboring AS. As such, in the example design shown in Figure 15-10, Routers E1 and E2 would configure MED. Example 15-9 shows E1's configuration.

Example 15-9 *MED Configuration on Router E1*

```
route-map set-med-to-I1-1 permit 10
 match ip address prefix only-public
 set metric 10
!
route-map set-med-to-I1-4 permit 10
 match ip address prefix only-public
 set metric 20
!
ip prefix-list only-public permit 128.107.0.0/19
!
```

```
router bgp 11
 neighbor 1.1.1.1 route-map set-med-I1-1 out
 neighbor 192.168.1.2 route-map set-med-I1-4 out
```

Both the configuration and the **show ip bgp** command output refers to MED as metric. Note that the route map in Example 15-8 uses the **set metric** command, rather than **set med** (which does not exist). And as shown in I1-1's output for the **show ip bgp** command in Example 15-10, the output lists MED under the heading metric. In particular, note that even the **show ip route** command lists the MED value in brackets as the metric for the BGP route.

Example 15-10 *BGP Table and IP Routing Table on Router I1-1*

```
I1-1# show ip bgp 128.107.0.0/19
BGP routing table entry for 128.107.0.0/19, version 13
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  11, (aggregated by 11 128.107.9.1), (received & used)
    11.11.11.11 from 11.11.11.11 (128.107.9.1)
      Origin IGP, metric 10, localpref 100, valid, external, atomic-aggregate, best

I1-1# sh ip bgp 128.107.0.0/19 longer-prefixes
BGP table version is 13, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 128.107.0.0/19   11.11.11.11             10             0 11 i

I1-1# show ip route 128.107.0.0 255.255.224.0 longer-prefixes
! Legend omitted for brevity

Gateway of last resort is not set

     128.107.0.0/19 is subnetted, 1 subnets
B       128.107.0.0 [20/10] via 11.11.11.11, 00:02:18
```

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F.

Design Review Table

Table 15-8 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 15-8 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Influence outbound route from an Enterprise toward prefixes in the Internet (3).	
Influence outbound route from an Enterprise toward prefixes in the Internet so that multiple Internet-connected Enterprise routers make the same choice based on the same information (2).	
Influence inbound routes into the Enterprise from the neighboring AS (2).	

Implementation Plan Peer Review Table

Table 15-9 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 15-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A plan shows two Enterprise routers, R1 and R2, connected to two different ISPs, with iBGP between R1 and R2. The plan shows R1 setting Weight for routes learned from an ISP. Will R2 react to those settings? Why or why not?	
A plan shows two Enterprise routers, R1 and R2, connected to two different ISPs, with iBGP between R1 and R2. The plan shows R1 setting Local_Pref for routes learned from an ISP. Will R2 react to those settings? Why or why not?	
The Plan calls for the use of BGP Weight, but the incomplete plan lists no configuration yet. What configuration alternatives exist? (2)	
The Plan calls for the use of BGP Local Preference, but the incomplete plan lists no configuration yet. What configuration alternatives exist?	
A plan shows two Enterprise routers, R1 and R2, connected to different ISPs. The plan calls for using MED to influence inbound routes. Which configuration options exist?	
A plan shows the use of BGP Weight, Local Preference, AS_Path prepending, and MED to influence the best path algorithm. Which of these can be set and advertised to eBGP peers?	

Create an Implementation Plan Table

This chapter does not focus on implementation or verification, but it did review one concept about static routes, as listed in Table 15-10.

Table 15-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure a route map that sets Weight.	
Enable a route map to set BGP weight.	
Enable a router to set BGP weight for all routes received from a neighbor.	
Configure a route map that sets BGP Local Preference.	
Enable a route map to set BGP Local Preference.	
Configure a route map that prepends ASNs to an AS_Path.	
Enable a route map to perform AS_Path prepending.	
Configure a route map that sets MED.	
Enable a route map to set MED.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own BGP verification plan, list in Table 15-11 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 15-11 *Verification Plan Memory Drill*

Information Needed	Commands
Display the BGP table, including the chosen best path for each prefix. (State how to identify the best paths.)	
List 1 line per BGP route but for the prefixes within a range.	
Identify a BGP table entry's BGP Weight. (Specify where to find output.)	
Identify a BGP table entry's BGP Local Preference. (Specify where to find output.)	
Identify a BGP table entry's AS_Path length. (Specify where to find output.)	
Identify a BGP table entry's MED. (Specify where to find output.) (4 methods)	
Display routes received from a neighbor before being processed by an inbound filter.	
The same as the previous item but after applying the inbound filter.	
Display BGP routes sent to a neighbor but after applying the inbound filter.	
Display BGP best paths that were not added to the IP routing table.	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 15-12 lists a reference of these key topics and the page numbers on which each is found.

Table 15-12 *Key Topics for Chapter 15*

Key Topic Element	Description	Page Number
Table 15-2	BGP path attributes that affect the BGP Best Path algorithm	495
Figure 15-1	Figure showing the location of BGP PA information in the output of **show ip bgp**	496
Table 15-3	Summary of BGP Best Path algorithm	496
List	Four items commonly set for the purpose of influencing the BGP best path decision	498
Table 15-4	Reference Table for BGP Weight	500
Table 15-5	Reference Table for BGP Local_Pref	507
Figure 15-7	Example of how Local_Pref influences best path choice	511
Figure 15-9	Example of how AS_Path Prepending works	518
Table 15-7	Reference Table for BGP MED	521

Complete the Tables and Lists from Memory

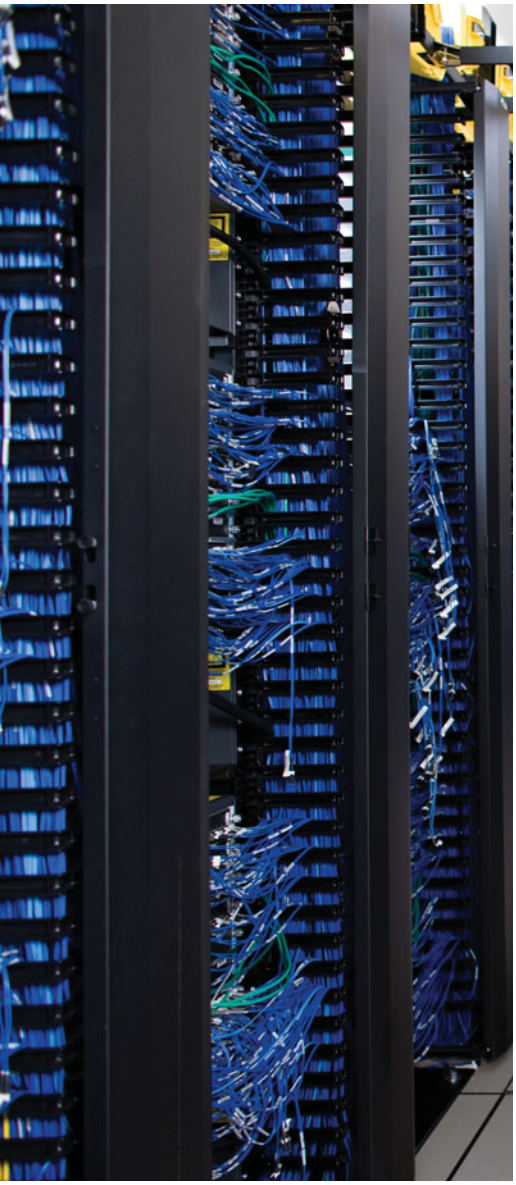
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

BGP Weight, Local Preference, AS_Path Prepending, Multi Exit Discriminator, Best Path algorithm, Routing Table Manager, RIB Failure, path attribute

This page intentionally left blank



This chapter covers the following subjects:

Global Unicast Addressing, Routing, and Subnetting: This section introduces the concepts behind unicast IPv6 addresses, IPv6 routing, and how to subnet using IPv6, all in comparison to IPv4.

IPv6 Global Unicast Address Assignment: This section examines how global unicast addresses can be assigned to hosts and other devices.

Survey of IPv6 Addressing: This section examines all types of IPv6 addresses.

Configuring IPv6 Addresses on Cisco Routers: This section shows how to configure and verify static IPv6 addresses on Cisco routers.

CHAPTER 16

IP Version 6 Addressing

IP Version 6 (IPv6), the replacement protocol for IPv4, is well known for a couple of reasons. IPv6 provides the ultimate solution for the problem of running out of IPv4 addresses in the global Internet by using a 128-bit address–approximately 10^{38} total addresses, versus the mere (approximate) $4* 10^{9}$ total addresses in IPv4. However, many articles over the years have discussed when, if ever, would a mass migration to IPv6 take place. IPv6 has been the ultimate long-term solution for more than 10 years, in part because the interim IPv4 solutions, including NAT/PAT, have thankfully delayed the day in which we truly run out of public unicast IP addresses.

As more IPv6 deployments continue to emerge, Cisco has steadily added more and more IPv6 content to its certification exams. CCNA now includes basic IPv6 concepts and configuration, with this version of the CCNP ROUTE exam examining IPv6 to greater depth than its predecessor BSCI exam. In particular, ROUTE includes the basic concepts, configuration, and verification of most of the router-based tools needed to deploy IPv6 in the Enterprise. As with most of the other topics, ROUTE's coverage of IPv6 does not include all skills and knowledge to plan an Enterprise IPv6 deployment, but it does include enough to take an IPv6 design plan and make a good start on developing an IPv6 implementation and verification plan.

Part VI of this book includes three chapters. This chapter examines the concepts around IPv6 addressing, and how to configure and verify those addresses on Cisco routers. Chapter 17, "IPv6 Routing Protocols and Redistribution," looks at IPv6 routing protocols, as well as route redistribution and static routing. Both these chapters focus on pure IPv6, essentially ignoring issues related to migration from IPv4 and coexistence with IPv4. The final chapter in this part, Chapter 18, "IPv4 and IPv6 Coexistence," looks at the migration and coexistence issues.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 16-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 16-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Global Unicast Addressing, Routing, and Subnetting	1–2
IPv6 Global Unicast Address Assignment	3–4
Survey of IPv6 Addressing	5–6
Configuring IPv6 Addresses on Cisco Routers	7–8

1. Which of the following is the shortest valid abbreviation for FE80:0000:0000:0000:0010:0000:0000:0123?

   a. FE80::10::123
   b. FE8::1::123
   c. FE80:0:0:0:10::123
   d. FE80::10:0:0:123

2. An ISP has assigned prefix 3000:1234:5678::/48 to Company1. Which of the following terms would typically be used to describe this type of public IPv6 prefix?

   a. Subnet prefix
   b. ISP prefix
   c. Global routing prefix
   d. Registry prefix

3. Which of the following answers lists either a protocol or function that can be used by a host to dynamically learn its own IPv6 address? (Choose two.)

   a. Stateful DHCP
   b. Stateless DHCP
   c. Stateless autoconfiguration
   d. Neighbor Discovery Protocol

4. Which of the following is helpful to allow an IPv6 host to learn the IP address of a default gateway on its subnet?

   a. Stateful DHCP
   b. Stateless RS
   c. Stateless autoconfiguration
   d. Neighbor Discovery Protocol

5. Which of the following answers lists a multicast IPv6 address?

   a. 2000::1:1234:5678:9ABC
   b. FD80::1:1234:5678:9ABC

c. FE80::1:1234:5678:9ABC

d. FF80::1:1234:5678:9ABC

6. Router R1 has two LAN interfaces and three serial interfaces enabled for IPv6. All the interfaces use link local addresses automatically generated by the router. Which of the following could be the link local address of R1's interface S0/0?

a. FEA0::200:FF:FE11:0

b. FE80::200:FF:FE11:1111

c. FE80::0213:19FF:FE7B:0:1

d. FEB0::211:11FF:FE11:1111

7. Router R1 has the following configuration. Assuming R1's F0/0 interface has a MAC address of 0200.0011.1111, what IPv6 addresses will R1 list for interface F0/0 in the output of the **show ipv6 interface brief** command?

```
interface f0/0
 ipv6 address 2345:0:0:8::1/64
```

a. 2345:0:0:8::1

b. 2345:0:0:8:0:FF:FE11:1111

c. FE80::8:0:FF:FE11:1111

d. FE80:0:0:8::1

8. Router R1 lists the following output from a **show** command. Which of the following is true about R1?

```
R1# show ipv6 interface f0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE12:3456
  No Virtual link-local address(es):
  Global unicast address(es):
    2000::4:213:19FF:FE12:3456, subnet is 2000:0:0:4::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF:12:3456
```

a. R1's solicited node multicast address is FF02::1:FF:12:3456.

b. R1's 2000::4:213:19FF:FE12:3456 address is a global unicast with all 128 bits statically configured.

c. Address FF02::2 is R1's solicited node multicast.

d. R1's solicited node multicast, not listed in this output, would be FF02::213:19FF:FE12:3456.

Foundation Topics

The world has changed tremendously over the last 10–20 years as a result of the growth and maturation of the Internet and networking technologies in general. As recently as 1990, a majority of the general public did not know about nor use global networks to communicate, and when businesses needed to communicate, those communications mostly flowed over private networks. During the 1990s, the public Internet grew to the point where people in most parts of the world could connect to the Internet; many companies connected to the Internet for a variety of applications, with the predominate applications being email and web. During the first decade of the 21st century, the Internet has grown further to billions of addressable devices with the majority of people on the planet having some form of Internet access. With that pervasive access came a wide range of applications and uses, including voice, video, collaboration, and social networking, with a generation that has grown up with this easily accessed global network.

The eventual migration to IPv6 will likely be driven by the need for more and more IP addresses. Practically every mobile phone supports Internet traffic, requiring the use of an IP address. Most new cars have the capability to acquire and use an IP address, along with wireless communications, allowing the car dealer to contact the customer when the car's diagnostics detect a problem with the car. Some manufacturers have embraced the idea that all their appliances need to be IP-enabled.

Although the two biggest reasons why networks might migrate from IPv4 to IPv6 are the need for more addresses and mandates from government organizations, at least IPv6 includes some attractive features and migration tools. Some of those advantages are

- **Address assignment features:** IPv6 supports a couple of methods for dynamic address assignment, including DHCP and Stateless Autoconfiguration.
- **Built-in support for address renumbering:** IPv6 supports the ability to change the public IPv6 prefix used for all addresses in an Enterprise, using the capability to advertise the current prefix with a short timeout and the new prefix with a longer lease life.
- **Built-in support for mobility:** IPv6 supports mobility such that IPv6 hosts can move around the Internetwork and retain their IPv6 address without losing current application sessions.
- **Provider independent and dependent public address space:** ISPs can assign public IPv6 address ranges (dependent), or companies can register their own public address space (independent).
- **Aggregation:** IPv6's huge address space makes for much easier aggregation of blocks of addresses in the Internet, making routing in the Internet more efficient.
- **No need for NAT/PAT:** The huge public IPv6 address space removes the need for NAT/PAT, which avoids some NAT-induced application problems and makes for more efficient routing.

- **IPsec:** Unlike IPv4, IPv6 requires that every IPv6 implementation support IPsec. IPv6 does not require that each device use IPsec, but any device that implements IPv6 must also have the ability to implement IPsec.
- **Header improvements:** Although it might seem like a small issue, the IPv6 header actually improves several things compared to IPv4. In particular, routers do not need to recalculate a header checksum for every packet, reducing per-packet overhead. Additionally, the header includes a flow label that allows for easy identification of packets sent over the same single TCP or UDP connection.
- **No broadcasts:** IPv6 does not use Layer 3 broadcast addresses, instead relying on multicasts to reach multiple hosts with a single packet.
- **Transition tools:** As covered in Chapter 18, the IPv6 has many rich tools to help with the transition from IPv4 to IPv6.

This list includes many legitimate advantages of IPv6 over IPv4, but the core difference, and the main topic of this chapter, is IPv6 addressing. The first two section of this chapter examine one particular type of IPv6 addresses, global unicast addresses, which have many similarities to IPv4 addresses–particularly public IPv4 addresses. The third section broadens the discussion to include all types of IPv6 addresses, and protocols related to IPv6 address assignment, default router discovery, and neighbor discovery. The final section looks at the router configuration commands for IPv6 addressing.

Global Unicast Addressing, Routing, and Subnetting

The original Internet design called for all organizations to register and be assigned one or more public IP networks (Class A, B, or C). By registering to use a particular public network number, the company or organization using that network was assured by the numbering authorities that no other company or organization in the world would be using the same addresses. As a result, all hosts in the world would have globally unique IP addresses.

From the perspective of the Internet infrastructure, in particular the goal of keeping Internet routers' routing tables from getting too large, assigning an entire network to each organization helped to some degree. The Internet routers could ignore all subnets as defined inside an Enterprise, instead having a route for each classful network. For instance, if a company registered and was assigned Class B network 128.107.0.0/16, the Internet routers just needed one route for that whole network.

Over time, the Internet grew tremendously. It became clear by the early 1990s that something had to be done, or the growth of the Internet would grind to a halt when all the public IP networks were assigned, and no more existed. Additionally, the IP routing tables in Internet routers were becoming too large for the router technology of that day. So, the Internet community worked together to come up with both some short-term and long-term solutions to two problems: the shortage of public addresses and the size of the routing tables.

The short-term solutions included a much smarter public address assignment policy, in which public addresses were not assigned as only Class A, B, and C networks, but as smaller subdivisions (prefixes), reducing waste. Additionally, the growth of the Internet

routing tables was reduced by smarter assignment of the actual address ranges based on geography. For example, assigning the class C networks that begin with 198 to only a particular ISP in a particular part of the world allowed other ISPs to use one route for 198.0.0.0/8—in other words, all addresses that begin with 198—rather than a route for each of the 65,536 different Class C networks that begin with 198. Finally, NAT/PAT achieved amazing results by allowing a typical home or small office to consume only one public IPv4 address, greatly reducing the need for public IPv4 addresses.

IPv6 provides the long-term solution to both problems (address exhaustion and Internet routing table size). The sheer size of IPv6 addresses takes care of the address exhaustion issue. The address assignment policies already used with IPv4 have been refined and applied to IPv6, with good results for keeping the size of IPv6 routing tables smaller in Internet routers. This section provides a general discussion of both issues, in particular how *global unicast* addresses, along with good administrative choices for how to assign IPv6 address prefixes, aid in routing in the global Internet. This section concludes with a discussion of subnetting in IPv6.

Global Route Aggregation for Efficient Routing

By the time the Internet community started serious work to find a solution to the growth problems in the Internet, many people already agreed that a more thoughtful public address assignment policy for the public IPv4 address space could help keep Internet routing tables much smaller and more manageable. IPv6 public address assignment follows these same well-earned lessons.

Note: The descriptions of IPv6 global address assignment in this section provides a general idea about the process. The process may vary from one RIR to another, and one ISP to another, based on many other factors.

The address assignment strategy for IPv6 is elegant, but simple, and can be roughly summarized as follows:

- Public IPv6 addresses are grouped (numerically) by major geographic region.
- Inside each region, the address space is further subdivided by ISPs inside that region.
- Inside each ISP in a region, the address space is further subdivided for each customer.

The same organizations handle this address assignment for IPv6 as for IPv4. The Internet Corporation for Assigned Network Numbers (ICANN, www.icann.org) owns the process, with the Internet Assigned Numbers Authority (IANA) managing the process. IANA assigns one or more IPv6 address ranges to each Regional Internet Registries (RIR), of which there are five at the time of publication, roughly covering North America, Central/South America, Europe, Asia/Pacific, and Africa. These RIRs then subdivide their assigned address space into smaller portions, assigning prefixes to different ISPs and other smaller registries, with the ISPs then assigning even smaller ranges of addresses to their customers.

The IPv6 global address assignment plan results in more efficient routing, as shown in Figure 16-1. The figure shows a fictitious company (Company1) which has been assigned an IPv6 prefix by a fictitious ISP, NA-ISP1 (meaning for North American ISP number 1). The figure lists the American Registry for Internet Numbers (ARIN), which is the RIR for North America.

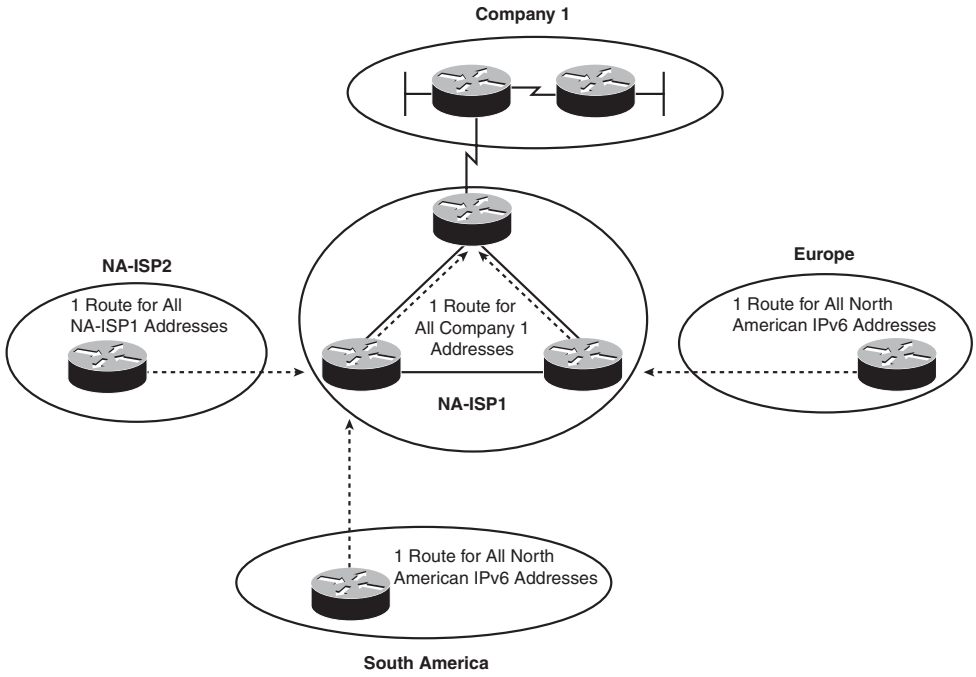


Figure 16-1 *Conceptual View of IPv6 Global Routes*

Key Topic

As shown in the figure, the routers installed by ISPs in other major geographies of the world can have a single route that matches all IPv6 addresses in North America. Although there might be hundreds of ISPs operating in north America, and hundreds of thousands of Enterprise customers of those ISPs, and tens of millions of individual customers of those ISPs, all the public IPv6 addresses can be from one (or a few) very large address blocks–requiring only one (or a few) routes on the Internet routers in other parts of the world. Similarly, routers inside other ISPs in North America (for example, NA-ISP2, meaning North American ISP number 2 in the figure), can have one route that matches all address ranges assigned to NA-ISP2. And the routers inside NA-ISP1 just need to have one route that matches the entire address range assigned to Company1, rather than needing to know about all the subnets inside Company1.

Besides keeping the routers' routing table much smaller, this process also results in fewer changes to Internet routing tables. For example, if NA-ISP1 signed a service contract with another Enterprise customer, NA-ISP1 could assign another prefix inside the range of addresses already assigned to NA-ISP1 by ARIN. The routers outside NA-ISP1's

network–the majority of the Internet–do not need to know any new routes, because their existing routes already match the address range assigned to the new customer. The NA-ISP2 routers (another ISP) already have a route that matches the entire address range assigned to NA-ISP1, so they do not need any more routes. Likewise, the routers in ISPs in Europe and South America already have a route that works as well.

Conventions for Representing IPv6 Addresses

IPv6 conventions use 32 hexadecimal numbers, organized into 8 quartets of 4 hex digits separated by a colon, to represent a 128-bit IPv6 address, for example:

2340:1111:AAAA:0001:1234:5678:9ABC

Each hex digit represents 4 bits, so if you want to examine the address in binary, the conversion is relatively easy if you memorize the values shown in Table 16-2.

Table 16-2 *Hexadecimal/Binary Conversion Chart*

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

Writing or typing 32 hexadecimal digits, although more convenient writing or typing 128 binary digits, can still be a pain. To make things a little easier, two conventions allow you to shorten what must be typed for an IPv6 address:

Key Topic

- Omit the leading 0s in any given quartet.
- Represent one or more consecutive quartets of all hex 0s with "::" but only for one such occurrence in a given address.

Note: For IPv6, a quartet is one set of 4 hex digits in an IPv6 address. There are 8 quartets in each IPv6 address.

For example, consider the following address. The bold digits represent digits in which the address could be abbreviated.

FE00:**0000:0000:0001:0000:0000:0000:00**56

This address has two different locations in which one or more quartets have 4 hex 0s, so two main options exist for abbreviating this address–using the :: abbreviation in one or the other location. The following two options show the two briefest valid abbreviations:

FE00::1:0:0:0:56

FE00:0:0:1::56

In particular, note that the "::" abbreviation, meaning "one or more quartets of all 0s," cannot be used twice because that would be ambiguous. So, the abbreviation FE00::1::56 would not be valid.

Conventions for Writing IPv6 Prefixes

IPv6 prefixes represent a range or block of consecutive IPv6 addresses. Just like routers use IPv4 subnets in IPv4 routing tables to represent ranges of consecutive addresses, routers use IPv6 prefixes to represent ranges of consecutive IPv6 addresses as well. The concepts mirror those of IPv4 addressing when using a classless view of the IPv4 address. Figure 16-2 reviews both the classful and classless view of IPv4 addresses, compared to the IPv6 view of addressing and prefixes.

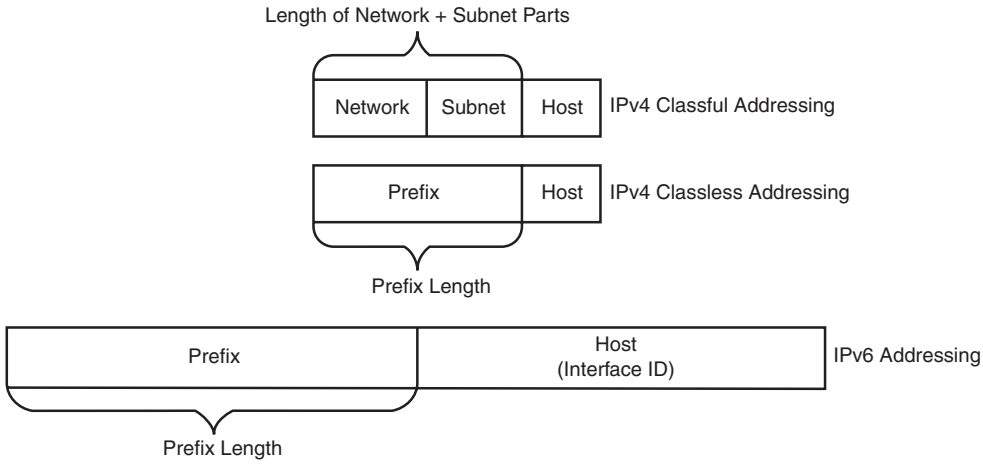


Figure 16-2 *IPv4 Classless and Classful Addressing, IPv6 Addressing*

First, for perspective, compare the classful and classless view of IPv4 addresses. Classful IPv4 addressing means that the class rules always identify part of the address as the network part. For example, the written value 128.107.3.0/24 (or 128.107.3.0 255.255.255.0) means 16 network bits (because the address is in a class B network), 8 host bits (because the mask has 8 binary 0s), leaving 8 subnet bits. The same value, interpreted with classless rules, means prefix 128.107.3.0, prefix length 24. Classless addressing and classful addressing just give slightly different meaning to the same numbers.

IPv6 uses a classless view of addressing, with no concept of classful addressing. Like IPv4, IPv6 prefixes list some prefix value, a slash, and then a numeric prefix length. Like IPv4 prefixes, the last part of the number, beyond the length of the prefix, will be represented by binary 0's. And finally, IPv6 prefix numbers can be abbreviated with the same rules as IPv4 addresses.

Note: IPv6 prefixes are often called IPv6 subnets as well. This book uses these terms interchangeably.

For example, consider the following IPv6 address that is assigned to a host on a LAN:

2000:1234:5678:9ABC:1234:5678:9ABC:1111/64

This value represents the full 128-bit IP address–there are no opportunities to even abbreviate this address. However, the /64 means that the prefix (subnet) in which this address resides is the subnet that includes all addresses that begin with the same first 64 bits as the address. Conceptually, it is the same logic as an IPv4 address, for example, address 128.107.3.1/24 is in the prefix (subnet) whose first 24 bits are the same values as address 128.107.3.1.

As with IPv4, when writing or typing a prefix, the bits past the end of the prefix length are all binary 0s. In the IPv6 address previously shown, the prefix in which the address resides would be

2000:1234:5678:9ABC:0000:0000:0000:0000/64

Which, when abbreviated, would be

2000:1234:5678:9ABC::/64

Next, one last fact about the rules for writing prefixes before seeing some examples and moving on. If the prefix length is not a multiple of 16, then the boundary between the prefix and the interface ID (host) part of the address is inside a quartet. In such cases, the prefix value should list all the values in the last octet in the prefix part of the value. For example, if the address just shown with a /64 prefix length instead had a /56 prefix length, the prefix would include all of the first 3 quartets (a total of 48 bits), plus the first 8 bits of the fourth quartet. The next 8 bits (last 2 hex digits) of the fourth octet should now be binary 0s, as part of the host portion of the address. So, by convention, the rest of the fourth octet should be written, after being set to binary 0s, as follows:

2000:1234:5678:9A**00**::/56

The following list summarizes some key points about how to write IPv6 prefixes.

Key Topic

- The prefix has the same value as the IP addresses in the group for the number of bits in the prefix length.
- Any bits after the prefix length number of bits are binary 0s.
- The prefix can be abbreviated with the same rules as IPv6 addresses.
- If the prefix length is not on a quartet boundary, write down the value for the entire quartet.

Examples can certainly help in this case. Table 16-3 shows several sample prefixes, their format, and a brief explanation.

Table 16-3 *Example IPv6 Prefixes and Their Meanings*

Prefix	Explanation	Incorrect Alternative
2000::/3	All addresses whose first 3 bits are equal to the first 3 bits of hex number 2000 (bits are 001).	2000/3 (omits ::) 2::/3 (omits the trailing 0s in the first quartet)
2340:1140::/26	All addresses whose first 26 bits match the listed hex number.	2340:114::/26 (omits trailing 0 in the second quartet)
2340:1111::/32	All addresses whose first 32 bits match the listed hex number.	2340:1111/32 (omits ::)

Note which options are not allowed. For example, 2::/3 is not allowed instead of 2000::/3, because it omits the rest of the octet, and a device could not tell if 2::/3 means "hex 0002" or "hex 2000".

Now that you understand a few of the conventions about how to represent IPv6 addresses and prefixes, a specific example can show how IANA's IPv6 global unicast IP address assignment strategy can allow the easy and efficient routing shown in Figure 16-1.

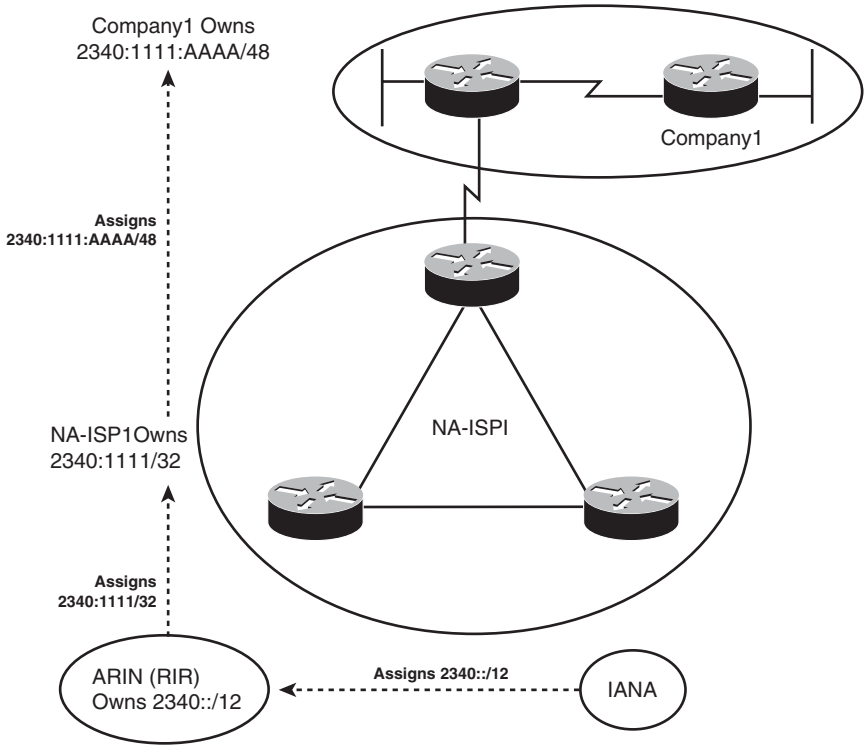
Global Unicast Prefix Assignment Example

IPv6 standards reserve the range of addresses inside the 2000::/3 prefix as global unicast addresses. This address range includes all IPv6 addresses that begin with binary 001, or as more easily recognized, all IPv6 addresses that begin with a 2 or 3. IANA assigns global unicast IPv6 addresses as public and globally unique IPv6 addresses, as discussed using the example previously shown in Figure 16-1, allowing hosts using those addresses to communicate through the Internet without the need for NAT. In other words, these addresses fit the purest design for how to implement IPv6 for the global Internet.

Figure 16-3 shows an example set of prefixes that could result in a company (Company1) being assigned a prefix of 2340:1111:AAAA::/48.

The process starts with IANA, who owns the entire IPv6 address space and assigns the rights to *registry prefix* to one of the RIRs (ARIN in this case, in North America). For the purposes of this chapter, assume that IANA assigns prefix 2340::/12 to ARIN. This assignment means that ARIN has the rights to assign any IPv6 addresses that begin with the first 12 bits of hex 2340 (binary value 0010 0011 0100). For perspective, that's a large group of addresses: 2^{116} to be exact.

Next, NA-ISP1 asks ARIN for a prefix assignment. After ARIN ensures that NA-ISP1 meets some requirements, ARIN might assign *site prefix* 2340:1111::/32 to NA-ISP1. This too is a large group: 2^{96} addresses to be exact. For perspective, this one address block may



Key Topic

Figure 16-3 *Example IPv6 Prefix Assignment in the Internet*

well be enough public IPv6 addresses for even the largest ISPs, without that ISP ever needing another IPv6 prefix.

Finally, Company1 asks its ISP, NA-ISP1, for the assignment of an IPv6 prefix. NA-ISP1 assigns Company1 the site prefix 2340:1111:AAAA::/48, which is again a large range of addresses: 2^{80} in this case. A little later in this section the text shows what Company1 could do with that prefix, but first, examine Figure 16-4, which presents the same concepts as in Figure 16-1, but now with the actual prefixes shown.

The figure shows the perspectives of routers outside North America, routers from another ISP in North America, and other routers in the same ISP. Routers outside North America can use a route for prefix 2340::/12, knowing the IANA assigned this prefix to be used only by ARIN. This one route could match all IPv6 addresses assigned in North America. Routers in NA-ISP2, an example alternative ISP in North America, need one route for 2340:1111::/32, the prefix assigned to NA-ISP1. This one route could match all packets destined for all customers of NA-ISP1. Inside NA-ISP1, its routers need to know to which NA-ISP1 router to forward packets to for that particular customer (named ISP-1 in this case), so the routes inside NA-ISP1's routers lists a prefix of 2340:1111:AAAA::/48.

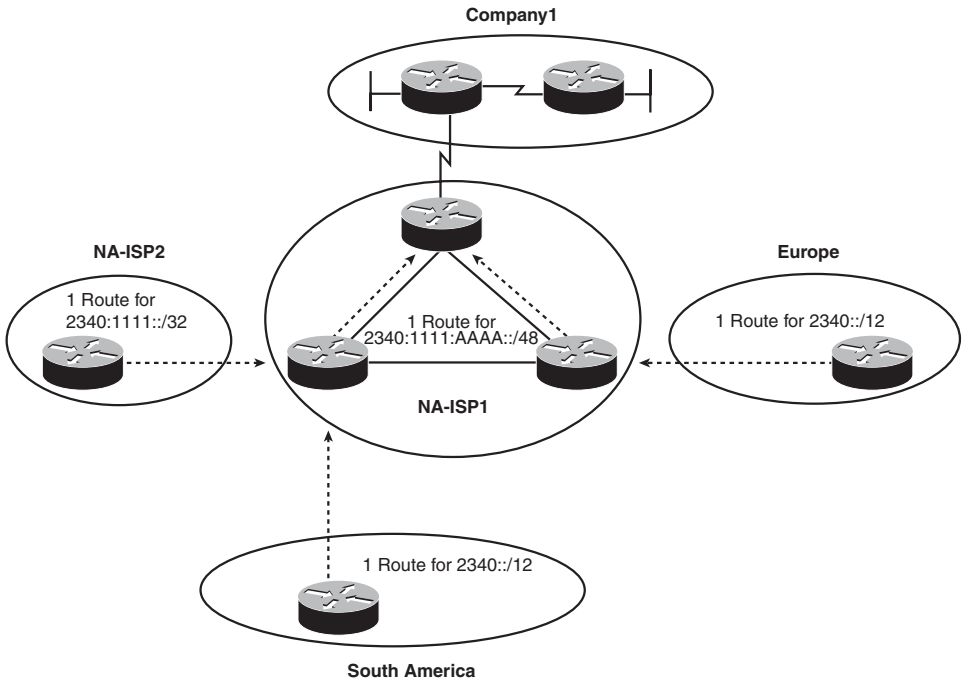


Figure 16-4 *IPv6 Global Routing Concepts*

Note: The /48 prefix assigned to a single company is called either a *global routing prefix* or a *site prefix*.

Subnetting Global Unicast IPv6 Addresses Inside an Enterprise

The original IPv4 Internet design called for each organization to be assigned a classful network number, with the Enterprise subdividing the network into smaller address ranges by subnetting the classful network. This same concept of subnetting carries over from IPv4 to IPv6, with the Enterprise subnetting its assigned global unicast prefix into smaller prefixes.

To better understand the IPv6 subnetting, you can draw on either classful or classless IPv4 addressing concepts, whichever you find most comfortable. From a classless perspective, you can view the IPv6 addresses as follows:

- The prefix assigned to the Enterprise by the ISP (the global routing prefix) acts like the prefix assigned for IPv4.
- The Enterprise engineer extends the prefix length, borrowing host bits, to create a subnet part of the address with which to identify individual subnets.
- The remaining part of the addresses on the right, called either the interface ID or host part, works just like the IPv4 host part, uniquely identifying a host inside a subnet.

Key Topic

For example, Figure 16-5 shows a more detailed view of the Company1 Enterprise network shown in several of the previous figures in this chapter. The design concepts behind how many subnets are needed with IPv6 are identical as with IPv4–a subnet is needed for each VLAN, and for each serial link, with the same options for subnets with Frame Relay. In this case, two LANs and two serial links exist, so Company1 needs four subnets.

Key Topic

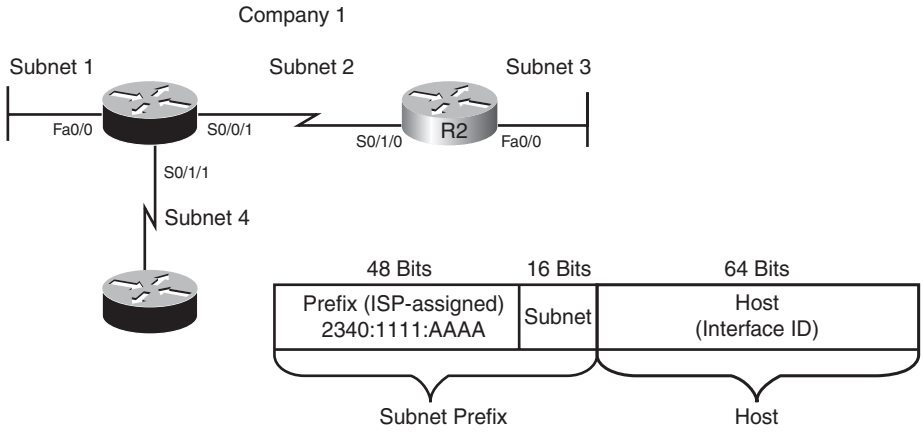


Figure 16-5 *Company1–Needs Four Subnets*

The figure also shows how the Enterprise engineer extended the length of the prefix as assigned by the ISP (/48) to /64, thereby creating a 16-bit subnet part of the address structure. To create this extra 16-bit subnet field, the engineer uses the same concept as with IPv4 when choosing a subnet mask by borrowing bits from the host field of an IPv4 address. In this case, think of the original host field (before subnetting) as having 80 bits, because the site prefix is 48-bits long, leaving 80 bits. The design in Figure 16-5 borrows 16 bits for the subnet field, leaving a measly 64 bits for the host field.

A bit of math about the design choices can help provide some perspective on the scale of IPv6. The 16-bit subnet field allows for 2^{16}, or 65,536, subnets–overkill for all but the very largest organizations or companies. (There are no worries about a zero or broadcast subnet in IPv6!) The host field is seemingly even more overkill: 2^{64} hosts per subnet, which is more than 1,000,000,000,000,000,000 addresses per subnet. However, there is a good reason for this large host or interface ID part of the address because it allows one of the automatic IPv6 address assignment features to work well, as covered later in the section "Assigning IPv6 Global Unicast IP Addresses."

Figure 16-6 takes the concept to the conclusion, assigning the specific four subnets to be used inside Company1. Note that the figure shows the subnet fields and prefix lengths (64 in this case) in bold.

Note: The subnet numbers in Figure 16-6 could be abbreviated slightly, removing the three leading 0s from the last shown quartets. The figure includes the leading 0s to show the entire subnet part of the prefixes.

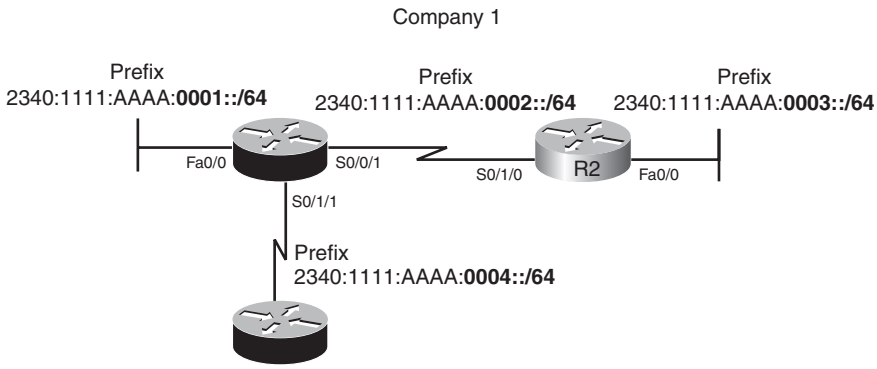


Figure 16-6 *Company1–4 Subnets Assigned*

Figure 16-6 just shows one option for subnetting the prefix assigned to Company1. However, any number of subnet bits could be chosen if the host field retained enough bits to number all hosts in a subnet. For example, a /112 prefix length could be used, extending the /48 prefix by 64 bits (4 hex quartets). Then, for the design in Figure 16-6, you could choose the following four subnets:

2340:1111:AAAA::0001:0000/112

2340:1111:AAAA::0002:0000/112

2340:1111:AAAA::0003:0000/112

2340:1111:AAAA::0004:0000/112

By using global unicast IPv6 addresses, Internet routing can be very efficient; Enterprises can have plenty of IP addresses and plenty of subnets with no requirement for NAT functions to conserve the address space.

Prefix Terminology

Before wrapping up this section, a few new terms need to be introduced. The process of global unicast IPv6 address assignment examines many different prefixes with many different prefix lengths. The text scatters a couple of more specific terms, but for easier study, Table 16-4 summarizes the four key terms with some reminders of what each means.

Table 16-4 *Example IPv6 Prefixes and Their Meanings*

Term	Assignment	Example from Chapter 16
Registry prefix	By IANA to an RIR	2340::/12
ISP prefix	By an RIR to an ISP[1]	2340:1111/32
Site prefix or global routing prefix	By an ISP or registry to a customer (site)	2340:1111:AAAA/48
Subnet prefix	By an Enterprise engineer for each individual link	2340:1111:AAAA:0001/64

[1]Although an RIR can assign a prefix to an ISP, an RIR may also assign a prefix to other Internet registries, which might subdivide and assign additional prefixes, until eventually an ISP and then their customers are assigned some unique prefix.

Assigning IPv6 Global Unicast Addresses

This section still focuses on global unicast IPv6 addresses but now examines the topic of how a host, router interface, or other device knows what global unicast IPv6 address to use. Also, hosts (and sometimes routers) also need to know a few other facts that can be learned at the same time as learning their IPv6 address. So, this section also discusses how hosts can get all the following relevant information that lets them use their global unicast addresses:

- IP address
- IP subnet mask (prefix length)
- Default router IP address
- DNS IP address(es)

IPv6 actually has four major options for IPv6 global unicast address assignment. This section looks at these options in the same order as listed in Table 16-5. Each method can use dynamic processes or static configuration, and each method can differ in terms of how a host or router gathers the other pertinent information (such as DNS IP addresses). Table 16-5 summarizes these main methods for easier review.

Table 16-5 *Summary of IPv6 Address Assignment for Global Unicast Addresses*

Method	Dynamic or static	Prefix and length learned from...	Host learned from...	Default router learned from...	DNS addresses learned from...
Stateful DHCP	Dynamic	DHCP Server	DHCP Server	Router, using NDP	(Stateful) DHCP Server
Stateless autoconfig	Dynamic	Router, using NDP	Derived from MAC	Router, using NDP	Stateless DHCP
Static configuration	Static	Local config	Local config	Router, using NDP	Stateless DHCP
Static config with EUI-64	Static	Local config	Derived from MAC	Router, using NDP	Stateless DHCP

The rest of this section develops more detail about the topics in the table. Some of the processes work much like IPv4, and some do not. Regardless, as you work through the

material, keep in mind one key fact about how IPv6 protocols approach the address assignment process:

> IPv6 address assignment processes may split the IPv6 address assignment into two parts: the prefix/length assignment and the host (interface ID) assignment.

Stateful DHCP for IPv6

IPv6 hosts can use stateful DHCP to learn and lease an IP address and corresponding prefix length (mask), the IP address of the default router, and the DNS IP address(es). The concept works basically like DHCP for IPv4; the host sends a (multicast) packet searching for the DHCP server. When a server replies, the DHCP client sends a message asking for a lease of an IP address, and the server replies, listing an IPv6 address, prefix length, and DNS IP addresses. (Note that Stateful DHCPv6 does not supply the default router information, instead relying on Neighbor Discovery Protocol between the client and local routers.) The names and formats of the actual DHCP messages have changed quite a bit from IPv4 to IPv6, so DHCPv4 and DHCPv6 actually differ in detail, but the basic process remains the same. (The term DHCPv4 refers to the version of DHCP used for IPv4, and the term DHCPv6 refers to the version of DHCP used for IPv6.)

DHCPv4 servers retain *state information* about each client, such as the IP address leased to that client, and the length of time for which the lease is valid. In other words, DHCPv4 tracks the current state of DHCP clients. DHCPv6 servers happen to have two operational modes: stateful, in which the server does track state information, and stateless, in which the server does not track any state information. Stateful DHCPv6 servers fill the same role as the older DHCPv4 servers, whereas stateless DHCPv6 servers fill a different purpose as one part of the stateless autoconfiguration process. (Stateless DHCP, and its purpose, is covered in the upcoming section "Finding the DNS IP Addresses Using Stateless DHCP.")

One difference between DHCPv4 and stateful DHCPv6 is that IPv4 hosts send IP broadcasts to find DHCP servers, whereas IPv6 hosts send IPv6 multicasts. IPv6 multicast addresses have a prefix of FF00::/8, meaning that if the first 8 bits of an address are binary 11111111, or FF in hex. The multicast address FF02::1:2 (longhand FF02:0000:0000:0000:0000:0000:0001:0002) has been reserved in IPv6 to be used by hosts to send packets to an unknown DHCP server, with the routers working to forward these packets to the appropriate DHCP server.

Stateless Autoconfiguration

The second of the two options for dynamic IPv6 address assignment uses a built-in IPv6 feature called *stateless autoconfiguration* as the core tool. Stateless autoconfiguration allows a host to automatically learn the key pieces of addressing information–prefix, host, and prefix length–plus the default router IP address and DNS IP addresses. To learn or derive all these pieces of information, stateless autoconfiguration actually uses the following functions:

Step 1. IPv6 Neighbor Discovery Protocol (NDP), particularly the router solicitation and router advertisement messages, to learn the prefix, prefix length, and default router

Step 2. Some math to derive the interface ID (host ID) portion of the IPv6 address, using a format called EUI-64

Step 3. Stateless DHCP to learn the DNS IPv6 addresses

This section examines all three topics in order.

Learning the Prefix/Length and Default Router with NDP Router Advertisements

The IPv6 Neighbor Discovery Protocol (NDP) has many functions. One function allows IPv6 hosts to multicast a message that asks all routers on the link to announce two key pieces of information: the IPv6 addresses of routers willing to act as a default gateway and all known IPv6 prefixes on the link. This process uses ICMPv6 messages called a Router Solicitation (RS) and a Router Advertisement (RA).

For this process to work, before a host sends an RS message on a LAN, some router connected to that same LAN must already be configured for IPv6. The router must have an IPv6 address configured, and it must be configured to route IPv6 traffic. At that point, the router knows it can be useful as a default gateway, and it knows at least one prefix that can be useful to any clients on the LAN.

For example, Figure 16-7 shows a subset of the internetwork seen in Figures 16-5 and 16-6, with the same IPv6 addresses and subnets used. Router R1's Fa0/0 has already been configured with an IPv6 address (2340:1111:AAAA:1:213:19FF:FE7B:5004/64) and has been configured to route IPv6 with the **ipv6 unicast-routing** global command.

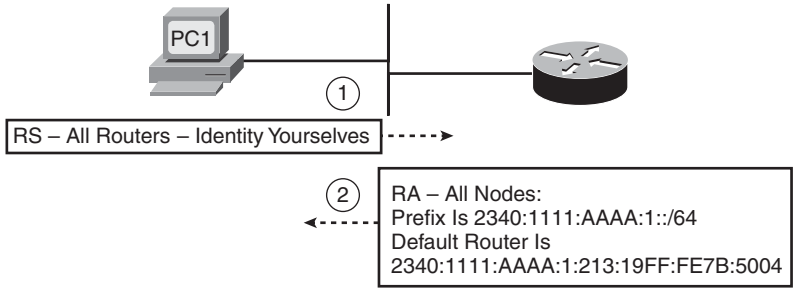


Figure 16-7 *Example NDP RS/RA Process to Find the Default Routers*

In the figure, host PC1, using stateless autoconfig, sends the RS message as an IPv6 multicast message destined to all IPv6 routers on the local link. The RS asks all routers to respond to the questions "What IPv6 prefix(s) is used on this subnet?" and "What is the IPv6 address(s) of any default routers on this subnet?" The figure also shows R1's response (RS), listing the prefix (2340:1111:AAAA:1::/64), and with R1's own IPv6 address as a potential default router.

Note: IPv6 allows for multiple prefixes and multiple default routers to be listed in the RA message; Figure 16-7 just shows one of each for simplicity's sake. One router's RA would include IPv6 addresses and prefixes advertised by other routers on the link as well.

IPv6 does not use broadcasts. In fact, there is no such thing as a subnet broadcast address, a networkwide broadcast address, or an equivalent of the all-hosts 255.255.255.255 broadcast IPv4 address. Instead, IPv6 makes use of multicast addresses. By defining a different multicast IPv6 address for different functions, an IPv6 host that has no need to participate in a particular function can simply ignore those particular multicasts, reducing the impact to the host.

For example, the RS message needs to be received and processed only by routers, so the RS message's destination IP address is FF02::2, which IPv6 reserves for use only by IPv6 routers. IPv6 defines that routers send RA messages to a multicast address intended for use by all IPv6 hosts on the link (FF02::1); routers do not forward these messages to other links. As a result, not only does the host that sent the RS message learn the information, all other hosts on the link learn the details as well. Table 16-6 summarizes some of the key details about the RS/RA messages.

Table 16-6 *Details of the RS/RA Process*

Message	RS	RA
Multicast destination	FF02::2	FF02::1
Meaning of Multicast address	All routers on this link	All IPv6 nodes on this link

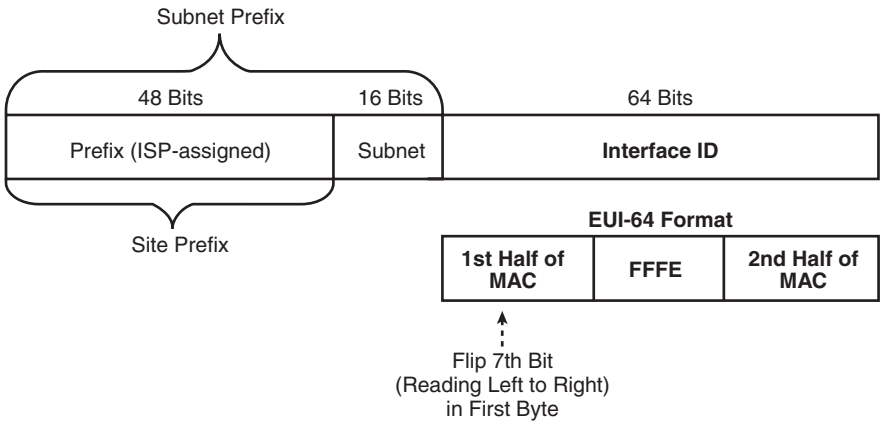
Calculating the Interface ID Using EUI-64

Earlier in the chapter, Figure 16-5 shows the format of an IPv6 global unicast address with the second half of the address called the host ID or interface ID. The value of the interface ID portion of a global unicast address can be set to any value if no other host in the same subnet attempts to use the same value.

To automatically create a guaranteed-unique interface ID, IPv6 defines a method to calculate a 64-bit interface ID derived from that host's MAC address. Because the burned-in MAC address should be literally globally unique, the derived interface ID should also be globally unique as well.

The EUI-64 process takes the 6-byte (48-bit) MAC address and expands it into a 64-bit value. To do so, IPv6 fills in 2 more bytes into the middle of the MAC address. IPv6 separates the original MAC address into two 3-byte halves and inserts hex FFFE in between the halves to form the Interface ID field of the IPv6 address. The conversion also requires flipping the seventh bit inside the IPv6 address, resulting in a 64-bit number that conforms to a convention called the EUI-64 format. The process is shown in Figure 16-8.

Although it may seem a bit convoluted, it works. Also, with a little practice, you can look at an IPv6 address and quickly notice the FFFE late in the address and then easily find the two halves of the corresponding interface's MAC address.



Key Topic

Figure 16-8 *IPv6 Address Format with Interface ID and EUI-64*

For example, the following two lines list a host's MAC address, and corresponding EUI-64 format Interface ID, assuming the use of an address configuration option that uses the EUI-64 format:

0034:5678:9ABC

0234:56FF:FE78:9ABC

Note: To change the seventh bit (left-to-right) in the example, hex 00 converts to binary 00000000, change the seventh bit to 1 (00000010), convert back to hex, for hex 02 as the first two digits.

At this point in the stateless autoconfig process, a host knows its full IPv6 address and prefix length, plus a local router to use as default gateway. The next section discusses how to complete the process using stateless DHCP.

Finding the DNS IP Addresses Using Stateless DHCP

Although the DHCP server function for IPv4 does not explicitly use the word "stateful" in its name, IPv4 DHCP servers keep state information about DHCP clients. The server keeps a record of the leased IP addresses, and when the lease expires. The server typically releases the addresses to the same client before the lease expires, and if no response is heard from a DHCP client in time to renew the lease, the server releases that IP address back into the pool of usable IP addresses–again keeping that state information. The server also has configuration of the subnets in uses and a pool of addresses in most subnets from which the server can assign IP addresses. It also serves other information, such as the default router IP addresses in each subnet, and the DNS server IP addresses.

The IPv6 stateful DHCP server, as previously discussed in the section "Stateful DHCP for IPv6", follows the same general idea. However, for IPv6, this server's name includes the word stateful, to contrast it with the stateless DHCP server function in IPv6.

The stateless DHCP server function in IPv6 solves one particular problem: It supplies the DNS server IPv6 address(es) to clients. Because all hosts typically use the same small

number of DNS servers, the stateless DHCP server does not need to keep track of any state information. An engineer simply configures the stateless DHCP server to know the IPv6 addresses of the DNS servers, and the servers tells any host or other device that asks, keeping no record of the process.

Hosts that use stateless autoconfig also use stateless DHCP to learn the DNS server IPv6 addresses.

Table 16-7 summarizes some of the key features of stateful and stateless DHCPv6.

Table 16-7 *Comparing Stateless and Stateful DHCPv6 Services*

Feature	Stateful DHCP	Stateless DHCP
Remembers IPv6 address (state information) of clients that make requests	Yes	No
Assigns IPv6 address to client	Yes	No
Supplies useful information, such as DNS server IP addresses	Yes	Yes
Most useful in conjunction with stateless autoconfiguration	No	Yes

Static IPv6 Address Configuration

Two options exist for static configuration of IPv6 addresses. For one option, you configure the entire 128-bit IPv6 address, and for the other, you just configure the 64-bit prefix and tell the device to use an EUI-64 calculation for the interface ID portion of the address. Both options result in the host or router interface knowing its full 128-bit IPv6 address and prefix length.

When a host uses either form of static IPv6 address configuration, the host does not need to statically configure the other key pieces of information (default router and DNS IP addresses). The host can use the usual NDP process to discover any default routers and stateless DHCP to discover the DNS IPv6 addresses.

When a router uses static IPv6 address configuration, it may still use stateless DHCP to learn the DNS IP addresses. The upcoming section "Configuring IPv6 Addresses on Cisco Routers" shows several examples of this configuration.

Survey of IPv6 Addressing

So far, this chapter has focused on the IPv6 addresses that most closely match the concept of IPv4 addresses: the global unicast IPv6 address. This section now takes a broader look at IPv6 addressing, including some concepts that can be tied to older IPv4 concepts, and some that are unique to IPv6.

This section begins with a brief overview of IPv6 addressing. It then looks at unicast IPv6 addresses, along with a brief look at some of the commonly used multicast addresses. This section ends with a discussion of a couple of related protocols, namely Neighbor Discovery Protocol (NDP) and Duplicate Address Detection (DAD).

Overview of IPv6 Addressing

The whole concept of global unicast addressing does have many similarities as compared with IPv4. If viewing IPv4 addresses from a classless perspective, both IPv4 and IPv6 global unicast addresses have two parts: subnet plus host for IPv4 and prefix plus interface ID for IPv6. The format of the addresses commonly list a slash followed by the prefix length–a convention sometimes referred to as *CIDR notation*, and other times as *prefix notation*. Subnetting works much the same, with a public prefix assigned by some numbering authority, and the Enterprise choosing subnet numbers, extending the length of the prefix to make room to number the subnets.

IPv6 addressing, however, includes several other types of unicast IPv6 addresses beside the global unicast address. Additionally, IPv6 defines other general categories of addresses, as summarized in this list.

- **Unicast:** Like IPv4, hosts and routers assign these IP addresses to a single interface for the purpose of allowing that one host or interface to send and receive IP packets.
- **Multicast:** Like IPv4, these addresses represent a dynamic group of hosts, allowing a host to send one packet that is then delivered to every host in the multicast group. IPv6 defines some special-purpose multicast addresses for overhead functions (such as NDP). IPv6 also defines ranges of multicast addresses for application use.
- **Anycast:** This address type allows the implementation of a nearest server among duplicate servers concept. This design choice allows servers that support the exact same function to use the exact same unicast IP address. The routers then forward a packet destined for such an address to the nearest server that is using the address.

Two big differences exist when comparing general address categories for IPv4 and IPv6. First, IPv6 adds the formal concept of Anycast IPv6 addresses as shown in the preceding list. IPv4 does not formally define an Anycast IP address concept, although a similar concept may be implemented in practice. Second, IPv6 simply has no Layer 3 broadcast addresses. For example, all IPv6 routing protocols send Updates either to Unicast or Multicast IPv6 addresses, and overhead protocols such as NDP make use of multicasts as well. In IPv4, ARP still uses broadcasts, and older routing protocols such as RIP-1 also used broadcasts. With IPv6, there is no need to calculate a subnet broadcast address (hoorah!) and no need to make hosts process overhead broadcast packets meant only for a few devices in a subnet.

Finally, note that IPv6 hosts and router interfaces typically have at least two IPv6 addresses and may well have more. Hosts and routers typically have a Link Local type of IPv6 address (as described in the upcoming section "Link Local Unicast Addresses"). A router may or may not have a global unicast address, and may well have multiple. IPv6 simply allows the configuration of multiple IPv6 addresses with no need for or concept of secondary IP addressing.

Unicast IPv6 Addresses

IPv6 supports three main types of unicast addresses: link local, global unicast, and unique local. This section takes a brief look at link local and unique local addresses.

Unique Local IPv6 Addresses

Unique local unicast IPv6 addresses have the same function as IPv4 RFC 1918 private addresses. RFC 4193 states that these addresses should be used inside a private organization, and should not be advertised into the Internet. Unique local unicast addresses begin with hex FD (FD00::/8), with the format shown in Figure 16-9.

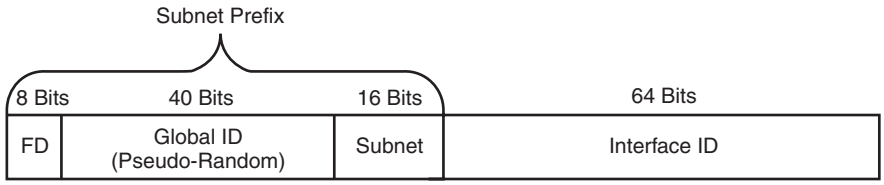


Figure 16-9 *Unique Local Address Format*

To use these addresses, an Enterprise engineer would choose a 40-bit global ID in a pseudo-random manner rather than asking for a registered public prefix from an ISP or other registry. To form the complete prefix, the chosen 40 bits would be combined with the initial required 8 bits (hex FD) to form a 48-bit site prefix. The engineer can then use a 16-bit subnet field to create subnets, leaving a 64-bit Interface ID. The interface ID could be created by static configuration or by the EUI-64 calculation.

This type of unicast address gives the engineer the ability to create the equivalent of an IPv4 private address structure, but given the huge number of available public IPv6 addresses, it may be more likely that engineers plan to use global unicast IP addresses throughout an Enterprise.

Link Local Unicast Addresses

IPv6 uses link local addresses for sending and receiving IPv6 packets on a single subnet. Many such uses exist; here's just a small sample:

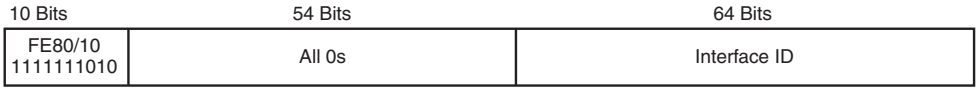
- Used as the source address for RS and RA messages for router discovery (as previously shown in Figure 16-7)
- Used by Neighbor discovery (the equivalent of ARP for IPv6)
- As the next-hop IPv6 address for IP routes

By definition, routers use a link local scope for packets sent to a link local IPv6 address. The term *link local* scope means exactly that–the packet should not leave the local link, or local subnet if you will. When a router receives a packet destined for such a destination address, the router does not forward the packet.

The link local IPv6 addresses also help solve some chicken-and-egg problems because each host, router interface, or other device can calculate its own link local IPv6 address without needing to communicate with any other device. So, before sending the first packets, the host can calculate its own link local address, so the host has an IPv6 address to use when doing its first overhead messages. For example, before a host sends an NDP RS

(router solicitation) message, the host will have already calculated its link local address, which can be used as the source IPv6 address on the RS message.

Link local addresses come from the FE80::/10 range, meaning the first 10 bits must be 1111 1110 10. An easier range to remember is that all hex link local addresses begin FE8, FE9, FEA, or FEB. However, practically speaking, for link local addresses formed automatically by a host (rather than through static configuration), the address always starts FE80, because the automatic process sets bits 11-64 to binary 0s. Figure 16-10 shows the format of the link local address format under the assumption that the host or router is deriving its own link local address, therefore using 54 binary 0s after the FE80::/10 prefix.



Key Topic

Figure 16-10 *Link Local Address Format*

IPv6 Unicast Address Summary

You may come across a few other types of IPv6 addresses in other reading. For example, earlier IPv6 RFCs defined the Site Local address type, which was meant to be used like IPv4 private addresses. However, this address type has been deprecated (RFC 3879). Also, the IPv6 migration and coexistence tools discussed in Chapter 18 use some conventions for IPv6 unicast addresses such that IPv4 addresses are imbedded in the IPv6 address.

Additionally, it is helpful to know about other special unicast addresses. An address of all hex 0s, written ::/128, represents an unknown address. This can be used as a source IPv6 address in packets when a host has no suitable IPv6 address to use. The address ::1/128, representing an address of all hex 0s except a final hex digit 1, is a loopback address. Packets sent to this address will be looped back up the TCP/IP stack, allowing for easier software testing. (This is the equivalent of IPv4's 127.0.0.1 loopback address.)

Table 16-8 summarizes the IPv6 unicast address types for easier study.

Key Topic

Table 16-8 *Common Link-Local Multicast Addresses*

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet	2000::/3	2 or 3
Unique local	Unicast packets inside one organization	FD00::/8	FD

Table 16-8 *Common Link-Local Multicast Addresses*

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Link local	Packets sent in the local subnet	FE80::/10	FE8
Site local	Deprecated; originally meant to be used like private IPv4 addresses	FECO::/10	FEC, FED, FEE, FEF
Unspecified	An address used when a host has no usable IPv6 address	::/128	N/A
Loopback	Used for software testing, like IPv4's 127.0.0.1	::1/128	N/A

IPv6 RFCs define the FE80::/10 prefix, which technically means that the first three hex digits could be FE8, FE9, FEA, or FEB. However, bit positions 11-64 of link local addresses should be 0, so in practice, link local addresses should always begin with FE80.

Multicast and Other Special IPv6 Addresses

IPv6 supports multicasts on behalf of applications and multicasts to support the inner workings of IPv6. To aid this process, IPv6 defines ranges of IPv6 addresses and an associated scope, with the scope defining how far away from the source of the packet that the network should forward a multicast.

All IPv6 multicast addresses begin with FF::/8 – in other words, with FF as the first two digits. Multicasts with a link local scope, like most of the multicast addresses referenced in this chapter, begin with FF02::/16; the 2 in the fourth hex digit identifies the scope as link local. A fourth digit of hex 5 identifies the broadcast as site local scope, with those multicasts beginning with FF05::/16.

For reference, Table 16-9 lists some of the more commonly seen IPv6 multicast addresses. Of particular interest are the addresses chosen for use by RIP, OSPF, and EIGRP, which somewhat mirror the multicast addresses each protocol uses for IPv4. Note also that all but the last two entries have link local scope.

Table 16-9 *Common Multicast Addresses*

Purpose	IPv6 Address	IPv4 Equivalent
All IPv6 nodes on the link	FF02::1	subnet broadcast address

All IPv6 routers on the link	FF02::2	N/A
OSPF messages	FF02::5, FF02::6	224.0.0.5, 224.0.0.6
RIP-2 messages	FF02::9	224.0.0.9
EIGRP messages	FF02::A	224.0.0.10
DHCP relay agents (routers that forward to the DHCP server)	FF02:1:2	N/A
DHCP servers (site scope)	FF05::1:3	N/A
All NTP servers (site scope)	FF05::101	N/A

Layer 2 Addressing Mapping and Duplicate Address Detection

As with IPv4, any device running IPv6 needs to determine the data link layer address used by devices on the same link. IPv4 uses Address Resolution Protocol (ARP) on LANs and Inverse ARP (InARP) on Frame Relay. IPv6 defines a couple of new protocols that perform the same function. These new functions use ICMPv6 messages and avoid the use of broadcasts, in keeping with IPv6's avoidance of broadcasts. This section gives a brief explanation of each protocol.

Neighbor Discovery Protocol for Layer 2 Mapping

When an IPv6 host or router needs to send a packet to another host or router on the same LAN, the host/router first looks in its neighbor database. This database contains a list of all neighboring IPv6 addresses (addresses in connected links) and their corresponding MAC addresses. If not found, the host or router uses the Neighbor Discovery Protocol (NDP) to dynamically discover the MAC address.

Figure 16-11 shows a sample of such a process, using the same host and router seen earlier in Figure 16-8.

The process acts like the IPv4 ARP process, just with different details. In this case, PC1 sends a multicast message called a Neighbor Solicitation (NS) ICMP message, asking R1 to reply with R1's MAC address. R1 sends a Neighbor Advertisement (NA) ICMP message, unicast back to PC1, listing B's MAC address. Now PC1 can build a data link frame with R1's MAC listed as the destination address and send encapsulated packets to R1.

The NS message uses a special multicast destination address called a *solicited node multicast* address. On any given link, the solicited node multicast address represents all hosts with the same last 24 bits of their IPv6 addresses. By sending packets to the solicited node multicast address, the packet reaches the correct host, but it may also reach a few other hosts–which is fine. (Note that packets sent to a solicited node multicast address have a link local scope.)

The solicited node multicast address begins with FF02::1:FF:0/104. The final 24 bits (6 hex digits) of the address is formed by adding the last 24 bits of the IPv6 address to which the message is being sent. This convention allows convenient use of LAN multicast addresses, which begin with hex 01005E hex, followed by one additional binary 0, and then an addi-

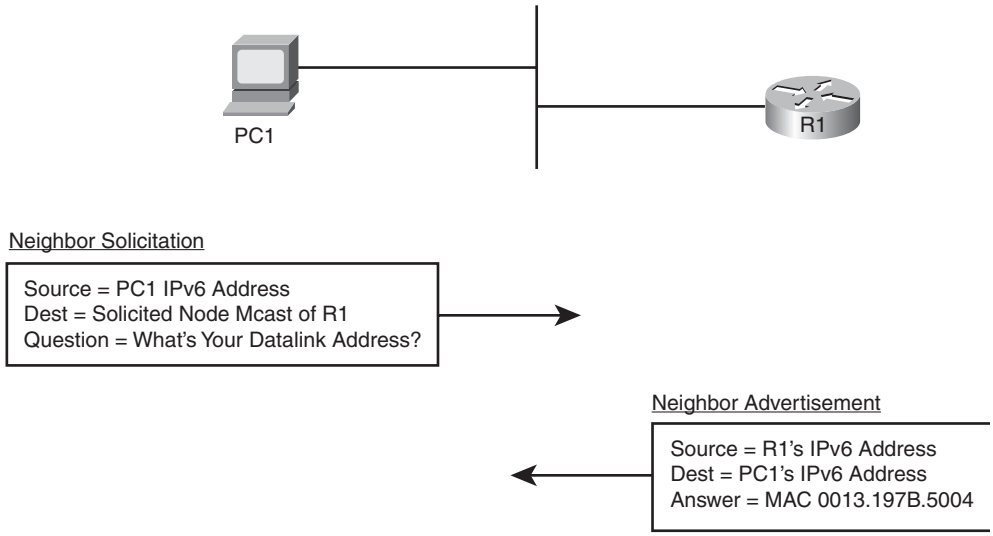


Figure 16-11 *Neighbor Discovery Protocol*

Key Topic

tional 23 bits–in this case taken from the low order 23 bits of the IPv6 address. All IPv6 listen for frames sent to their own solicited node multicast address, so that when a host or router receives such a multicast, the host can realize that it should reply. For example, in this case, based on R1's IPv6 address previously seen in Figure 16-8:

- R1's IPv6 address: 2340:1111:AAAA:1:213:19FF:FE7B:5004
- R1's solicited node address: FF02::1:FF:7B:5004

Note: The corresponding Ethernet multicast MAC address would be 0100.5E7B.5004.

Duplicate Address Detection (DAD)

When an IPv6 interface first learns an IPv6 address, or when the interface begins working after being down for any reason, the interface performs duplicate address detection (DAD). The purpose of this check is to prevent hosts from creating problems by trying to use the same IPv6 address already used by some other host on the link.

To perform such a function, the interface uses the same NS message shown in Figure 16-11 but with small changes. To check its own IPv6 address, a host sends the NS message to the solicited node multicast address based on its own IPv6 address. If some host sends a reply, listing the same IPv6 address as the source address, the original host has found that a duplicate address exists.

Inverse Neighbor Discovery

The ND protocol discussed in this section starts with a known neighbor's IPv6 address and seeks to discover the link layer address used by that IPv6 address. On Frame Relay

networks, and with some other WAN data link protocols, the order of discovery is reversed. A router begins with knowledge of the neighbor's link layer address and instead needs to dynamically learn the IPv6 address used by that neighbor.

IPv4 solves this discovery problem on LANs using ARP, and the reverse problem over Frame Relay using Inverse ARP (InARP). IPv6 solves the problem on LANs using ND, and now for Frame Relay, IPv6 solves this problem using Inverse Neighbor Discovery (IND). IND, also part of the ICMPv6 protocol suite, defines an Inverse NS (INS) and Inverse NA (INA) message. The INS message lists the known neighbor link layer address (DLCI for Frame Relay), and the INS asks for that neighboring device's IPv6 addresses. The details inside the INS message include the following:

- Source IPv6: IPv6 unicast of sender
- Destination IPv6: FF02::1 (all IPv6 hosts multicast)
- Link layer addresses
- Request: Please reply with your IPv6 address(es)

The IND reply lists all the IPv6 addresses. As with IPv4, the **show frame-relay map** command lists the mapping learned from this process.

Configuring IPv6 Addresses on Cisco Routers

Most IPv6 implementation plans make use of both static IPv6 address configuration and dynamic configuration options. As is the case with IPv4, the plan assigns infrastructure devices with static addresses, with client hosts using one of the two dynamic methods for address assignment.

IPv6 addressing includes many more options than IPv4, and as a result, many more configuration options exist. A router interface can be configured with a static global unicast IPv6 address, either with or without using the EUI-64 option. Although less likely, a router could be configured to dynamically learn its IPv6 address with either stateful DHCP or stateless autoconfig. The router interface could be configured to either not use a global unicast address, instead relying solely on its link local address, or to borrow another interface's address using the IPv6 unnumbered feature.

This section summarizes the address configuration commands and shows several examples of configuration and verification commands for IPv6. To that end, Table 16-10 summarizes the IPv6 configuration commands and their meanings.

Table 16-10 *Router IOS IPv6 Configuration Command Reference*

Command	Description
ipv6 address *address/length*	Static configuration of the entire IPv6 unicast address.
ipv6 address *prefix/length* **eui-64**	Static configuration of the first 64 address bits; the router derives the last 64 bits with EUI-64.

ipv6 address autoconfig	Router uses stateless autoconfig to find address.
ipv6 address dhcp	Router uses stateful DHCP to find address.
ipv6 unnumbered *interface-type number*	Uses the same IPv6 unicast address as the referenced interface.
ipv6 enable	Enables IPv6 on the interface, but results in only a link local address.
ipv6 address *address* **link-local**	Overrides the automatically created link local address. The configured value must conform to the FE80::/10 prefix.
ipv6 address *address/length* **anycast**	Designates that the unicast address is an anycast.

Note: All the interface subcommands in Table 16-10 enable IPv6 on the interface, which means the router derives an IPv6 link local address for the interface. The description shows what the command does in addition to enabling IPv6.

Configuring Static IPv6 Addresses on Routers

The configuration examples in this section use the internetwork shown in Figure 16-12. The figure shows a diagram you might see in an implementation plan, with the five IPv6 subnet numbers shown over the five links. The interface ID of each interface is then abbreviated, or shown as eui-64, as a reminder of whether to configure the whole 128-bit address or to rely on the EUI-64 feature.

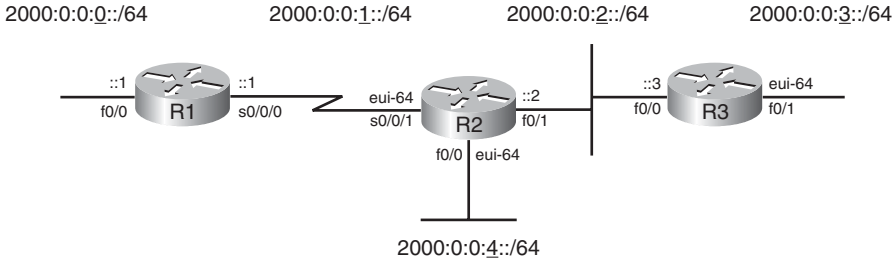


Figure 16-12 *Sample IPv6 Address Planning Diagram*

Example 16-1 shows the configuration process on Router R2, which uses EUI-64 on two interfaces, and a complete IPv6 address on another. Also, note that the configuration includes the **ipv6 unicast-routing** global configuration command, which enables the router to route IPv6 traffic. (The addresses can be configured without also configuring **ipv6 unicast-routing**, but without this command, the router acts more like an IPv6 host, and it will not forward IPv6 packets until this command has been configured.)

Example 16-1 *R2's IPv6 Configuration*

```
R2# show running-config
! lines omitted for brevity

interface FastEthernet0/0
 ipv6 address 2000:0:0:4::/64 eui-64
!
interface FastEthernet0/1
 ipv6 address 2000:0:0:2::2/64
!
interface Serial0/0/1
 ipv6 address 2000:0:0:1::/64 eui-64
!
!
R2# show ipv6 interface brief
FastEthernet0/0             [up/up]
    FE80::213:19FF:FE7B:5004
    2000::4:213:19FF:FE7B:5004
FastEthernet0/1             [up/up]
    FE80::213:19FF:FE7B:5005
    2000:0:0:2::2
Serial0/0/0                 [administratively down/down]
    unassigned
Serial0/0/1                 [up/up]
    FE80::213:19FF:FE7B:5004
    2000::1:213:19FF:FE7B:5004
Serial0/1/0                 [administratively down/down]
    unassigned
Serial0/1/1                 [administratively down/down]
    unassigned

R2# show interfaces fa0/0

FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is 0013.197b.5004 (bia 0013.197b.5004)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
! lines omitted for brevity
```

The **ipv6 address** commands both enable IPv6 on the associated interfaces and define either the prefix (with the **eui-64** option) or the entire address. The **show** commands listed after the configuration confirm the IPv6 addresses. Of particular note

- All three interfaces now have link local addresses that begin FE80.
- F0/1 has the address exactly as configured.

- S0/0/1 and F0/0 have the configured prefixes (2000:0:0:1 and 2000:0:0:4, respectively), but with EUI-64 derived interface-IDs.
- S0/0/1 uses Fa0/0's MAC address (as shown in the **show interfaces f0/0** command) when forming its EUI-64.

On this last point, whenever IOS needs a MAC address for an interface, and that interface does not have a built-in MAC address, the router uses the MAC address of the lowest-numbered LAN interface on the router–in this case, F0/0. The following list shows the derivation of the last 64 bits (16 digits) of R2's IPv6 interface IDs for its global unicast IPv6 addresses on F0/0 and S0/0/1:

Step 1. Use F0/0's MAC Address: 0013.197B.5004.

Step 2. Split and insert FFFE: 0013:19FF:FE7B:5004.

Step 3. Invert bit 7: Hex 00 = 00000000 binary, flip for 00000010, and convert back to hex 02, resulting in 0213:19FF:FE7B:5004.

Multicast Groups Joined by IPv6 Router Interfaces

Next, consider the deeper information held in the **show ipv6 interface f0/0** command on Router R2, as shown in Example 16-2. Not only does it list the same link local and global unicast addresses, but it lists other special addresses as well.

Example 16-2 *All IPv6 Addresses on an Interface*

```
R2# show ipv6 interface f0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE7B:5004
  No Virtual link-local address(es):
  Global unicast address(es):
    2000::4:213:19FF:FE7B:5004, subnet is 2000:0:0:4::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF7B:5004
  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 22807)
  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.
```

The three joined multicast groups should be somewhat familiar after reading this chapter. The first multicast, FF02::1, represents all IPv6 devices, so router interfaces must listen for packets sent to this address. FF02::2 represents all IPv6 routers, so again, R2 must listen for packets sent to this address. Finally, the FF02::1:FF beginning value is the range for an address' solicited node multicast address, used by several functions, including the duplicate address detection (DAD) and neighbor discovery (ND).

Connected Routes and Neighbors

The third example shows some new concepts with the IP routing table. Example 16-3 shows R2's current IPv6 routing table that results from the configuration shown in Example 16-1. Note that no IPv6 routing protocols have been configured, and no static routes have been configured.

Example 16-3 *Connected and Local IPv6 Routes*

```
R2# show ipv6 route
IPv6 Routing Table - Default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   2000:0:0:1::/64 [0/0]
     via Serial0/0/1, directly connected
L   2000::1:213:19FF:FE7B:5004/128 [0/0]
     via Serial0/0/1, receive
C   2000:0:0:2::/64 [0/0]
     via FastEthernet0/1, directly connected
L   2000:0:0:2::2/128 [0/0]
     via FastEthernet0/1, receive
C   2000:0:0:4::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2000::4:213:19FF:FE7B:5004/128 [0/0]
     via FastEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

First, the IPv6 routing table lists the expected connected routes, but a new type of route–a "local" route–designated by an L in the output of the **show ipv6 route** command. The connected routes occur for any unicast IPv6 addresses on the interface that happen to have more than link local scope. So, R2 has routes for subnets 2000:0:0:1::/64, 2000:0:0:2::/64, and 2000:0:0:4::/64, but no connected subnets related to R2's link local addresses. The Local routes, all /128 routes, are essentially host routes for the router's unicast IPv6 addresses. These local routes allow the router to more efficiently process packets directed to the router itself, rather than for packets directed toward connected subnets.

The IPv6 Neighbor Table

The IPv6 neighbor table replaces the IPv4 ARP table, listing the MAC address of other devices that share the same link. Example 16-4 shows a **debug** that lists messages during the NDP process, a ping to R3's F0/0 IPv6 address, and the resulting neighbor table entries on R2.

Example 16-4 *Creating Entries and Displaying the Contents of R2's IPv6 Neighbor Table*

```
R2# debug ipv6 nd
  ICMP Neighbor Discovery events debugging is on
R2# ping 2000:0:0:2::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2000:0:0:2::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms
R2#
*Sep  2 17:07:25.807: ICMPv6-ND: DELETE -> INCMP: 2000:0:0:2::3
*Sep  2 17:07:25.807: ICMPv6-ND: Sending NS for 2000:0:0:2::3 on FastEthernet0/1
*Sep  2 17:07:25.807: ICMPv6-ND: Resolving next hop 2000:0:0:2::3 on interface
FastEthernet0/1
*Sep  2 17:07:25.811: ICMPv6-ND: Received NA for 2000:0:0:2::3 on FastEthernet0/1
from 2000:0:0:2::3
*Sep  2 17:07:25.811: ICMPv6-ND: Neighbour 2000:0:0:2::3 on FastEthernet0/1 : LLA
0013.197b.6588

R2# undebug all
All possible debugging has been turned off

R2# show ipv6 neighbors
IPv6 Address                                 Age Link-layer Addr State Interface
2000:0:0:2::3                                  0 0013.197b.6588  REACH Fa0/1
FE80::213:19FF:FE7B:6588                     0 0013.197b.6588  REACH Fa0/1
```

The example shows the entire NDP process by which R2 discovers R3's Fa0/0 MAC address. The example begins with a **debug ipv6 nd** command, which tells R2 to issue messages related to NDP messages. The **ping 2000:0:0:2::3** command that follows tells IOS to use IPv6 ping R3's F0/0 address–but R2 does not know the corresponding MAC address. The debug output that follows shows R2 sending an NS, with R3 replying with an NA message, listing R3's MAC address.

The example ends with the output of the **show ipv6 neighbor** command, which lists the neighbor table entries for both R3's IPv6 addresses.

Stateless Autoconfiguration

The final example in this section demonstrates stateless autoconfiguration using two routers, R2 and R3. In Example 16-5, R2's F0/1 configuration will be changed, using the

ipv6 address autoconfig subcommand on that interface. This tells R2 to use stateless autoconfig process, with R2 learning its prefix from Router R3. R2 then builds the rest of its IPv6 address using EUI-64.

Example 16-5 *Using Stateless Autoconfig on Router R2*

```
R2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# interface fa0/1
R2(config-if)# no ipv6 address
R2(config-if)# ipv6 address autoconfig
R2(config-if)#^Z

R2# show ipv6 interface brief
FastEthernet0/0             [up/up]
    FE80::213:19FF:FE7B:5004
    2000::4:213:19FF:FE7B:5004
FastEthernet0/1             [up/up]
    FE80::213:19FF:FE7B:5005
    2000::2:213:19FF:FE7B:5005
Serial0/0/0                 [administratively down/down]
    unassigned
Serial0/0/1                 [up/up]
    FE80::213:19FF:FE7B:5004
    2000::1:213:19FF:FE7B:5004
Serial0/1/0                 [administratively down/down]
    unassigned
Serial0/1/1                 [administratively down/down]
    unassigned

R2# show ipv6 router
Router FE80::213:19FF:FE7B:6588 on FastEthernet0/1, last update 0 min
  Hops 64, Lifetime 1800 sec, AddrFlag=0, OtherFlag=0, MTU=1500
  HomeAgentFlag=0, Preference=Medium
  Reachable time 0 (unspecified), Retransmit time 0 (unspecified)
  Prefix 2000:0:0:2::/64 onlink autoconfig
    Valid lifetime 2592000, preferred lifetime 604800
```

Starting with the configuration, the **no ipv6 address** command actually removes all configured IPv6 addresses from the interface and also disables IPv6 on interface F0/1. Then, the **ipv6 address autoconfig** command again enables IPv6 on F0/1 and tells R2 to use stateless autoconfig.

The **show** commands confirm that R2 does indeed learn its IPv6 address: 2000:0:0:2:0213:19FF:FE7B:5005. The **show ipv6 router** command, which lists the cached contents of any received RA messages, lists the information received from R3's RA message, including R3's link local address (used to identify the routers) and R3's advertised prefix (2000:0:0:2::/64).

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F.

Design Review Table

Table 16-11 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 16-11 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
An IPv6 design suggests that all client hosts should dynamically learn their IPv6 addresses. Which tools can be used? (2)	
A plan shows the use of stateless autoconfiguration. What functions should we expect the IPv6 DHCP server to perform?	

Implementation Plan Peer Review Table

Table 16-12 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 16-12 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
An implementation plan states that router IPv6 address should be assigned as obvious values, using the lowest numbers in the range per each assigned prefix. What configuration methods could be used to configure these low address values?	
A plan calls for the use of stateless autoconfig for client hosts. What must be configured on the routers to support this process?	

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 16-13 all configuration commands related to the configuration of the following features. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 16-13 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure the full global unicast address on an interface.	
Configure the unicast IPv6 prefix on an interface, and let the router add the interface ID.	
Configure an interface to find its unicast IPv6 address using stateless autoconfig.	
Configure an interface to enable IPv6 and use another interface's IPv6 address as needed.	
Enable IPv6 on an interface and do not configure a unicast IPv6 address.	
Configure the link local address of an interface.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 16-14 all commands that supply the requested information. You may want to record your answers outside the book, and set a goal to be able to complete this table (and others like it) from memory during your final reviews before taking the exam.

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Table 16-14 *Verification Plan Memory Drill*

Information Needed	Commands
All IPv6 routes	
A single line per IPv6 address	
Detailed information about IPv6 on an interface, including multicast addresses	
The MAC address used by an interface	
The MAC addresses of neighboring IPv6 hosts	
The information learned from another router in an RA message	

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 16-15 lists a reference of these key topics and the page numbers on which each is found.

Table 16-15 *Key Topics for Chapter 16*

Key Topic Element	Description	Page Number
Figure 16-1	Conceptual view of IPv6 global routes	535
List	Rules for abbreviating IPv6 addresses	536
List	Rules about how to write IPv6 prefixes	538
Figure 16-3	IPv6 public prefix assignment concepts	540
List	IPv6 subnetting process	541
Figure 16-5	IPv6 subnetting concepts	542
List	Three steps used by the stateless autoconfig feature	545
Figure 16-8	IPv6 Address format when using EUI-64	548
Table 16-7	Comparisons of Stateful and Stateless DHCP	549
List	IPv6 address types (unicast, multicast, and anycast)	550
Figure 16-10	Link local address format	552
Table 16-9	Address types and prefixes	552
Figure 16-11	NDP concepts	555

Complete the Tables and Lists from Memory

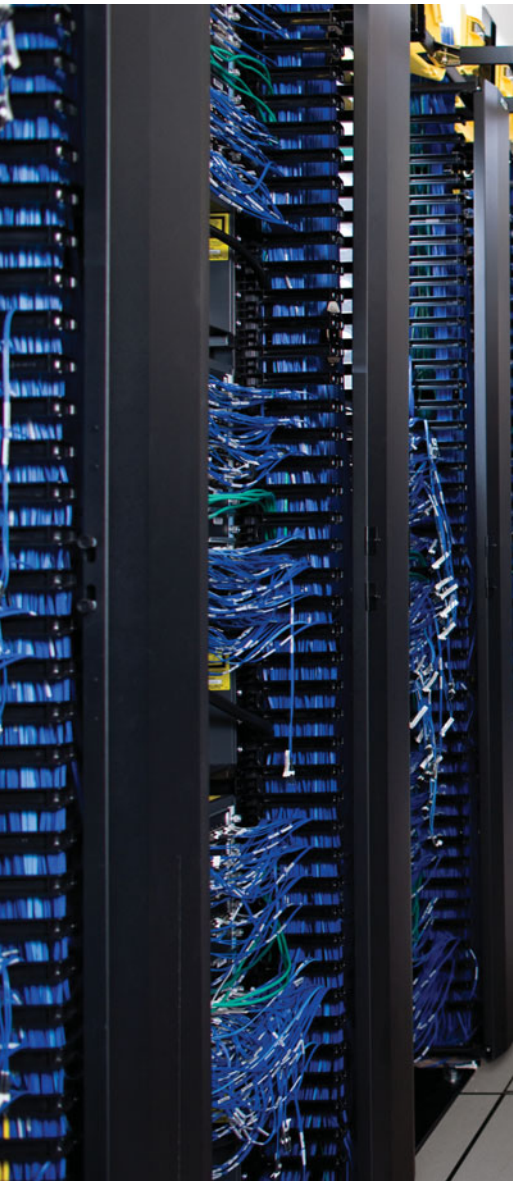
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

global unicast address, link local address, unique local address, stateful DHCP, stateless DHCP, stateless autoconfig, Neighbor Discovery Protocol (NDP), Neighbor Solicitation (NS), Neighbor Advertisement (NA), Router Solicitation (RS), Router Advertisement (RA), solicited node multicast address, duplicate address detection (DAD), inverse neighbor discovery

This page intentionally left blank



This chapter covers the following subjects:

RIP Next Generation (RIPng): This section compares and contrasts IPv4's RIPv2 and IPv6's RIPng routing protocols and shows how to configure RIPng.

EIGRP for IPv6: This section compares and contrasts EIGRP for IPv4 and for IPv6 and shows how to configure EIGRP for IPv6.

OSPF Version 3: This section compares and contrasts OSPFv2, which applies to IPv4, and OSPFv3, which applies to IPv6, along with how to configure OSPFv3.

IPv6 IGP Redistribution: This section compares and contrasts IPv4 and IPv6 IGP redistribution concepts and configuration.

IPv6 Static Routes: This short section shows the syntax of the **ipv6 route** command and how you can use it to configure static IPv6 routes.

CHAPTER 17

IPv6 Routing Protocols and Redistribution

IPv6 uses an updated version of the three popular IGPs (RIP, EIGRP, and OSPF) to exchange routes inside an Enterprise. Additionally, updates to the BGP Version 4 standard, called multiprotocol extensions for BGP-4 (RFC 4760), allow the exchange of IPv6 routing information in the Internet.

As you might imagine, these updated routing protocols have many features in common with their IPv4 cousins. This chapter examines each of the three IPv6 IGPs, with an eye toward the similarities and differences between the IPv4 and IPv6 versions of each IGP protocol. This chapter also examines IPv6 route redistribution between IPv6 IGPs and static IPv6 routes.

Note that as in Chapter 16, "IP Version 6 Addressing," this chapter also discusses the issues of native support of IPv6, ignoring the transition and translation features discussed in Chapter 18, "IPv4 and IPv6 Coexistence."

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these eight self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 17-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 17-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
RIP Next Generation (RIPng)	1–2
EIGRP for IPv6	3–4
OSPF Version 3	5–6
IPv6 IGP Redistribution	7
IPv6 Static Routes	8

1. Which of the following features work the same in both RIP-2 and RIPng? (Choose three.)

    a. Distance Vector Logic
    b. Uses UDP
    c. Uses RIP-specific authentication
    d. Maximum useful metric of 15
    e. Automatic route summarization

2. Router R1 currently has no configuration related to IPv6 or IPv4. The following configuration exists in a planning document, intended to be used to copy/paste into Router R1 to enable RIPng and IPv6 on interfaces F0/0 and S0/0/0. No other related configuration exists. Which of the following is true about RIPng on R1 after this configuration has been pasted into R1?

```
ipv6 unicast-routing
interface f0/0
ipv6 rip one enable
ipv6 address 2000::1/64
interface s0/0/0
ipv6 address 2001::/64 eui-64
ipv6 rip one enable
```

    a. RIPng will be enabled on no interfaces.
    b. RIPng will be enabled on one interface.
    c. RIPng will be enabled on two interfaces.
    d. RIPng will advertise about prefixes connected to S0/0/0 and F0/0, but only send Updates on one interface.

3. Router R1 currently has no configuration related to IPv6 or IPv4. The following configuration exists in a planning document intended to be used to copy/paste into Router R1 to enable EIGRP for IPv6 on interfaces F0/0 and S0/0/0. No other related configuration exists. Assuming F0/0 and S0/0/0 reach an up/up state, which of the following is true about EIGRP for IPv6 on R1 after this configuration has been pasted into R1?

```
ipv6 router eigrp 1
ipv6 unicast-routing
interface f0/0
ipv6 address 2000::1/64
ipv6 eigrp 1
interface s0/0/0
ipv6 address 2001::/64 eui-64
ipv6 eigrp 1
```

   a. EIGRP works on F0/0 and S0/0/0 without further configuration.
   b. EIGRP works with the addition of one command: a **no shutdown** command in EIGRP router configuration mode.
   c. EIGRP works with the addition of one command: an **eigrp router-id** command in EIGRP router configuration mode.
   d. EIGRP for IPv6 needs at least two more configuration commands before it works on R1.

4. Router R1 connects to Router R2 over an Ethernet LAN with both routers using their F0/0 interfaces. R1 learns a route from R2 using EIGRP for IPv6. That route lists F0/0 as the outgoing interface with R2 as the next hop. The configuration excerpt shows all relevant configuration on R2's F0/0 interface. Which of the following is true about R1's route?

```
interface f0/0
mac-address 1111.1111.1111
ipv6 address 2000::/64 eui-64
ipv6 address 2001::1/64
```

   a. The next hop is 2000::1311:11FF:FE11:1111
   b. The next hop is FE80::1311:11FF:FE11:1111
   c. The next hop is FE80::5111:11FF:FE11:1111
   d. The next hop is 2001::1

5. Which of the following are true of both OSPFv2 and OSPFv3? (Choose two.)

   a. The method of choosing an OSPF router ID
   b. Verification checks that must be validated before two routers can become OSPF neighbors
   c. Support for route tags
   d. Support for multiple instances per interface

6. Router R1 currently has no configuration related to IPv6 or IPv4. The following configuration exists in a planning document, intended to be used to copy/paste into Router R1 to enable OSPFv3 on interfaces F0/0 and S0/0/0. No other related configuration exists. Assuming F0/0 and S0/0/0 reach an up/up state, which of the following is true about OSPFv3 on R1 after this configuration has been pasted into R1?

```
ipv6 router ospf 1
ipv6 unicast-routing
interface f0/0
ipv6 address 2000::1/64
ipv6 ospf 1 area 1
interface s0/0/0
ipv6 address 2001::/64 eui-64
ipv6 ospf 1 area 0
```

   a. OSPF works on F0/0 and S0/0/0 without further configuration.
   b. OSPF works with the addition of one command: a **no shutdown** command in OSPF router configuration mode.
   c. OSPF works with the addition of one command: an **router-id** command in OSPF router configuration mode.
   d. OSPFv3 needs at least two more configuration commands before it works on R1.

7. The following output occurs on Router R1, which runs both EIGRP for IPv6 and OSPFv3, with redistribution from EIGRP into OSPF configured with the **redistribute eigrp 1 metric 25** command. Interface S0/0/1 has been enabled for EIGRP ASN 1. Which of the following should be true of redistribution in this case?

```
D   2000::/64 [90/1422516]
     via FE80::213:19FF:FE7B:5026, Serial0/0/1
C   2000:0:0:1::/64 [0/0]
     via Serial0/0/1, directly connected
L   2000::1:213:19FF:FE7B:5004/128 [0/0]
     via Serial0/0/1, receive
```

   a. Route 2000::/64 will be redistributed.
   b. Route 2000:0:0:1::/64 will be redistributed.
   c. Route 2000:1:213:19FF:FE7B:5004/128 will be redistributed.
   d. No routes will be redistributed because of the omission of the **subnets** parameter of the **redistribute** command.

8. Router R1 has been configured with an **ipv6 route 2000::/64 S0/0/0 64** command. Which of the following does the 64 at the end of the command represent?

   a. Metric
   b. Administrative distance
   c. Timeout (seconds)
   d. Prefix length
   e. Interface ID

Foundation Topics

To support IPv6, all the IPv4 routing protocols had to go through varying degrees of changes, with the most obvious being that each had to be changed to support longer addresses and prefixes. The actual messages used to send and receive routing information have changed in some cases, using IPv6 headers instead of IPv4 headers, and using IPv6 addresses in those headers. In particular, like their IPv4 versions, each IPv6 IGP uses IPv6 multicast addresses—for example, RIPng sends routing updates to the IPv6 destination address FF02::9 instead of the old RIP-2 IPv4 224.0.0.9 address. Also, the routing protocols typically advertise their link-local IP address as the next hop in a route.

Even with these changes, each IPv6 IGP has more similarities than differences compared to their respective IPv4 cousins. For example, RIPng, based on RIP-2, is still a Distance Vector protocol, with hop count as the metric and 15 hops as the longest valid route (16 is infinity). OSPF Version 3 (OSPFv3), created specifically to support IPv6, uses Link State logic like OSPFv2, uses cost as the metric, and retains the LSA types–but there are some changes to how the LSAs work. However, most of the core OSPF operational concepts remain the same.

Table 17-2 lists the IPv6 routing protocols and their new RFCs (as appropriate).

Table 17-2 *Updates to Routing Protocols for IPv6*

Routing Protocol	Full Name	RFC
RIPng	RIP Next Generation	2080
OSPFv3	OSPF Version 3	5340
MP-BGP4	Multiprotocol BGP-4	4760
EIGRP for IPv6	EIGRP for IPv6	Proprietary

This chapter examines the three IGPs, the redistribution between the IGPs, and static IPv6 routes.

RIP Next Generation (RIPng)

Routing Information Protocol (RIP) began life as one of the earliest efforts in the field of dynamic IP routing protocols. It eventually became the first dynamic routing protocol for the emerging IP protocol back in the 1970s. Later, in the mid-1990s, the RIP Version 2 (RIP-2) specifications enhanced RIP, with the original version becoming known as RIP Version 1, or simply RIP-1.

Also in the mid-1990s, the process of defining IPv6 was drawing toward completion, at least for the original IPv6 standards. To support IPv6, the IETF committees defined a new version of RIP to support IPv6. But rather than number this updated flavor of RIP as RIP Version 3, the creators chose to number this new protocol as version 1, treating it like a new protocol. However, no one bothered to put "version 1" in the name, simply calling it

RIP Next Generation (RIPng), or even simply RIP. To date, no new version of RIPng has been defined, making the original RIPng still the most recent version of the protocol.

Note: For you Star Trek TV show fans, yes, the name in part came from "Star Trek: The Next Generation."

RIPng–Theory and Comparisons to RIP-2

The RIPng RFC states that the protocol uses many of the same concepts and conventions as the original RIP-1 specification, also drawing on some RIP-2 concepts. However, knowing that many of you might not remember a lot of details about RIP-2, particularly because RIP-2 is included in the CCNA certification rather than CCNP, Table 17-3 lists a variety of facts about RIP-2 and RIPng.

Key Topic

Table 17-3 *Comparing RIP-2 to RIPng*

Feature	RIP-2	RIPng
Advertises routes for...	IPv4	IPv6
RIP messages use these Layer 3/4 protocols	IPv4, UDP	IPv6, UDP
UDP Port	520	521
Use Distance Vector	Yes	Yes
Default Administrative distance	120	120
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses Split Horizon	Yes	Yes
Uses Poison Reverse	Yes	Yes
30 second periodic full updates	Yes	Yes
Uses triggered updates	Yes	Yes
Uses Hop Count metric	Yes	Yes
Metric meaning infinity	16	16
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.9	FF02::9
Authentication	RIP-specific	uses IPv6 AH/ESP

The overall operation of RIPng closely matches RIP-2. In both, routers send periodic full updates with all routes, except for routes omitted due to Split Horizon rules. No neighbor relationships occur; the continuing periodic Updates, on a slightly-variable 30 second period, also serve the purpose of confirming that the neighboring router still works. The

metrics work exactly the same. When a router ceases to see a route in received updates, ceases to receive updates, or receives a poisoned (metric 16) route, it reacts to converge, but relatively slowly compared to EIGRP and OSPF.

Some differences relate specifically to IPv6. First, the messages themselves list IPv6 prefixes/lengths, rather than subnet/mask. In RIP-1 and RIP-2, RIP encapsulated RIP Update messages inside an IPv4 and UDP header; with IPv6, the encapsulation uses IPv6 packets, again with a UDP header. Some small differences in the Update message format exist as well, with the most obvious difference being that the updates list IPv6 prefixes and prefix lengths.

The last difference of note is that because IPv6 supports authentication using the IPsec Authentication Header (AH), RIPng does not natively support authentication, instead relying on IPsec.

Configuring RIPng

RIPng uses a new command style for the basic configuration, but most of the optional features and verification commands look much like the commands used for RIP for IPv4. This section first takes a look at the basic RIPng configuration, accepting as many defaults as possible.

The big difference between RIP-2 and RIPng configuration is that RIPng discards the age-old RIP **network** command in deference to the **ipv6 rip** *name* **enable** interface subcommand, which enables RIPng on the interface. Another difference relates to the routing of IPv4 and IPv6: IOS routes IPv4 by default (due to a default global configuration command of **ip routing**), but IOS does not route IPv6 by default (a default of **no ipv6 unicast routing**). Finally, RIPng allows multiple RIPng processes on a single router, so IOS requires that each RIPng process is given a text name that identifies each RIPng process for that one router–another difference compared to RIP-2.

The following list shows the basic configuration steps for RIPng, including steps to enable IPv6 routing and enabling IPv6 on the interfaces.

Step 1. Enable IPv6 routing with the **ipv6 unicast-routing** global command.

Step 2. Enable RIPng using the **ipv6 router rip** *name* global configuration command. The name must be unique on a router but does not need to match neighboring routers.

Step 3. Enable IPv6 on the interface, typically with one of these two methods:

`Configure an IPv6 unicast address on each interface using the` **`ipv6 address`** *`address/prefix-length`* `[`**`eui-64`**`] interface command.`
`Configure the` **`ipv6 enable`** `command, which enables IPv6 and causes the router to derive its link local address.`

Step 4. Enable RIP on the interface with the **ipv6 rip** *name* **enable** interface subcommand (where the name matches the **ipv6 router rip** *name* global configuration command).

Note: For a complete discussion of options for enabling IPv6 on an interface, as mentioned at Step 3, refer to Chapter 16's Table 16-11.

The list includes just a few straightforward configuration commands, but a few subtle interactions also exist. The list shows steps related directly to RIPng (Steps 2 and 4), plus other steps related to making IPv6 itself work (Steps 1 and 3). The list also pairs two sets of dependent steps with each other, as follows:

- Step 2 relies on Step 1 because IOS rejects the command at Step 2 (**ipv6 router rip** *name*) if the command at Step 1 (**ipv6 unicast-routing**) has been omitted.
- Step 4 relies on Step 3 because IOS rejects the command at Step 4 if IPv6 has yet been enabled on the interface, either statically as mentioned at Step 3 or with one of the other methods listed in Chapter 16's Table 16-11.

Finally, note that although the **ipv6 rip** *process-name* **enable** interface subcommand (Step 4) refers to the process name configured at Step 2 (the **ipv6 router rip** *process-name* command), IOS creates the RIP process in reaction to the **ipv6 rip** *process-name* **enable** interface subcommand if that RIPng process name does not yet exist. In other words, if you followed the previous steps in order, but forgot to do Step 2, the command at Step 4 causes IOS to automatically create the command at Step 2.

As with RIP-1 and RIP-2, for any interface on which RIPng has been enabled, the RIP process does three main actions. First, it starts sending RIP updates on that interface. It also starts processing any RIP updates received on that interface. Finally, it advertises about the connected routes on that interface. In particular, because IPv6 allows the configuration of multiple IPv6 unicast addresses on an interface, RIP advertises about most IPv6 unicast prefixes associated with the interface. The notable exceptions are that RIP does not advertise about any link local addresses, nor does RIP advertise about the local host routes–routes with a /128 prefix length–created for each interface IPv6 address. In short, RIP advertises all routable subnets associated with the interface.

Figure 17-1 shows a sample internetwork with IPv6 global unicast IPv6 subnets shown. The sample internetwork uses addressing values that are both memorable and make for shorter IPv6 addresses when abbreviated. All the subnets use /64 prefix length, with quartets 2, 3, and 4 composed of all 0 values. The interface ID portion of each address uses all hex 0's in the first three quartets (quartets 5, 6, and 7 in the overall address), with the final digit in the final quartet used to identify each router. This last digit matches the name of each router in most cases.

For example, all R1's IPv6 addresses' last four octets are 0000:0000:0000:0001. R1's S0/0/0.3 subinterface, which connects with a PVC to Router R3, uses a prefix of 2003:0000:0000:0000::/64, making the entire IPv6 address on this interface, when abbreviated, 2003::1/64–a convenient value for sifting through all the output in the upcoming examples.

Example 17-1 shows the RIPng configuration on Router R1 in this design. The RIP process name is fred.

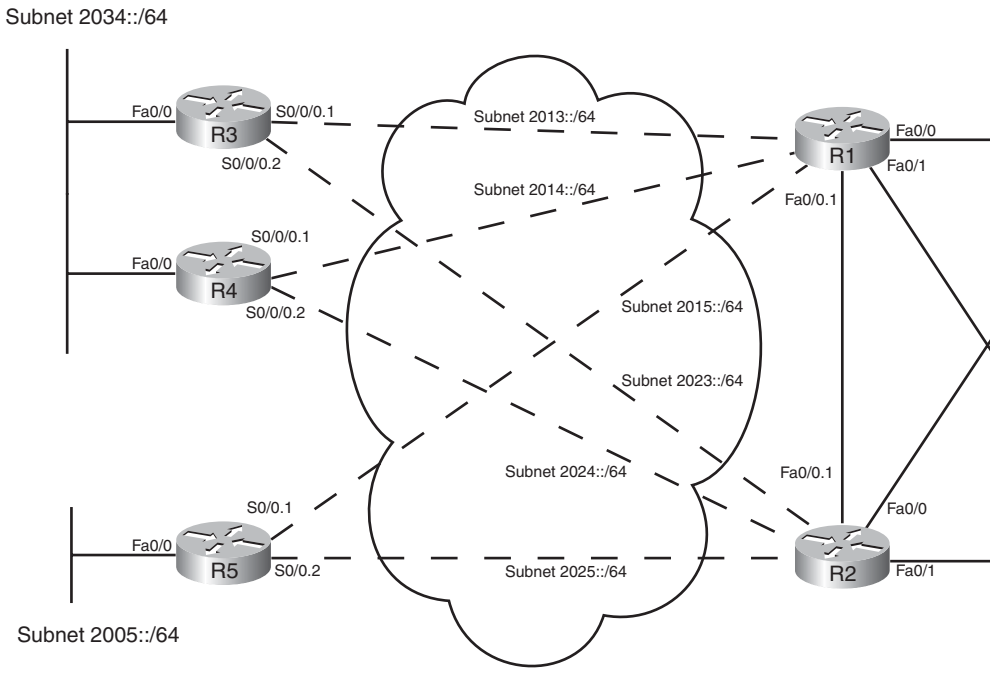


Figure 17-1 *Sample Internetwork for IPv6 Routing Protocol Configuration*

Example 17-1 *Configuring IPv6 Routing and Routing Protocols on R1*

```
R1# show running-config
! The output is edited to remove lines not pertinent to this example.
! Next, step 1's task: enable IPv6 routing
ipv6 unicast-routing
!
! Next, on 5 interfaces, steps 3 and 4: configuring an IPv6 address,
! and enable RIPng, process "fred".
interface FastEthernet0/0.1
 ipv6 address 2012::1/64
 ipv6 rip fred enable
!
interface FastEthernet0/0.2
 ipv6 address 2017::1/64
 ipv6 rip fred enable
!
interface FastEthernet0/1.18
 ipv6 address 2018::1/64
 ipv6 rip fred enable
!
interface Serial0/0/0.3
 ipv6 address 2013::1/64
```

```
 ipv6 rip fred enable
!
interface Serial0/0/0.4
 ipv6 address 2014::1/64
 ipv6 rip fred enable
!
interface Serial0/0/0.5
 ipv6 address 2015::1/64
 ipv6 rip fred enable
!
! Next, step 2's task, creating the RIPng process named "fred"
ipv6 router rip fred
```

Verifying RIPng

The **show** commands related to RIPng have the same general kinds of information as seen with RIP-2. However, some of the commands used to get to the same piece of information differ, and of course, some obvious differences exist due to the different IPv6 address structure. Table 17-4 lists a cross-reference comparing all commands related to RIP that begin either **show ip** or **show ipv6**. It also lists the similar **debug** commands used to display RIP routing information.

Table 17-4 *Comparing Verification Commands:* **show ip** *and* **show ipv6**

Function	IPv4	IPv6
All routes	**... route**	**... route**
All RIP learned routes	**... route rip**	**... route rip**
Details on the routes for a specific prefix	**... route** *subnet mask*	**... route** *prefix/length*
Interfaces on which RIP is enabled	**... protocols**	**... protocols**
List of routing information sources	**... protocols**	**... rip next-hops**
Debug that displays sent and received Updates	**debug ip rip**	**debug ipv6 rip**

The most notable differences occur with the information seen with IPv4 in the **show ip protocols** command. The **show ip protocols** command displays a wide variety of information for IPv4 RIP, whereas the IPv6 commands spread the information over a couple of different commands, as listed in Table 17-4. Example 17-2 shows a sampling of the commands, taken from Router R3 in Figure 17-1. The explanatory comments are listed within the example in this case. Note that Router R3 used a RIPng process name of barney.

Example 17-2 *IPv6 RIPng* **show** *Commands*

```
! On R3, process name "barney" has two current routes to reach the
! datacenter prefix 2099::/64.

R3# show ipv6 route 2099::/64
Routing entry for 2099::/64
  Known via "rip barney", distance 120, metric 3
  Route count is 2/2, share count 0
  Routing paths:
    FE80::22FF:FE22:2222, Serial0/0/0.2
      Last updated 00:27:12 ago
    FE80::11FF:FE11:1111, Serial0/0/0.1
      Last updated 00:27:10 ago

! Note that the next command lists only RIP-learned routes. It lists
! two next-hops for 2099::64. Note the next-hop information lists
! link-local addresses.
R3# show ipv6 route rip
IPv6 Routing Table - Default - 19 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2005::/64 [120/3]
     via FE80::11FF:FE11:1111, Serial0/0/0.1
     via FE80::22FF:FE22:2222, Serial0/0/0.2
R   2012::/64 [120/2]
     via FE80::11FF:FE11:1111, Serial0/0/0.1
     via FE80::22FF:FE22:2222, Serial0/0/0.2

! lines omitted for brevity...
R   2099::/64 [120/3]
     via FE80::22FF:FE22:2222, Serial0/0/0.2
     via FE80::11FF:FE11:1111, Serial0/0/0.1

! Unlike show ip protocols, show ipv6 protocols displays little info.
R3# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "rip barney"
  Interfaces:
    Serial0/0/0.2
    Serial0/0/0.1
    FastEthernet0/0
```

```
  Redistribution:
    None

! This command lists the timers displayed for RIP-2 with show ip protocols.
R3# show ipv6 rip
RIP process "barney", port 521, multicast-group FF02::9, pid 258
     Administrative distance is 120. Maximum paths is 16
     Updates every 30 seconds, expire after 180
     Holddown lasts 0 seconds, garbage collect after 120
     Split horizon is on; poison reverse is off
     Default routes are not generated
     Periodic updates 57, trigger updates 10
  Interfaces:
    Serial0/0/0.2
    Serial0/0/0.1
    FastEthernet0/0
  Redistribution:
    None

! This command lists the equivalent of the information in the
! show ip protocols commands' "Routing Information Sources" heading.
! Note the link local addresses are listed.
R3# show ipv6 rip next-hops
 RIP process "barney", Next Hops
  FE80::11FF:FE11:1111/Serial0/0/0.1 [9 paths]
  FE80::44FF:FE44:4444/FastEthernet0/0 [3 paths]
  FE80::22FF:FE22:2222/Serial0/0/0.2 [9 paths]
```

Beyond the information emphasized in the comments inside the example, the next-hop IPv6 addresses in the example need to be scrutinized. RIPng uses the link local IPv6 address as the next-hop IP address. (Reminder: link local addresses begin with FE80.) The **show ipv6 route 2099::/64** and **show ipv6 route** commands early in the example, and the **show ipv6 rip next-hops** command at the end of the example, all show next-hop IP addresses that begin with FE80, confirming that RIP indeed uses link local addresses as next-hop addresses.

To discover which routers use which link local addresses, and make it easier to work with link local addresses, you have a couple of options. First, you can set the MAC address of each LAN interface to something noticeable. For Example 17-2, the routers each used a recognizable MAC: R1 used 0200.1111.1111, R2 used 0200.2222.2222, and so on. Alternatively, you can just configure the link local address with the **ipv6 address** command, using the **link-local** keyword at the end, and make each link local address be more recognizable. Regardless, to find the router whose link local address is listed in the IPv6 routing table, the **show cdp entry** *name* command can be useful because it lists both the IPv4 and IPv6 addresses, including the neighbor's link local address.

EIGRP for IPv6

Cisco originally created EIGRP to advertise routes for IPv4, IPX, and AppleTalk. This original EIGRP architecture easily allowed for yet another Layer 3 protocol, IPv6, to be added. As a result, Cisco did not have to change EIGRP significantly to support IPv6, so many similarities exist between the IPv4 and IPv6 versions of EIGRP.

Note: Many documents, including this chapter, refer to the IPv6 version of EIGRP as EIGRP for IPv6. However, some documents at www.cisco.com also refer to this protocol as EIGRPv6, not because it is the sixth version of the protocol, but because it implies a relationship with IPv6.

As with the previous section "RIP Next Generation (RIPng)," this section begins with a discussion of the similarities and differences between the IPv4 and IPv6 versions of EIGRP. The remaining coverage of EIGRP focuses on the changes to EIGRP configuration and verification in support of IPv6.

EIGRP for IPv4 and IPv6–Theory and Comparisons

For the most part, EIGRP for IPv4 and for IPv6 have many similarities. The following list outlines some of the key differences:

- EIGRP for IPv6 advertises IPv6 prefixes/lengths, rather than IPv4 subnet/mask information.
- EIGRP for IPv6 uses the neighbor's link local address as the next-hop IP address; EIGRP for IPv4 has no equivalent concept.
- EIGRP for IPv6 encapsulates its messages in IPv6 packets, rather than IPv4 packets.
- Like RIPng and OSPFv3, EIGRP for IPv6 authentication relies on IPv6's built-in authentication and privacy features.
- EIGRP for IPv4 defaults to use automatic route summarization at the boundaries of classful IPv4 networks; IPv6 has no concept of classful networks, so EIGRP for IPv6 cannot perform any automatic summarization.
- EIGRP for IPv6 does not require neighbors to be in the same IPv6 subnet as a requirement to become neighbors.

Other than these differences, most of the details of EIGRP for IPv6 works like EIGRP for IPv4. For reference, Table 17-5 compares the features of each.

Table 17-5 *Comparing EIGRP for IPv4 and IPv6*

Feature	EIGRP for IPv4	EIGRP for IPv6
Advertises routes for...	IPv4	IPv6
Layer 3 protocol for EIGRP messages	IPv4	IPv6

Table 17-5 *Comparing EIGRP for IPv4 and IPv6*

Feature	EIGRP for IPv4	EIGRP for IPv6
Layer 3 header protocol type	88	88
UDP Port	N/A	N/A
Uses Successor, Feasible Successor logic	Yes	Yes
Uses Dual	Yes	Yes
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses triggered updates	Yes	Yes
Uses composite metric, default using bandwidth and delay	Yes	Yes
Metric meaning infinity	$2^{32} - 1$	$2^{32} - 1$
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.10	FF02::10
Authentication	EIGRP-specific	Uses IPv6 AH/ESP

Configuring EIGRP for IPv6

EIGRP for IPv6 follows the same basic configuration style as for RIPng, plus a few additional steps, as follows:

Key Topic

Step 1. Enable IPv6 routing with the **ipv6 unicast-routing** global command.

Step 2. Enable EIGRP using the **ipv6 router eigrp** *{1 – 65535}* global configuration command.

Step 3. Enable IPv6 on the interface, typically with one of these two methods:

```
Configure an IPv6 unicast address on each interface, using the ipv6
address address/prefix-length [eui-64] interface command.
Configure the ipv6 enable command, which enables IPv6 and causes the
router to derive its link local address.
```

Step 4. Enable EIGRP on the interface with the **ipv6 eigrp** *asn* interface subcommand (where the name matches the **ipv6 router eigrp** *asn* global configuration command).

Step 5. Enable EIGRP for IPv6 with a **no shutdown** command while in EIGRP configuration mode.

Step 6. If no EIGRP router ID has been automatically chosen, due to not having at least one working interface with an *IPv4* address, configure an EIGRP router ID with the **eigrp router-id** *rid* command in EIGRP configuration mode.

Note: For a complete discussion of options for enabling IPv6 on an interface, as mentioned at Step 3, refer to Chapter 16's Table 16-11.

The first four steps essentially mirror the four steps in the RIPng configuration process previously listed in this chapter. The same interdependencies exist for EIGRP for IPv6 as well; in particular, the command at Step 2 works only if Step 1's command has been configured, and the command at Step 4 fails if the command at Step 3 has not yet been completed.

EIGRP for IPv6 requires Steps 5 and 6, whereas RIPng does not need equivalent steps. First, at Step 5, IOS supports the ability to stop and start the EIGRP process with the **shutdown** and **no shutdown** router mode subcommands. After initial configuration, the EIGRP for IPv6 process starts in shutdown mode, so to make the process start, IOS requires a **no shutdown** command.

Step 6 shows the other difference as compared to RIPng configuration, but this step may or may not be needed. The EIGRP for IPv6 process must have a router ID (RID) before the process works. EIGRP for IPv6 uses the same process as EIGRP for IPv4 for choosing the RID. The EIGRP for IPv6 RID is indeed a 32-bit number, with two of the EIGRP RID decision steps based on IPv4 configuration. The following list defines how EIGRP for IPv6 picks its RID, listed in the order of preference:

Step 1. Use the configured value (using the **eigrp router-id** *a.b.c.d* EIGRP subcommand under the **ipv6 router eigrp** command).

Step 2. Use the highest IPv4 address on an up/up loopback interface.

Step 3. Use the highest IPv4 address on an up/up nonloopback interface.

Note that although most installations already have IPv4 addresses configured, it is possible that the EIGRP for IPv6 process cannot derive a RID value. If the router has no working interfaces that have IPv4 addresses, and the EIGRP for IPv6 RID is not explicitly configured, then the EIGRP for IPv6 process simply does not work. So, the six-step configuration process includes a mention of the EIGRP RID; more generally, it may be prudent to configure a RID explicitly as a matter of habit.

After being enabled on an interface, EIGRP for IPv6 performs the same two basic tasks as it does with EIGRP for IPv4: it discovers neighbors and advertises about connected subnets. EIGRP for IPv6 uses the same set of neighbor checks as do routers using EIGRP for IPv4, except that EIGRP for IPv6 does not require that neighboring IPv6 routers have IPv6 addresses in the same subnet. (See Table 2-4 in Chapter 2, "EIGRP Overview and Neighbor Relationships," for the list of neighbor validations performed by EIGRP for IPv4). Also, as with EIGRP for IPv4, EIGRP for IPv6 advertises about any and all connected subnets on the interface, with the exception of the link local addresses and the local routes (the host routes for a router's own interface IPv6 addresses).

Example 17-3 shows a sample configuration on Router R1 from Figure 17-1. All neighboring routers must use the same ASN; ASN 9 will be used in this case.

Example 17-3 *Configuring EIGRP for IPv6 Routing and Routing Protocols on R1*

```
R1# show running-config
! output is edited to remove lines not pertinent to this example
! Configuration step 1: enabling IPv6 routing
ipv6 unicast-routing

! Next, configuration steps 3 and 4, on 5 different interfaces
interface FastEthernet0/0.1
 ipv6 address 2012::1/64
 ipv6 eigrp 9
!
interface FastEthernet0/0.2
 ipv6 address 2017::1/64
 ipv6 eigrp 9
!
interface FastEthernet0/1.18
 ipv6 address 2018::1/64
 ipv6 eigrp 9
!
interface Serial0/0/0.3
 ipv6 address 2013::1/64
 ipv6 eigrp 9
!
interface Serial0/0/0.4
 ipv6 address 2014::1/64
 ipv6 eigrp 9
!
interface Serial0/0/0.5
 ipv6 address 2015::1/64
 ipv6 eigrp 9
!
! Configuration steps 2, 5, and 6
ipv6 router eigrp 9
 no shutdown
 router eigrp 10.10.34.3
```

Verifying EIGRP for IPv6

The EIGRP for IPv6 **show** commands generally list the same kinds of information as the equivalent commands for EIGRP for IPv4, even more so than RIPng. In most cases, simply use the same **show ip...** commands applicable with IPv4 and EIGRP, and substitute **ipv6** instead of **ip**. Table 17-6 lists a cross-reference comparing the most popular EIGRP-related commands for both versions. Note that the table assumes that the commands begin either **show ip** or **show ipv6** in all but the last row of the table.

Table 17-6 *Comparing EIGRP Verification Commands:* **show ip** *and* **show ipv6...**

Function	show ip...	show ipv6...
All routes	... route	... route
All EIGRP learned routes	... route eigrp	... route eigrp
Details on the routes for a specific prefix	... route *subnet mask*	... route *prefix/length*
Interfaces on which EIGRP is enabled, plus metric weights, variance, redistribution, max-paths, admin distance	... protocols	... protocols
List of routing information sources	... protocols ... eigrp neighbors	... eigrp neighbors
Hello interval	... eigrp interfaces detail	... eigrp interfaces detail
EIGRP database	... eigrp topology [all-links]	... eigrp topology [all-links]
Debug that displays sent and received Updates	debug ip eigrp notifications	debug ipv6 eigrp notifications

Example 17-4 shows a few sample **show** commands taken from Router R3 in the internetwork shown in Figure 17-1. The explanatory comments are listed within the example in this case.

Example 17-4 *IPv6 EIGRP* **show** *Commands*

```
! On R3, as when using RIPng, the next-hop address is the
! link-local address of the next router.

R3# show ipv6 route 2099::/64
Routing entry for 2099::/64
  Known via "eigrp 9", distance 90, metric 2174976, type internal
  Route count is 2/2, share count 0
  Routing paths:
    FE80::22FF:FE22:2222, Serial0/0/0.2
      Last updated 00:24:32 ago
    FE80::11FF:FE11:1111, Serial0/0/0.1
      Last updated 00:07:51 ago

! Note that the next command lists only EIGRP-learned routes. It lists
! two next-hops for 2099::64. Note the next-hop information lists
! link-local addresses.
R3# show ipv6 route eigrp
IPv6 Routing Table - Default - 19 entries
```

```
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D   2005::/64 [90/2684416]
     via FE80::11FF:FE11:1111, Serial0/0/0.1
     via FE80::22FF:FE22:2222, Serial0/0/0.2
D   2012::/64 [90/2172416]
     via FE80::22FF:FE22:2222, Serial0/0/0.2
     via FE80::11FF:FE11:1111, Serial0/0/0.1
D   2014::/64 [90/2681856]
     via FE80::11FF:FE11:1111, Serial0/0/0.1
D   2015::/64 [90/2681856]
     via FE80::11FF:FE11:1111, Serial0/0/0.1

! lines omitted for brevity...
D   2099::/64 [90/2174976]
     via FE80::22FF:FE22:2222, Serial0/0/0.2
     via FE80::11FF:FE11:1111, Serial0/0/0.1

! show ipv6 protocols displays less info than its IPv4 cousin.
R3# show ipv6 protocols
IPv6 Routing Protocol is "eigrp 9"
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Interfaces:
    FastEthernet0/0
    Serial0/0/0.1
    Serial0/0/0.2
  Redistribution:
    None
  Maximum path: 16
  Distance: internal 90 external 170

! This command lists the equivalent of the information in the
! show ip protocols commands' "Routing Information Sources" heading.
! Note the link local addresses are listed.
R3# show ipv6 eigrp neighbors
IPv6-EIGRP neighbors for process 9
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                              (sec)         (ms)       Cnt Num
1   Link-local address:     Se0/0/0.2         14 01:50:51    3   200  0  82
```

```
    FE80::22FF:FE22:2222
0   Link-local address:     Se0/0/0.1          13 01:50:52   14   200  0  90
    FE80::11FF:FE11:1111

! The next command lists the EIGRP topology database, including
! feasible distance calculations, reported distance, and listing
! all successor and feasible successor routes.
R3# show ipv6 eigrp topology
IPv6-EIGRP Topology Table for AS(9)/ID(10.10.34.3)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2005::/64, 2 successors, FD is 2684416
         via FE80::11FF:FE11:1111 (2684416/2172416), Serial0/0/0.1
         via FE80::22FF:FE22:2222 (2684416/2172416), Serial0/0/0.2
P 2012::/64, 2 successors, FD is 2172416
         via FE80::11FF:FE11:1111 (2172416/28160), Serial0/0/0.1
         via FE80::22FF:FE22:2222 (2172416/28160), Serial0/0/0.2
P 2013::/64, 1 successors, FD is 2169856
         via Connected, Serial0/0/0.1

! lines omitted for brevity
P 2099::/64, 2 successors, FD is 2174976
         via FE80::11FF:FE11:1111 (2174976/30720), Serial0/0/0.1
         via FE80::22FF:FE22:2222 (2174976/30720), Serial0/0/0.2

! Finally, the link-local address of neighbor R1 is identified.
R3# show cdp entry R1
-------------------
Device ID: R1
Entry address(es):
  IP address: 10.10.13.1
  IPv6 address: 2013::1  (global unicast)
  IPv6 address: FE80::11FF:FE11:1111  (link-local)
Platform: Cisco 1841,  Capabilities: Router Switch IGMP
Interface: Serial0/0/0.1,  Port ID (outgoing port): Serial0/0/0.3
! lines omitted for brevity
```

The most notable fact listed in the example is that the output confirms that little difference exists with the **show** commands for EIGRP for IPv4 versus IPv6. The main differences relate to the **show ip protocols/show ipv6 protocols** commands and that EIGRP for IPv6 uses a link-local IP address for the next hop of each route.

OSPF Version 3

The original OSPF, created to exchange routing information for IPv4, began life as Version 1 and was later enhanced as Version 2. The original OSPFv2 RFC (1247) made draft standard status in 1991. Although some changes and additions have occurred in OSPFv2 over the years, the OSPF many companies have used for a long time is OSPFv2.

To support IPv6, an IETF working group took the OSPFv2 standard and made changes to the protocol to support IPv6, resulting in the new protocol named OSPF Version 3 (OSPFv3). OSPFv3 does advertises IPv6 prefixes, but it does not advertise IPv4 prefixes, so in most OSPF shops that begin migration to IPv6, the routers run OSPFv2 for IPv4 support and OSPFv3 for IPv6 support, just like a network running EIGRP for IPv4 would then add configuration for EIGRP for IPv6.

As with the earlier sections about RIPng and EIGRP for IPv6, this section first examines the similarities and differences between the IPv6 version of OSPF and the earlier IPv4 version. It then looks at the OSPFv3 configuration and finally OSPFv3 verification.

Comparing OSPFv2 and OSPFv3

Because OSPFv2 should already be somewhat familiar to you after reading Part 3 of this book, this section summarizes the similarities and differences, and then explains some of the differences briefly. To that end, Table 17-7 summarizes a wide variety of OSPF features, comparing OSPFv2 and OSPFv3.

Key Topic

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
Advertises routes for...	IPv4	IPv6
OSPF messages use this layer 3 protocol	IPv4	IPv6
IP Protocol Type	89	89
Uses Link State logic	Yes	yes
Supports VLSM	Yes	Yes
Process to choose RID, compared to OSPFv2	Same	Same
LSA flooding and aging compared to OSPFv2	Same	Same
Area structure compared to OSPFv2	Same	Same
Packet types and uses compared to OSPFv3 (Table 6-4)	Same	Same
LSA flooding and aging compared to OSPFv2	Same	Same
RID based on highest up/up loopback IPv4 address, or highest other IPv4 interface address?	Yes	Yes
32-bit LSID	Yes	Yes

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
Uses interface cost metric, derived from interface bandwidth	Yes	Yes
Metric meaning infinity	$2^{16} - 1$	$2^{16} - 1$
Supports route tags	Yes	Yes
Elects DR based on highest priority, then highest RID	Yes	Yes
Periodic reflooding every...	30 minutes	30 minutes
Multicast–all SPF routers	224.0.0.5	FF02::5
Multicast–All Designated routers	224.0.0.6	FF02::6
Authentication	OSPF-specific	Uses IPv6 AH/ESP
Neighbor checks compared to OSPFv2 (table 5-5)	Same	Same, except no "same subnet" check
Multiple instances per interface	No	Yes

Some of the comparisons in the table require further explanation:

- The multicast addresses used by OSPFv3 of course differ, but they keep similar numbers compared to OSPFv2.
- OSPF uses IPv6's inherent IPsec capabilities, rather than defining a separate authentication process.
- As with RIPng and EIGRP for IPv6, OSPFv3 does not require neighboring routers to be in the same subnet as a requirement for becoming neighbors. However, OSPFv3 does follow all other neighbor verification checks as compared with OSPFv2, as summarized in Table 5-5 in Chapter 5, "OSPF Overview and Neighbor Relationships."
- Support for multiple *instances* of OSPF on a single link, whereas OSPFv2 supports only a single instance per link.
- OSPF uses the neighbor's link local IPv6 address as the next-hop IP address.
- The RID is still based on IPv4 addresses, not IPv6 addresses. Like EIGRP for IPv6, a router must have a RID before OSPFv3 will work.

Other differences exist as well, particularly when examining the details listed in the OSPFv3 LSDB. However, keeping to a scope that includes general concepts, configuration, and verification, this chapter does not get into those details.

Configuring OSPFv3

OSPFv3 configuration requires no new steps and few changes syntactically when compared to the newer style of OSPFv2 configuration discussed in Chapter 5's section "Enabling OSPF Neighbor Discovery on LANs." That section explained how to configure OSPF using the interface **ip ospf area** interface subcommand to enable OSPF on an interface, rather than relying on the indirect reference made by OSPFv2's **network** router subcommand. With OSPFv3, only one configuration style is supported, using the similar **ipv6 ospf area** command to enable OSPFv3 on an interface. IOS does not support a **network** command for OSPFv3.

The basic OSPFv3 configuration matches the newer style of OSPFv2 configuration, except for substituting **ipv6** instead of **ip** in the configuration commands. The following list shows the basic configuration steps, including the assignment of an IPv6 address to the interface.

Key Topic

Step 1. Enable IPv6 routing with the **ipv6 unicast-routing** global command.

Step 2. Create an OSPFv3 routing process using the **ipv6 router ospf** *process-id* global configuration command.

Step 3. Enable IPv6 on the interface, typically by configuring static IPv6 addresses as follows:
Configure an IPv6 unicast address on each interface, using the **ipv6 address** *address*/*prefix-length* [**eui-64**] interface command.
Configure the **ipv6 enable** command, which enables IPv6 and causes the router to derive its link local address.

Step 4. Enable OSPFv3 on the interface with the **ipv6 ospf** *process-id* **area** *area-number* interface subcommand.

Step 5. If no OSPF router ID has been automatically chosen, due to not having at least one working interface with a IPv4 address, configure an OSPF router ID with the **router-id** *rid* command in OSPFv3 configuration mode.

Note: for a complete discussion of options for enabling IPv6 on an interface, as mentioned at Step 3, refer to Chapter 16's Table 16-11.

Note: Unlike a new EIGRP for IPv6 routing process, a newly created OSPFv3 routing process defaults to an administratively enabled state. As a result, the previous configuration checklist does not include the **no shutdown** OSPFv3 subcommand. However, the router mode subcommands **shutdown** and **no shutdown** can be used to disable and reenable the OSPFv3 routing process.

To be complete, Example 17-5 lists the OSPFv3 configuration on Router R1 from Figure 17-. Figure 17-2 shows the area design used for the configuration.

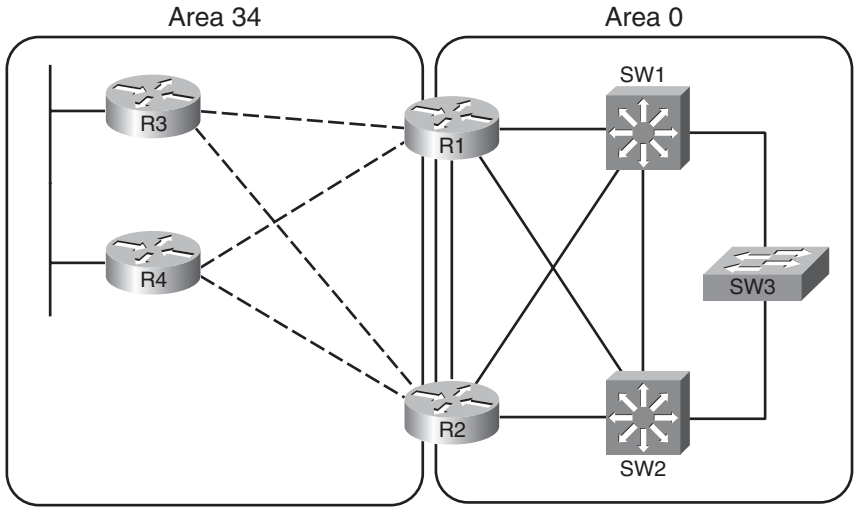


Figure 17-2 *Area Design Used for Example 17-5*

Example 17-5 *Configuring IPv6 Routing and Routing Protocols on R1*

```
R1# show running-config
! output is edited to remove lines not pertinent to this example
! Step 1: Enable IPv6 routing
ipv6 unicast-routing
!
! On five interfaces, do steps 3 and 4: configure static IPv6 addresses,
! and enable OSPFv3 process 5, in the appropriate areas
interface FastEthernet0/0.1
 ipv6 address 2012::1/64
 ipv6 ospf 5 area 0
!
interface FastEthernet0/0.2
 ipv6 address 2017::1/64
 ipv6 ospf 5 area 0
!
interface FastEthernet0/1.18
 ipv6 address 2018::1/64
 ipv6 ospf 5 area 0
!
interface Serial0/0/0.3
 ipv6 address 2013::1/64
 ipv6 ospf 5 area 34
!
interface Serial0/0/0.4
 ipv6 address 2014::1/64
 ipv6 ospf 5 area 34
```

```
! Steps 2 and 5 - creating the OSPFv3 process, and defining the RID
ipv6 router ospf 5
 router-id 1.1.1.1
```

The configuration example shows two interface subcommands on each interface: an **ipv6 address** command that defines a global unicast IPv6 address, plus the **ipv6 ospf area** command to enable OSPFv3 on the interface. The interface subcommands place each interface into the correct area. But these interface subcommands have no effect unless the configuration also includes a matching **ipv6 router ospf 5** global command, with the 5 in this case matching the **ipv6 ospf 5 area** commands on the interfaces.

Beyond this basic OSPFv3 configuration, many of the optional OSPF features match when comparing the OSPFv2 and OSPFv3 configuration as well, for example:

- The concept and commands related to OSPF stub areas are literally identical, using commands such as **area 34 stub**. (See Table 7-3 for a reminder of the commands for OSPFv2.)
- Like OSPFv2, OSPFv3 can only summarize routes on ABRs and ASBRs, using the similar command **area** *x* **range** *ipv6-prefix/length* router subcommand, with the only difference being that the command lists an IPv6 prefix rather than an IPv4 subnet and mask.
- Like OSPFv2, OSPFv3 uses the concept of OSPF interface types as configured with the **ipv6 ospf network** *type* interface subcommand. These types dictate whether OSPFv3 attempts to elect a DR, and whether routers need to configure neighbors with the **ipv6 ospf neighbor** interface subcommand, with the same issues seen with OSPFv2.

Verifying OSPFv3

Most of the **show** commands for OSPFv3 have similar output compared to the OSPFv2 versions of the commands. The biggest differences exist in the output related to the OSPF database, both due to the changes in LSA terminology, the addition of new LSA types, and that some LSDB information is located with a different LSA type when comparing OSPFv2 and OSPFv3. Table 17-8 lists a cross-reference comparing the key commands related to both OSPFv2 and OSPFv3, with the second column listing the parameters after **show ip**, and the third column listing parameters after **show ipv6**.

Table 17-8 *Comparing OSPF Verification Commands:* **show ip** *and* **show ipv6**...

Function	show ipv4...	show ipv6...
All OSPF-learned routes	... route ospf	... route ospf
Router ID, Timers, ABR, SPF statistics	... ospf	... ospf
List of routing information sources	... protocols ... ospf neighbor	... ospf neighbor

Interfaces assigned to each area	**... protocols** **... ospf interface brief**	**... protocols** **... ospf interface brief**
OSPF interfaces–costs, state, area, number of neighbors	**... interface brief**	**... interface brief**
Detailed information about OSPF interfaces	**... ospf interface**	**... ospf interface**
Displays summary of OSPF database	**... ospf database**	**... ospf database**

The first verification example focuses on some of the differences between OSPFv2 and OSPFv3 for the display of neighbor status. Example 17-6 shows output from Router R3, as shown in Figures 17-1 and 17-2. The output highlights the only noticeable difference between the output of the **show ip ospf neighbor** and **show ipv6 ospf neighbor** commands: OSPFv2 displays the interface IP address of neighbors, but OSPFv3 displays an interface ID. OSPFv3 routers create a locally significant interface ID, using that in LSAs that describe the intra-area topology, rather than using neighbor IPv4 addresses to describe the topology. OSPFv3 separates the topology information from the Layer 3 addressing information, so OSPFv3 uses the interface ID concept to identify how a router reaches a particular neighbor.

Example 17-6 *IPv6 OSPFv3 Interface IDs on Router R3*

```
! On R3, IPv4 OSPFv2 neighbors are listed, with interface IPv4 addresses.
R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           0   FULL/  -        00:00:31    10.10.23.2      Serial0/0/0.2
1.1.1.1           0   FULL/  -        00:00:31    10.10.13.1      Serial0/0/0.1
4.4.4.4           1   FULL/DR         00:00:39    10.10.34.4
FastEthernet0/0

! On R3, IPv6 OSPFv3 neighbors are listed, with interface IDs.
R3# show ipv6 ospf neighbor

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
2.2.2.2           1   FULL/  -        00:00:36    19              Serial0/0/0.2
1.1.1.1           1   FULL/  -        00:00:35    19              Serial0/0/0.1
4.4.4.4           1   FULL/DR         00:00:34    3
FastEthernet0/0

! Next, the LSA for R3 (RID 3.3.3.3) lists its own interface ID and
! the neighbor interface ID, for each link.
R3# show ipv6 ospf database router adv-router 3.3.3.3

            OSPFv3 Router with ID (3.3.3.3) (Process ID 5)
```

```
                Router Link States (Area 34)

  LS age: 996
  Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
  LS Type: Router Links
  Link State ID: 0
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000007
  Checksum: 0xDC04
  Length: 72
  Number of Links: 3

    Link connected to: another Router (point-to-point)
      Link Metric: 64
      Local Interface ID: 17
      Neighbor Interface ID: 19
      Neighbor Router ID: 2.2.2.2

    Link connected to: another Router (point-to-point)
      Link Metric: 64
      Local Interface ID: 16
      Neighbor Interface ID: 19
      Neighbor Router ID: 1.1.1.1

    Link connected to: a Transit Network
      Link Metric: 1
      Local Interface ID: 3
      Neighbor (DR) Interface ID: 3
      Neighbor (DR) Router ID: 4.4.4.4
```

Example 17-7 displays the output of a few other IPv6 **show** command, just for perspective. The **show ipv6 ospf interface brief** command lists the same kind of information shown in the similar **show ip ospf interface brief** command, but again listing the interface ID as new information. The **show ipv6 protocols** command lists much sparser information as compared to the similar **show ip protocols** command. Note that both commands identify which OSPFv3 interfaces have been assigned to each area.

Example 17-7 *Finding Interfaces and Areas for OSPFv3, on Router R3*

```
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 34):
    Serial0/0/0.1
    Serial0/0/0.2
```

```
    FastEthernet0/0
  Redistribution:
    None
R1# show ipv6 ospf interface brief
Interface     PID   Area            Intf ID    Cost  State Nbrs F/C
Se0/0/0.1     1     34              19          64    P2P   1/1
Se0/0/0.2     1     34              19          64    P2P   1/1
Fa0/0         1     34               3           1    BDR   1/1
```

IPv6 IGP Redistribution

IPv6 routing protocols can perform route redistribution, much like IPv4 route redistribution (as detailed in Chapter 9, "Basic IGP Redistribution," and Chapter 10, "Advanced IGP Redistribution"). This section shows a few examples to confirm those similarities and compares IPv4 and IPv6 route redistribution.

The following list summarizes some of those key similarities between both IPv4 and IPv6 route redistribution:

- Redistribution takes routes from the IP routing table, not from the topology tables and databases controlled by the source routing protocol.
- Route maps can be applied when redistributing for the purpose of filtering routes, setting metrics, and setting route tags.
- IPv6 routing protocols use the same default administrative distances, with the same basic mechanisms to override those defaults.
- The same basic mechanisms exist in IPv6 to defeat routing loop problems: administrative distance, route tags, and filtering.
- The routing protocols use the same default administrative distance (AD) settings for internal and external routes.
- The redistribution configuration uses practically the same syntax with the same commands.

Some differences do exist, both in configuration and in concept, as follows:

- Any matching done with distribution lists or route maps would use IPv6 prefix lists and IPv6 ACLs, which match based on IPv6 prefix and length.
- The IPv6 version of the **redistribute** command takes only routes learned from an IGP but by default does not take connected routes on interfaces enabled for that IGP. To also redistribute those connected routes, the **redistribute** command must include the **include-connected** parameter. When an IPv4 routing protocol redistributes from an IGP, it always attempts to take both the IGP-learned routes and the connected routes for interfaces enabled for that IGP.
- Unlike OSPFv2, OSPFv3 does not require a **subnets** parameter on the **redistribute** command, because IPv6 does not maintain the IPv4 concept of classful networks and the subnets inside those classful networks.

- IPv6 redistribution ignores the "local" routes in the IPv6 routing table (the /128 host routes for a router's own interface IPv6 addresses). IPv4 has no equivalent concept.

Redistributing without Route Maps

As with IPv4 redistribution, IPv6 redistribution can be done without applying a route map. The first redistribution example shows such a case, with router R2 performing the redistribution. In this case, R2 redistributes routes learned from an OSPF domain on the right, into a RIP domain on the left, without applying a route map. Figure 17-3 shows the two routing domains and the IPv6 subnets used.

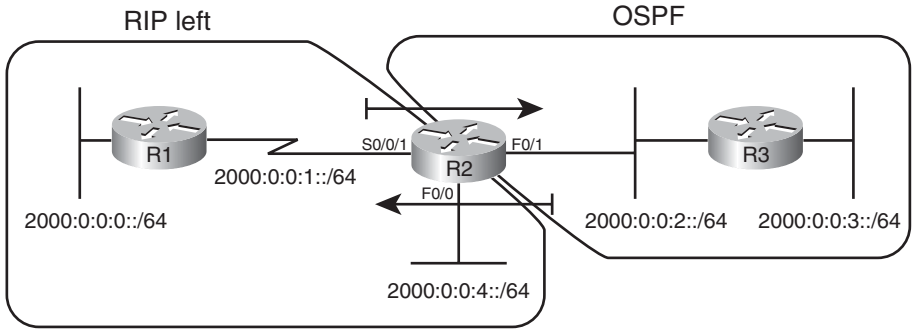


Figure 17-3 *Redistribution Plan*

Example 17-8 shows the redistribution configuration on Router R2, which takes routes from OSPF process number 5 and redistributes them into RIP process "left." The example begins with a **show ipv6 route** command on R2 to list the routes that could be redistributed, and ends with those routes being seen on Router R1.

Example 17-8 *Redistributing from OSPF into RIP, with Default Metrics*

```
R2# show ipv6 ospf interface brief
Interface    PID   Area            Intf ID    Cost  State Nbrs F/C
Fa0/1        5     0               4          1     BDR   1/1

R2# show ipv6 route
IPv6 Routing Table - Default - 9 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2000::/64 [120/2]
     via FE80::213:19FF:FE7B:5026, Serial0/0/1
C   2000:0:0:1::/64 [0/0]
```

```
      via Serial0/0/1, directly connected
L    2000::1:213:19FF:FE7B:5004/128 [0/0]
      via Serial0/0/1, receive
C    2000:0:0:2::/64 [0/0]
      via FastEthernet0/1, directly connected
L    2000:0:0:2::2/128 [0/0]
      via FastEthernet0/1, receive
O    2000:0:0:3::/64 [110/2]
      via FE80::213:19FF:FE7B:6588, FastEthernet0/1
C    2000:0:0:4::/64 [0/0]
      via FastEthernet0/0, directly connected
L    2000::4:213:19FF:FE7B:5004/128 [0/0]
      via FastEthernet0/0, receive
L    FF00::/8 [0/0]
      via Null0, receive

R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# ipv6 router rip left
R2(config-rtr)# redistribute ospf 5 include-connected
R2(config-rtr)# ^Z
R2#
```

```
! The next command is executed on router R1
R1# show ipv6 route rip
IPv6 Routing Table - Default - 8 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R    2000:0:0:2::/64 [120/2]
      via FE80::213:19FF:FE7B:5004, Serial0/0/0
R    2000:0:0:3::/64 [120/3]
      via FE80::213:19FF:FE7B:5004, Serial0/0/0
R    2000:0:0:4::/64 [120/2]
     via FE80::213:19FF:FE7B:5004, Serial0/0/0
```

The brief configuration shows both the simplicity and one difference in the IPv4 and IPv6 redistribution configuration. First, the redistribution function works with only the **redistribute ospf 5** command, which takes routes from R2's OSPF 5 process. However, the **redistribute** command has the **include-connected** keyword, which tells RIP to not only take the OSPF-learned routes for 2000:0:0:3::/64, but also the connected route for 2000:0:0:2::/64, the connected route off R2's F0/1 interface. The beginning of the example lists a **show ipv6 ospf interface brief** command, which lists that only F0/1 has been enabled for OSPF.

The **show ipv6 route rip** command on R1, at the end of the example, shows that R2 indeed redistributed the two routes from the OSPF side of Figure 17-3.

Redistribution into RIPng uses a metric as taken from the IPv6 routing table by default. In this case, the output from R1 at the end of the example lists the two redistributed routes with metrics 2 and 3. You can see from the earlier output in the example that R2's metric for its OSPF route to 2000:0:0:3::/64 was 2, so R3's metric is now listed as 3.

Using RIPng's default action to take the integer metric from the source IGP worked in this case, but in most cases, the metric should be set via configuration. OSPF metrics will often be more than the maximum usable RIP metric of 15, making these routes instantly unusable in RIP. Redistributing from EIGRP into RIP would not work either, given the relatively large integer metrics calculated by EIGRP. So when redistributing into RIP, change the configuration to set the metric. For instance, using the command **redistribute ospf 5 include-connected metric 3** would have set the default metric for all redistributed routes and avoided such problems.

Redistributing with Route Maps

IPv6 redistribution can also call route maps, for all the usual reasons: setting metrics for different routes, filtering routes, and setting route tags, to name a few. To demonstrate the configuration, Example 17-9 shows another example, using the same design in Figure 17-3. In this case, Router R2 redistributes RIP routes into OSPF, with the following criteria:

- Redistribute only LAN subnets, with OSPF metric 200.
- Filter all other subnets.

Example 17-9 shows the configuration on R2 and the resulting OSPF routes on R3.

Example 17-9 *Redistributing from RIP into OSPF, Using a Route Map*

```
R2# show run
! unrelated lines omitted
ipv6 router ospf 5
 router-id 2.2.2.2
 redistribute rip left route-map only-RIP-lan include-connected
!
ipv6 router rip left
 redistribute ospf 5 metric 3 include-connected
!
ipv6 prefix-list rip-to-ospf seq 5 permit 2000::/64
ipv6 prefix-list rip-to-ospf seq 10 permit 2000:0:0:4::/64
!
route-map only-RIP-lan permit 10
 match ipv6 address prefix-list rip-to-ospf
 set metric 200
```

```
R3# show ipv6 route ospf
IPv6 Routing Table - Default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
```

```
        B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external
        O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OE2 2000::/64 [110/200]
     via FE80::213:19FF:FE7B:5005, FastEthernet0/0
OE2 2000:0:0:4::/64 [110/200]
     via FE80::213:19FF:FE7B:5005, FastEthernet0/0
```

First, the configuration shows an IPv6 prefix list and a route map that uses a **match ipv6** command that refers to the prefix list. The prefix list works just like an IPv4 prefix list, other than matching with the IPv6 prefix format rather than IPv4. The route map works with the same logic as well, in this case referencing the IPv6 prefix list with the **match ipv6 address prefix rip-to-ospf** command. The route map matches the two LAN subnets in the RIP domain with the first route map clause and sets the metric to 200. The implied deny clause at the end of the route map matches all other routes, which makes R2 filter all other routes from being redistributed into OSPF. As a result, the serial IPv6 subnet, 2000:0:0:1::/64, is filtered by the redistribution process.

The **show ipv6 route ospf** command on R3 at the end of the example confirms that R3 learned routes for both LAN subnets in the RIP domain but no other routes. Of particular interest, note that OSPFv3 lists the route as OSPF external Type 2, because just like OSPFv2, OSPFv3 defaults to redistribute routes as external Type 2 routes. Note also that the output lists metrics for each route as 200, because R2 set the metric to 200, and OSPF does not add anything to the metric of E2 routes.

Static IPv6 Routes

IPv6 supports the configuration of static routes with similar syntax compared to the **ip route** command used to configured IPv4 static routes. The syntax of the **ipv6 route** command matches the **ip route** command, other than that the IPv6 destination is represented as prefix/length, and any next-hop IP address is an IPv6 address, of course. The generic syntax is

ipv6 route *prefix/length* {*outgoing-interface* [*next-hop-address*] ¦ *next-hop-address*} [*admin-distance*] [**tag** *tag-value*]

The parameters use the same concepts as the same parameters used for IPv4. However, with IPv6, a few interesting differences exist. First, when using the next-hop IP address, you can use any address on the neighboring router, including the neighbor's link local IPv6 address. Second, if using a link local address as the next-hop address, then you must configure both the outgoing interface and the link local address.

Example 17-10 shows such a case, with a static route configured on Router R3 in Figure 17-4. R3 adds a static route for R1's LAN prefix, 2000::/64, using R2 as next hop, specifically R2's F0/1 link local address as next hop.

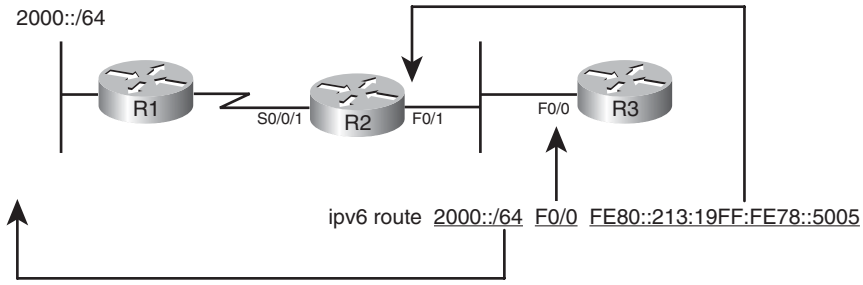


Figure 17-4 *Static IPv6 Route on Router R3*

Example 17-10 *Static Route on R3, with Link Local Next-Hop Address*

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# ipv6 route 2000::/64 FE80::213:19FF:FE7B:5005
% Interface has to be specified for a link-local nexthop

R3(config)# ipv6 route 2000::/64 f0/0 FE80::213:19FF:FE7B:5005
R3(config)# ^Z
R3# show ipv6 route
IPv6 Routing Table - Default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2000::/64 [1/0]
     via FE80::213:19FF:FE7B:5005, FastEthernet0/0
C   2000:0:0:2::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2000:0:0:2::3/128 [0/0]
     via FastEthernet0/0, receive
C   2000:0:0:3::/64 [0/0]
     via FastEthernet0/1, directly connected
L   2000::3:213:19FF:FE7B:6589/128 [0/0]
     via FastEthernet0/1, receive
OE2 2000:0:0:4::/64 [110/200]
     via FE80::213:19FF:FE7B:5005, FastEthernet0/0
L   FF00::/8 [0/0]
     via Null0, receive
R3# show ipv6 route 2000::/64
Routing entry for 2000::/64
  Known via "static", distance 1, metric 0
```

```
Backup from "ospf 1 [110]"
Route count is 1/1, share count 0
Routing paths:
  FE80::213:19FF:FE7B:5005, FastEthernet0/0
    Last updated 00:11:50 ago
```

The example shows a static route that uses a multiaccess interface as the outgoing interface, so using both the outgoing interface (F0/0) and the next-hop address in the **ipv6 route** command prevents ambiguity. Also, note that IPv6 routers do not actually have a route that matches the neighbor's link local address. As a result, the **ipv6 route** command must supply the outgoing interface as well, so the local router (R3 in this case) knows on which interface to use NDP to discover how to reach this neighboring IPv6 address. The example shows an attempt to configure the **ipv6 route** command with only R2's next-hop link local address, but R3 rejects the command. When both the outgoing interface (F0/0) has been listed, plus R2's link local address as next hop, R3 accepts the command and adds this static route to its IPv6 routing table.

Finally, the output of the **show ipv6 route 2000::/64** command at the end of the example lists some information not seen in the equivalent IPv4 command. The output reveals some insight into the operation of the routing table manager (RTM), which chooses the best route for a prefix among many sources. In this case, the static route for 2000::/64 used a default administrative distance of 1. The routers in this design were also still configured as they were at the end of Example 17-9, with RIP on the left of the figure, OSPF on the right, and redistribution between the two. So, R3 still has an OSPF route for 2000::/64, with default administrative distance 110. The output of this command lists not only the static route, but also the backup route, with a worse administrative distance, in this case the OSPF-learned route. When configuring a floating static route—a static route with a high administrative distance so that the route will not be used until the IGP routes have all failed—this command can be useful to confirm that this backup route is known to IOS.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Practice Planning Tables."

Implementation Plan Peer Review Table

Table 17-9 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 17-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A RIPng implementation plan lists two neighboring routers with unicast IPv6 addresses 2000::1/64 and 2001::2/64, respectively. Will this cause a neighborship issue?	
Same issues as in the previous row, but the plan uses EIGRP for IPv6.	
A plan shows a planned config for a new router, with no IPv4 addresses, IPv6 addresses on all interfaces, and EIGRP for IPv6 configuration. What potential issues should you look for in the configuration? (3)	
Same scenario as the previous row, but with OSPFv3.	
The plan shows an EIGRP for IPv6 and OSPFv3 domain with mutual redistribution. The configuration shows a **redistribute eigrp 1** command under the OSPF process. What kinds of routes should be redistributed? Which kinds will not?	

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 17-10 all configuration commands related to the configuration of the following features. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 17-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Assuming IPv6 routing and IPv6 addresses have already been configured, configure RIPng.	
Assuming IPv6 routing and IPv6 addresses have already been configured and no IPv4 addresses exist on the router, configure EIGRP for IPv6.	
Assuming IPv6 routing and IPv6 addresses have already been configured and no IPv4 addresses exist on the router, configure OSPFv3.	
Configure RIPng to redistribute routes from OSPF process 1 including subnets, and connected interfaces.	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own EIGRP verification plan, list in Table 17-11 all commands that supply the requested information. You may want to record your answers outside the book, and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 17-11 *Verification Plan Memory Drill*

Information Needed	Commands
All IPv6 routes	
Details about a given IPv6 prefix	
All routes within a given range of IPv6 addresses	
All RIP-learned IPv6 routes	
All next-hop IPv6 addresses used by RIP routes	
The interfaces on which RIP is enabled	
All EIGRP-learned IPv6 routes	
All EIGRP neighbors	
Summary of the EIGRP topology table	
OSPF router ID and SPF statistics	
List of OSPF neighbors	
All OSPF-learned IPv6 routes	
Interfaces enabled for OSPF and their assigned areas	
OSPF costs per interface	
Summary of the OSPF database	

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the Key Topics icon in the outer margin of the page. Table 17-12 lists a reference of these key topics and the page numbers on which each is found.

Key Topic

Table 17-12 *Key Topics for Chapter 17*

Key Topic Element	Description	Page Number
Table 17-3	Comparisons between RIP-2 and RIPng	574
List	Configuration steps for RIPng	575
Table 17-5	Comparisons between EIGRP for IPv4 and EIGRP for IPv6	581
List	Configuration steps for EIGRP for IPv6	582
List	Decision process for choosing an EIGRP for IPv6 router ID	583
Table 17-7	Comparisons between EIGRP for IPv4 and EIGRP for IPv6	588
List	Additional explanations of key differences between OSPFv3 and OSPFv2	589
List	Configuration steps for OSPFv3	590
List	Similarities and differences with IPv4 and IPv6 redistribution	595

Complete the Tables and Lists from Memory

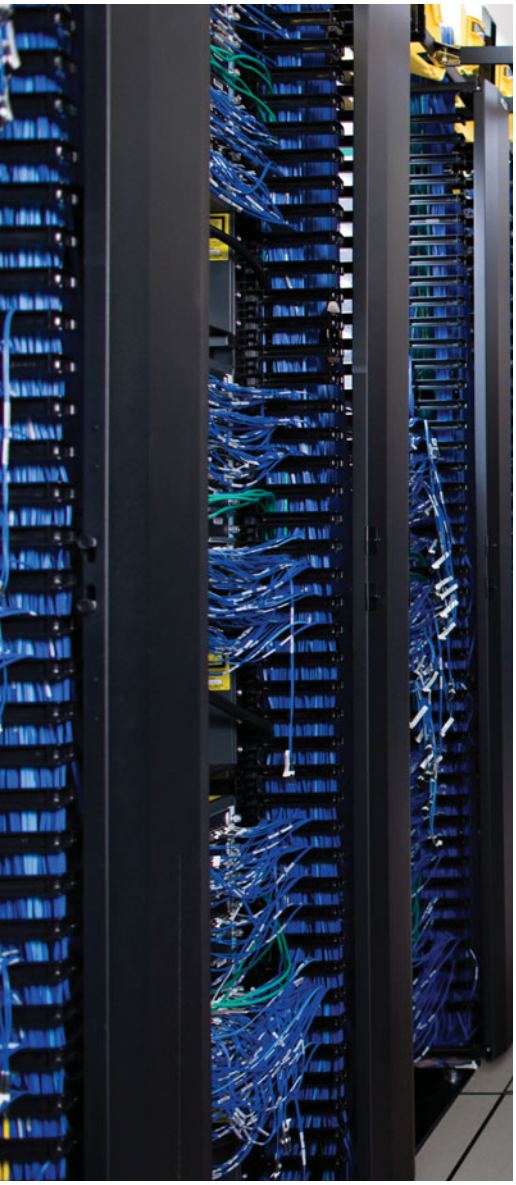
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

RIP Next Generation, OSPF Version 3, EIGRP for IPv6

This page intentionally left blank



This chapter covers the following subjects:

IPv4 and IPv6 Migration and Coexistence Concepts: This section introduces the three main branches of coexistence tools: dual stacks, tunnels, and protocol translation.

Static Point-to-Point IPv6 Tunnels: This section explains the concepts behind IPv6 point-to-point tunnels, their differences, and the configuration of each.

Dynamic Multipoint IPv6 Tunnels: This section explains the concepts behind IPv6 multipoint tunnels, their differences, and the configuration of each.

CHAPTER 18

IPv4 and IPv6 Coexistence

Most of the barriers for a migration to IPv6 have been removed. The IPv6 protocol set has been complete for more than a decade. Many products, including the most common client and server operating systems, support IPv6. Cisco routers have supported IPv6 long enough that IPv6 support exists in the most commonly deployed IOS versions. The product support means that IPv6 deployment should not be inhibited by the availability of IPv6 in commonly used software. Additionally, the Internet's support for IPv6 is growing. Barriers do still exist, but IPv6 adoption may have finally reached the point of gathering momentum worldwide.

Although the tools exist, the process of migrating billions of IPv4-based computers and devices to IPv6–devices under the control of millions of organizations and maybe billions of individual people–will not happen overnight. In the real world, the migration from IPv4 to IPv6 will probably take a generation to complete–arguably, it has been in progress for a decade already. Knowing that the migration would take years, the creators of IPv6 wisely planned for coexistence tools that allow both a single Enterprise, and the Internet, to use both IPv4 and IPv6 at the same time, coexisting peacefully.

This chapter discusses the three major categories of migration and coexistence tools for IPv6: dual stacks, tunneling, and protocol translation. However, the majority of the chapter examines four different options for tunneling IPv6 over IPv4 networks.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these nine self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 18-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 18-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
IPv4 and IPv6 Migration and Coexistence Concepts	1–3
Static Point-to-Point IPv6 Tunnels	4–6
Dynamic Multipoint IPv6 Tunnels	7–9

1. An enterprise has plans to start adding IPv6 support. For the first year, the IPv6 will be in small pockets spread around the existing large IPv4 network, with occasional IPv6 traffic while applications teams test IPv6-enabled servers and applications. Which of the following tools would be most appropriate?

   a. Native IPv6
   b. Point-to-point tunnels
   c. Multipoint tunnels
   d. NAT-PT

2. An enterprise has plans to start adding IPv6 support. The initial deployment requires support from some IPv6-only devices that need to access servers that support only IPv4. Which of the following tools would be most appropriate?

   a. Native IPv6
   b. Point-to-point tunnels
   c. Multipoint tunnels
   d. NAT-PT

3. A client host uses IPv4 to communicate with one server and IPv6 to communicate with another. Which of the following IPv6 coexistence features is likely at work on the host?

   a. Native IPv6
   b. Point-to-point tunnels
   c. Multipoint tunnels
   d. NAT-PT
   e. Dual stacks

4. The following configuration exists on a router on one end of an IPv6 tunnel. Which type of tunnel is created by this configuration?

```
interface loopback 1
 ip address 1.1.1.1 255.255.255.255
interface tunnel 2
 ipv6 address 2000::1::/64
 tunnel source loopback 1
 tunnel destination 2.2.2.2
 tunnel mode ipv6ip
 ipv6 eigrp 1
```

   a. Automatic 6to4
   b. Manually configured tunnel
   c. ISATAP
   d. GRE

5. An engineer is reviewing another engineer's sample configuration for a GRE tunnel used to pass IPv6 traffic. The tunnel has not yet been configured on the router. Which of the following commands is not required for the configuration to pass IPv6 traffic?

   a. **tunnel source**
   b. **tunnel destination**
   c. **tunnel mode**
   d. All these commands are required.

6. Which of the following IPv6 tunneling mechanisms support IPv6 IGP routing protocols? (Choose two.)

   a. Automatic 6to4
   b. Manually configured tunnel
   c. ISATAP
   d. GRE

7. The following configuration exists on a router on one end of an IPv6 tunnel. Although the configuration added so far is correct, the configuration is incomplete. Which type of tunnel is most likely to be intended by the network engineer?

```
interface loopback 1
 ip address 192.168.1.1 255.255.255.255
interface tunnel 2
 ipv6 address 2002:C0A8:101::1/64
 tunnel source loopback 1
```

   a. Automatic 6to4
   b. Manually configured tunnel
   c. ISATAP
   d. GRE

8. The answers each list a tunnel method and two consecutive IPv6 address quartets. Which answers identify a tunneling method that relies on an IPv4 address to be embedded into an IPv6 address, within the correct quartets listed? (Choose two.)

   a. Automatic 6to4, quartets 2 and 3
   b. Automatic 6to4, quartets 7 and 8
   c. ISATAP, quartets 2 and 3
   d. ISATAP, quartets 7 and 8

9. Router R1 uses MAC address 1111.1111.1111 for its Fa0/0 interface. An engineer sees the following configuration in the output of a **show running-config** command. Then, the engineer issues a **show ipv6 interface brief** command. What global unicast IPv6 address does this command display for interface tunnel 1?

```
interface loopback 1
 ip address 192.168.1.1 255.255.255.255
interface tunnel 1
 tunnel source loopback 1
 tunnel destination 192.168.1.2
 tunnel mode ipv6ip isatap
 ipv6 address 2000::/64 eui-64
```

   a. 2000::1311:11FF:FE11:1111
   b. 2000::C0A5:101
   c. 2000:C0A5:101::
   d. 2000::5EFE:C0A5:101

Foundation Topics

IPv4 and IPv6 Migration and Coexistence Concepts

Most Enterprises do not move from having no formal IPv6 support to creating a full native IPv6 implementation on all routers, hosts, servers, and other devices. In the real world, some Enterprise networks will begin with several locations that need consistent and working IPv6 support. Other companies may begin with a small set of people who want to experiment with IPv6 in anticipation of the day in which IT will take advantage of IPv6. Eventually, IPv6 may well become the predominate traffic in the Enterprise, and one day, IPv4 may go away altogether, but that day may be decades away. As a result, IPv4 and IPv6 will likely coexist in a given internetwork for a very long time.

During this possibly long migration, three main classes of tools may be used to allow IPv4 to continue to work well, while supporting IPv6:

- Dual IPv4/IPv6 stacks (*dual stacks*)
- Tunneling
- NAT Protocol Translator (NAT-PT)

This section looks at each type of tool in succession.

IPv4/IPv6 Dual Stacks

The term *dual stacks* means that the host or router uses both IPv4 and IPv6 at the same time. For hosts, this means that the host has both an IPv4 and IPv6 address associated with each NIC, that the host can send IPv4 packets to other IPv4 hosts, and that the host can send IPv6 packets to other IPv6 hosts. For routers, it means that in addition to the usual IPv4 IP addresses and routing protocols, the routers would also have IPv6 addresses and routing protocols configured as discussed in Chapters 16, "IP Version 6 Addressing," and 17, "IPv6 Routing Protocols and Redistribution." To support both IPv4 and IPv6 hosts, the router could then receive and forward both IPv4 packets and IPv6 packets.

Note: Although not used as often today, the general term *protocol stack* refers to the layers in a protocol suite, with a protocol stack referring to the implementation of all layers of that protocol on a computer. Historical examples include Novell Netware, AppleTalk, and IBM SNA protocol stacks.

The hosts using dual stacks follow the same general process of using DNS to resolve a name into an IP address. The DNS requests can return either an IPv4 address or an IPv6 address. The dual stack host can then choose to use IPv4 to communicate with an IPv4 host or IPv6 to communicate with an IPv6 host.

To support dual stack hosts, routers need to forward both IPv4 and IPv6 packets. To forward IPv6 packets to the various destinations, the design engineer can use one of two general approaches:

- **Native IPv6:** Configure IPv6 on most or all routers, on most or all production interfaces, making all routers use a dual stack.
- **IPv6 tunnels:** Configure some routers with IPv6, others without IPv6, and tunnel the IPv6 packets over the IPv4 network by encapsulating IPv6 packets inside IPv4 packets.

Configuring native IPv6 simply means that you configure IPv6 as discussed in detail in Chapters 16 and 17, giving each interface one or more IPv6 addresses, enabling IPv6 routing protocols, and so on. Assuming an IPv4 network already exists, the engineer could build and execute an implementation plan to configure native IPv6 by enabling IPv6 on the same interfaces as IPv4, configuring an IPv6 routing protocol, and the routers would be ready to forward both types of packets.

A router that has been configured to forward both IPv4 and IPv6 is a dual stack router. This works well, but it does require the implementation of IPv6 on all routers that might one day receive an IPv6 packet that needs to be forwarded. Alternatively, using tunnels may be more reasonable to support smaller pockets of IPv6 hosts, because tunnels require fewer routers to be configured with IPv6 at all.

From a practical exam-preparation perspective for the CCNP ROUTE exam, the implementation of dual stacks means that you need to configure IPv6 as detailed in Chapters 16 and 17, while leaving the existing IPv4 configuration in place. Therefore, this chapter does not discuss the dual stack option's configuration, instead relying on the detail in Chapters 16 and 17.

Tunneling

Tunneling refers to a process by which one router or host encapsulates the IPv6 packet inside an IPv4 packet. The networking devices forward the IPv4 packet, ignoring the fact that the packet's payload is an IPv6 packet. Some later device or host decapsulates the original IPv6 packet, forwarding it on to the final destination.

From a network design perspective, tunneling IPv6 over IPv4 results in fewer routers needing any IPv6 configuration at all. As such, it allows a quicker migration from no IPv6 support to enough support to get IPv6 packets between two sites. Fewer routers need new configuration, the smaller number of changes means less operational risk, and the end hosts can still forward IPv6 traffic to each other.

This section begins by examining the concepts behind IPv6 tunneling. Then the text more closely examines the two main categories of tunnels: point-to-point tunnels and multipoint tunnels.

General Tunneling Concepts

To appreciate the configuration of the many types of IPv6 tunnels discussed in this chapter, you need a solid understanding of the basics. Throughout this chapter, most examples will be built on a small internetwork that represents a pre-existing IPv4-only Enterprise

network. The IPv6 tunneling discussed in this chapter will be built on this simple network, as shown in Figure 18-1.

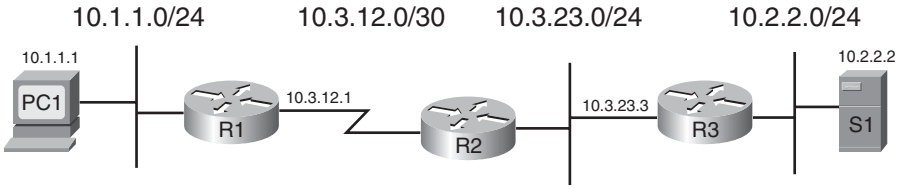


Figure 18-1 *Simple IPv4 Enterprise Network*

Next, consider the case in which an engineer must create an implementation plan to allow PC1 and server S1 to communicate using IPv6. They both may use IPv6 exclusively, or maybe one or both use a dual stack. In either case, for this example, the engineer's task is to ensure delivery of IPv6 packets sent between these two hosts. Additionally, the native mode option–in other words, configuring IPv6 on all three routers and all interface on each router–has been rejected, so IPv6 tunneling will be used.

To configure an IPv6 tunnel, the engineer could enable IPv6 on R1 and R3, and not at all on R2. The configuration on R1 and R3 would not need to enable IPv6 on the inner interfaces at all. Additionally, R1 and R3 would need the configuration related to a point-to-point tunnel. Figure 18-2 shows some of the concepts behind this point-to-point tunnel between R1 and R3.

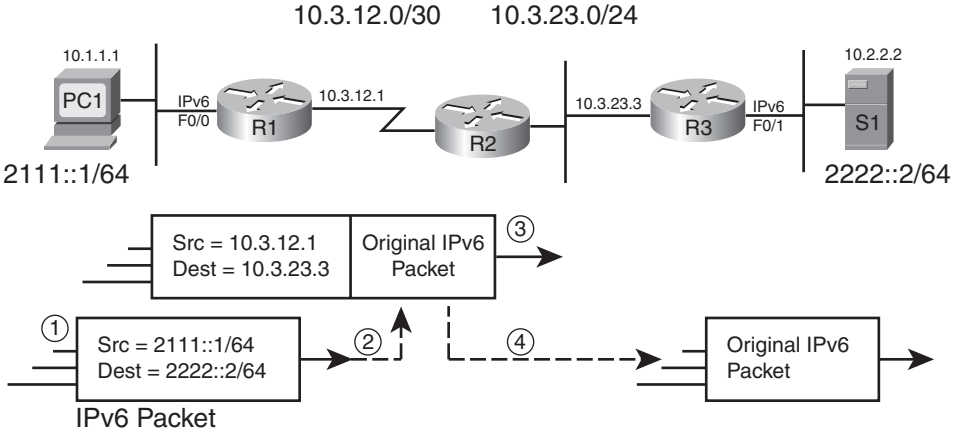


Figure 18-2 *IPv6 Tunneling Encapsulation and De-encapsulation*

Following the steps in Figure 18-2

Step 1. PC1, acting as an IPv6 host, sends an IPv6 packet to S1, IPv6 address 2222::2. (PC1 would have likely used a DNS process to find S1's IPv6 address at some point prior to this step to find S1's IPv6 address.)

Step 2. R1 encapsulates the IPv6 packet into an IPv4 packet, with R1's IPv4 address as source IPv4 address, and R3's IPv4 address as the destination IPv4 address.

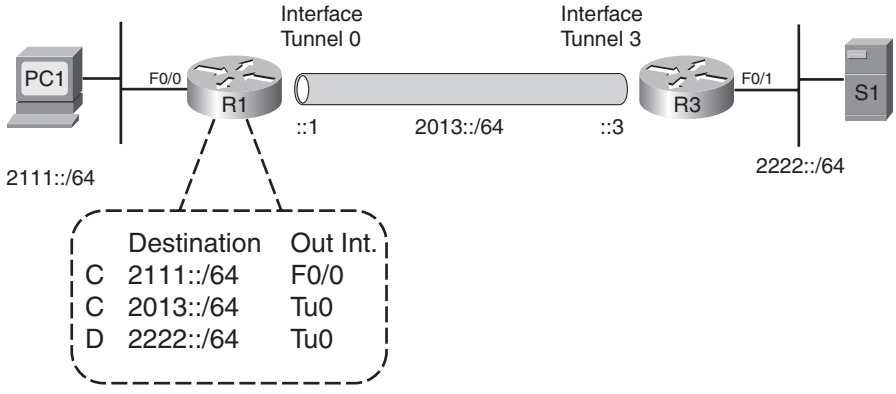
Step 3. R2 forwards the IPv4 packet to R3, with no knowledge of IPv6.

Step 4. R3 decapsulates the original IPv6 packet, forwarding the IPv6 packet on its IPv6-enabled F0/1 interface, to server S1.

Although the figure and steps show the mechanics of the IPv6 tunnel, with such a small sample network, it may be easier to just natively configure IPv6 on all three routers. However, imagine that the Enterprise has 500 branches, hundreds of servers, and only 10 branches and two servers need IPv6 support. Additionally, the distribution and core layers were rightfully designed with redundancy, so to support 10 branches, 30-40 routers may need IPv6 support added to support IPv6 natively just to cover all possible cases of links or routers failing. Alternatively, a tunnel could be created from each branch router to the new IPv6 server subnet. In such cases, the use of tunneling can give you a much quicker and easier start to the journey toward IPv6.

Point-to-Point IPv6 Tunnels

Some tunnels use a point-to-point concept, whereas others use a multipoint concept. For point-to-point, two devices (and only two) sit at the ends of the tunnel, as did routers R1 and R3 in Figure 18-2. These point-to-point tunnels work like a virtual point-to-point serial link. Figure 18-3 shows these concepts, plus a few other details, using the same underlying network as shown in Figures 18-1 and 18-2.



Key Topic

Figure 18-3 *IPv6 Point-to-Point Tunnel Concept*

To create the tunnel shown in the figure, each router configures a type of virtual interface called a *tunnel* interface. The configuration associated with the tunnel interfaces tells IOS the encapsulation details as previously shown in Figure 18-2. In the example of Figure 18-3, R1 uses tunnel interface 0, and R3 uses tunnel interface 3. The tunnel interface numbers can be any integer (up into the low billions), much like choosing loopback interface numbers, with no advantage or disadvantage of using any particular interface number.

The two routers on the ends of the tunnel treat the tunnel interfaces like serial interfaces on a point-to-point serial link, at least from a Layer 3 forwarding perspective. For example, to support IPv6, the engineer would actually enable IPv6 on the tunnel interfaces with the same commands shown in Chapter 16 and configure a routing protocol so that it runs over the tunnel interfaces, as shown in Chapter 17. In this case, Figure 18-3 shows IPv6 addresses assigned to the tunnel interfaces from within the 2013::/64 subnet.

When the configuration is complete, R1 and R2 exchange IPv6 routes and route IPv6 packets over the tunnel interface, which then triggers the encapsulation seen back in Figure 18-2.

Point-to-Multipoint IPv6 Tunnels

Multipoint IPv6 tunnels allow the sending router–the "point" if you will–to use a single tunnel interface to send packets to multiple remote routers. In some ways, a multipoint tunnel works much like a LAN, or even more like a Non-Broadcast Multi-Access (NBMA) network like Frame Relay. Multipoint tunnels still encapsulate the IPv6 packets, but they need additional logic so that the sending router (the "point") knows to which of several remote routers (the "multipoints") to send the encapsulating IPv4 packet.

The biggest leap in logic from point-to-point tunnels to point-to-multipoint tunnels is the logic in how a router chooses which of the many remote tunnel endpoints should receive a particular packet. Multipoint tunnels rely on either the IPv6 packet's destination address, or next-hop information in the IPv6 routing table, to determine which of the multiple remote devices should receive a given packet. This decision happens dynamically on the sending router. In some cases, this dynamic decision process can result in less configuration when adding a new member of the multipoint group. In other cases, the dynamic decision just happen to be how it works, but with no real advantage over point-to-point tunnels.

In all types of multipoint IPv6 tunnels, the tunneling process starts when the router receives an IPv6 packet and then tries to route that packet out the multipoint tunnel interface. This action triggers the logic by which the source router determines how to forward the IPv6 packet, inside an IPv4 packet, to the correct router. Figure 18-4 shows the general idea of how the logic works. In this case, R1 acts as the point–the encapsulating router that must dynamically decide to what IPv4 address to encapsulate and send the IPv6 packet. The figure shows one example of how R1 reacts when it receives an incoming IPv6 packet from the left.

Figure 18-4 illustrates the following steps:

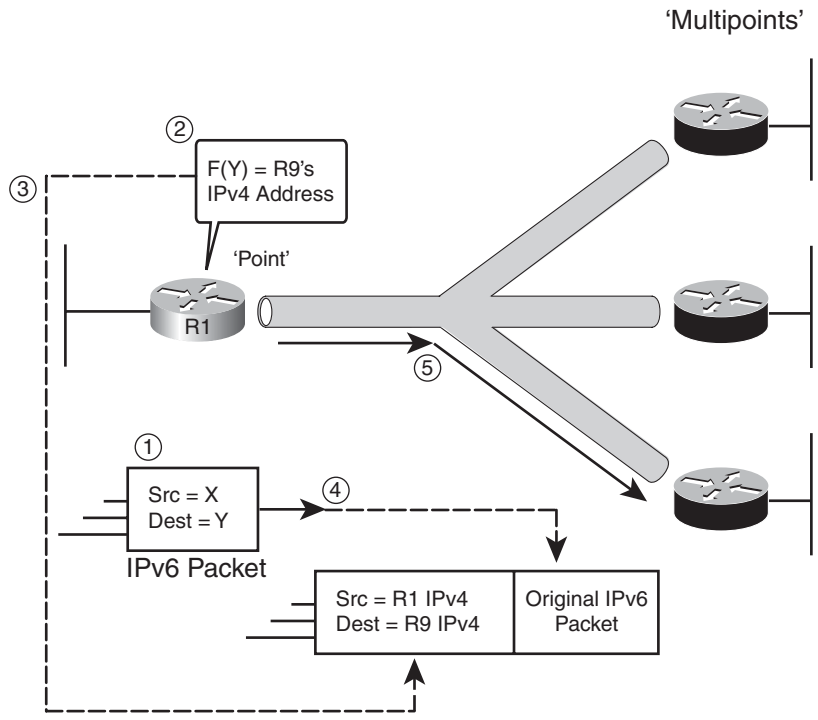
Step 1. R1 receives an IPv6 packet in its LAN interface and decides that the packet should be forwarded out its multipoint tunnel interface.

Step 2. R1 analyzes the destination IPv6 address (listed as Y in the figure), deriving the tunnel endpoint's IPv4 address (in this case, R9's IPv4 address).

Step 3. R1 builds an IPv4 packet header, with its own address as source address and using R9's IPv4 address as the destination (as derived at Step 2).

Step 4. R1 puts the original IPv6 packet into the new IPv4 packet.

Step 5. The IPv4 routers forward the IPv4 packet to R9.



Key Topic

Figure 18-4 *Dynamic Multipoint IPv6 Tunnel Concepts*

The particularly interesting part of this process relates to how the encapsulating router (R1 in this case) determines to what IPv4 address to send the IPv4 packet, shown as Step 2 in the figure. Two general options for these dynamic tunnels exist: automatic 6to4 tunnels and ISATAP tunnels. In both cases, the 32-bit IPv4 address of the destination is embedded in the destination IPv6 address. To make the whole process work, the network engineer must plan the IPv6 and IPv4 addresses so that they conform to various rules, so that the sending router can derive the correct destination IPv4 address.

Summary of IPv6 Tunneling Methods

Tunneling IPv6 over IPv4 gives engineers a tool with which to avoid a full native migration to IPv6 on all routers. However, all tunneling methods add more overhead to the routers that perform the tunneling encapsulation and decapsulation. The engineer designing each network needs to weigh the benefits of a full native IPv6 implementation versus a more gradual migration using IPv6 tunnels.

The tunnels themselves also have many pros and cons. Generally, the point-to-point tunnels work best when the IPv6 occurs regularly. In such cases, it should be worth the effort to run an IPv6 IGP over the tunnel. Multipoint tunnels generally work best when the IPv6 traffic occurs infrequently, or even in cases where the traffic volumes are less predictable. Also, the dynamic capabilities of multipoint tunnels allow hosts to form dynamic tunnels with routers, without requiring any additional router configuration.

The more detailed discussion of tunneling in the rest of this chapter more fully develops these pros and cons. For now, Table 18-2 lists the tunneling techniques discussed in more detail inside this chapter, along with some notes that will make more sense as you work through the chapter.

Table 18-2 *Summary of IPv6 Tunneling Options in This Chapter*

Method	Static or Dynamic	Topology	Advantages and Other Notes
Manually Configured	Static	Pt-pt	Acts like a virtual point-to-point link, supporting IPv6 IGPs. Good for more permanent tunnels. Supports IPv6 IGPs. Slightly less overhead than GRE.
GRE	Static	pt-pt	Generic Routing Encapsulation. Same advantages as previous row, plus it can support other Layer 3 protocols over the same tunnel.
6to4	Dynamic	Mpt	It may require less configuration than all other types when adding a new site. Supports global unicasts, with some extra configuration. Uses second and third quartets to store IPv4 address.
ISATAP	Dynamic	Mpt	It easily supports global unicast addresses for all prefixes. Uses seventh and eighth quartets to store IPv4 address.

NAT Protocol Translation

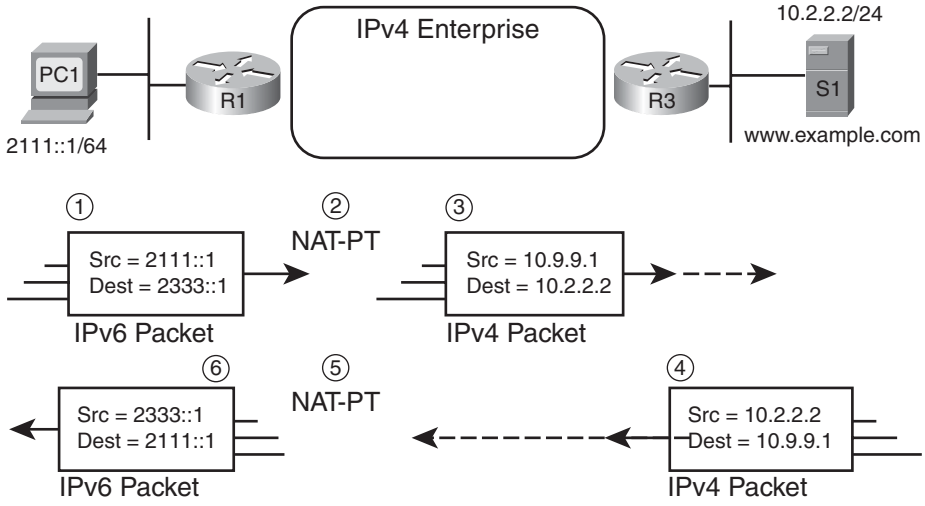
The migration toward IPv6 typically begins with no IPv6 installed but with pervasive IPv4 support on hosts, routers, and other devices. Next, engineers start adding IPv6 support, typically making hosts dual stacked. To support those hosts, network engineers either configure IPv6 natively in the network, or use some form of tunneling. Eventually, some devices migrate fully to IPv6. Those devices may be new devices that speak only to IPv6 servers. They may be traditional end-user clients that need to communicate only with IPv6-capable servers.

In all these cases, when any pair of hosts communicate, they both support the same protocol, whether IPv4 and IPv6. Because they send and receive either IPv4 packets, or both send and receive IPv6 packets, the network simply needs to deliver those packets. The packets may be encapsulated at some point in its journey, but if the packet left one host as a particular version of IP, the packet received by the other host is the same version.

However, at some time during the migration toward IPv6, the network may need to support the ability for an IPv4-only host to communicate with an IPv6-only host. The IPv6

migration and coexistence RFCs actually make allowance for such a feature, but to do so, something between the two hosts must translate between the two different protocols (IPv4 and IPv6). The solution is the *Network Address Translation–Protocol Translation* (NAT-PT) feature.

NAT-PT gets the first part of its name from the old IPv4 Network Address Translation (NAT) feature. The venerable IPv4 NAT translates the IP addresses inside an IPv4 header, most often changing the Enterprise host's private IPv4 address into a public, Internet-routable IPv4 address. NAT-PT translates both the source and destination IP address, translating between an IPv4 and IPv6 address for both. Not only does NAT-PT translate IP addresses, but also it translates the entire IPv4 and IPv6 header, plus other headers as well, such as TCP, UDP, and ICMP. Figure 18-5 shows an example of the results of NAT-PT on Router R1, with PC1 sending a packet to Server S1, and S1 returning the packet.



Key Topic

Figure 18-5 *NAT-PT Translation on Router R1*

Following the steps in the figure

Step 1. PC1, an IPv6-only host, sends an IPv6 packet to destination 2333::1, source 2111::1. And 2333::1 actually resides on R1, so the IPv6 network forwards the packet to R1.

Step 2. R1, with NAT-PT configured, listens for packets sent to 2333::1. R1 converts the IPv6 header and other packet headers to IPv4 standards.

Step 3. The IPv4 network forwards the packet to IPv4 host Server S1.

Step 4. S1 sends an IPv4 packet back to what it thinks of as PC1, source IPv4 address 10.2.2.2, destination IPv4 address 10.9.9.1. 10.9.9.1 actually resides on Router R1, so the IPv4 network forwards the IPv4 packet to R1.

Step 5. R1, with NAT-PT configured, listens for IPv4 packets sent to 10.9.9.1. R1 converts the IPv4 header and other packet headers to IPv6 standards.

Step 6. R1 forwards the IPv6 packet through the IPv6 side of the network to PC1, the IPv6-onlt host.

For the process in Figure 18-5 to work, NAT-PT must also be heavily involved in DNS flows as well. For example, before the flow in Figure 18-5, PC1 will have sent a DNSv6 request to resolve the name of the server (www.example.com in this case). The router performing NAT-PT must convert the requests between DNSv4 and DNSv6, keeping track of the names and address bindings so that the NAT-PT translation process converts to the correct addresses.

Note: NAT-PT, while interesting, has actually been moved to historic status (RFC 2766). IOS still supports NAT-PT, and the ROUTE course's e-learning component still includes coverage of NAT-PT. However, NAT-PT will likely fade away over time. Protocols meant to replace the function of NAT-PT are under development as of the writing of this book.

This completes the overview of IPv6 coexistence tools. The final two major sections of this chapter examine the various IPv6 tunneling methods in depth.

Static Point-to-Point IPv6 Tunnels

This section examines the configuration and some basic verification for two different types of IPv6 point-to-point tunnels:

- Manually configured tunnels (MCT)
- Generic Routing Encapsulation (GRE) tunnels

Note: The phrase *manually configured tunnels* refers to an RFC standard encapsulation of IPv6 inside IPv4 packets. There is no formal name for this feature, so most documents refer to it as "manually configured tunnels," "manual tunnels," or "configured tunnels." This chapter uses the term manually configured tunnels (MCT) for consistency and easier reference.

MCT and GRE tunnels have many similarities, including the configuration. They both create a virtual point-to-point link between two IPv4 routers for the purpose of supporting the forwarding of IPv6 packets. IPv6 IGP routing protocols can run over these virtual links. To support the IGPs, plus other features, the routers will assign link local addresses on these links and allow the forwarding of IPv6 multicast traffic. Both types allow the configuration of additional security features over the tunnel interfaces. Finally, both require static configuration of both the tunnel source and the tunnel destination IPv4 addresses.

To the depth used for these topics in this book, these two tunneling options have only minor differences. Manually configured tunnels simply conform to the rules about generic IPv6 tunnels outlined in RFC 4213, which defines how to encapsulate IPv6 inside an IPv4 packet, without additional headers. GRE tunnels use a generic encapsulation originally defined by Cisco and later standardized in RFC 2784. GRE uses an additional stub header

between the IPv4 and IPv6 header; this extra header includes a protocol type field, which allows a GRE tunnel to carry many passenger protocols. (The passenger protocol is the protocol encapsulated inside another protocol's header.) GRE also supports several transport protocols (transport protocols encapsulate the passenger protocol), although IPv4 typically is the only transport protocol used for tunneling IPv6. GRE's flexibility allows a single GRE tunnel to carry IPv6 plus other traffic as well, whereas manually configured tunnels cannot.

This section examines manually configured tunnels in detail and then compares GRE tunnels to manually configured tunnels.

Manually Configured Tunnels

The main planning task for an implementation plan that includes manually configured tunnels requires a little thought about the tunnel source and destination IPv4 addresses. Other than that, the configuration uses straightforward commands that refer to the key elements in the conceptual drawings in Figures 18-2 and 18-3. However, even though the individual commands are straightforward, the configuration requires several steps. So, this section breaks the configuration into two sections: tunnel configuration steps and IPv6 configuration steps.

Configuring and Verifying a Manually Configured Tunnel

To configure the tunnel, first you need to choose what IPv4 addresses will be used when encapsulating packets (as shown in Figure 18-2). A router's tunnel interface borrows the IPv4 address on some other interface; the router then uses that IPv4 address as the source address when encapsulating packets. Comparing the configuration on the two routers on either end of the tunnel, the two routers must agree to the correct IPv4 addresses: the source address used by one router should match the other router's tunnel destination IPv4 address and vice versa. Finally, for better availability, if any IPv4 redundancy exists between the two routers, the engineer should choose to use loopback interface IPv4 addresses, because the tunnel interface fails if the interface associated with the source IP address fails.

The configuration centers around the configuration of a virtual interface called a tunnel interface. The tunnel interface creates an object that can be configured like an interface, so that additional commands can set parameters for a given tunnel. For instance, the engineer configures the source and destination IPv4 addresses (two different interface subcommands), with another command to configure the tunnel encapsulation (GRE or manually configured tunnel). The following list outlines the steps to configure the tunnel.

Step 1. Find the tunnel IPv4 addresses planned for the tunnel, and ensure that each router can forward IPv4 packets between the addresses. If using a new loopback interface, create the loopback using the **interface loopback** *number* command, assign it an IPv4 address with the **ip address** command, and confirm that routes for this interface will be advertised by IPv4.

Step 2. Create a tunnel interface using the **interface tunnel** *number* command, selecting a locally significant integer as the tunnel interface number.

Step 3. Define the source IPv4 address of the tunnel using the **tunnel source** {*interface-type interface-number* | *ipv4-address*} interface subcommand. (This address must be an IPv4 address configured on the local router.)

Step 4. Define the destination IPv4 address for the encapsulation using the **tunnel destination** *ipv4-address* interface subcommand; the address must match the **tunnel source** command on the other router.

Step 5. Define the tunnel as a manually configured tunnel (not GRE), as defined in RFC 4213, using the **tunnel mode ipv6ip** interface subcommand.

For example, consider Figure 18-6, which shows an updated representation of the Enterprise network shown in Figures 18-1, 18-2, and 18-3. In this case, even though no redundancy exists, the engineer plans to use loopback addresses (10.9.9.1 on R1 and 10.9.9.3 on R3). Example 18-1 that follows shows the tunnel configuration on R1 and R3.

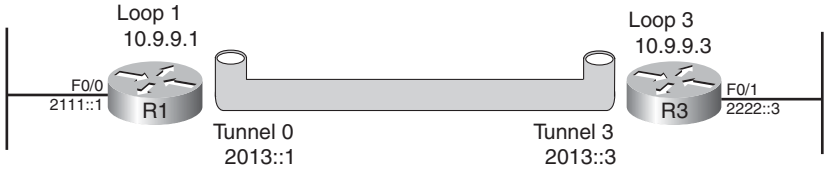


Figure 18-6 *Figure Showing Configuration Items in Example 18-1*

Example 18-1 *Configuring an IPv6IP Tunnel on R1 and R3*

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface loopback 1
R1(config-if)# ip address 10.9.9.1 255.255.255.255
R1(config-if)# interface tunnel 0
R1(config-if)# tunnel source loopback 1
R1(config-if)# tunnel destination 10.9.9.3
R1(config-if)# tunnel mode ipv6ip
```

```
! Next, on router R3
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface loopback 3
R3(config-if)# ip address 10.9.9.3 255.255.255.255
R3(config-if)# interface tunnel 3
R3(config-if)# tunnel source loopback 3
R3(config-if)# tunnel destination 10.9.9.1
R3(config-if)# tunnel mode ipv6ip
```

At this point, assuming IPv4 connectivity exists between the two tunnel endpoints, the tunnel interfaces should be up. A quick **show** command, as shown in Example 18-2, confirms the tunnel interface's status on R1, plus a few other interesting tidbits. Beside the tunnel interface being up, the output confirms the source and destination IPv4 addresses.

It also confirms that the tunnel mode uses IPv6 over IP (the word "IP" implying IPv4) to match the **tunnel mode ipv6ip** configuration command.

Example 18-2 *Verification of the Tunnel*

```
R1# show interfaces tunnel0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.9.9.1 (Loopback1), destination 10.9.9.3
  Tunnel protocol/transport IPv6/IP
  Tunnel TTL 255
  Tunnel transport MTU 1480 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
! the rest of the output is the typical counter information
```

Configuring and Verifying the Manually Configured Tunnel

The configuration shown in the previous section prepares the tunnel to encapsulate traffic inside IPv4 packets, but it does complete the configuration. The configuration also needs to enable IPv6 on the routers that create the tunnel, treating the tunnel interfaces just like you would for serial interfaces connected with a leased line.

The configuration steps mirror those discussed at some length in Chapters 16 and 17, so the specific commands will not be discussed here. However, Example 18-3 shows the configuration of both the tunnel and the IPv6 details on R1, with the newly added IPv6 commands highlighted.

Example 18-3 *Completed Tunnel and IPv6 Configuration on Router R1*

```
R1# show running-config
! Only pertinent portions retained
ipv6 unicast-routing
!
interface Loopback1
 ip address 10.9.9.1 255.255.255.255
!
interface Tunnel0
 no ip address
 ipv6 address 2013::1/64
 ipv6 eigrp 1
 tunnel source Loopback1
 tunnel destination 10.9.9.3
 tunnel mode ipv6ip
```

```
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 ipv6 address 2111::1/64
 ipv6 eigrp 1
!
ipv6 router eigrp 1
 eigrp router-id 1.1.1.1
 no shutdown
```

The new configuration should be somewhat familiar from the last few chapters, but a few items could use some emphasis. First, note that the tunnel interface does indeed have an IPv6 address configured, and EIGRP enabled, just as if it were a physical interface. However, the tunnel interface does not have, nor does not need, an IPv4 address. The tunnel interface needs only a Layer 3 address for the passenger protocols – in other words, the protocols carried as data inside the encapsulating packets. So, only an IPv6 address is needed on the interface. The rest of the highlighted IPv6 configuration uses the same logic discussed in Chapters 16 and 17.

The same verification commands seen in Chapters 16 and 17 can confirm the IPv6 details on the tunnel. Example 18-4 lists the output from a few of these commands, highlighting some of the key details.

Example 18-4 *Verifying IPv6 Works over the Tunnel*

```
R1# show ipv6 interface brief
FastEthernet0/0             [up/up]
    FE80::213:19FF:FE7B:5026
    2111::1
! irrelevant lines omitted for brevity
Loopback1                   [up/up]
    unassigned
Tunnel0                     [up/up]
    FE80::A09:901
    2013::1
R1# show ipv6 interface tunnel0
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::A09:901
  No Virtual link-local address(es):
  Global unicast address(es):
    2013::1, subnet is 2013::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::A
    FF02::1:FF00:1
```

```
    FF02::1:FF09:901
  MTU is 1480 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 42194)
  Hosts use stateless autoconfig for addresses.
R1# traceroute ipv6 2222::3

Type escape sequence to abort.
Tracing the route to 2222::3
  1 2013::3 0 msec 4 msec 4 msec
```

The first two commands show how routers create the link local IPv6 addresses for manually configured tunnel interfaces. Routers normally form a serial interface's link local IPv6 address using EUI-64 rules based on the MAC address of the first LAN interface on the router. In this case, the router forms the link local address with a FE80::/96 prefix and then adds the 32-bit tunnel source IPv4 address as the last 32 bits. For example, the first two **show** commands in Example 18-4 list R1's link local address on its tunnel interface: FE80::A90:901. Adding back the leading 0s in the last two quartets, the last two quartets look like this:

0A09:0901

Converting each pair of hex characters to decimal, you can find the IPv4 address of 10.9.9.1. (Appendix B, "Conversion Tables," includes a hex/decimal conversion table.)

The **show ipv6 interface tunnel0** command lists the usual IPv6 addresses. It lists the link local address, along with the statically configured global unicast address (2013::1). It lists multicast addresses, including EIGRP's FF02::A, and the (highlighted) solicited node multicast address associated with both the link local and the global unicast address on the interface. (When a multicast needs to be sent out a manually configured tunnel's interface, the router encapsulates the IPv6 multicast and forwards it inside a unicast IPv4 packet to the other end of the tunnel.)

The **traceroute ipv6 2222::3** command at the end of the example confirms that IPv6 traffic can pass over the tunnel.

GRE Tunnels

Only one difference exists in the configuration between manually configured tunnels and point-to-point GRE tunnels: the tunnel mode. IOS uses the **tunnel mode ipv6ip** command for manually configured tunnels, which also tells IOS to use the encapsulation as defined in RFC 4213. IOS uses the **tunnel mode gre ip** command to instead configure a GRE tunnel, meaning that IOS uses the RFC 2784 GRE encapsulation. This encapsulation adds the GRE header to the encapsulation. (Also, because IOS defaults to use GRE over IP, you can alternatively just omit the **tunnel mode** command as well.)

The configurations are indeed almost identical. For example, to migrate from the configuration shown in Example 18-3, which shows all of R1's configuration for both the tunnel and IPv6 with a manually configured tunnel, simply use one of the following two options:

- Issue the **tunnel mode gre ip** command on both routers' tunnel interfaces.
- Issue the **no tunnel mode ipv6ip** command on both routers' tunnel interfaces, which reverts to the default of GRE over IP.

Note: If the two routers' tunnel modes do not match, the tunnel interfaces can stay up/up, but the routers cannot forward IPv6 packets because of the mismatched encapsulations.

The verification command output looks almost identical as well, but with just a few differences to note. IOS uses a different convention for the link local address created for a GRE tunnel interface. It works as if the tunnel interface is a serial interface, deriving the interface ID using EUI-64 rules and the MAC address of the first LAN interface on the router. The second difference relates to how IOS automatically sets the MTU of the passenger protocols (IPv6 in this case) to 1476 for GRE tunnels; with manually configured tunnels, the passenger MTU was set to 1480. These settings allow space in both modes for the 20-byte additional IPv4 header that encapsulates the packet, plus in the case of GRE, the additional 4-byte GRE header. Example 18-5 highlights some of these minor differences.

Example 18-5 *Verifying GRE over IP Tunnels That Support IPv6*

```
R1# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.9.9.1 (Loopback1), destination 10.9.9.3
Tunnel protocol/transport GRE/IP
Key disabled, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255
Fast tunneling enabled
Tunnel transport MTU 1476 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
! the remaining omitted lines list interface counters
```

Point-to-Point IPv6 Tunnel Summary

Point-to-point IPv6 tunnels give engineers a means to implement IPv6 connectivity without requiring extensive deployment of native IPv6. At the same time, these tunnels act like native IPv6 links, running IPv6 IGPs, using link local addresses, and requiring no static

route configuration. The two options in this category have almost no practical difference. Table 18-3 summarizes the key features, with the small differences highlighted in the table.

Key Topic

Table 18-3 *Comparing Manual and GRE IPv6-over-IP Tunnels*

	Manual Tunnels	GRE
RFC	4213	2784
Tunnel mode command	**tunnel mode ipv6ip**	**tunnel mode gre ip**
Passenger MTU default	1480	1476
Supports IPv6 IGPs?	Yes	Yes
Forwards IPv6 multicasts?	Yes	Yes
Uses static configuration of tunnel destination?	Yes	Yes
Supports multiple passenger protocols?	No	Yes
Link local based on...	FE80::/96, plus 32 bits from tunnel source IPv4 address	IPv6 EUI-64, using lowest numbered interface's MAC address

Dynamic Multipoint IPv6 Tunnels

Multipoint tunnels give engineers a convenient tool to use when irregular or infrequent IPv6 traffic occurs between sites. The multipoint topology creates the possibility that new sites can join into the tunnel without requiring additional configuration on the existing routers. Additionally, these multipoint tunnels in some cases allow IPv6 hosts to act as tunnel endpoints, allowing a host at a remote site to connect into the Enterprise IPv6 network even if the local router has no knowledge of IPv6. This flexibility and ability to add routers and hosts with potentially no extra configuration makes multipoint IPv6 tunnels a very useful migration tool.

Multipoint tunnels also have some disadvantages. To take advantage of the possibility to limit future configuration changes, IPv6 address planning must follow some additional rules and constraints. These tunnels also do not support IPv6 IGPs, requiring the use of either static routes or multiprotocol BGP. The dynamic forwarding logic requires more work per packet as compared with point-to-point tunnels, which is one of the main reasons multipoint tunnels are best used for less frequent traffic, with point-to-point tunnels best used for more frequent traffic. Finally, the additional addressing rules and considerations require a bit more of a learning curve to become comfortable with these tools, at least as compared with point-to-point tunnels.

This section examines the configuration of two types of multipoint tunnels: automatic 6to4 tunnels, as defined in RFC 3056, and ISATAP tunnels, as defined in RFC 4214.

Automatic 6to4 Tunnels

As previously described in the section "Point-to-Multipoint IPv6 Tunnels," a multipoint tunnel does not explicitly define the tunnel endpoint IPv4 addresses. Instead, the incoming IPv6 packet's destination IPv6 address implies the IPv4 address that a router should use when encapsulating and forwarding the packet. Because the tunnels rely on the IPv6 address to determine the destination IPv4 address for these tunnels, network engineers must spend more time initially planning IPv6 and IPv4 addresses used to deploy IPv6.

The first big planning and design choice when using automatic 6to4 tunnels relates to whether to use global unicast addresses for the end user subnets, or whether to use a special reserved range of addresses (2002::/16). If the Enterprise expects all Internet traffic to/from the Enterprise to remain IPv4-only for the foreseeable future, then the IPv6 addresses used in the Enterprise do not matter much, and the network engineer can take advantage of the 2002::/16 reserved range. Using this range allows new sites to be added to the multipoint tunnel at a later time, without requiring new configuration on the existing routers in a multipoint tunnel. However, if the Enterprise needs to use its registered global unicast site prefix, automatic 6to4 tunnels can still be used, just with a little more configuration effort over time.

This section first examines the case where the Enterprise needs no IPv6 Internet connectivity, using the 2002::/16 reserved range of addresses. Following that, this section examines the same tool, this time using a registered global unicast site prefix.

Using the Automatic 6to4 Prefix for All Devices

RFC 3056 defines a reserved range of IPv6 addresses for use with automatic 6to4 tunnels: 2002::/16. Even though this range appears to come from the range of global unicast IPv6 addresses (2000::/3), IANA reserves the 2002::/16 prefix as a set of addresses that will never be assigned as global unicast addresses.

By starting with the 2002::/16 prefix, a network engineer can then create a /48 prefix:

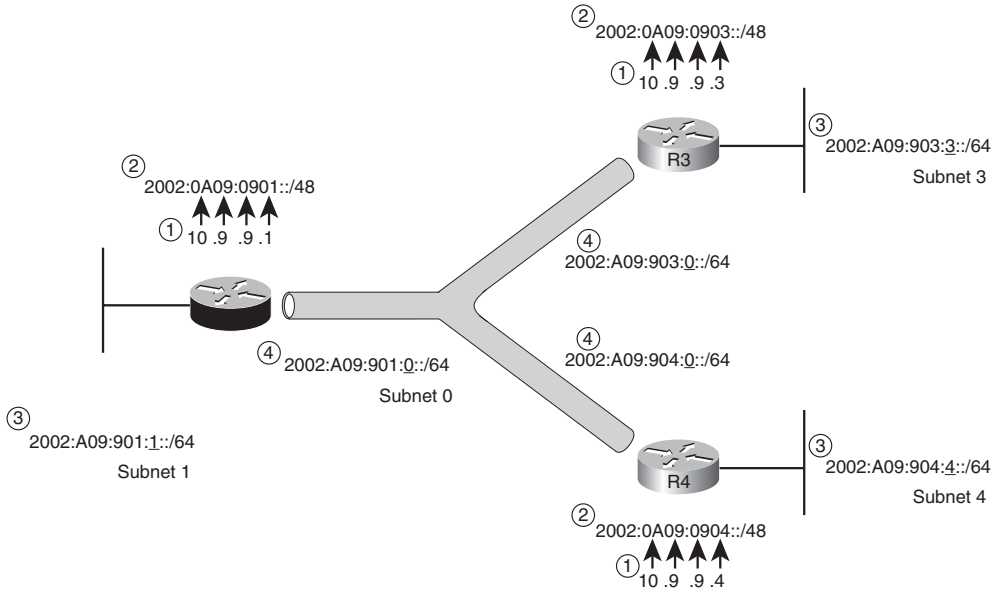
- The network engineer can assign each tunnel endpoint (router or host) its own /48 prefix, used for all prefixes connected to that local router, by adding the hex version of the router's IPv4 address as bits 17 through 48 (quartets 2 and 3).
- The engineer can allocate /64 prefixes for each required subnet connected to each router by allocating a unique subnet number in the fourth quartet (much like when an Enterprise receives a /48 site prefix from an IPv6 registrar).

Figure 18-7 shows the format of the automatic 6to4 tunnel IPv6 addresses.

Figure 18-7 *Reserved Automatic 6to4 IPv6 Addresses*

The first half of the address has three major parts, with the second half of the address structure used for the host ID as with most implementations of global unicast addresses. The addresses always begin with a first quartet of 2002. The second and third quartet list the hex version of the IPv4 address for that site–usually the IPv4 address of a loopback interface on a router. (In this case, the hex value AABBCCDD represents 170.187.204.221, found by converting each pair of hex digits to the decimal equivalent.) The fourth quartet can be conveniently used as a subnet field so that the engineer can assign the various subnets connected to each router. The /48 prefix chosen for each router works much like when a registrar gives a company a /48 global unicast prefix, leaving 16 bits for the company to use for subnetting.

For example, consider the case with a multipoint tunnel with three routers, R1, R3, and R4, as shown in Figure 18-8. The figure depicts the planning steps taken by the engineer when using the 2002::/16 prefix for all IPv6 addresses.



Key Topic **Figure 18-8** *Example Planning Diagram for 2002::/16 Prefixes*

Following the steps in Figure 18-8

Step 1. The engineer chooses an IPv4 address on each router to use as the tunnel endpoints. In this case, he chose loopback IPv4 addresses, which is typical.

Step 2. The engineer then derived the /48 prefix used for allocating subnets off each router by converting each octet of the IPv4 addresses to hex to form the /48 prefix used on each router.

Step 3. The engineer then allocates the first /64 prefixes, one for each router's LAN subnet. The IPv6 hosts on these LANs uses IPv6 addresses from these prefixes.

Step 4. The engineer picks an IPv6 address to use on each tunnel interface. With automatic 6to4 tunnels, these IPv6 addresses typically come from each router's own prefix, so each router's tunnel IPv6 address will actually be in different IPv6 subnets.

That completes an example of the address planning, but the underlying logic is not yet complete. Multipoint IPv6 tunnels do not support IGP routing protocols, but to trigger the dynamic encapsulation process for the tunnel, the routers must route IPv6 traffic out the tunnel interface. The solution is actually simple: because all the IPv6 addresses start with 2002::/16 (by implementation choice), the engineer plans a static route for prefix 2002::/16, forwarding all packets destined for these special 2002::/16 addresses out the multipoint tunnel interface. This process triggers the tunnel as previously discussed for Figure 18-4.

So, what happens when an IPv6 packet now arrives at one of these routers? With these well-chosen IPv6 address, the following occurs:

Step 1. The packet's IPv6 destination address begins with 2002, so the router tries to forward the packet out its tunnel interface, triggering the process.

Step 2. The router notices the tunnel type (automatic 6to4), which tells IOS to encapsulate and send the IPv6 packet to the destination IPv4 address found in quartets 2 and 3.

Configuring the Automatic 6to4 Tunnel

The configuration of the automatic 6to4 tunnel has some similarities and some differences compared to point-to-point tunnels. For the tunnel itself, unlike point-to-point tunnels, the multipoint tunnel configuration does not need a configured **tunnel destination** command, because the destination IPv4 address is instead embedded in the destination IPv6 address. Also, the setting of the **tunnel mode** command differs. Otherwise, the tunnel configuration details remain the same.

Some differences also exist for the IPv6-specific configuration related to the tunnel. An IPv6 IGP is not needed, because the router finds the destination IPv4 address on the other end of the tunnel embedded in the destination IPv6 address in the received packet. Although IPv6 IGPs cannot be configured on the automatic 6to4 tunnel, static IPv6 routes and multiprotocol BGP (for IPv6) can be configured. And even though the tunnel interface must be enabled for IPv6, as with point-to-point tunnels, the IPv6 addresses are not in the same subnet as the other routers' tunnel interfaces.

The following list outlines the steps to configure the automatic 6to4 tunnel and to enable IPv6 forwarding across the tunnel. Note that the steps list the use of a loopback interface because most sites use a loopback, but any interface's IPv4 address may be used.

Step 1. Configure the planned loopback interface (**interface loopback** *number* global command), and assign the planned IPv4 address. (Ensure that the IPv4 IGP advertises a route for this address.)

Step 2. Create a tunnel interface using the **interface tunnel** *number* command, selecting a locally significant integer as the tunnel interface number.

Step 3. Define the source IPv4 address of the tunnel using the **tunnel source** {*interface-type interface-number* | *ipv4-address*} interface subcommand using the loopback IP address from Step 1.

Step 4. Do NOT define a tunnel destination with the **tunnel destination** interface subcommand.

Step 5. Identify the tunnel as an automatic 6to4 tunnel using the **tunnel mode ipv6ip 6to4** interface subcommand.

Step 6. Enable IPv6 on the tunnel interface, typically with the **ipv6 address** interface subcommand.

Step 7. Complete the normal IPv6 configuration, include defining the LAN interface IPv6 addresses per the planning chart, and enable IPv6 routing with the **ipv6 unicast-routing** command.

Step 8. Define a static route for 2002::16, with outgoing interface of the tunnel interface, using the **ipv6 route 2002::/16 tunnel** *number* global command.

Example 18-6 shows the configuration on Router R1 from Figure 18-8. Note that although not shown, the routers in the figure do exchange IPv4 routes for their respective loopback addresses so that all IPv4 loopbacks in Figure 18-7 are pingable. The routers also use IPv6 addresses on their interfaces, rather than just having IPv6 enabled with the **ipv6 enable** command: 2002:A09:901::1 on R1, 2002:A09:903::3 on R3, and 2002:A09:904::4 on R4.

Example 18-6 *R1 Configuration for an Automatic 6to4 Tunnel*

```
ipv6 unicast-routing
!
interface Loopback1
 ip address 10.9.9.1 255.255.255.255
!
interface Tunnel0
 no ip address
 ipv6 address 2002:a09:901::/128
 tunnel source Loopback1
 tunnel mode ipv6ip 6to4
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 ipv6 address 2002:A09:901:1::1/64
!
ipv6 route 2002::/16 Tunnel0
```

This configuration defines all the small pieces listed in the configuration checklist, but the configuration does not spell out the underlying logic. Looking at the configuration, along

with the design and the knowledge of how automatic 6to4 tunnels work, the following logic occurs in the background:

Step 1. R1 will have two connected IPv6 routes (F0/0's and tunnel0's) in the 2002::/16 range, plus a static route for the entire 2002::/16 range.

Step 2. When R1 receives an IPv6 packet, destination in the 2002::/16 range, and the destination is not in one of the connected subnets, R1 will try to forward the packet out tunnel0.

Step 3. The **tunnel mode ipv6ip 6to4** command tells R1 to look to the 2nd/3rd octets to find the destination IPv4 address, and perform the tunneling.

Also, take another moment to review the configuration in Example 18-6, looking specifically for any configuration on R1 that references anything specific about either R3 or R4 in Figure 18-8. No such specifics exist. R1's logic works without any preconfigured addressing information about R3, and to R4, and to hundreds of future routers, if the engineer plans the IPv6 addresses correctly. So, if this Enterprise decided to enable the automatic tunnel on yet another router, only the configuration in that one new router would need to be changed, but the configuration on R1, R3, and R4 would be unchanged. The rest of the routers would not need any more configuration because the static route for 2002::/16 triggers the routing of IPv6 packets, tunneled in IPv4, to the correct remote routers.

Example 18-7 lists the **show** commands that confirm some of these facts.

Example 18-7 *R1's State When Using the Automatic 6to4 Tunnel*

```
R1# show ipv6 route
IPv6 Routing Table - Default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2002::/16 [1/0]
     via Tunnel0, directly connected
LC  2002:A09:901::/128 [0/0]
     via Tunnel0, receive
C   2002:A09:901:1::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2002:A09:901:1::1/128 [0/0]
     via FastEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive

R1# show ipv6 interface brief
FastEthernet0/0            [up/up]
```

```
    FE80::213:19FF:FE7B:5026
    2002:A09:901:1::1
FastEthernet0/1             [administratively down/down]
    unassigned
! unrelated lines omitted
Tunnel0                     [up/up]
    FE80::A09:901
    2002:A09:901::

R1# trace 2002:a09:903:3::3

Type escape sequence to abort.
Tracing the route to 2002:A09:903:3::3

  1 2002:A09:903::3 4 msec 4 msec 4 msec
```

The **show ipv6 route** command lists the static route and both connected routes. Note that in this case, the configuration used a /128 prefix length for Tunnel0's IPv6 address, so the connected route shows up on the same line as the local route, both of which have a /128 prefix length. The **show ipv6 interface brief** command confirms the configured IPv6 address from the 2002::/16 range, plus the link local address derived for the Tunnel0 interface. Note that link local address uses an unusual convention–instead of using the usual EUI-64 logic, the link local address starts with FE80::/96, with the last two quartets the same as the tunnel's IPv4 source address. Finally, the **traceroute** command at the end of the example proves the tunnel works.

Using Global Unicasts with Automatic Tunnels

The previous example of automatic 6to4 tunnels assumed that the engineer chose all IPv6 addresses from the reserved 2002::/16 range because no need existed for IPv6 Internet connectivity. This section looks at the same automatic 6to4 concepts but with global unicasts used for the LAN subnets.

The big conceptual difference that occurs when using global unicast addresses relates to when using an assigned global unicast prefix, the second and third quartets are set, are the same for all subnets in the Enterprise, and can no longer be used to store the tunnel destination's IPv4 address. For example, if an Enterprise were assigned 2000:0:1::/48 as a site prefix, then all the subnets at the Enterprise would need to begin 2000:0:1, instead of encoding the IPv4 address of the site in the second and third quartets.

The problem can be overcome, however, through a piece of logic used when the router forwards packets out the tunnel interface after doing a recursive route lookup:

> When the router matches a route with an outgoing 6to4 tunnel interface, but with no next-hop IPv6 address, AND this route was matched due to route recursion, then derive the tunnel's destination IPv4 address based on the previously matched route's next-hop IPv6 address.

Although true, the formal definition is a bit much to work through. So, consider Figure 18-9, which shows a revised version of Figure 18-8. The figure uses the same three routers, same IPv4 loopback addresses, and same IPv6 addresses on the tunnel interfaces. However, for the three LANs, the engineer chose IPv6 prefixes from within the site's 2000:0:1::/48 prefix, essentially making R1's LAN Subnet 1, R3's LAN Subnet 3, and R4's LAN Subnet 4.

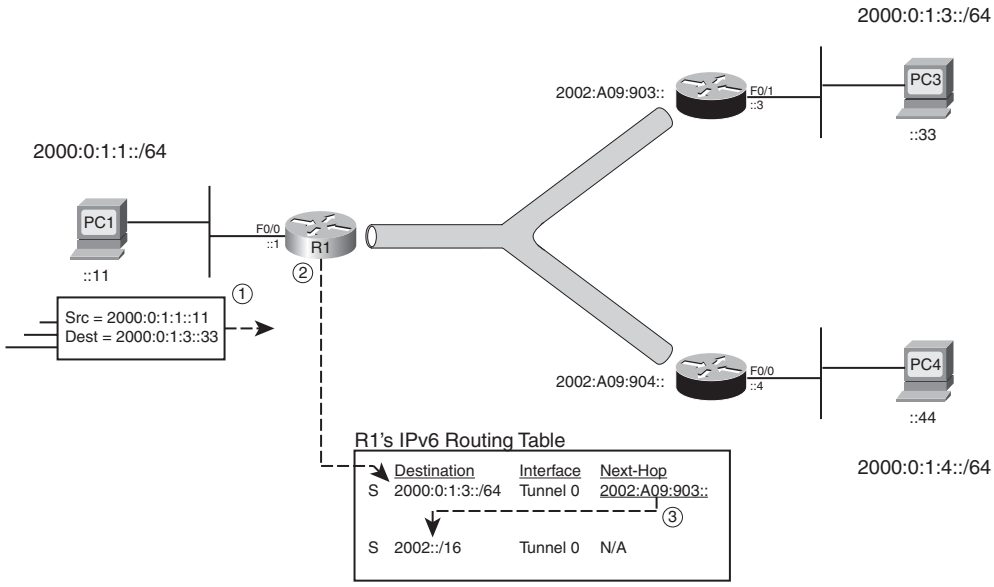


Figure 18-9 *Automatic Tunnels, Using Global Unicast IPv6 Addresses*

The underlying logic hinges on the two static routes in R1's IPv6 routing table, as shown in Figure 18-9. One route is the old route for 2002::/16, whereas the other route, added because of the use of global unicast addresses, matches the prefix for R3's LAN. Following the numbered sequence in the figure

Step 1. PC1 sends an IPv6 packet to PC3, destination address 2000:0:1:3::33.

Step 2. R1 compares the destination IPv6 address, matching the first route (destination 2000:0:1:3::/64) with outgoing interface Tunnel0 and next-hop router 2002:A09:903:: (R3's tunnel IPv6 address).

Step 3. R1 needs to decide how to forward packets to 2002:A09:903::, so R1 performs route recursion to find the matching route for this destination. R1 matches the

static route for 2002::/16 with outgoing interface tunnel0 and with no next-hop address listed.

At this point, the usual automatic 6to4 tunnel logic kicks in but based on the first route's next-hop address of 2002:a09:903::.

Summarizing, the differences in planning and configuration for using global unicasts with automatic 6to4 tunnels are

Step 1. Plan the prefixes and addresses for the LANs using the global unicast range assigned to the Enterprise.

Step 2. Configure an additional static route for each remote subnet, configuring the tunnel as outgoing interface and configuring the next-hop IPv6 address. That next-hop must be the remote router's tunnel IPv6 address, which embeds the destination IPv4 address as the second and third octets.

Note: You also can use BGP for IPv6 to learn the route listed in Step 2.

For R1 to forward traffic to the IPv6 hosts PC3 and PC4 in Figure 18-9, R1 would need the following two additional routes:

- **ipv6 route 2000:0:1:3::/64 tunnel0 2002:a09:903::**
- **ipv6 route 2000:0:1:4::/64 tunnel0 2002:a09:904::**

Finally, the introduction to this section mentioned that the use of global unicast addresses required more configuration changes. When a new router is added to the multipoint tunnel, each router already on the tunnel needs to add additional static routes or the alternative additional BGP configuration.

IPv6 ISATAP Tunnels

You can use ISATAP–the Intra-site Automatic Tunnel Addressing Protocol–to identify the IPv4 address of the remote site for the purposes of tunneling IPv6 packets. As a result, you can create dynamic multipoint tunnels using ISATAP, in general a concept much like the multipoint tunnels created using automatic 6to4 tunnels.

Comparing ISATAP and Automatic 6to4 Concepts

ISATAP tunnels differ in some ways with automatic 6to4 tunnels. However, ISATAP IPv6 tunnels use concepts that closely match those used by automatic 6to4 tunnels when using global unicast addresses (as previously shown in Figure 18-9). The following list makes some important comparisons between these two options with the items that differ between ISATAP tunnels and automatic 6to4 tunnels highlighted in gray:

Key Topic

Step 1. ISATAP tunnels use global unicast prefixes for user subnets.

Step 2. ISATAP tunnel interfaces use IPv6 addresses that embed the tunnel's destination IPv4 address inside the IPv6 address.

Step 3. The routers need static routes for the destination end-user IPv6 prefixes; the route must list a next-hop IPv6 address, which in turn embeds the tunnel destination IPv4 address.

Step 4. ISATAP tunnel interface IPv6 addresses embed the IPv4 address in the last two quartets.

Step 5. ISATAP tunnels do not use a special reserved range of IPv6 addresses at all, instead using just normal IPv6 unicast prefixes.

Step 6. ISATAP tunnels typically use a single prefix to which all tunnel interfaces connect, so all routers have a connected IPv6 route to that same subnet.

Step 7. ISATAP tunnels can automatically derive the tunnel interface's interface ID by using modified EUI-64 rules.

As usual, an example can help sift through some of the details; Figure 18-10 shows just such an example. The figure is the same as Figure 18-9's automatic 6to4 tunnel example with the same user IPv6 prefixes from the global unicast range. However, the design and configuration has been changed to work with ISATAP tunnels, as follows:

- The three tunnel interfaces now have IPv6 addresses in common IPv6 subnet 2000:0:1:9::/64. (The actual subnet number is not important; just choose a currently unused IPv6 subnet.)
- The tunnel interfaces' IPv6 addresses conform to *modified* EUI-64 rules (explained following the figure) embedding the IPv4 address in the last two quartets.

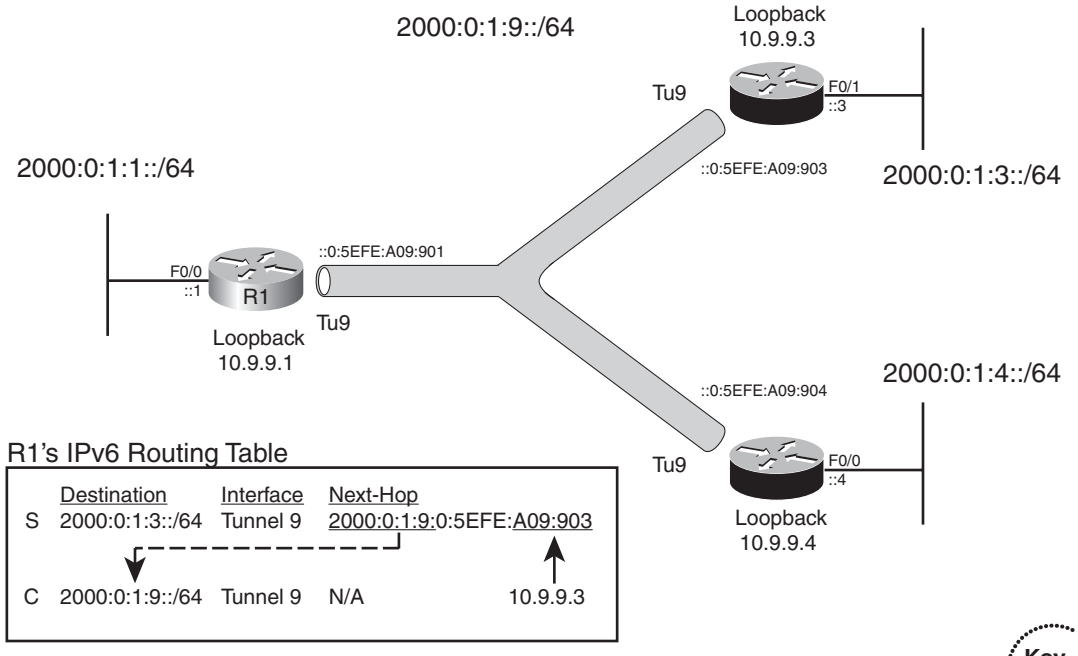


Figure 18-10 *ISATAP Tunnel Example, with Minimal Changes Compared to Figure 18-9*

Key Topic

- The routers no longer need a route for 2002::/16, instead relying on the connected route created for subnet 2000:0:1:9::/64.

First, examine the four IPv6 global unicast prefixes in the figure. The three LAN-based prefixes match the same ones used in the example of Figure 18-9. The fourth prefix includes all three routers' tunnel interfaces, much like a typical LAN. All three router's tunnel interfaces use an IPv6 address from this same prefix (2000:0:1:9::/64), and each will configure their address using a /64 prefix length.

Next, consider R1's routing logic when it receives an IPv6 packet destined for R3's LAN IPv6 prefix. R1's routing table lists a static route for R3's LAN subnet (2000:0:1:3::/64), with R3's tunnel IPv6 address listed as next hop (2000:0:1:9:0:5EFE:A09:903). When R1 receives an IPv6 packet destined for R3's LAN IPv6 subnet, R1 matches this static route. R1 also notices that the outgoing interface is not only a tunnel, but also an ISATAP tunnel, so R1 derives the tunnel's destination IPv4 address from the last two quartets of the next-hop address of the route (A09:903). (These values convert to 10.9.9.3.) R1 can then encapsulate and send the IPv4 transport packet to 10.9.9.3.

Beyond the detail in Figure 18-10, the only other item that needs some explanation is the format of the ISATAP-defined IPv6 address. The addresses can be configured manually but can also be derived by the router using *modified EUI-64 rules*. The rules work as follows:

Key Topic

- Configure a 64-bit prefix on the tunnel interface, and use the **eui-64** parameter, telling the router to derive the second half (interface ID) of the address.
- The router adds 0000:5EFE as quartets 5 and 6.
- The router finds the tunnel's source IPv4 address, converts it to hex, and adds that as quartets 7 and 8.

Configuring ISATAP IPv6 Tunnels

The overall configuration includes many steps, as usual, but most of the steps mirror the steps seen in the automatic 6to4 tunnel configurations. The following list outlines the steps. It also highlights in gray the steps not used when configuring automatic 6to4 tunnels for the case that used addresses in the 2002::/16 range exclusively.

Key Topic

Step 1. Configure the planned loopback interface and its IPv4 address, ensuring the IPv4 IGP advertises a route for this address.

Step 2. Create a tunnel interface using the **interface tunnel** *number* command.

Step 3. Define the tunnel source (**tunnel source** {*interface-type interface-number* | *ipv4-address*}) using the loopback IP address from Step 1.

Step 4. Do NOT define a tunnel destination with the **tunnel destination** interface subcommand.

Step 5. Identify the tunnel as an ISATAP tunnel using the **tunnel mode ipv6ip isatap** interface subcommand.

Step 6. Configure an IPv6 prefix with EUI-64 option using the e **ipv6 address***prefix/length***eui-64** interface subcommand.

Step 7. Complete the normal IPv6 configuration, include defining the LAN interface IPv6 addresses per the planning chart and enabling IPv6 routing with the **ipv6 unicast-routing** command.

Step 8. Define static IPv6 routes (using the **ipv6 route** global command) for each destination IPv6 prefix, with an outgoing interface and next-hop address. (The next-hop should be the destination router's IPv6 address that embeds the IPv4 address as the last two octets.)

Example 18-8 shows the configuration on Routers R1 and R3 from Figure 18-10 with the tunnel details highlighted.

Example 18-8 *Configuring R1 and R3 for an ISATAP Tunnel per Figure 18-10*

```
! First, on router R1:
R1# show running-config
! only relevant portions shown
ipv6 unicast-routing
!
interface Loopback1
 ip address 10.9.9.1 255.255.255.255
!
interface Tunnel9
 no ip address
 ipv6 address 2000:0:1:9::/64 eui-64
 tunnel source Loopback1
 tunnel mode ipv6ip isatap
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 ipv6 address 2000:0:1:1::1/64
!
ipv6 route 2000:0:1:3::/64 2000:0:1:9:0:5EFE:A09:903
ipv6 route 2000:0:1:4::/64 2000:0:1:9:0:5EFE:A09:904
```

```
! Next, on router R3:

R3# show running-config
! only relevant portions shown
ipv6 unicast-routing
!
interface Loopback3
 ip address 10.9.9.3 255.255.255.255
!
interface Tunnel9
 no ip address
 ipv6 address 2000:0:1:9::/64 eui-64
 tunnel source Loopback3
 tunnel mode ipv6ip isatap
```

```
!
interface FastEthernet0/1
 ip address 10.1.3.3 255.255.255.0
 ipv6 address 2000:0:1:3::3/64
!
ipv6 route 2000:0:1:1::/64 2000:0:1:9:0:5EFE:A09:901
ipv6 route 2000:0:1:4::/64 2000:0:1:9:0:5EFE:A09:904
```

The most important parts of the configurations are the tunnel configuration and the static routes. First, on R1's new tunnel interface (Tunnel 9), the configuration sets the mode to ISATAP (**tunnel mode ipv6ip isatap**). This command, in combination with the **ipv6 address 2000:0:1:9::/64 eui-64** command, tells R1 to use the modified EUI-64 rules to give R1's Tunnel 9 interface an IPv6 address of 2000:0:1:9:0:5EFE:A09:901. R1 derives the last two octets based on the indirect reference made in the **tunnel source loopback 1** command, using loopback 1's IPv4 address (10.9.9.1) to complete Tunnel 9's IPv6 address.

The two static routes on R1 simply define routes to the LAN subnet on each remote router with the important part being the next-hop IPv6 address. The listed addresses exist on R3 and R4, generated by the modified EUI-64 process on those routers.

Example 18-9 shows some supporting verification commands on Router R1.

Example 18-9 *Verifying Tunnel Operation on R1*

```
R1# show ipv6 interface brief
FastEthernet0/0             [up/up]
    FE80::213:19FF:FE7B:5026
    2000:0:1:1::1
! irrelevant interfaces removed
Loopback1                   [up/up]
    unassigned
Tunnel9                     [up/up]
    FE80::5EFE:A09:901
    2000:0:1:9:0:5EFE:A09:901

R1# show ipv6 route
IPv6 Routing Table - Default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   2000:0:1:1::/64 [0/0]
     via FastEthernet0/0, directly connected
L   2000:0:1:1::1/128 [0/0]
     via FastEthernet0/0, receive
```

```
S    2000:0:1:3::/64 [1/0]
      via 2000:0:1:9:0:5EFE:A09:903
S    2000:0:1:4::/64 [1/0]
      via 2000:0:1:9:0:5EFE:A09:904
C    2000:0:1:9::/64 [0/0]
      via Tunnel9, directly connected
L    2000:0:1:9:0:5EFE:A09:901/128 [0/0]
      via Tunnel9, receive
L    FF00::/8 [0/0]
      via Null0, receive

R1# traceroute
Protocol [ip]: ipv6
Target IPv6 address: 2000:0:1:3::3
Source address: 2000:0:1:1::1
Insert source routing header? [no]:
Numeric display? [no]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Priority [0]:
Port Number [0]:
Type escape sequence to abort.
Tracing the route to 2000:0:1:3::3

  1 2000:0:1:9:0:5EFE:A09:903 0 msec 0 msec 4 msec
```

The example begins with the **show ipv6 interfaces brief** command, which shows the two IPv6 addresses R1 derives. It lists the 2000:0:1:9::5EFE:A09:901 address, derived using the modified EUI-64 rules. It also lists a link local address on the interface, formed not with traditional EUI-64 rules but instead with the same modified EUI-64 rules used for the global unicast address.

The **show ipv6 route** command output highlights the same routes shown in Figure 18-10. It lists the static route for R3's LAN subnet, a route to 2000:0:1:3::/64, with R3's modified EUI-64 IPv6 address as the next hop. The output also shows R1's connected route for the tunnel subnet (2000:0:1:9::/64). Finally, the **traceroute** command at the end confirms that R1 can send packets to R3's LAN IPv6 address (2000:0:1:3::3) from R1's LAN IPv6 address (2000:0:1:1::1).

Multipoint IPv6 Tunnel Summary

Multipoint IPv6 tunnels give engineers a good means to implement IPv6 connectivity for short periods of time. The tunnels can allow easier addition of new sites, with less configuration on existing routers. These tunnels also support tunneling with IPv6 hosts. However, these tunnels do not support IPv6 IGPs. Also, the extra processing required to

forward each IPv6 packet makes this solution most useful as a short-term temporary solution, with native IPv6 and point-to-point tunnels more reasonable for more extended periods of time. Table 18-4 summarizes the key features with the small differences highlighted in the table.

Key Topic

Table 18-4 *Comparing IPv6 Multipoint Tunnels*

	Automatic 6to4	ISATAP
RFC	3056	4214
Uses a reserved IPv6 address prefix.	Yes (2002::/16)	No
Supports the use of global unicast addresses?	Yes	Yes
Quartets holding the IPv4 destination address.	2/3	7/8
End-user host addresses embed the IPv4 destination?	Sometimes	No
Tunnel endpoints IPv6 addresses embed IPv4 destination.	Sometimes	Yes
Uses modified EUI-64 to form tunnel IPv6 addresses?	No	Yes

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans. This section provides some exercises that may help you to take a step back from the minute details of the topics in this chapter so that you can think about the same technical topics from the planning perspective.

For each planning practice table, simply complete the table. Note that any numbers in parentheses represent the number of options listed for each item in the solutions in Appendix F, "Completed Practice Planning Tables."

Design Review Table

Table 18-5 lists several design goals related to this chapter. If these design goals were listed in a design document, and you had to take that document and develop an implementation plan, what implementation options come to mind? You should write a general description; specific configuration commands are not required.

Table 18-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design states that an Enterprise needs IPv6 support for most LANs, with a regular high-volume of IPv6 traffic. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	
The design states that an Enterprise needs IPv6 support for a set small subset of LANs but that their traffic will be regular. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	
The design states that an Enterprise needs IPv6 support for a set small subset of LANs but that their traffic will be irregular and occasional. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	
The plan calls for IPv6 tunneling so that new routers, when added to the tunnel, do not require additional configuration on existing routers. What type tunnel would you choose, and what IPv6 address ranges?	

Implementation Plan Peer Review Table

Table 18-6 shows a list of questions that others might ask, or that you might think about, during a peer review of another network engineer's implementation plan. Complete the table by answering the questions.

Table 18-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan calls for the use of OSPFv3 along with the implementation of IPv6 tunnels. What tunnel types do you expect to find the sample configurations? (2)	
The planning diagrams show multipoint tunnels, with IPv6 addresses that embed an IPv4 address in the last two quartets. What type of tunneling do you expect to see in the sample configurations?	
The plan lists a sample configuration with the command **tunnel mode ipv6ip** under a tunnel interface. What type of tunneling is used in this case?	
Same question as the previous row, but the command listed is **tunnel mode ipv6ip isatap**.	
Same question as the previous row, but the command listed is **tunnel mode gre ip**.	
Same question as the previous row, but the command listed is **tunnel mode ipv6ip 6to4**.	
A plan shows the use of a manually configured tunnel and an ISATAP tunnel. What tunnel subcommand would you expect to see for the point-to-point tunnel, but not the multipoint tunnel?	

Create an Implementation Plan Table

To practice skills useful when creating your own implementation plan, list in Table 18-7 all configuration commands related to the configuration of the following features. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Table 18-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an IPv6 manually configured tunnel using a loopback IPv4 address. Ignore IPv6 addressing and routing configuration.	
Add IPv6 addressing and routing configuration to the previous row's list. Assume EIGRP for IPv6 ASN 1 is preconfigured.	
Configure an IPv6 GRE tunnel using a loopback IPv4 address. Ignore IPv6 addressing and routing configuration.	
Configure an IPv6 automatic 6to4 tunnel using a loopback IPv4 address. Include only IPv6 configuration required for the tunnel to pass IPv6 traffic. Assume all hosts use addresses in the 2002::/16 range.	
List steps to migrate from the automatic 6to4 tunnel from the previous row to a comparable ISATAP tunnel	

Choose Commands for a Verification Plan Table

To practice skills useful when creating your own verification plan, list in Table 18-8 all commands that supply the requested information. You may want to record your answers outside the book and set a goal to complete this table (and others like it) from memory during your final reviews before taking the exam.

Note: Some of the entries in this table may not have been specifically mentioned in this chapter but are listed in this table for review and reference.

Table 18-8 *Verification Plan Memory Drill*

Information Needed	Commands
Tunnel interface status for IPv6.	
Tunnel interface's IPv6 address(es).	
Connected routes related to the tunnel.	
The tunnel source and destination IPv4 addresses.	
Test the tunnel to see if it can pass traffic.	

Review all the Key Topics

Review the most important topics from inside the chapter noted with the Key Topics icon in the outer margin of the page. Table 18-9 lists a reference of these key topics and the page numbers on which each is found.

Key Topic

Table 18-9 *Key Topics for Chapter 10*

Key Topic Element	Description	Page Number
Figure 18-3	Point-to-point IPv6 tunnel concept	614
Figure 18-4	Multipoint IPv6 tunnel concept	616
Table 18-2	Comparisons of four IPv6 tunnel types	617
Figure 18-5	NAT-PT concepts	618
List	Manually configured tunnel configuration checklist	620
List	Configuration differences between GRE and manually configured IPv6 tunnels	625
Table 18-3	Comparisons of manually configured tunnels and GRE tunnels	626
Figure 18-8	Address planning for automatic 6to4 tunnels	628
List	Configuration checklist for automatic 6to4 tunnels	629
List	Comparisons of automatic 6to4 and ISATAP tunnels	634
Figure 18-10	ISATAP tunnel logic	635
List	Modified EUI-64 rules for forming ISATAP IPv6 addresses	636
List	ISATAP tunnel configuration checklist	636
Table 18-4	Comparisons of Automatic 6to4 and ISATAP tunnels	640

Complete the Tables and Lists from Memory

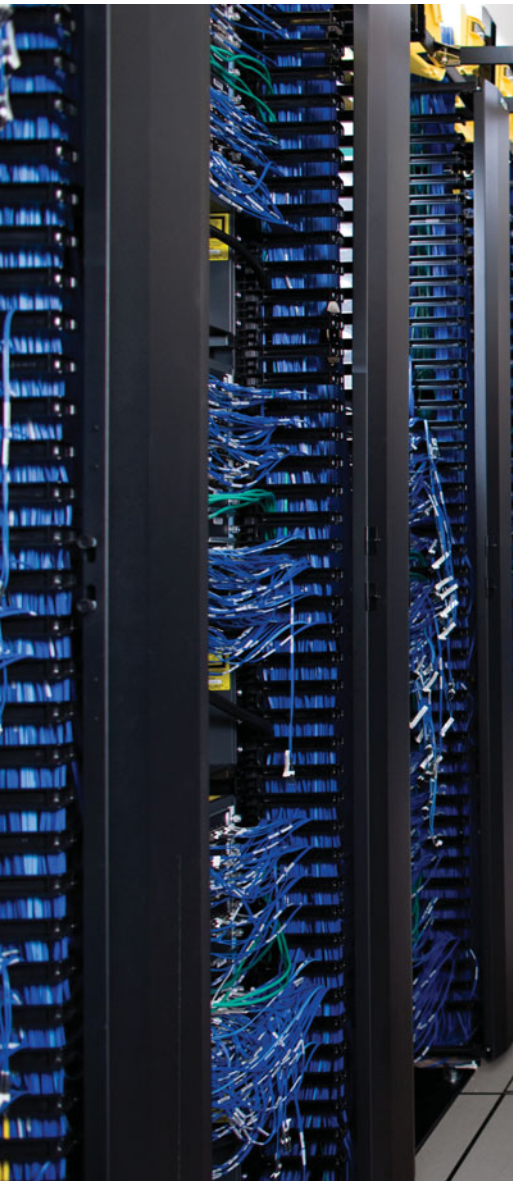
Print a copy of Appendix D, "Memory Tables," (found on the CD), or at least the section for this chapter, and complete the tables and lists from memory. Appendix E, "Memory Tables Answer Key," also on the CD, includes completed tables and lists to check your work.

Define Key Terms

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

Dual stacks, Network Address Translation–Protocol Translation (NAT-PT), tunneling, tunnel, tunnel interface, point-to-point tunnel, multipoint tunnel, ISATAP tunnel, ISATAP, automatic 6to4 tunnel, manually configured tunnel, GRE tunnel, Modified EUI-64

This page intentionally left blank



This chapter covers the following subjects:

Branch Office Broadband Internet Access: This section introduces some of the issues related to using the Internet as a WAN connectivity option for an Enterprise network.

Broadband Branch Access Configuration: This section generally discusses the configuration of a DSL connection on a router, along with the associated NAT and DHCP configuration.

VPN Configuration: This section generally discusses IPSec VPN and GRE tunnel configuration, particularly the routing aspects as they apply to branch office routers.

CHAPTER 19

Routing over Branch Internet Connections

This chapter discusses routing from the Enterprise branch office perspective, specifically for cases in which the branch uses the Internet for its connectivity back to the rest of the Enterprise network. Such a branch router needs to supply many functions: NAT, DHCP services, firewall, and even dynamic routing when multiple connections exist to the Enterprise. This chapter examines each of these and some basics regarding Internet access technologies as well.

From an exam preparation perspective, this chapter differs from the other 17 technology-focused chapters (Chapters 2 through 18) in this book. For the most part, the other chapters not only introduce some networking topics, the descriptions, figures, and examples go quite deep into the subject. This chapter, however, discusses more generalized concepts and some configuration samples without getting into every command option. Some topics mentioned in this chapter are actually part of the prerequisite CCNA topics, and some fit into other Cisco certifications, such as CCNA Security and CCSP. This chapter attempts to maintain the intended depth and breadth of the exam for this small set of remaining topics.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess if you should read the entire chapter. If you miss no more than one of these six self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks." Table 19-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so that you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A.

Table 19-1 *"Do I Know This Already?" Foundation Topics Section-to-Question Mapping*

Foundations Topics Section	Questions
Branch Office Broadband Internet Access	1-2
Branch Router Configuration for Broadband Access	3-4
VPN Configuration	5

1. Router R1 sits at an Enterprise branch office, using the Internet for its only connectivity back to the rest of the Enterprise. Which of the following is not a benefit of using

an IPsec tunnel for packets sent through the Internet, between R1 and the rest of the Enterprise?

a. Privacy
b. Authentication
c. Allows using an IGP between R1 and the Enterprise
d. Secure communications

2. Router R1 sits at an Enterprise branch office, using both the Internet and a leased line to another Enterprise router for its two connectivity options back into the rest of the Enterprise network. The engineer planning for this branch decided to use the leased line for all Enterprise traffic, unless it fails, in which case the Internet connection should be used to pass traffic to the Enterprise. Which of the following is most likely to be useful on the branch router? (Choose two.)

a. IPsec tunnel
b. GRE tunnel
c. Floating static route
d. An IGP

3. Router R1, a branch router, connects to the Internet using DSL. The engineer plans to use a configuration with a dialer interface. The answers list a feature and interface on which the feature could be configured. Which combinations accurately describe the interface under which a feature will be configured?

a. PPP on the ATM interface
b. VPI/VCI on the dialer interface
c. IP address on the ATM interface
d. CHAP on the dialer interface

4. Router R1, a branch router, connects to the Internet using DSL. Some traffic flows through a GRE and IPsec tunnel, over the DSL connection, and into the core of an Enterprise network. The branch also allows local hosts to communicate directly with public sites in the Internet over this same DSL connection. Which of the following answers defines how the branch NAT config avoids performing NAT for the Enterprise-directed traffic but does perform NAT for the Internet-directed traffic?

a. By not enabling NAT on the IPsec tunnel interface
b. By not enabling NAT on the GRE tunnel interface
c. By configuring the NAT-referenced ACL to not permit the Enterprise traffic
d. By asking the ISP to perform NAT in the cloud

5. Router R1, a branch router, connects to the Internet using DSL. Some traffic flows through a GRE and IPsec tunnel, over the DSL connection, destined for an Enterprise network. Which of the following answers best describes the router's logic that tells the router, for a given packet, to apply GRE encapsulation to the packet?

   a. When the packet received on the LAN interface is permitted by the ACL listed on the **tunnel gre** *acl* command under the incoming interface

   b. When routing the packet, matching a route whose outgoing interface is the GRE tunnel interface

   c. When routing the packet, matching a route whose outgoing interface is the IPsec tunnel interface

   d. When permitted by an ACL that was referenced in the associated crypto map

Foundation Topics

Branch Office Broadband Internet Access

Many options exist today for private connectivity between an Enterprise branch office and the core of an Enterprise network. These options include leased lines, Frame Relay, MPLS VPNs, and Metro Ethernet. Although each differs in some way, they all share an important characteristic: They provide an inherently private path over which two Enterprise routers can send packets to each other.

Several other public options exist for branch office connectivity. All these options use the Internet for connectivity between the branch office and the core of the Enterprise network. Regardless of the particular physical Internet access technology–typically digital subscriber line (DSL), cable, or wireless broadband–all these options use a public Internet to forward the packets.

The differences between the public Internet and private connectivity mean that the branches need to use several additional functions just to make the connectivity work, plus the branches need to add other functions to make the connection secure. This chapter focuses on the functions required, and how they impact routing between the branch and the rest of the Enterprise.

The branch routing for the Internet-connected branch differs in part depending on the design. Figure 19-1 shows three examples of branch offices of different sizes. The figure labels these branches as small, medium, and large, but frankly, those descriptions are a bit subjective. For the purposes of this chapter, as the size increases, the number of connections or routers increases, and with that, the number of issues to consider also increases.

For the small branch design, most of the implementation challenges fall into two main categories: features needed for communication with hosts in the public Internet, and features needed to support secure communications with hosts in the Enterprise. The first category includes the Internet access details–how to make DSL, cable, and so on work–plus services likely to be required to learn, allocate, and translate the public IP addresses learned from the ISP, such as NAT and DHCP. The second category focuses on virtual private network (VPN) options that allow the Enterprise to trust that a packet comes from a legitimate branch office, and to prevent attackers from reading the contents of the packet as it crosses the Internet.

The rest of this first section of the chapter gives an overview of the various pieces of the puzzle of supporting private Enterprise traffic, over the public Internet, between the branch and the rest of the Enterprise.

Broadband Internet Access Basics

The term *broadband* has been around in the world of networking for a long time. The original meaning related to the frequency bands used by some Layer 1 standards that used a wider (broader) range of frequencies to achieve a higher bit rate. Today, the term broadband has grown to become synonymous with *high speed*.

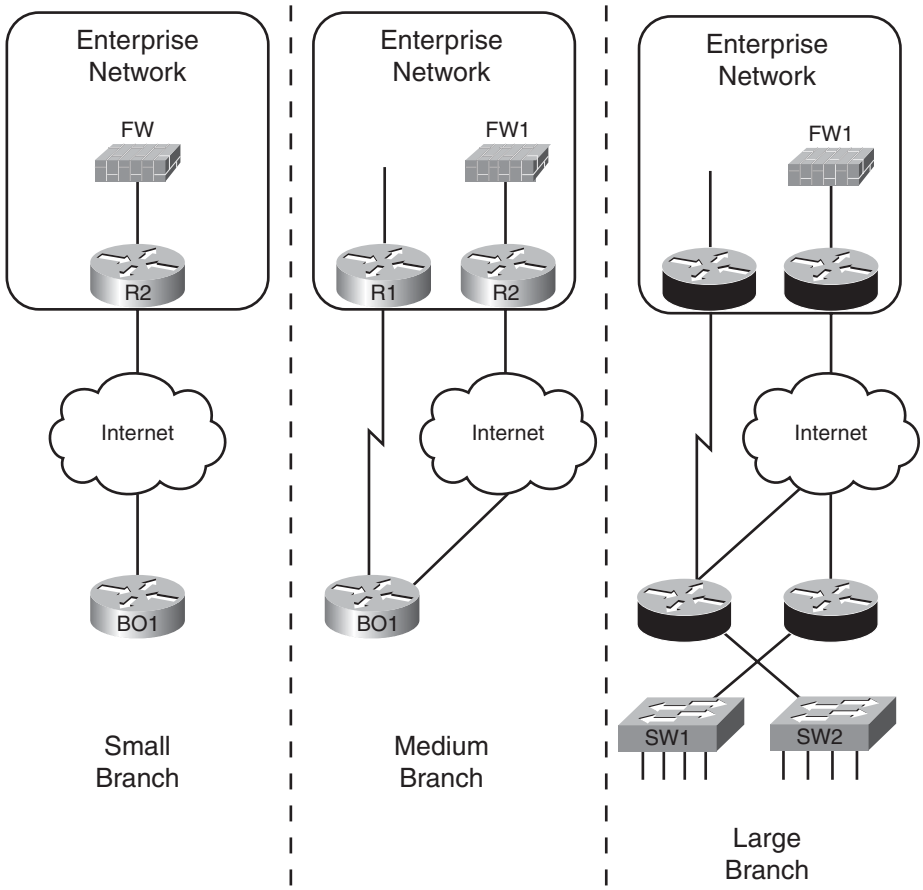


Figure 19-1 *Example Small, Medium, and Large Branch Designs*

The term *broadband Internet access technology* simply refers to a class of high-speed (a relative term) communications methods that allows a device to access some ISP, and in turn the Internet. Today, this term generally refers to (high-speed) cable Internet access, (high-speed) DSL Internet access, and (high-speed) wireless Internet access, with DSL and cable as the most commonly used options.

The hardware used at the branch office for each of these access technologies varies. However, whether a single piece of hardware does all the work, or whether several pieces of hardware combine to do the work, the hardware typically includes an IP router and either a cable modem or DSL modem. Again, like the term broadband, the term *modem* has taken new meaning over the years compared to its original historic use. However, a cable modem or DSL modem sits in the same location as a traditional analog modem, so the term has grown to a broader meaning. For example, Figure19-2 shows a typical branch office LAN with a router and a cable modem.

Figure 19-2 shows a typical connection at a small office/home office (SOHO) site, with a separate router and cable modem. The router in this case uses two LAN interfaces, routing

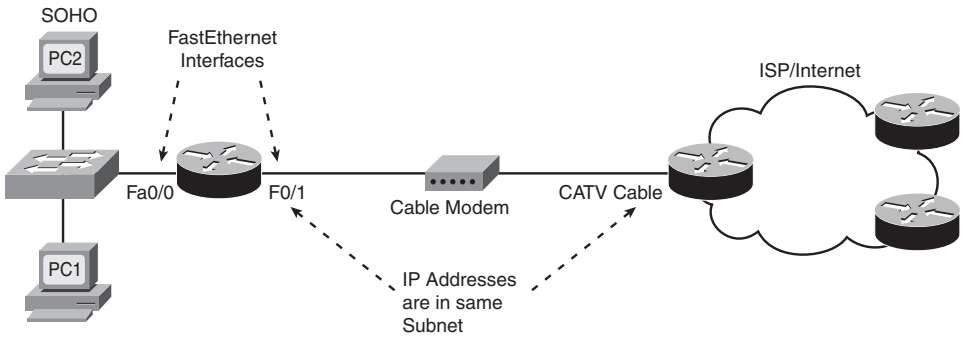


Figure 19-2 *Typical Cabling with Cable Modem at a Small Office/Home Office (SOHO)*

over those interfaces. However, on the LAN interface connected toward the ISP, the router uses a slightly different convention on the Ethernet called *PPP over Ethernet (PPPoE)*. The frames sent by the router still use an Ethernet header, but then include a PPP header, and then an IP header.

The PPP header gives the router and ISP the capability to use PPP features such as the Challenge Handshake Authentication Protocol (CHAP). CHAP allows the ISP to perform authentication, which allows the ISP to confirm the identity of the router that sent the packet. This allows the ISP to check the user's account, make accounting records of the access, and confirm that the customer is not late in paying their high-speed cable bill. (DSL uses a similar protocol, PPP over ATM (PPPoA), also in part to support PPP CHAP.)

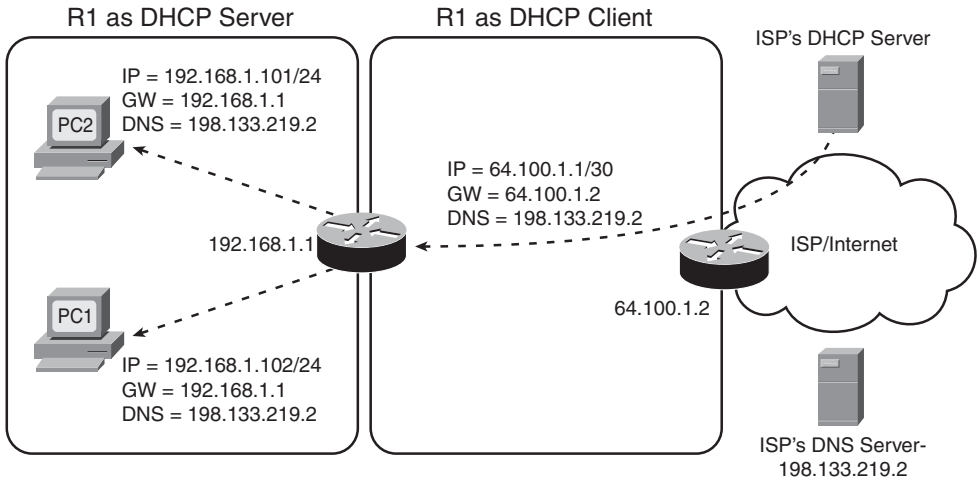
Branch Router as DHCP Server and Client

When planning the permanent connection from an Enterprise to the Internet, as discussed at some length in Chapter 12, "Internet Connectivity and BGP," the ISP uses a range of registered public IP addresses. The ISP often assigns this range, or the Enterprise may register a prefix directly. Most Enterprises then use RFC 1918 private IPv4 addresses for most hosts inside the Enterprise. To make the whole package work, the Enterprise must also configure NAT, typically with the port address translation (PAT) feature, to translate between the private and public addresses.

Each Internet-connected branch follows the same public IP addressing model in general, but with some differences. First, the ISP typically assigns one and only one public IP address for the branch site, expecting the branch to use PAT. That public IP address may change over time; even though most broadband Internet access technologies use "always on" logic, the underlying protocols act more like a dial connection that never stops. So, the ISP dynamically assigns the public IP address so that if the subscriber stops using the connection, fails to pay the bill, or cancels the account, the public IP address can be easily and automatically reclaimed into the pool of available addresses.

The IP address assignment process differs slightly for the branch LAN as well. Hosts still learn their IP address, mask, default gateway, and DNS server addresses, but rather than require a separate host at each branch to act as DHCP server, the branch router may act as

DHCP server. Figure 19-3 shows the idea of the branch router learning its public address, plus acting as DHCP server for local LAN hosts.



Key Topic

Figure 19-3 *Branch Office Router Acting as DHCP Client and DHCP Server*

As you can see, the two branch hosts both received the appropriate addressing information for private network 192.168.1.0/24. R1 itself learns similar information from the ISP's DHCP server. R1 will use this learned IP address on the interface it uses to connect to the ISP. R1 also adds a default route to its IP routing table, using the ISP1's IP address as next hop–in this case, 64.100.1.2.

Finally, R1 must enable NAT/PAT. As a reminder, the term NAT today refers to all NAT functions, including the port address translation feature. PAT allows a little more than 65,000 concurrent flows to use a single public IP address by using a different TCP or UDP port number for each flow. As a result, this single public IP address should be more than enough for a typical branch office.

Note: Consumer-class products enable by default most of the features described in this section, such as DHCP client, server, and NAT/PAT. Enterprise-class routers typically rely on explicit configuration by a network engineer for these same features.

Branch Office Security

With a permanent Internet connection near the core of the Enterprise network, the Enterprise typically uses one or more border security products. These products include stateful firewalls such as the ASA appliance from Cisco, plus Intrusion Prevention Systems (IPS).

Now that a branch has a connection to the Internet, most Enterprises implement similar features at the branch. Again, for economic reasons, it makes more sense to implement those features on the same router platform already needed for so many other reasons. So, Cisco includes firewall and IPS features in IOS. These features include the older Context

Based Access Control (CBAC) and newer IOS Zone-Based Firewall. Cisco also supplies the IOS IPS feature as well.

Although it may seem that running such features on a branch router might overload the router, Cisco designed its Integrated Services Routers (ISR) to have the power to concurrently run many services for exactly this type of application. ISRs include the older model series 800, 1800, 2800, and 3800 ISR families, plus model series 1900, 2900, and 3900, introduced in 2009.

Using IPsec Tunnels

The technologies reviewed up to this point, when implemented at the branch, give hosts at the branch the capability to access hosts in the Internet, and to do so securely. However, the hosts at the branch typically cannot access applications inside the Enterprise at this point.

Several problems prevent the branch hosts from accessing hosts with private IP addresses inside the Enterprise core. One problem is that packets from branch hosts at this point look like packets from any other Internet-based hosts, and the Enterprise firewall could (and should) discard such packets. For example, if a branch host tries to send a packet to a host inside the Enterprise core, the branch would NAT the packet to use the branch's public IP address. When that packet arrives at the Enterprise's firewall at the company's main Internet connection, the firewall would have no way to know whether the packet came from a legitimate host at the branch. One solution would be to allow packets past the firewall, making the server available to the Internet, and rely on host security to prevent access, but that may be too risky.

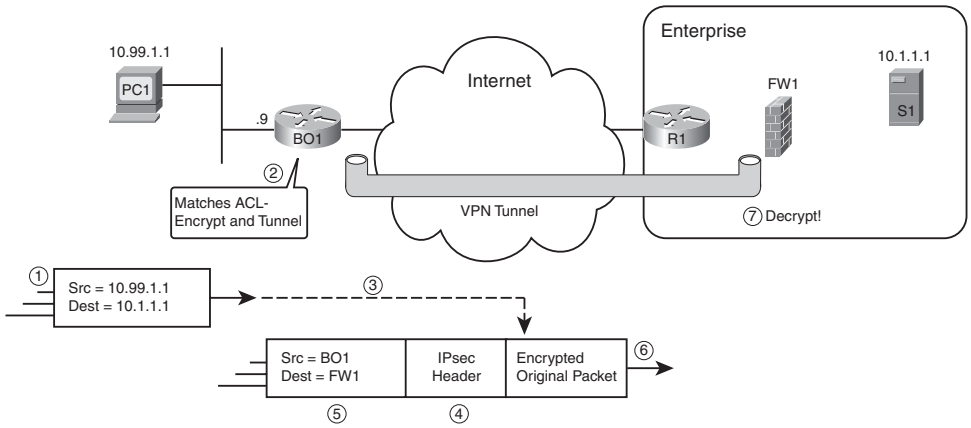
To solve this problem, typically the engineer configures a tunnel between the branch and the Enterprise core. Then the branch router directs traffic destined for the Enterprise through the tunnel. For such packets, the branch router will not translate the IP address of the original packet using NAT. Instead, the branch router will encrypt the IP packet and encapsulate the encrypted packet inside another IPv4 header, using public source and destination addresses in this new packet. Then, the branch router can forward this packet through the public Internet, with a device in the Enterprise core receiving and decrypting the packet. That device can tell from additional security headers in the encrypted packet whether it comes from a trusted branch router. Figure 19-4 shows an example of the process.

Following the steps in Figure 19-4:

Step 1. PC1 creates a packet, with its own 10.99.1.1 address as source, and server S1's 10.1.1.1 IP address as destination. PC1 sends the packet to its default gateway (BO1).

Step 2. When Router BO1 tries to route the packet toward the Internet, the router's logic tells it to check an ACL. All packets permitted by the ACL go through the tunnel–in this case, all packets destined for 10.0.0.0/8, the private network used by this Enterprise.

Step 3. Router BO1 encrypts the original packet so that no one in the Internet can read the data.



Key Topic

Figure 19-4 *IPsec Tunnel Concept*

Step 4. Router BO1 adds a security header (IPsec header), which helps the receiver decrypt the packet, and know that it came from a trusted router.

Step 5. Router BO1 adds a new IPv4 header, this time with BO1's public IP address as source, and the tunnel destination's public IPv4 address as destination. In this case, the destination is a public IPv4 address used by Firewall FW1.

Step 6. Router BO1 sends the packet into the Internet, with the various routers forwarding the packet to FW1's public IP address.

Step 7. Firewall FW1 de-encapsulates and decrypts the original packet, leaving the original packet as described at Step 1. FW1 forwards the packet toward Server S1.

The process of tunneling the packet gives the branch office the logical equivalent of a point-to-point link between the branch router and the firewall. The tunnel logic works like the point-to-point tunnel logic discussed throughout Chapter 18, "IPv4 and IPv6 Coexistence," except for two key differences. First, this tunnel uses IPv4 as both the transport protocol and passenger protocol, rather than using IPv4 to transport IPv6 as the passenger protocol. Also, this tunnel uses additional security features, like the encryption shown in Figure 19-4.

IP Security (IPsec) defines the details of how this particular tunnel works. IPsec defines which security standards may be used to encrypt the packet–for example, IPsec allows the use of triple DES (3DES) and Advanced Encryption Standard (AES). IPsec also allows the use of authentication protocols and headers as well, which the firewall can use to verify that the packet came from a legitimate branch office and not some attacker.

Although the security protocols may be interesting, this book does not delve into the details, other than to say that the protocols ensure that the data in the tunnel is private (through encryption) and comes from a legitimate branch (through authentication).

Branch Routing for the Small Branch

Figure 19-1 shows a sample small branch whose only WAN connectivity is to the Internet. It may seem that such a router has no need to be concerned about routing. However, with the addition of an IPsec tunnel from the branch to the Enterprise core, the branch does have a kind of routing decision to make: to send a packet through the tunnel, or to send it to the local ISP after performing NAT. Both actions result in the packet flowing toward the ISP, but one option results in the original packet successfully arriving at the Enterprise, with private IP addresses. The other routing choice results in the packet being changed with NAT and forwarded to an address somewhere in the public Internet.

Figure 19-5 shows an example of such a routing decision, again using branch Router BO1. Host PC1 has two windows open—one using a web-based application with a server inside the Enterprise core, with another web browser window open and connected to the user's favorite public website.

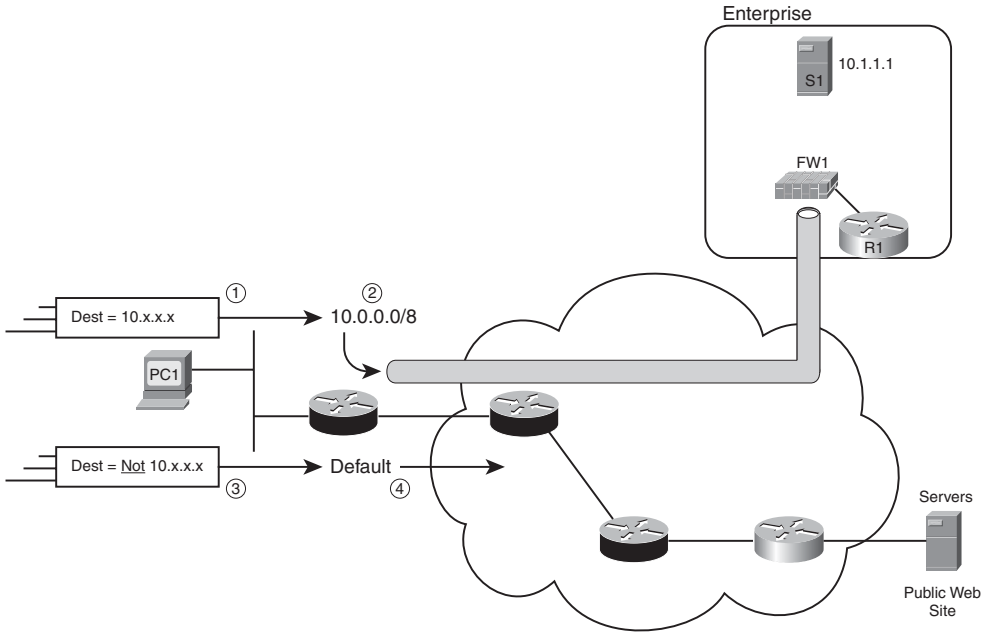


Figure 19-5 *Routing over the IPsec Tunnel or Through the Public Internet*

Following the steps in the Figure 19-5

Step 1. PC1 sends a packet destined to the Enterprise server's 10.x.x.x IP address.

Step 2. Router BO1 uses logic that matches destination addresses in the 10.0.0.0/8 range and directs the packet over the IPsec tunnel. This action results in the encapsulation shown in Figure 19-4 and in the original packet being forwarded into the Enterprise core.

Step 3. PC1 sends another packet; this one is destined for the public web server's public IP address.

Step 4. Router BO1 does not match the packet with its IPsec ACL, so BO1 just routes the packet out its Internet connection, after using NAT.

Routing in Medium and Large Branches

The final concept in this broad overview section has to do with routing in medium and large branches, after the branch has a working Internet connection plus an IPsec tunnel. In the medium- and large-branch examples, another private WAN connection exists between the branch and the Enterprise core. So, the branch needs to make choices about when to use routes that use the private connection and when to use routes that use the public connection. Likewise, the routers in the Enterprise need to make good routing choices about how to route packets back to the branch.

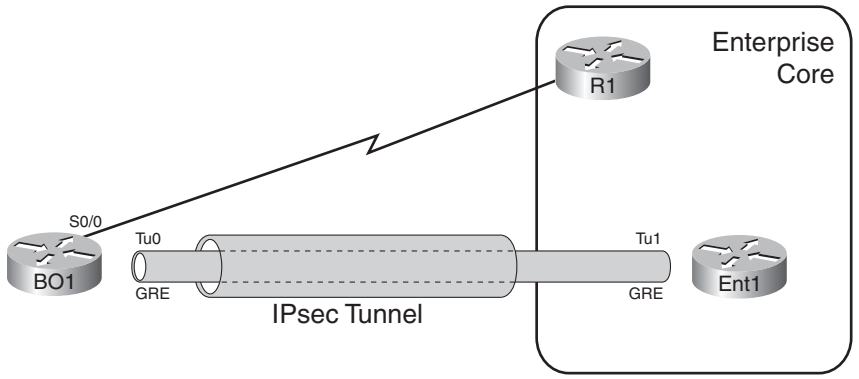
Focusing on the branch router's logic, two main options exist: using static routes and using an IGP. Static routes can be used to forward traffic over the private connection and over the IPsec tunnel. However, IPsec does not directly support IGP protocols, because the IPsec tunnel cannot natively forward IPv4 multicasts. To overcome this restriction, you can use a GRE tunnel that actually runs over the IPsec tunnel. GRE supports multicasts by encapsulating them in unicast packets, so GRE supports IGPs. Summarizing, by using GRE, you get the following features:

- A GRE tunnel acts like a point-to-point link from a Layer 3 perspective.
- A GRE tunnel supports many passenger protocols, including IPv4.
- A GRE tunnel encapsulates/forwards broadcasts and multicasts, therefore supporting IPv4 IGPs.
- Although not mentioned in Chapter 18, GRE tunnels can run through IPsec tunnels.

After configuring a GRE tunnel to run over the IPsec tunnel on a medium or large branch, from a routing perspective, the branch now has two Layer 3 paths to the rest of the Enterprise network: out the GRE tunnel interface and out the private WAN interface. Then you can run an IGP on each, and the branch can choose which path is best using the usual IGP criteria. Figure 19-6 shows an example with a single branch router, a leased line, and a GRE tunnel running through the Internet.

For the network in the figure, branch Router BO1 has two possible outgoing interface choices for any static or IGP-learned route: interface S0/0 (the leased line) or tunnel0 (the GRE tunnel). When a packet arrives at BO1, if the route forwards the packet over tunnel0, the next step in the logic triggers the GRE tunnel logic, encapsulating the packet. Then, that packet matches the IPsec ACL for all packets destined for network 10.0.0.0/8 (in this case) and causes the IPsec process to occur. The IPsec-generated packet can then be routed through the public Internet.

Note: Figure 19-12, in the section "Summary–Branch Routing from PC1 to Enterprise Server S1," shows the encapsulation used to support the design shown in Figure 19-6.



Key Topic

Figure 19-6 *Competing Routes–Private Link and GRE Tunnel*

Next, the text examines the static routing option at the branch, followed by the option to use IGPs and GRE tunnels.

Routing Using Floating Static Routes

A floating static route occurs when the **ip route** command configures a static route in which the command overrides the default AD value of 1. With a higher AD for the static route, the router may consider some other routes for that same prefix as being better. For example, if the IGP learns a route for the same prefix as defined in the static route, the router does not use the static route with the higher AD, instead uses the dynamically learned route with the lower AD. Should the dynamically learned route fail, the router adds the static route back to the IP routing table. The term *floating static* refers to that the route may float in and out of the routing table.

A branch router can use a floating static route directing Enterprise traffic over the Internet, but only using that route when the IGP-learned route over the private link fails. Often, the branch design uses the Internet connection just as a backup for the private link into the Enterprise. To implement such a design, a branch router can use a floating static route for the route over the Internet, only using that route when the IGP-learned route over the private link fails. For example, in Figure 19-6, Router BO1 could use EIGRP over the leased line and configure a floating static route for 10.0.0.0/8 that routes traffic over the Internet connection. Router BO1 would prefer an EIGRP learned, AD 90 route for 10.0.0.0/8 instead, until that route through S0/0 failed.

Note that to use the floating static route option, a GRE tunnel would not be required, because the routers do not need to use an IGP through the Internet.

Dynamic Routing over the GRE Tunnel

If the design engineer decides that the branch router should use both paths into the Enterprise, a somewhat simple solution exists. Simply configure an IGP to run as normal, over both the private WAN connection and over a GRE tunnel. For example, with Figure 19-6, enable EIGRP on the branch Router BO1 on both interface S0/0 and Tu0. The design can then use all the usual tools to manipulate the choice of routes, including tuning the metrics and configuring variance to influence load sharing.

This concludes the overview of branch office routing. The next two major sections of this chapter examine broadband access and IPsec VPNs. These sections go a little deeper on the concepts and show sample configurations for perspective.

Branch Router Configuration for Broadband Access

This section focuses on a branch office router configuration using DSL and the services that must be configured to support Internet traffic. The discussion starts with a little more background about DSL concepts, followed by some sample configurations.

Understanding DSL Concepts

DSL uses the Telco local loop: the phone line that runs between the phone company's nearby facility (called the central office, or CO) and the customer site. In other words, DSL uses the same phone line that runs to most people's homes and to most office buildings. The phone company has used these types of lines for the better part of a century to carry analog electrical signals for voice traffic, but DSL uses other frequency ranges to carry a digital signal for the purpose of sending data.

From the customer premise perspective, the customer can still use the same old analog phones, which still use frequencies below 4000 Hz. The DSL router connects to another RJ-11 socket in the wall, just as if it were just another analog telephone. However, the router sends digital signals at frequencies above 4000 Hz, which does not interfere with the voice traffic. So, both the voice and DSL electrical signals flow over the same cable, at the same time, just at different frequencies.

Note: Although the human voice generates frequencies below 4000 Hz, the human ear can hear some higher frequencies, so some DSL installations require the use of filters on the lines connected to the phones. These filters prevent humans from hearing some of the higher frequency DSL tones.

From the telco perspective, the Telco has to separate the voice and DSL signals. To do so, the Telco uses a device called a DSL Access Multiplexor (DSLAM). It splits the analog signal off to the switch that handles traditional analog voice calls and splits the digital traffic to a router. Figure 19-7 shows these ideas, both for the customer premise and central office.

At Layer 1, the DSL uses digital signals that use some encoding that does not matter to the discussions in this book. However, at Layer 2, DSL uses two different data link protocols: Asynchronous Transfer Mode (ATM) and Point-to-Point Protocol (PPP).

DSL uses ATM in the traditional role of a data link protocol, and PPP for several reasons, but particularly for its CHAP authentication. For DSL, ATM controls the use of the Layer 1 medium so that data can be successfully sent over the link and to the right device. ATM defines the headers used on the link, the data link addresses, and the rules for passing Layer 3 data up and down the protocol stack. For instance, the ATM cell headers enable the DSLAM to know where to send the data received from a customer over a DSL link.

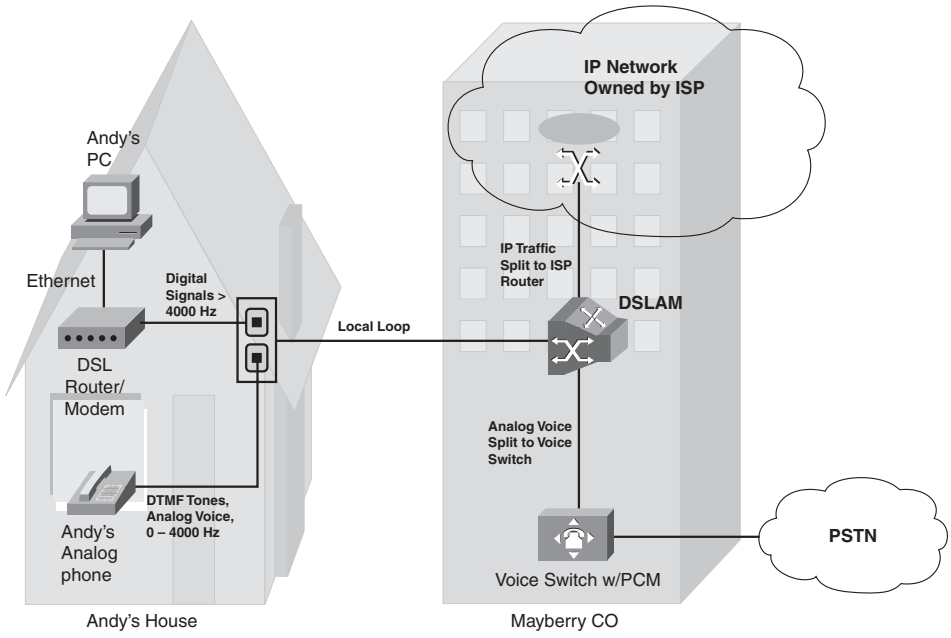


Figure 19-7 *DSL at the Home and the Telco Local Loop*

ATM uses a PVC concept similar to a Frame Relay PVC. With DSL, an ATM PVC would exist between the DSL router at the branch and the ISP router in Figure 19-7. Similar to Frame Relay, an ATM address identifies the PVC, but the address value is local, so the value used by one endpoint may be different than the address used on the other end of the PVC. Frame Relay uses the DLCI as the identifier of a PVC, whereas ATM uses a two-part address, called the Virtual Path Identifier/Virtual Connection Identifier (VPI/VCI).

Frame Relay and ATM differ quite a bit in relation to Layer 2 encapsulation. Frame Relay uses simpler conventions than ATM: When a packet needs to be sent out a Frame Relay interface, the router adds a Frame Relay header and trailer and sends the frame. ATM adds a short extra header to the packet and then performs *segmentation*. The segmentation process breaks the data into 48-byte segments and adds a 5-byte header to each to create ATM *cells*. The cell header holds the VPI/VCI pair, which allows the network to forward the cells to the correct destination end of the PVC. Figure 19-8 shows an example of the encapsulation.

Follow the steps in Figure 19-8:

Step 1. The router wants to send a packet out the DSL ATM interface. (This logic occurs after any tunneling logic and NAT, and works the same whether using IPSec, GRE over IPsec, or the packet is just destined for a public host in the Internet.)

Step 2. The router, using PPPoA concepts, adds a PPP header.

Step 3. The router adds another short header (details unimportant here).

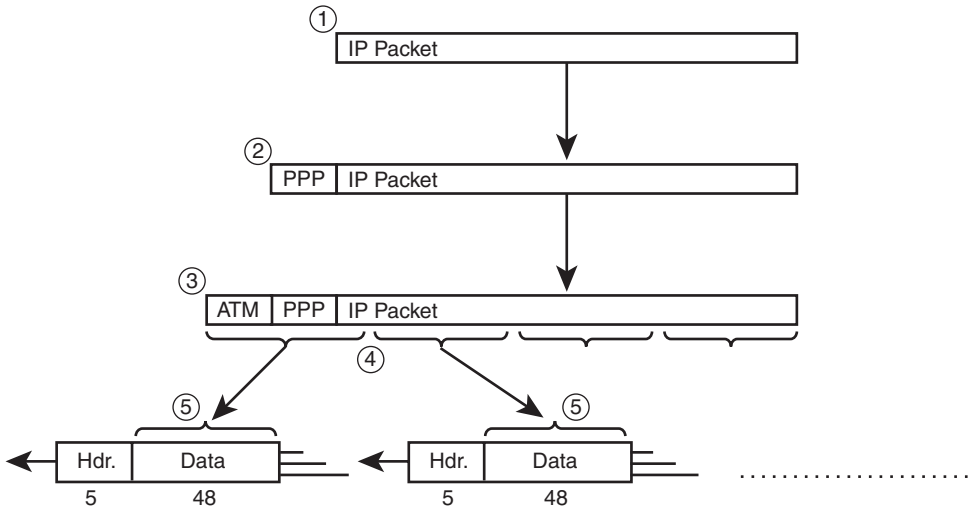


Figure 19-8 *ATM Encapsulation and Segmentation*

Step 4. The router's segmentation and reassembly (SAR) chip segments the frame from the previous step into 48-byte segments.

Step 5. The SAR chip encapsulates these segments inside 5-byte cell headers and sends the cells over the DSL link.

The DSLAM then receives the cells and forwards them on to the router. The router at the other end of the PVC reassembles the cells–the finishing touch on the ATM SAR process. The receiving router can then begin interpreting the various headers and de-encapsulate the packet.

Briefly, note that the encapsulation process also adds a PPP header to the IP packet before sending the data over the DSL link. The routers use the PPP header for several reasons, including PPP authentication with CHAP, and for dynamic address assignment and discovery. This convention to use both PPP and ATM protocols together is called PPP over ATM (PPPoA).

Configuring DSL

DSL configuration–even ignoring related services like DHCP and NAT–requires several steps. The goal of this section is to give you a general idea of the configuration by showing one example, just to give you a sense of the configuration pieces.

To appreciate the sample configuration, first consider that DSL is a switched connection. Most people think DSL (and cable) provide an *always on* or leased Internet connection, because typically the user does not need to do anything to start and stop the connection. However, a router can start and stop the DSL connection–or using the traditional terms, the router can dial and hang-up the connection. The idea that DSL routers do something to dial the connection means that the connection is actually switched.

Because cable and DSL connections use switched logic, IOS implements DSL configuration using some older switched network commands, including dialer interfaces and virtual templates. The option used in this section's example, the dialer interface, has been around IOS for a long time as a place to configure the logic and features related to a dialed connection.

The main pieces of the DSL configuration as shown in this section are as follows:

- The configuration creates a dialer interface.
- The Layer 3 and PPP configuration related to DSL is applied to the dialer interface.
- The ATM configuration is applied to the physical ATM interface.
- The ATM interface is linked to the dialer interface.
- An IP route forwards traffic out the dialer interface, which triggers the DSL encapsulation process, as shown in Figure 19-8.

Figure 19-9 shows the configuration, with some notes about the interactions of the various pieces of the configuration.

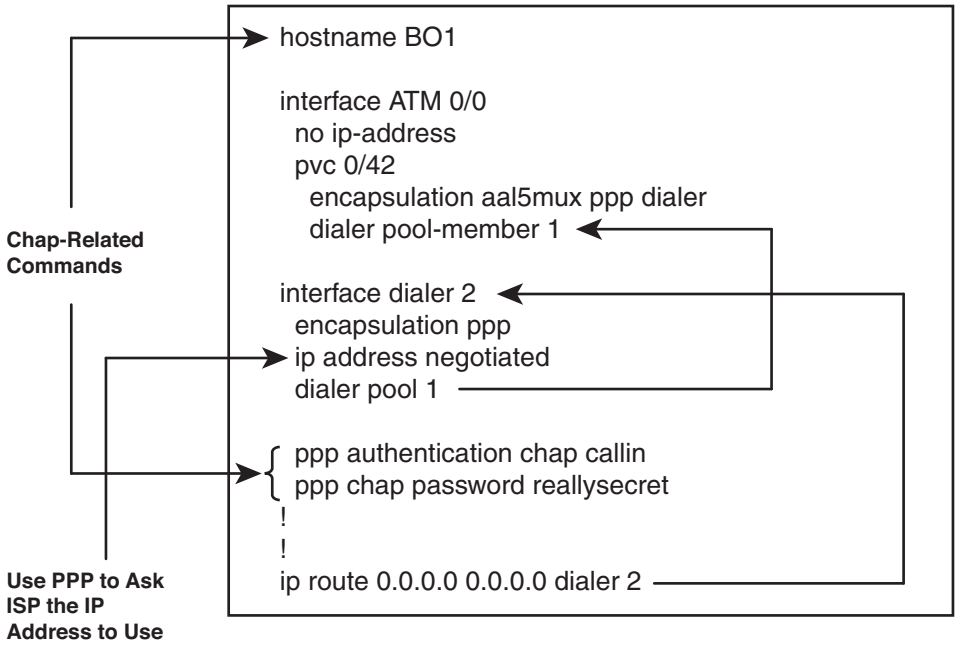


Figure 19-9 *DSL Configuration on Router BO1*

First examine the ATM interface. The configuration defines the VPI/VCI as 0/42; the ISP needs to match this value, or more likely, dictates the value to the customer. The **encapsulation** command defines that PPP will also be used (as shown in Step 2 of Figure 19-8's encapsulation), and it defines the style of ATM header added at Step 3 of that

same figure (AAL5MUX). The **encapsulation aal5mux ppp dialer** command's **dialer** parameter defines that this PVC will use the logic of a dialer interface. Finally, the **dialer pool-member 1** command associates the ATM interface with the dialer interface as noted in the figure.

The dialer interface has five subcommands in this case, including three related to PPP. One command tells the router to use PPP to learn its IP address from the ISP (**ip address negotiated**). The **dialer pool 1** command tells the dialer interface that when it needs to signal a new connection, look for interfaces with **dialer pool-member 1** configured, such as interface ATM 0/0.

Finally, the static default route sends traffic out the dialer interface. Packets forwarded out this DSL connection will match this route, causing IOS to try to forward the packet using the dialer interface, triggering the encapsulation and logic described in Figure 19-8.

Configuring NAT

When the branch router receives a packet over the LAN interface, it has several options of how to process the packet. For instance, using the medium-sized branch from Figure 19-1, which has a leased line into the Enterprise plus a DSL Internet connection, the router could do the following:

- Forward the packet out the serial interface, unchanged, to the rest of the Enterprise network.
- Forward the packet out the tunnel, changed somewhat (encrypted, encapsulated, and so on), to the rest of the Enterprise network.
- Forward the packet over the Internet link (the DSL dialer interface), after using NAT to change the source private address to a public address, to some public IP destination address.

Only the third option requires NAT. Thankfully, NAT configuration easily supports the concept of performing NAT for traffic going to Internet destinations and not performing NAT for traffic in the tunnel. Example 19-1 shows a sample configuration, again using Router BO1. This configuration assumes that BO1 was already configured, as shown in Figure 19-9.

Example 19-1 *NAT Configuration for Router BO1*

```
interface fastethernet 0/0
 ip address 10.99.1.9 255.255.255.0
 ip nat inside

interface dialer 2
 ip nat outside
ip nat inside source list local-lan interface dialer2 overload
ip access-list extended local-lan
 permit ip 10.99.1.0 0.0.0.255 any
```

The configuration shows NAT overload, using a single public IP address–namely, dialer2's dynamically learned IP address. ACL local-lan matches all packets whose source IP address is from the branch's local LAN subnet (10.99.1.0/24). The ACL, referenced by the **ip nat inside** global command, tells the router to NAT traffic permitted by this ACL. The traffic going through the tunnel will already be encapsulated in a new IP header, and no longer have a source address from the LAN subnet, so only traffic destined for Internet destinations will have NAT applied. Finally, the interface subcommands **ip nat inside** and **ip nat outside** tell the interfaces on which to attempt the translation.

Configuring DHCP Server

The branch router also may need to act as the DHCP server. If so, the router needs to have a pool of IP addresses appropriate for the local branch LAN. It needs to know the IP addresses of the DNS servers–both inside the Enterprise and the ISP's DNS server. It also needs to assign a default gateway, typically that same branch router's LAN IP address. Example 19-2 continues the same configuration.

Example 19-2 *DHCP Configuration for Router BO1*

```
ip dhcp pool fred
 network 10.99.1.0 255.255.255.0
 default-router 10.99.1.9
 ip dhcp exclude-address 10.99.1.9
 dns-server 10.2.2.2 128.107.2.1
```

Note that no interface configuration is needed on the LAN interface–the router notices the incoming interface of the DHCP request, compares the connected subnets to the pool, and picks a pool that matches the correct address range.

VPN Configuration

Earlier, this chapter introduced the concept of an IPsec VPN between the branch router and another device in the Enterprise core. This VPN, sometimes called a *VPN tunnel*, gives the Enterprise engineer a way to send a packet native to the Enterprise, with private IP addresses, through the Internet. Additionally, the VPN provides privacy (through encryption) and verification that the sender is legitimate (through authentication).

Although extremely useful, the IPsec tunnel unfortunately does not allow IGP traffic to flow directly over the IPsec VPN tunnel. One solution is to also use a GRE tunnel, which does support IGPs because it can encapsulate the IGP's multicasts inside a unicast IP packet. As previously shown in Figure 19-6, GRE solves the problem by routing over the GRE tunnel interface, whose traffic is, in turn, processed by the IPSsec tunnel.

Other alternatives exist for supporting routing over an IPsec tunnel. These options include

- **Virtual Tunnel Interfaces:** Similar in concept to GRE tunnels but it uses an encapsulation that does not add and extra 4-byte header. (GRE adds such a header.)
- **Dynamic Multipoint VPN (DMVPN):** Creates a multipoint VPN concept, allowing less configuration to add new sites.

- **Group Encrypted Transport (GET) VPN:** A more recent addition to IOS, also supporting multipoint VPNs with less configuration to add new sites.

This section shows a sample configuration for both IPsec and GRE tunnels, just to complete the perspectives on the Internet-connected branch router.

Configuring an IPsec VPN

To fully understand the IPsec configuration, you need a deeper understanding of the security protocols than the detail included in this book. However, if you ignore the particulars about security protocols, a sample configuration can reveal some interesting facts about branch routing, which is the focus of this chapter.

Figure 19-10 shows a sample IPsec configuration for Router BO1. Again, this configuration assumes the configuration in the previous examples, plus Figure 19-9, have already occurred.

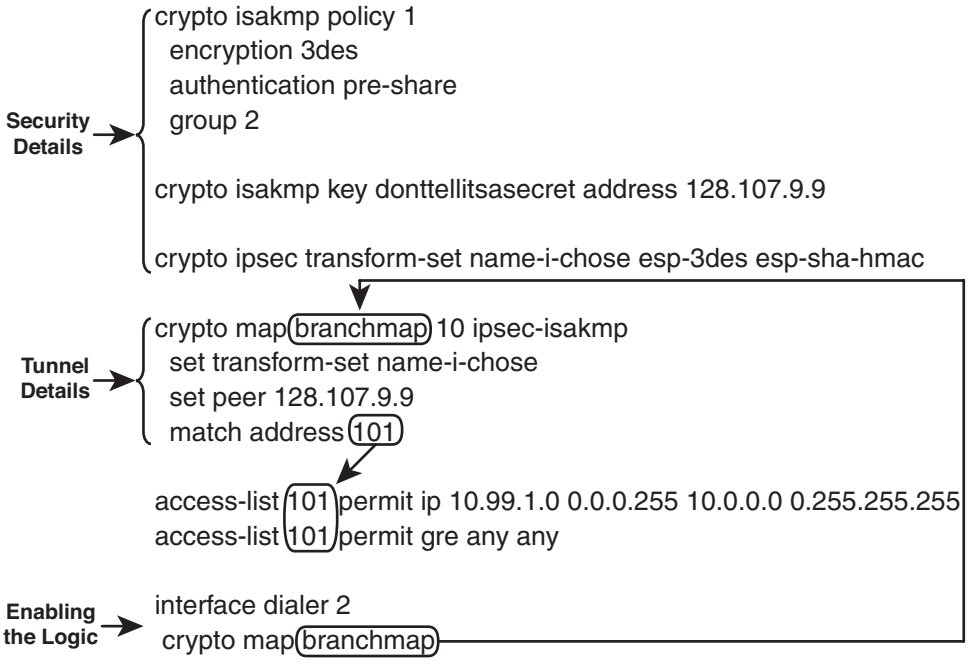


Figure 19-10 *IPsec Configuration*

Key Topic

Focus first on the crypto map (named branchmap) and the dialer interface. The dialer interface enables IPsec with the **crypto map branchmap** command, causing IOS to consider applying IPsec to packets *exiting* Dialer 2. The crypto map causes IOS to only encrypt and tunnel the packets that are matched by ACL 101 in this case. (See the arrows in the figures as to how the crypto map is linked to using ACL 101.) The crypto map also identifies the destination IP address used when the encapsulation takes place (128.107.9.9). This

address is the public IP address of the device on the other end of the tunnel; the earlier figures showed that as Router Ent1.

Next, think about a packet received by BO1 over the LAN, in light of ACL 101, and in light of the crypto map processing outbound traffic on interface Dialer 2. The packet arrives in BO1's F0/0 interface. The packet may be processed by a GRE tunnel first, or it may not. Then, some route must route the packet out Dialer 2. At that point, the logic of the commands in Figure 19-10 finally begins.

Continuing with this same packet, the ACL matches packets that were tunneled by GRE, or other packets that come from the LAN and are going toward the rest of the Enterprise. The first line in the ACL matches the packets from the local LAN (10.99.1.0/8) going to another destination in the Enterprise. The second line in the ACL matches all GRE packets. Note that packets destined to some public IP address in the Internet would not match the ACL with a permit action. So, only packets destined for the Enterprise network match ACL 101; only the packets permitted by ACL 101 will be processed by the IPsec tunnel logic.

Configuring GRE Tunnels

The GRE tunnel configuration on the branch router does not require any additional commands as compared with the GRE tunnels discussed in Chapter 18, which showed how to tunnel IPv6 (the passenger protocol) over IPv4 (the transport protocol). In this case, the GRE tunnel carries IPv4 as the passenger protocol, inside an IPv4 packet.

The fact that an IPSec tunnel exists, plus the issues related to the public and private addresses used over the Internet connection, does make the application of the tunnel a bit more challenging. First, to make more sense of what will be configured, Figure 19-11 shows the concepts and parameters related to configuring a GRE tunnel in this case.

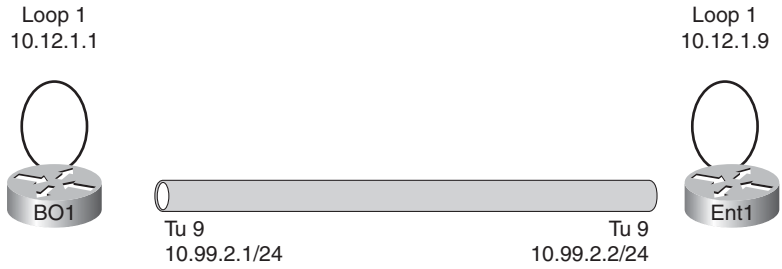


Figure 19-11 *GRE Tunnel Topology and Addresses*

The link that appears as a tunnel between the branch router (BO1) and central site Enterprise router (Ent1) acts like a point-to-point serial link. In the many examples in Chapter 18, this link would have had IPv6 addresses because IPv6 was the passenger protocol. In this case, the tunnel interfaces have IPv4 addresses because IPv4 is the passenger protocol, with the addresses in a new subnet allocated just for this tunnel.

The configuration also uses loopback interfaces, with those interfaces and their IP addresses used as the tunnel endpoints. This configuration means that the new IP packet header created by GRE will use addresses 10.12.1.1 and 10.12.1.9. Finally, any routes

learned over the tunnel will list tunnel 9 as an outgoing interface, with next-hop address 10.99.2.2.

Example 19-3 shows a sample configuration, again on Router BO1.

Example 19-3 *GRE Tunnel Configuration*

```
interface tunnel 9
ip address 10.99.2.1 255.255.255.255
 tunnel source loopback 1
 tunnel destination 10.12.1.9

interface loopback 1
 ip address 10.12.1.1 255.255.255.0

router eigrp 1
 network 10.12.1.1 0.0.0.0
 network 10.99.2.1 0.0.0.0

ip route 10.12.1.9 255.255.255.255 dialer2
```

The tunnel configuration just uses three subcommands: one to define the source IP address (indirectly, as loopback 1's 10.12.1.1), the tunnel destination (10.12.1.9), and the interface's passenger protocol address (IP address 10.99.2.1). The **tunnel mode** command is not needed, because IOS defaults to use IPv4 as the transport protocol, which then allows any of the supported passenger protocols.

The configuration also requires two main branches of logic for routing to work correctly. First, for the tunnel to function, the tunnel destination must be reachable; in this example, a static route was added for this purpose. Additionally, the routers need to exchange routes that will list the tunnel interface as the outgoing interface, which in turn directs packets through the tunnel. The example includes the EIGRP configuration that enables EIGRP on tunnel 9 just as a reminder that one of the primary motivations for bothering with the GRE tunnel is to support IGP routing protocols.

Summary–Branch Routing from PC1 to Enterprise Server S1

To complete the chapter, this section works through an example where a host at the branch sends a packet to a server inside the Enterprise. For the sake of argument, the branch prefers to send this packet over the Internet. It is immaterial whether the branch does not have a private link into the Enterprise, or if the engineer chose to implement routing so that the path through the Internet is currently preferred. The example then gives us a chance to work through the logic when the packet is sent through a GRE tunnel, and an IPsec tunnel, and then out the DSL ATM interface.

Figure 19-12 shows the example. The example begins with the arrival of a packet from PC1, destined from servers S1, 10.1.1.1. The packet has arrived at Router BO1, which has removed the incoming frame's data link header/trailer. The figure picks up the story as BO1 makes its first routing decision about this packet.

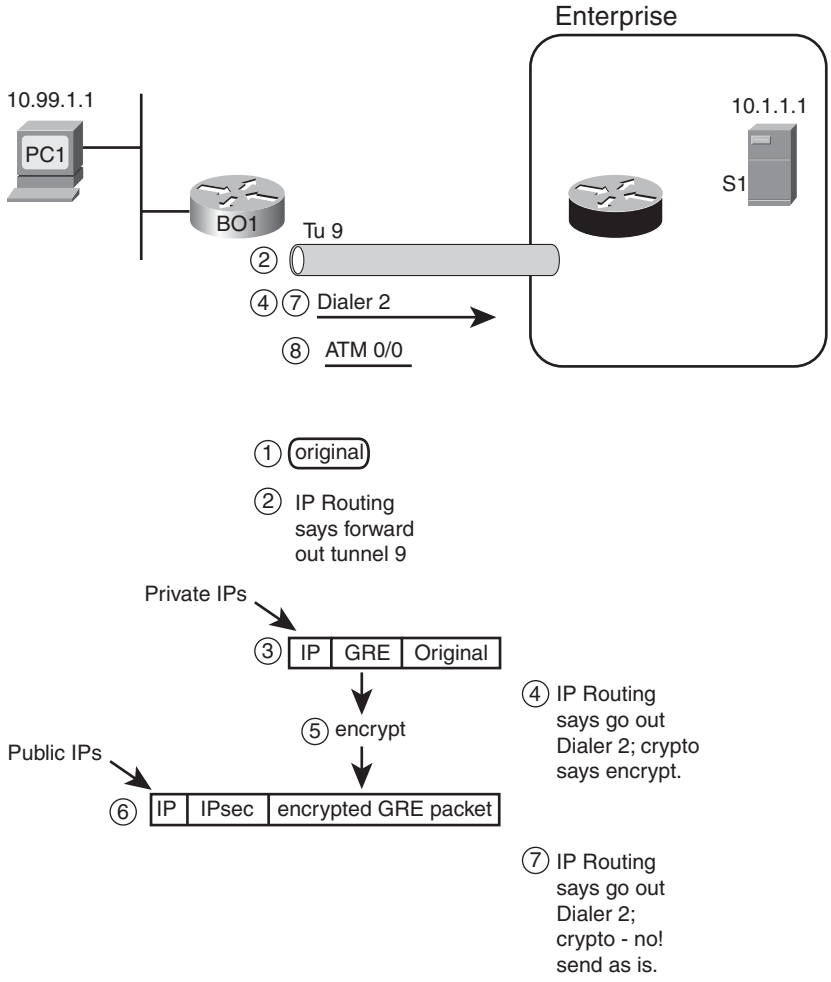


Figure 19-12 *Example of Routing and Encapsulation with GRE over IPsec*

Following are the steps in Figure 19-12:

Step 1. R1 has the original packet in memory, source 10.99.1.1 (PC1), destination 10.1.1.1 (S1).

Step 2. BO1's best route for destination 10.1.1.1 uses outgoing interface tunnel 9. This route may have been learned by an IGP running over this GRE tunnel.

Step 3. BO1 adds a new IPv4 header and GRE header to the original packet. This new packet as a destination based on BO1's tunnel 9 subcommand **tunnel destination**, per the previous Example 19-3, is address 10.12.1.9.

Step 4. BO1 routes the packet formed in the previous step. This best route for 10.12.1.9 lists Dialer 2 as the outgoing interface. The crypto map on interface Dialer 2 refers to an ACL, and ACL matches this packet with a permit action. This combination of logic tells BO1 to use IPsec to encrypt this packet for transmission over the IPsec tunnel.

Step 5. BO1 encrypts the packet that was created in Step 3–in other words, it encrypts the GRE-created packet.

Step 6. BO1 encapsulates the encrypted data, adding several IPsec headers, plus a new IPv4 header. The new IPv4 header uses BO1's public IPv4 address as source and the configured public IPv4 address of the other end of the IPsec tunnel as destination. Per the example in Figure 19-10, the destination IP address would be 128.107.9.9.

Step 7. BO1 routes this latest packet, with its destination IP address of 128.107.9.9, matching a route (probably a default route) that lists Dialer 2 (again) as the outgoing interface. However, the crypto map's ACL does not match the packet with a permit action, so BO1 bypasses any further IPsec functions and simply tries to forward the packet.

Step 8. Forwarding out the dialer interface then causes this DSL-connected router to forward the packet out the underlying ATM interface, which performs the encapsulation and segmentation previously shown in Figure 19-8.

Interestingly, this process drives the branch router to make comparisons to the routing table three separate times when forwarding this data. The most important thing to remember from this example is to get a sense for how the pieces work together and how the steps add additional headers.

Exam Preparation Tasks

Planning Practice

The CCNP ROUTE exam expects test takers to review design documents, create implementation plans, and create verification plans for most topics. However, the CCNP ROUTE exam topics require less depth for the topics in this chapter. For example, the exam topics do not specifically mention the verification task for any of the topics in this chapter, so the chapter does not examine **show** command output.

Because of this key difference, the typical tables at the end of each chapter are not as useful for this chapter. However, some review can be helpful. Table 19-2 lists a series of questions for the purpose of reviewing the content in this chapter. Refer to the same appendix you have been using throughout this book for the suggested answers.

Table 19-2 *Design Review*

Question	Answer
When a branch uses its broadband Internet connection to communicate into the rest of an Enterprise network, what benefits does an IPsec tunnel provide? (3)	
To make the basic broadband connection work, and to support flows from the branch office hosts to/from public websites, what features discussed in this chapter might the router need to configure? (5)	
What method allows a branch to statically route over the IPsec tunnel to the rest of the Enterprise, but only when routes through the private WAN connection fail?	
For what reasons might a network engineer consider also using a GRE tunnel when connecting a branch router, over the Internet, with the rest of the Enterprise network?	
When configuring DSL using dialer interfaces, in what configuration mode are the ATM details configured? PPP details? Layer 3 details?	
A branch router has configured its one LAN interface as a NAT inside interface and its DSL dialer interface as an outside interface. What prevents packets in the IPsec tunnel from being NATted?	
When configuring an IPsec tunnel on a branch router, identify the three main configuration components that link the interface to the matching logic that determines which packets the router processes into the tunnel.	

Note that all the questions assume that a branch office router exists and that it uses some form of broadband access to the Internet.

Review all the Key Topics

Review the most important topics from inside the chapter, noted with the key topics icon in the outer margin of the page. Table 19-3 lists a reference of these key topics and the page numbers on which each is found.

Table 19-3 *Key Topics for Chapter 10*

Key Topic Element	Description	Page Number
Figure 19-3	Branch router's role as DHCP server and client	653
Figure 19-4	IPsec tunnel concepts	655
Figure 19-6	Comparing the competing routes–private and tunnel	658
Figure 19-10	IPsec configuration	665

Define Key Terms

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

IPsec, IPsec tunnel, GRE tunnel, Floating static route, Digital Subscriber Line (DSL), Cable, PPP over ATM (PPPoA), PPP over Ethernet (PPPoE), Challenge Handshake Authentication Protocol (CHAP)

CHAPTER 20

Final Preparation

The first 19 chapters of this book cover the technologies, protocols, commands, and features required to be prepared to pass the ROUTE exam. Although these chapters supply the detailed information, most people need more preparation than simply reading the first 19 chapters of this book. This chapter details a set of tools and a study plan to help you complete your preparation for the exams.

This short chapter has two main sections. The first section lists the exam preparation tools useful at this point in the study process. The second section lists a suggested study plan now that you have completed all the preceding chapters in this book.

Note: Appendixes D, E, and F exist as soft-copy appendixes on the CD included in the back of this book.

Tools for Final Preparation

This section lists some information about the available tools and how to access the tools.

Exam Engine and Questions on the CD

The CD in the back of the book includes the Boson Exam Environment (BEE). The BEE is the exam-engine software that delivers and grades a set of free practice questions written by Cisco Press. The BEE supports multiple-choice questions, drag-and-drop questions, and many scenario-based questions that require the same level of analysis as the questions on the ROUTE exam. The installation process has two major steps. The first step is installing the BEE software; the CD in the back of this book has a recent copy of the BEE software, supplied by Boson Software (http://www.boson.com). The second step is activating and downloading the free practice questions. The practice questions written by Cisco Press for the ROUTE exam are not on the CD. Instead, the practice questions must be downloaded from www.boson.com.

Note: The CD case in the back of this book includes the CD and a piece of paper. The paper contains the activation key for the practice questions associated with this book. Do not lose this activation key.

Install the Software from the CD

Following are the steps you should perform to install the software:

Step 1. Insert the CD into your computer.

Step 2. From the main menu, click the option to install the Boson Exam Environment (BEE). The software that automatically runs is the Cisco Press software needed to access and use all CD-based features, including the BEE, a PDF of this book, and the CD-only appendixes.

Step 3. Respond to the prompt windows as you would with any typical software installation process.

The installation process might give you the option to register the software. This process requires that you establish a login at the www.boson.com website. You need this login to activate the exam; therefore, you should register when prompted.

Activate and Download the Practice Exam

After the Boson Exam Environment (BEE) is installed, you should activate the exam associated with this book.

Step 1. Launch the BEE from the Start menu.

Step 2. The first time you run the software, you should be asked to either log in or register an account. If you do not already have an account with Boson, select the option to register a new account. You must register to download and use the exam.

Step 3. After you register or log in, the software might prompt you to download the latest version of the software, which you should do. Note that this process updates the BEE, not the practice exam.

Step 4. From the Boson Exam Environment main window, click the Exam Wizard button to activate and download the exam associated with this book.

Step 5. From the Exam Wizard dialog box, select Activate a purchased exam and click the Next button. Although you did not purchase the exam directly, you purchased it indirectly when you bought the book.

Step 6. In the EULA Agreement window, click Yes to accept the terms of the license agreement and click the Next button. If you do not accept the terms of the license agreement, you cannot install or use the software.

Step 7. In the Activate Exam Wizard dialog box, enter the activation key from the paper inside the CD holder in the back of the book, and click the Next button.

Step 8. Wait while the activation process downloads the practice questions. When the exam has been downloaded, the main BEE menu should list a new exam. If you do not see the exam, click the My Exams tab on the menu. You might also need to click the plus sign icon (+) to expand the menu and see the exam.

At this point, the software and practice questions are ready to use.

Activating Other Exams

You need to install the exam software and register only once. Then, for each new exam, you need to complete only a few additional steps. For instance, if you bought this book along with *CCNP SWITCH 642-813 Official Certification Guide* or *CCNP TSHOOT 642-832 Official Certification Guide*, you would need to perform the following steps:

Step 1. Launch the BEE (if it is not already open).

Step 2. Perform the preceding Steps 4 through 7 under Activate and Download the Practice Exam.

Step 3. Repeat Steps 1 and 2 for any exams in other Cisco Press books.

You can also purchase Boson ExSim-Max practice exams that are written and developed by Boson Software's subject-matter experts at www.boson.com. The ExSim-Max practice exams simulate the content on the actual certification exams allowing you to gauge whether you are ready to pass the real exam. When you purchase an ExSim-Max practice exam, you receive an activation key; you can then activate and download the exam by performing the preceding Steps 1 and 2.

The Cisco Learning Network

Cisco provides a wide variety of CCNP preparation tools at a Cisco website called the Cisco Learning Network. This site includes a large variety of exam preparation tools, including sample questions, forums on each Cisco exam, learning video games, and information about each exam.

To reach the Cisco Learning Network, go to www.cisco.com/go/learnnetspace, or just search for "Cisco Learning Network." You need to use the login you created at www.cisco.com. If you don't have such a login, you can register for free. To register, simply go to www.cisco.com, click Register at the top of the page, and supply some information.

Memory Tables

Like most Certification Guides from Cisco Press, this book purposefully organizes information into tables and lists for easier study and review. Re-reading these tables can be very useful before the exam. However, it is easy to skim over the tables without paying attention to every detail, especially when you remember having seen the table's contents when reading the chapter.

Instead of simply reading the tables in the various chapters, this book's Appendixes D and E give you another review tool. Appendix D, "Memory Tables," lists partially completed versions of many of the tables from the book. You can open Appendix D (a PDF on the CD that comes with this book) and print the appendix. For review, you can attempt to complete the tables. This exercise can help you focus on the review. It also exercises the memory connectors in your brain; plus it makes you think about the information without as much information, which forces a little more contemplation about the facts.

Appendix E, "Memory Tables Answer Key," also a PDF located on the CD, lists the completed tables to check yourself. You can also just refer to the tables as printed in the book.

Chapter-Ending Review Tools

Chapters 2 through 19 each have several features in the Exam Preparation Tasks section at the end of the chapter. You may have used some of or all these tools at the end of each chapter. It can also be useful to use these tools again as you make your final preparations for the exam.

Suggested Plan for Final Review/Study

This section lists a suggested study plan from the point at which you finish reading through Chapter 19 until you take the ROUTE exam. Certainly, you can ignore this plan, use it as is, or just take suggestions from it.

The plan uses six steps. If following the plan verbatim, you should proceed by part through the steps as previously listed. That is, starting with Part 2 (EIGRP), do the following six steps. Then, for Part 3 (OSPF), do the following six steps, and so on. The steps are as follows:

Step 1. **Review key topics and DIKTA questions:** You can use the table that lists the key topics in each chapter, or just flip the pages looking for the Key Topics icons. Also, reviewing the DIKTA questions from the beginning of the chapter can be helpful for review.

Step 2. **Complete memory tables:** Open Appendix D on the CD, and print the entire appendix, or print the tables by major part. Then complete the tables and check your answers in Appendix E, which also appears on the CD.

Step 3. **Hands-on practice:** Most people practice CCNP configuration and verification before the exam. Whether you use real gear, a simulator, or an emulator, practice the configuration and verification commands.

Step 4. **Build configuration checklists:** Glance through the Table of Contents, looking for major configuration tasks. Then from memory create your own configuration checklists for the various configuration commands.

Step 5. **Planning practice:** Even if you use the "Planning Practice" tables when you initially read each chapter, repeat the process, particularly the tables related to interpreting a design and reviewing another engineer's implementation plan.

Step 6. **Subnetting practice:** If you can no longer do subnetting well and quickly without a subnetting calculator, take some time to get better and faster before going to take the ROUTE exam.

Step 7. **Use the exam engine to practice:** The exam engine on the CD can be used to study using a bank of 100 unique exam-realistic multiple-choice questions available only with this book.

The rest of this section describes Steps 1, 3, 6, and 7 for which a little more explanation may be helpful.

Step 1: Review Key Topics and DIKTA Questions

This review step focuses on the core facts related to the ROUTE exam. The exam certainly covers other topics as well, but the DIKTA questions and the Key Topics items attempt to focus attention on the more important topics in each chapter.

As a reminder, if you follow this plan after reading the first 19 chapters, working a major part at a time (EIGRP, Chapters 2 through 4, for example) helps you pull each major topic together.

Step 3: Hands-On Practice

Although this book gives you many configuration checklists, specific configuration examples, examples of output, and explanations for the meaning of that output, there is no substitute for hands-on practice. This short section provides a couple of suggestions regarding your efforts at practice from the CLI.

First, most people use one or more of the following options for hands-on skills:

- **Real gear:** Either purchased (often used), borrowed, or rented
- **Simulators:** Software that acts like real gear
- **Emulators:** Software that acts like router hardware, with IOS running inside that environment

First, a few words about finding these tools. For real gear, this book makes no attempt at suggesting how to go about getting, borrowing, or renting gear. However, the topic of what gear to buy for a home Cisco certification lab tends to be a popular topic from time-to-time on my (the author's) blog. Because the blog makes a good place for discussion, but a poor place for reference material, some of the base information about used gear, IOS versions/feature sets, relative to CCNP ROUTE (and other exams) is listed at www.thecertzone.com. The site also lists a link to the blog as well.

As for emulators, two exist of note. For the general public, a group of three free software offerings cooperate to allow you to run multiple instances of IOS on a PC: Dynagen and Dynamips (see www.dynagen.org) for the emulation, and GNS3 for the graphical interface (see www.gns3.net). Many websites devote attention to how to best use these tools; this book simply mentions the tools and their websites to get you started if interested.

Step 6: Subnetting Practice

This book assumes that you have mastered subnetting and the related math. However, many people who progress through CCNA, and move on to CCNP, follow a path like this:

Step 1. Learn subnetting conceptually.

Step 2. Get really good at doing the math quickly.

Step 3. Pass CCNA.

Step 4. Don't practice regularly and therefore become a lot slower at doing the subnetting math.

Step 5. Study for CCNP ROUTE.

Although subnetting should not be assessed as an end to itself on CCNP ROUTE, many questions require that you understand subnetting math and can do that math just as quickly as you did when you passed CCNA. If you are a little slow on doing subnetting math, before you go to the ROUTE exam, try some of the following exercises:

- Practice finding the subnet number, broadcast address, and range of addresses in a subnet. To do so, pick a number and mask, calculate the values, and use your favorite subnet calculator to check your work. Look at the Cisco Learning Network for a calculator if you don't have one.
- Use the Cisco Subnetting Game, also at the Cisco Learning Network.
- Practice choosing the best summary route for a range of subnets. Pick three to four addresses/masks. Calculate the subnet number and range. Then, try to choose the summary (subnet number/mask) that includes those three to four subnets, without including any more subnets than what is required. You can check your math with a subnet calculator as well.

If you like using binary/decimal conversions when you work through these problems, but just need to go faster, check out the Cisco Binary game, also at the Cisco Learning Network.

Step 7: Use the Exam Engine

The Boson Exam Engine (BEE) software on the CD includes a 100-question database of questions created specifically for this book. You can use the BEE software either in study mode or simulation mode, as follows:

- **Study mode:** Study mode is most useful when you want to use the questions to learn and practice. In study mode, you can select options such as whether you want to randomize the order of the questions, randomize the order of the answers, automatically see answers to the questions, refer to specific sections of the text that resides on the CD, and many other options.
- **Simulation mode:** Simulation mode presents questions in a timed environment, providing you with a more exam realistic experience. It also restricts your ability to see your score as you progress through the exam, view answers to questions as you take the exam, and refer to sections of the text. These timed exams not only allow you to study for the actual ROUTE exam, but they also help you simulate the time pressure that can occur on the actual exam.

When doing your final preparation, you can use study mode, simulation mode, or both. However, after you have seen each question a couple of times, you will likely start to

remember the questions, and the usefulness of the exam database may go down. So, consider the following options when using the exam engine:

- Use this question database for review. Use study mode, and study the questions by major book part, just as with the other final review steps listed in this chapter. Plan on getting another exam (possibly from Boson, who supplies the BEE software–www.boson.com) if you want to take simulated exams.
- Save the question database, not using it for review during your review of each book part. Save it until the end, so you will not have seen the questions before. Then, use simulation mode to simulate the exam.

Picking the correct mode from the exam engine's user interface is quite obvious. The following steps show how to move to the screen from which to select study or simulation mode:

Step 1. Click the Choose Exam button, which should list the exam under the title ROUTE/Cisco Press CCNP Route.

Step 2. Click the name of the exam once, which should highlight the exam name.

Step 3. Click the Load Exam button.

By taking these actions, the engine should display a window from which you can choose Simulation Mode or Study Mode with a radio button on the right side of the window and the exam bank in the left side of the window. When selecting Custom Exam, you can further choose the book chapters by using the Modify Settings button, limiting the questions to those explained in the specified chapters of the book.

Summary

The tools and suggestions listed in this chapter have been designed with one goal in mind: to help you develop the skills required to pass the ROUTE exam. This book has been developed from the beginning to not just tell you the facts, but also help you learn how to apply the facts. No matter what your experience level is leading up when you take the exams, it is our hope that the broad range of preparation tools, and even the structure of the books, can help you pass the exams with ease. I hope you do well on the exam.

APPENDIX A

Answers to "Do I Know This Already?" Quizzes

Chapter 2

1. B and C. The **network 172.16.1.0 0.0.0.255** command tells IOS to match the first three octets when comparing the interface IP addresses to the configured "172.16.1.0" value. Only two answers match in the first three octets. The other two answers have a 0 in the 3rd octet, making the addresses not match the **network** command.

2. B. The **show ip eigrp interfaces** command displays working (up/up) interfaces on which EIGRP has been enabled but omits passive interfaces. A failure of the interface, or making the interface passive, would omit the interface from the output of this command.

3. D. The **show ip eigrp interfaces detail** command does display a router's EIGRP Hello timer setting for each enabled interface. The other listed commands do not display the timer. Also, EIGRP routers do not have to have matching Hello timers to become neighbors.

4. C. The **show ip eigrp neighbors failure** command is not a valid IOS command. The debug displays messages that state when a neighborship fails due to authentication each time Hellos are exchanged. The **show key chain** command lists the specific keys that are currently valid, allowing you to determine if both routers use the same key values in currently valid keys. The **show clock** command displays the current time-of-day clock setting on a router, allowing you to check the valid times of the various keys versus the two router's clocks.

5. C. The **neighbor 172.16.2.20 fa0/0** command would only be rejected if the IP address (172.16.2.20) is not inside the range of addresses in the subnet (172.16.2.0/26, range 172.16.2.0–172.16.2.63). This command does not impact interface state. The command does disable all EIGRP multicasts, and because the three dynamically discovered neighbors require the EIGRP multicasts, all three neighbors fail. Although 172.16.2.20 is a valid potential neighbor, both routers must be configured with static **neighbor** commands, and we know that 172.16.2.20 was not previously configured with a static **neighbor** command; otherwise, it could not have been a neighbor with R1.

6. A and D. Table 2-4 lists the issues. For EIGRP, Router IDs do not have to be unique for EIGRP routers to become neighbors, and the hold timer does not have to match between the two neighbors. However, making an interface passive disables the processing of all EIGRP messages on the interface, preventing all neighborships. Mismatched IP subnets also prevent neighborships from forming.

7. A. The configuration requires the **ip authentication mode eigrp** *asn* **md5** command, which is currently missing. This command enables MD5-style authentication, rather than the default of no authentication. Adding this one command completes the configuration. Any valid key numbers can be used. Also, the 9 in the **ip authentication key-chain eigrp 9 fred** command refers to the EIGRP ASN, not an authentication type.

8. A. EIGRP forms neighborships only when two routers can communicate directly over a data link. As a result, with Frame Relay, EIGRP neighborships occur only between routers on the ends of a PVC, so in this case, 100 neighborships exist.

Chapter 3

1. B and C. Other than the two listed correct answers, the local router also adds connected routes for which the **network** command matches the corresponding interfaces, so it may not add all connected routes. Also, EIGRP does not add static routes to the EIGRP topology table, unless those routes are redistributed, as discussed in Chapter 9, "Basic IGP Redistribution."

2. B and D. EIGRP sends bandwidth, delay, reliability, load, MTU, and hop-count in the message. The formula to calculate the metric includes bandwidth, delay, reliability, and load.

3. A. EIGRP performs WAN bandwidth control without any explicit configuration, using default settings. Because no **bandwidth** commands have been configured, each subinterface uses the default 1544 Kbps setting. For S0/0.1, WAN bandwidth control divides the 1544 by 3 (515 Kbps), and then takes the (default) WAN bandwidth of 50 percent, meaning about 250 Kbps for each of the three DLCIs. For the two subinterfaces with one PVC, the default 1544 is multiplied by the 50 percent default WAN bandwidth, meaning that each could use about 750 Kbps.

4. A. This command lists all successor and feasible successor routes. The output states that two successors exist, and only two routes (listed with the "via..." text) exist. So, no feasible successor routes exist.

5. A and C. By default, the metric weights cause EIGRP to consider bandwidth and delay in the metric calculation, so changing either bandwidth or delay impacts the calculation of the feasible distance and reported distance, and impacts choice of feasible successor routes. Offset lists also change the metric, which in turn can change whether a route is an FS route. Link loading would impact the metrics, but not without changing the metric weights to nonrecommended values. Finally, variance impacts which routes end up in the IP routing table, but it is not considered by EIGRP when determining which routes are FS routes.

6. C and E. The EIGRP metric calculation treats bandwidth and delay differently. For bandwidth, EIGRP takes the lowest bandwidth, in Kbps, which is in this case 500 Kbps. For delay, EIGRP takes the cumulative delay, which is 20100 per the various **show interfaces** commands. However, the **show interfaces** command uses a unit of microseconds, and the interface **delay** command, and the EIGRP metric formula uses a unit of tens-of-microseconds, making the delay that feeds into the formula be 2010.

7. C and E. R1, as a stub router with the **connected** option, still advertises routes, but only routes for connected subnets. R1 announces its stub attribute to R2, so R2 chooses to not send Query messages to R1, knowing that R1 cannot be a transit router for other subnets anyway.

8. B. Of the five options, the **show ip route eigrp all-links** and **show ip eigrp topology all-learned** are not valid commands. Both **show ip eigrp topology** and **show ip route eigrp** can show at most successor and feasible successor routes. However, **show ip eigrp topology all-links** shows also nonfeasible successor routes, making it more likely to show all possible neighbors.

9. D. EIGRP considers only successor and feasible successor routes. Each of those routes must have metrics such that variance * metric is less than the best route's metric; the best route's metric is called the feasible distance (FD).

Chapter 4

1. D and E. The two listed commands correctly configure EIGRP route filtering such that prefixes matched by the ACL's permit clause will be allowed. All other prefixes will be filtered due to the implied deny all at the end of the ACL. The ACL permits numbers in the range 10.10.32.0–10.10.47.255, which leaves 10.10.48.0 and 10.10.60.0 unmatched by the permit clause.

2. B, C, and E. Sequence number 5 matches prefixes from 10.1.2.0–10.1.2.255, with prefix lengths between 25–27, and denies (filters) those prefixes. This results in answer A being incorrect, because the prefix length (/24) is not in the correct range. Clause 15 matches prefixes from 10.2.0.0–10.2.255.255, with prefix length exactly 30, matching answer C. Clause 20 matches only prefix 0.0.0.0 with length /0, so only a default route would match this entry. As a result, 10.0.0.0/8 does not match any of the three clauses.

3. C. When used for route filtering, the route map action (permit or deny) defines the filtering action, and any referenced **match** commands' permit or deny action just defines whether the prefix is matched. By not matching ACL 1 with a permit action, EIGRP does not consider a match to have occurred with clause 10, so it moves to clause 20. The prefix list referenced in clause 20 has a permit action, matching prefixes from 10.10.10.0–10.10.11.255, with prefix lengths from 23–25. Both criteria match the prefix in question, making answer C correct.

4. B and C. Answer A is invalid–the **ge** value must be larger than /24 in this case, so the command is rejected. Answer B implies a prefix length range from 24–28, inclusive. Answer C implies a range from 25–32 inclusive, because no **le** parameter exists to limit the prefix length lower than the full length of an IPv4 subnet mask. The same logic applies with answer D, but with a range from 28–32, so this final list could not match prefix lengths of /27.

5. B. 10.1.0.0/18 implies a range from 10.1.0.0–10.1.63.255, which includes none of the four subnets. 10.1.64.0/18 implies a range from 10.1.64.0–10.1.127.255, which includes all subnets. 10.1.100.0/24 implies range 10.1.100.0–10.1.100.255, which leaves out two of the subnets. Finally, 10.1.98.0/22 does not actually represent a summary–instead, 10.1.96.0/22 represents range 10.1.96.0–10.1.99.255, with 10.1.98.0 as listed in answer D being an IP address in that range. As such, IOS would actually accept the command, and change the parameter from 10.1.98.0 to 10.1.96.0, and would not include the four listed subnets.

6. B. The **ip summary-address** command does reset neighborships, but only on the interface under which it is configured. After those neighborships come up, R1 will advertise the summary route, but none of the subordinate routes inside that summary. The summary route will us a metric equal to the metric of the lowest metric subordinate route, approximately 1,000,000 in this case.

7. B and D. R2 has interfaces only in class A network 10.0.0.0, so the auto-summary setting has no effect. R3 has interfaces in both class A network 10.0.0.0 and class B network 172.16.0.0, so auto-summary causes R3 to summarize all subnets of 172.16.0.0/16 as a summary route when advertising to R2.

8. D. The phrase quoted in the question means that R1 is using its route for class A network 2.0.0.0 to decide where to send packets by default. R1's route for network 2.0.0.0 must have 1.1.1.1 as its next-hop router. This phrase occurs when EIGRP has learned a route for class A network 2.0.0.0 that has been flagged as a candidate default route by another router. The router flagging a route as a candidate default route, using the **ip default-network** command, does not actually use the route as its default route.

9. C and E. With the suggested configuration style, the static route must first be configured statically, as shown in answer A. Then, either this route must be redistributed as a static route into EIGRP (answer B), or pulled into EIGRP by virtue of the **network 0.0.0.0** EIGRP subcommand (answer D). The other two options have no effect on default route creation and advertisement.

Chapter 5

1. A and D. The wildcard mask is used for matching the prefix only, and not the prefix length. As such, 172.16.1.0 0.0.0.255 matches all addresses that begin with 172.16.1, and 172.16.0.0 0.0.255.255 matches all addresses that begin 172.16. Also, OSPF reviews the **network** command with the most specific wildcard masks (wildcard masks with the most binary 0's) first, so an interface IP address beginning with 172.16.1 matches the command that references area 8.

2. D. ABRs, by definition, connect the backbone area to one or more nonbackbone areas. To perform this function, a router must have at least one interface assigned to the backbone area, and at least one interface assigned to a nonbackbone area.

3. B and C. First, for the two correct answers: **show ip ospf interface brief** explicitly lists all OSPF-enabled interfaces that are not passive. **show ip protocols** lists either the details of the configured **network** commands, or if configured using the **ip ospf area** command, it lists the interfaces on which OSPF is enabled. This command also lists the passive interfaces, so armed with interface IP address information, the list of OSPF-enabled nonpassive interfaces could be derived. Of the three wrong answers, **show ip ospf database** does not list enough detail to show the OSPF-enabled interfaces. **show ip route ospf** lists only routes learned with OSPF, so if no routes use a particular OSPF-enabled interface as an outgoing interface, this command would not indirectly identify the interface. Finally, an interface may be OSPF-enabled but with no neighbors reachable on the interface, so the **show ip ospf neighbor** command may not identify all OSPF-enabled interfaces.

4. B and C. On a LAN, the non-DR routers form fully adjacent neighborships with only the DR and BDR, giving R1 two neighbors in the FULL state. The other two neighbors settle into the 2WAY state.

5. C and D. The **show ip ospf interface** command displays a router's OSPF Hello Interval setting for each enabled interface. The other listed commands do not display the timer. Also, OSPF routers do need to have matching Hello timers to become neighbors, so the neighborship would fail.

6. B and D. The **area 0 authentication** command tells R1 to use simple text password authentication on all interfaces in area 0 unless overridden by an interface subcommand. So, R1 must have configured the **ip ospf authentication message-digest** on its Fa0/0 interface, enabling MD5 authentication instead. The other correct answer is the command that correctly configures the MD5 authentication key. Of the two incorrect answers that list an authentication key, **ip ospf authentication-key** defines the clear-text key, and the other is not a valid IOS command.

7. E. Table 5-5 in Chapter 5 lists the issues. For OSPF, Router IDs must be unique; the interfaces must not be passive; the dead timers must match; and the primary IP addresses must be in the same subnet, with the same subnet mask. However, the process IDs, found on the **router ospf** *process-id* command, do not have to match.

8. A. Frame Relay is a Layer 2 service and as such does not participate in customer routing protocols. Because the design uses a separate subnet per PVC, and one point-to-point subinterface per PVC/subnet, OSPF will use a point-to-point network type. That means that the two routers on either end of a PVC will become neighbors, and become fully adjacent, meaning the central site router will have 100 fully adjacent neighborships.

Chapter 6

1. D. As an ABR connected to areas 0 and 2, ABR2 will have LSDB entries for both area 0 and area 2. In Area 0, ABR2 learns Type 1 LSAs from the four routers internal to area 0, plus ABR1, and plus 1 for the area 0 Type 1 LSA ABR2 creates for itself. In area 2, ABR2 learns 1 each for the five routers internal to area 2, plus the 1 Type 1 LSA ABR2 created for itself inside area 2. The total is 12.

2. E. OSPF creates a Type 2 LSA for a subnet when the router interface connected to the subnet calls for the election of a designated router (DR), and at least two routers have discovered each other and elected a DR. Then, the DR creates and floods the Type 2 LSA. IOS by default does not elect a DR on point-to-point topologies. It does on router LAN interfaces. One answer states that one router only exists in the subnet, so it does not actually find a second router and elect a DR. In the other case, a DR and BDR have been elected, but the router described in the answer is the BDR, not the DR. So, none of the other answers is correct.

3. C. Each ABR, by definition, creates a single Type 3 LSA to represent a subnet known in one area to be advertised into another area. Assuming 10.100.0.0 is a subnet in area 0, both ABR1 and ABR2 would advertise a Type 3 LSA into area 100. The **show ip ospf database summary** command specifically lists type 3 network summary LSAs.

4. C. The Database Description (DD) packet lists a short LSA header but not the entire LSA. The Link State Request (LSR) packet asks the neighbors for a copy of an LSA. The Link State Update (LSU) holds the LSAs. LSAck simply acknowledges received LSAs, and Hello is used for neighbor discovery and neighbor state maintenance.

5. B and D. Because the subnet was stable before R5 arrived, the other routers will have elected a DR and BDR. OSPF does not preemptively elect a new DR nor BDR, so R5 will be neither (DROther). As a result, R5's messages to the DR will be sent to the 224.0.0.6 all-DR-routers multicast address, and the DR's messages directed to R5 will be sent to the 224.0.0.5 all-SPF-router address.

6. E. R1, internal to area 1, can use LSAs only in the area 1 LSDB. R2's Type 1 LSA exists only in area 2's LSDB. The Type 2 LSA for subnet 10.1.1.0/24, if one exists, also only exists in area 2's LSDB. R1 will use ABR1's Type 1 LSA in area 1 to calculate the possible intra-area routes inside area 1, but R1 will use ABR1's Type 1 LSA in area 1. Finally, the Type 3 LSA, created for 10.1.1.0/24, and flooded into area 1, is also needed to calculate the metric.

7. A and B. OSPF builds the SPF tree based on the topology information Type 1 and Type 2 LSAs. Changes therefore require another SPF run. Changes to the other LSA types do not require an SPF calculation.

8. A and B. Because none of the interfaces have a **bandwidth** command configured, the only commands that can influence the OSPF cost are the **auto-cost reference-bandwidth** router subcommand and the **ip ospf cost** interface subcommand. To give the output shown in the question, either the interface cost could be set directly on all three interfaces listed. Alternatively, the reference-bandwidth could be set (in router

configuration mode) to cause one of the interface costs to be as shown in the output, with the other two interfaces having their costs set directly.

For the wrong answers, the **ip ospf cost interface s0/0/0.1** router subcommand does not exist—instead, it is an interface subcommand. An auto-cost of 64700, used as the numerator in the ref-bw/bandwidth cost calculation, does not result in any of the three listed interface costs.

For the two correct answers, with a default bandwidth of 1544 (Kbps) on the serial subinterfaces, a reference bandwidth of 1000 (Mbps) implies the math 1,000,000 / 1544, for an Interface cost of 647. With a default bandwidth of 100,000 Kbps (100 Mbps) on Fa0/0, a reference bandwidth of 2000 (MBps) implies math of 2,000 / 100 = 20.

9. A, B, and C. OSPF uses Types 1, 2, and 3 for calculating routes internal to the OSPF domain. OSPF uses types 4, 5, and 7 for external routes redistributed into the OSPF domain, as discussed in Chapter 9, "Basic IGP Redistribution."

Chapter 7

1. C. The output lists all R1's routes for subnets within the range of 10.1.0.0–10.1.255.255, whose prefix lengths are longer than /16. One answer lists subnet 10.2.2.0/24, which is not in this range, so the output cannot be used to confirm nor deny whether the subnet was filtered. R1's route for 10.1.2.0/24 is an intra-area route by virtue of not listing an IA code by the route; Type 3 LSA filtering only filters Type 3 LSAs, which routers use to calculate interarea routes, so the output tells us nothing about any filtering of 10.1.2.0/24. The output shows a single interarea route for 10.1.3.0/24, so at least one ABR has flooded a Type 3 LSA for this route. Additionally, the output confirms that at least one ABR flooded a type 3 LSA for 10.1.3.0/24, or the output would not show an IA route for 10.1.3.0/24. So, the type 3 LSA for 10.1.3.0/24 was not filtered by both ABRs.

2. C. When referenced from a distribute list, OSPF filters routes from being added to that router's IP routing table but has no impact on the flow of LSAs. As such, neither A nor B is correct. An OSPF **distribute-list** command does attempt to filter routes from being added to the IP routing table by OSPF, so the two answers that mention the IP routing table might be correct. Sequence number 5 matches prefixes from 10.1.2.0–10.1.2.255, with prefix lengths between 25–27, and denies (filters) those prefixes. So, the prefix list will match 10.1.2.0/26 with the first line, with a deny action. The 10.1.2.0/24 subnet does not match the first line of the prefix list, but it does match the third line, the match all line, with a permit action. Because 10.1.2.0/26 is matched by a deny clause, this route is indeed filtered, so it is not added to R1's IP routing table. 10.1.2.0/24, matched with a permit clause, is allowed and would be in the IP routing table.

3. A. When referenced from an **area filter-list** command, OSPF filters Type 3 LSAs created on that router, preventing them from being flooded into area 1 (per the configuration command). As an ABR, R1 would calculate intra-area routes to these area 0 subnets, so this filtering will have no effect on R1's routes. Sequence number 5

matches prefixes from 10.1.2.0–10.1.2.255, with prefix lengths between 25–27, and denies (filters) those prefixes. So, the prefix list will match 10.1.2.0/26 with the first line, with a deny action. The 10.1.2.0/24 subnet does not match the first line of the prefix list because the prefix length does not match, but it does match the third line, the match all line, with a permit action. By matching subnet 10.1.2.0/26 with a deny action, the filter-list does prevent R1 from flooding a Type 3 LSA for that subnet. By matching 10.1.2.0/24 with a permit action, R1 does not filter the Type 3 LSA for that subnet.

4. B and D. The **area range** command does not cause a failure in neighborships. Because at least one intra-area subordinate subnet of 10.1.0.0/16 exists in R1, R1 both creates a summary route for 10.1.0.0/16 and stops advertising LSAs for the (three) subordinate subnets. By default, the metric of the summary is the metric of the lowest-metric component subnet.

5. D. The **show ip ospf database summary** command lists only Type 3 LSAs. The **summary-address** command creates Type 5 LSAs on ASBRs, ruling out one answer. The output does not specify whether the LSA was created as a summary route; all references to the word "summary" refer to Type 3 Summary LSAs. If created by an **area range** command, the metric defaults to be the best metric of all subordinate subnets, but it may also be explicitly set, ruling out another of the possible answers. In short, this LSA may represent a route summarized by the area range command, but that fact cannot be proved or disproved by the output as shown..

6. B. Without the **always** parameter, the **default-information originate** command generates an LSA for a default route, with prefix 0.0.0.0/0, but only if its own IP routing table has a route for 0.0.0.0/0. It does not flag another LSA as being used as a candidate default route.

7. C and D. Both types of NSSA stubby areas allow the redistribution of external routes into the area, but these routes are advertised as Type 7 LSAs. As a totally NSSA area, the ABR should flood no Type 5 LSAs into the area and flood no Type 3 LSAs into the area, except for the Type 3 LSAs used to advertise the default route into the area. As such, a router internal to a totally stubby are should see zero Type 5 LSAs, and a small number of Type 3 LSAs for the default route(s) advertised by the ABR(s).

8. B. The **stub** keyword means either a stub area or totally stubby. The **no-summary** command means the area is totally stubby.

Chapter 8

1. D. The answer with **area 0 virtual-link 4.4.4.4 cost 3** is incorrect because the show command output lists a transit area of 1, but the answer's **area** parameter refers to area 0 as the transit area. (There is also no **cost** parameter on the **area virtual-link** command.) The RID of the router on the other end of the virtual link, 4.4.4.4 per the **show** command output, does not have to be pingable for the virtual link to work. The cost of the virtual link is 3, but that cost is calculated as the cost to reach the other router through the transit area, so the command output listed with the question can-

not be used to predict Fa0/0's OSPF interface cost alone. However, because the output lists area 1 as the transit area, and because the neighbor RID is listed as 4.4.4.4, R1 will use the area 1 LSDB entries to calculate the cost to reach 4.4.4, a process that will include the area 1 Type 1 LSA for RID 4.4.4.4.

2. B. The **area virtual-link** command defines the virtual link, with the transit area—the area through which the virtual link passes—listed as the first parameter. The other parameter is the RID of the other router. Two of the wrong answers are not IOS commands.

3. D. Of the four types listed, only point-to-multipoint nonbroadcast does not use a DR but does require the static definition of neighbors.

4. C. Of the four types listed, only point-to-multipoint does not use a DR and dynamically discovers neighbors.

5. A and C. For routers to use their OSPF routes in a multipoint design, each router needs mapping to each other router in the Frame Relay subnet. In this case, R3–R10 all need **frame-relay map** commands to define the mapping to other routers with which they do not have a PVC. This network type requires static definition of neighbors, but the neighbor relationships match the PVC topology, so R3–R10 need only two **neighbor** commands. This OSPF network type does not use a DR, so the **ip ospf priority** commands have no effect and would be unnecessary.

Chapter 9

1. D. The three incorrect answers list typical reasons for using route redistribution. The correct answer—the least likely reason among the answers for using route redistribution—lists a problem for which an OSPF virtual link is often used. Route redistribution could be attempted to solve a problem with a discontiguous OSPF area, but the redistribution completely changes the LSAs that would have otherwise been known and could have negative impacts on route summaries and cause routing loops, and have other problems as well.

2. B and D. For a router to redistribute routes between two routing protocols, the router must have both routing protocols configured, have a working link into each routing domain, and configure **redistribute** commands under each routing process. The **redistribute** command, issued in routing protocol configuration mode, pulls routes into that routing process, from another routing process as referenced on the **redistribute** command.

3. B and C. Because the metrics come from a different routing protocol than EIGRP, the metric must be set. The metric must be set with five components; EIGRP will then use those components as it would for an internal route. The metric components may be set as listed in the two correct answers, plus using a route-map as referenced by the **redistribute** command.

4. C. This output is the external data section of a detailed view of an EIGRP topology table entry for an external route. This output confirms that this route was redistributed into EIGRP. If R1 were the redistributing router, the output would include the phrase "(this system)"; this example does not include that notation. The output means that on the router that did the redistribution, the route was redistributed from OSPF process 1, and the OSPF metric was 64. R1's metric is not based on the OSPF metric of the route.

5. B. The **redistribute ospf** command will attempt to redistribute OSPF routes and connected routes from interfaces on which OSPF is enabled. The metric components include 1000 Kbps (or 1 Mbps), 100 tens-of-microseconds (or 1000 microseconds), 10 for the loading, 1 for the reliability, and 1500 for MTU. The EIGRP version of the **redistribute** command does not include a subnets option.

6. A and C. Because the routes come from OSPF and feed into OSPF, the metrics can be set with the usual tools, or the metric can default. When taking routes from OSPF into another OSPF process, the default metric is taken from the source route's OSPF cost. Alternatively, the metric can be set for all routes, regardless of the route source, using the **default-metric** OSPF subcommand. The **metric transparent** keywords cannot be used for an OSPF **redistribute** command.

7. D. This command lists the output of Type 4 Summary ASBR LSAs. The LSID identifies the redistributing ASBR (9.9.9.9). The advertising router is the ABR that created and flooded the LSA (3.3.3.3), and the metric is the ABR's best metric route to reach the ASBR.

8. D. Routers add internal and external costs for E1 routes and use only external costs for E2 routes, so the cost for the route through R22 will always be lower. However, for a given prefix/length, OSPF always prefers intra-area routes first, then interarea, then E1, and finally, E2, all regardless of metric.

Chapter 10

1. E. Because OSPF does not use hop count as a metric, the information about the number of hops is not available in OSPF routes in the IP routing table. The other answers list items that can be matched with the route map **match** subcommand.

2. A. The deny clauses in the route map mean that the route map will filter routes matched by that clause. The permit or deny action of the referenced ACLs just defines whether the route is matched. So, routes permitted by ACL "two" will be matched and then filtered due to the route-map clause deny action. Routes denied by ACL "one" simply do not match the route map clause numbered 10; such routes may or may not be redistributed depending on the next two clauses. Clause number 100 does not have a **match** command, meaning it matches all routes not otherwise matched, with a permit action, allowing these routes to be redistributed.

3. A and C. The problem states that R1 has learned OSPF intra-area routes for 10.1.1.0/24, so **show ip route** will display that subnet. As an intra-area route based on a Type 2 LSA, the **show ip ospf topology** command lists the summary of the LSAs, including the 10.1.1.0 subnet number for that Type 2 LSA. However, because the redistribution filtering discards subnet 10.1.1.0/24, this value will not be included in the EIGRP topology table.

4. B. The **external 2** parameters on the **redistribute** command act as matching logic; only routes from the source routing protocol (in this case OSPF 2) that match this extra logic will be considered for redistribution by this **redistribute** command. The **set metric-type type-1** route-map subcommand sets the route type as it is injected into the destination routing protocol (in this case OSPF 1); this logic is not used for matching the source routes. The routes permitted by ACL 1 will be redistributed, but only those that are also E2 routes from the (source) OSPF 2 domain. The redistribute function will not change the attributes of routes inside a single routing domain, but only in the destination routing domain (OSPF 1), so the configuration has no effect on the OSPF 2 routes that remain in OSPF 2.

5. C. EIGRP, by default, sets a different AD for internal (90) and external (170) routes. The rest of the answers are accurate regarding default settings.

6. A. All the answers list reasonable options in some cases, but the only feature listed that is useful with all three routing protocols is the route tag feature. RIPv2 does not support the concept of differentiating between internal and external routes, so the two answers that suggest setting administrative distance (AD) based on the route type (internal or external) could not be used in all three routing domains, as required by the question. All three routing protocols support setting route tags and setting the AD per route; however, because RIPv2 cannot match based on the route type (internal/external), the option to set the route tags is the only option that applies to all three routing domains.

7. D. AD can be used to prevent the domain loop problem with two routing domains by making each routing protocol's AD for internal routes be better (lower) than the other routing protocol's AD for external routes. RIP uses AD 120 for all routes, with no distinction of internal or external. As such, OSPF's internal default AD settings of 110 meet the requirement that OSPF's internal AD (110) is better than RIP's external (120). However, RIP's default of 120 is not better than OSPF's default for externals (110), so the **distance ospf external 180** command changes that setting to meet both requirements. The three wrong answers, while syntactically valid, do not help meet the requirements.

8. E. Route tags are unitless integers that can be given to a route and even passed between different routing protocols by routers that perform redistribution.

Chapter 11

1. A and C. PBR supports processing packets on an interface, for the inbound direction only. The referenced route map causes PBR to attempt policy routing of packets that match a permit clause in the route map.
2. B and E. Packets created by Telnet use TCP, so the packet will match ACL 101 with a permit action. So, PBR will match the only route map clause shown in the configuration, with that permit route map clause listing a **set** command. The **set** command lists S0/0/1 as the outgoing interface, and without a **default** parameter. So, Router R1 will first attempt to forward the packet based on the **set** command (interface S0/0/1), but if the interface is down, then try to forward based on the IP routing table (interface S0/1/1).
3. D. The output from the **show ip policy** command shows the interfaces on which PBR has been enabled, and the name of the route map enabled for PBR on each interface. For the purposes of this question, the output tells us the interfaces on which PBR has been enabled. Two answers mention packets exiting the interface, so these answers cannot be correct, because PBR applies to packets entering an interface. For the two interfaces that mention inbound packets, one suggests that all packets will be forwarded per the PBR configuration; some may not be forwarded per PBR, depending on the configuration of the route map. The correct answer specifically mentions that PBR will consider all packets with PBR, which is the more accurate statement about PBR operations.
4. A and B. The IP SLA feature focuses on IP traffic, so IOS does not include Novell's older IPX protocol as part of IP SLA. IP SLA uses SNMP MIBs to store statistics, but it does not use SNMP as an operation.
5. C. The three lines shown create the operation number (first command), define the operation (second command), and start the operation (third command). All commands are correct. After the operation is started, IP SLA stores the data in the RTTMON MIB—no additional configuration necessary.
6. D. The up timers on the tracking object defines how long to wait, when in a down state, after seeing the IP SLA object transition to an OK state. Similarly, the down timer defines how long to wait, when in an OK state, after seeing the IP SLA object move to a down state, before moving the tracking object to a down state.

Chapter 12

1. B and E. The private IPv4 address space consists of Class A network 10.0.0.0, Class B networks 172.16.0.0–172.31.0.0, and the 256 Class C networks that begin 192.168.
2. B. ICANN and IANA manage the assignment of public IPv4 address space such that large address blocks (often called CIDR blocks) exist in a particular geography or

are assigned to particular ISPs. As such, Internet routers can more easily create summary routes to help keep the routing table small in the Internet. 200.1.2.0/24 would likely also be allocated to some registrar, ISP, or customer in Asia. Because of the large route summaries, in this case possibly a summary for 200.0.0.0/8, routers in North America would not see an increase in the size of their routing tables.

3. A. The router in ASN 22, R22, advertises the BGP update with (at least) 22 in the AS_Path Path Attribute (PA). When R1 advertises the route to R2, also in ASN 11, R1 does not add an ASN. As a result, R2's AS_Path has at least ASN 22 and not ASN 11.

4. A and C. The public range of 16-bit BGP ASNs is 1 through 64,495.

5. D. The question asks which answers are true about the eBGP peer but also not true about an iBGP peer. Both iBGP and eBGP use TCP port 179. An eBGP peer uses a different ASN than the local router, by definition, making that answer incorrect. The correct answer refers to the fact that an eBGP peer adds its own ASN to the BGP AS_Path PA before sending routing information to another router, whereas iBGP peers do not.

6. A. Although using BGP does avoid some static configuration at the Enterprise and the ISP, the primary reason to consider using BGP in the Enterprise is to influence and react to Path Attributes for the purpose of choosing the best path. Typically, engineers do not redistribute BGP routes into the IGP due to scalability problems. And although it may be interesting to monitor the size of the Internet BGP table, it is not a primary motivation for choosing to use BGP on a router.

7. C and D. The terms "homed" makes reference to a single homed ISP, and "multihomed" to multiple ISPs. The terms "single" and "dual" refer to the number of connections to each ISP.

Chapter 13

1. B and C. The **router bgp** command lists the local ASN, and the **neighbor remote-as** command lists the neighbor's ASN. Because the neighbor relationship uses the IP addresses on the common link, the routers do not need to identify the update source interface, because each will default to use their S0/0 interfaces (in this case) as the update source.

2. D. Three of the commands list valid commands. The **neighbor 2.2.2.2 multihop 2** command is syntactically incorrect; it should be **neighbor 2.2.2.2 ebgp-multihop 2**.

3. D. The **show ip bgp** command lists the BGP neighbor state in the last column of output, listing the literal state, unless in an established state. In that state, the output lists the number of prefixes learned from the neighbor, so a numeric value implies an established state.

4. A and D. The output lists R2's local ASN as ASN 11, a value that is configured in the **router bgp** *asn* command. The line for neighbor 1.1.1.1 lists that router's ASN as 1, so a **neighbor 1.1.1.1 remote-as 1** command should exist on R2 instead of the **neighbor 1.1.1.1 remote-as 11** command. The state for neighbor 1.1.1.1 lists "Idle (Admin)," implying that the **neighbor 1.1.1.1 shutdown** command has been configured. The other answer lists a nonexistent command.

5. A. The BGP Update message lists a set of PAs, plus any prefixes/lengths that use those PAs. It can also list withdrawn routes in the same Update message as newly advertised routes. It can also list multiple prefixes in a single Update message.

6. C. The "Known via" text refers to the local router's (R1's) **router bgp** command, which identifies the local router's ASN. The rest of the output does not identify the neighboring ASN, nor the rest of the AS_Path details. It does list that the route is external, with the text "type external", and the AS Hops (which is the AS_Path length).

7. A. The third character in each line for each router is either blank, meaning the route is an eBGP route, or an "i," meaning an iBGP-learned route. The contents of the AS_Path can be determined (1, 2, 3, 4), but the answer about AS_Path does not suggest 4 ASNs. The best route for each prefix has a ">" in the second character, and this route does not.

8. D. The **network** command will take the route from the IP routing table and put the equivalent into the BGP table, if that exact route exists. The output does not show a route for 130.1.16.0/20, so the **network 130.1.16.0 mask 255.255.240.0** command does not match a specific route. The other answer with a **network** command is syntactically incorrect. Redistribution without aggregation would redistribute the three routes, but all three subordinate routes would be advertised into eBGP. By also using BGP route summarization, a single route for 130.1.16.0/20 can be advertised.

Chapter 14

1. C. R1 needs to be configured with **router bgp 1**, **neighbor 2.2.2.2 remote-as 1**, and **neighbor 2.2.2.2 update-source loopback1**. The **neighbor 2.2.2.2 ibgp-multihop 2** and **neighbor 2.2.2.2 ibgp-mode** commands are simply unsupported commands. The **neighbor 1.1.1.1 remote-as 1** command has correct syntax and is used as a command in R2's configuration but not on R1. The **neighbor 2.2.2.2 remote-as 2** command has correct syntax but with the wrong ASN (2 instead of 1).

2. D. The small letter "i" in the third character position implies the route was learned with iBGP. Of the five lines, four have an "i" in the third column.

3. B and C. The line reading "1.1.1.1 from 2.2.2.2..." implies the BGP RID of the neighbor is 1.1.1.1, with neighbor ID–the IP address on the local router's **neighbor** command–of 2.2.2.2. The end of the output shows that the route is internal (iBGP learned) and is best, so both the > and i will be displayed for this route by the **show ip bgp** command. Finally, the output does not identify the local ASN, although it does list the AS_Path of the route (1, 2, 3, 4).

4. B. By default, when a router advertises an iBGP route, it leaves the Next-Hop PA unchanged. By default, R2's next hop for routes learned from I2 will be I2's IP address used on the R2–I2 neighbor relationship.

5. A and C. The Enterprise core routers need to know which exit point (R1 or R2) is best; the correct answers supply those routes to the routers internal to the company. Note that redistribution from BGP into the IGP is not recommended, but it does defeat this particular problem.

6. B. The show ip bgp neighbors 2.2.2.2 advertised-routes command does list the post-outbound-filter BGP Update; however, the user did not issue a **clear** command, so the filter has not yet taken effect. As such, the output still lists the original three prefixes as if the filter had not yet been applied.

7. B, D, and E. The neighbor distribute-list out command refers to an ACL, but for the ACL to match on both prefix and prefix length, the ACL must be an extended ACL. The neighbor filter-list command refers to an AS-path filter and cannot match based on prefix/length.

8. A and B. The router resets the BGP neighborship when performing a hard reset of the peer. See Table 14-3 in the chapter for a list of several variations of the **clear** command and whether they perform a hard or soft reset.

Chapter 15

1. B. Weight and Local_Pref were created for the purpose of giving engineers tools to influence the BGP best path choice. AS_Path was created for loop avoidance, but AS_Path length can also be manipulated (for instance, with AS_Path prepend) to influence the best path choice. Although the Origin PA can be changed by configuration for the purpose of influencing the best path decision, the intent of this PA is to identify the source from which the route was introduced into BGP. Additionally, the best path algorithm considers the Origin PA after the other PAs listed in the answers, making Origin the least useful of these answers for influencing path choice.

2. A. Of the items listed in the question, Weight is the first one considered in the best path algorithm, with a bigger weight being better. As a result, Route 1 is the better route of the two.

3. B. Of the items listed in the question, Weight is the first one considered in the best path algorithm, and it is a tie. The next item considered, Local Preference, uses bigger-is-better logic, so Route 2 will be considered best.

4. B and D. Weight, a Cisco-proprietary feature of BGP on Cisco routers, cannot be transmitted in a BGP Update, so setting Weight on an outbound route map at the ISPs will have no effect. Also, the goals call for setting Weight for all routes from an ISP to the same number, so creating a prefix list to match a subset of reachable prefixes, in this case all class C networks, is not useful. However, two methods of configuring Weight do exist: the **neighbor weight** command and configuring an inbound route map with a **set weight** command in the route map.

5. B and C. The line reading 1.1.1.1 from 2.2.2.2... implies the BGP RID of the neighbor is 1.1.1.1, with neighbor ID–the IP address on the local router's **neighbor** command–of 2.2.2.2. The end of the output shows that the route is internal (iBGP learned), and the output lists the word "best," so the **show ip bgp** command will display both the > and i for this route. Finally, the output does not identify the local ASN, although it does list the AS_Path of the route (1, 2, 3, 4).

6. B. The output shows the results of AS_Path prepending. The repetitive 1's cannot mean that the route has been advertised into and out of the same ASN repeatedly because loop prevention would have prevented such an advertisement. With AS_Path prepending, the neighboring ASN typically adds its own ASN to the end of the AS_Path (as listed on the left of the output).

7. C. The command lists the administrative distance as the first number inside the square brackets and the MED values as the second number in brackets. The AD of 20 implies an eBGP route instead of iBGP. The output says nothing about the Weight or AS_Path length.

Chapter 16

1. D. Inside a quartet, any leading 0s can be omitted, and one sequence of one or more quartets of all 0s can be replaced with "::". The correct answer replaces the longer three-quartet sequence of 0s with ::.

2. C. The name of the prefix generally represents the group to which the prefix is given, with the exception of the term *global routing*. IANA assigns a prefix to a registry (registry prefix). The registry may assign a subset of that range as a prefix to an ISP (ISP prefix). That ISP then subdivides that range of addresses into prefixes and assigns a prefix to one of its customers (site prefix, also called global routing prefix). The Enterprise network engineers then further subdivides the range, often with prefix length 64, into subnet prefixes.

3. A and C. IPv6 supports stateful DHCP, which works similarly to IPv4's DHCP protocol to dynamically assign the entire IP address. Stateless autoconfiguration also allows for the assignment by finding the prefix from some nearby router and calculating the Interface ID using EUI-64 format. Stateless DHCP simply supplies the DNS server IP addresses, and NDP supplies Layer 2 mapping information.

4. D. Stateless autoconfiguration only helps a host learn and form its own IP address, but it does not help the host learn a default gateway. Stateless RS is not a valid term or feature. Neighbor Discovery Protocol (NDP) is used for several purposes, including the same purpose as ARP in IPv4, plus to learn configuration parameters such as a default gateway IP address.

5. D. Global unicast addresses begin with 2000::/3, meaning the first three bits match the value in hex 2000. Similarly, unique local addresses match FD00::/8, and link local addresses match FE80::/10 (values that begin with FE8, FE9, FEA, and FEB hex). Multicast IPv6 addresses begin FF00::/8, meaning the first two hex digits are F.

6. B. When created automatically, link local addresses begin FE80::/64, because after the prefix of FE80::/10, the device builds the next 54 bits as binary 0s. Statically assigned link local addresses simply need to confirm to the FE80::/10 prefix. As a result, only two answers are candidates with a beginning quartet of FE80. Of these, only one has only hex 0s in the second, third, and fourth quartets, making answer B the only valid answer.

7. A and C. The **ipv6 address** command does not list an **eui-64** parameter, so R1 does not form its global unicast address using the EUI-64 format. However, it does form its link local address using EUI-64. The **show ipv6 interface brief** command lists both the global unicast and link local addresses in its output.

8. A. The group addresses listed in the output are the all IPv6 hosts address (FF02::1), the all IPv6 routers address (FF02::2), and the solicited node address that is based on R1's global unicast address (FF02::1:FF:12:3456). Also, R1's global unicast address is listed correctly in answer B, but the "[EUI]" notation implies that R1 derived the interface ID portion using EUI-64 conventions.

Chapter 17

1. A, B, and D. RIP-2 and RIPng both use UDP, both use Distance Vector logic, and both use the same metric, with the same maximum (15) and same metric that means infinity (16). RIPng does not perform automatic route summarization because IPv6 has no concept of a classful network. RIPng also uses the built-in IPv6 authentication mechanisms, rather than a RIP-specific authentication such as RIP-2.

2. B. That the configuration will be copied/pasted into a router means that the order of the commands matters. In this case, that the **ipv6 rip one enable** command precedes the **ipv6 address** command on interface f0/0 means that IOS will reject the first of these commands, therefore not enabling RIPng on F0/0. The correct order listed under S0/0/0 means that RIPng will be enabled on S0/0/0. As a result, RIPng on R1 will advertise about S0/0/0's connected IPv6 prefixes, and send Updates on S0/0/0, but will do nothing related for F0/0.

3. D. Because the question states that no IPv4 configuration exists, EIGRP for IPv6 cannot derive a 32-bit EIGRP router ID. Before EIGRP will work, R1 needs to define an EIGRP router ID (using the **eigrp router-id** command) and enable EIGRP using the **no shutdown** router subcommand.

4. B. EIGRP uses the link local address as the next hop for routing protocols. Based on R2's MAC address, R2's link local address on F0/0 will be FE80::1311:11FF:FE11:1111. This value is derived by splitting the MAC, inserting FFFE, and flipping bit 7, making the initial hex 11 become hex 13.

5. A and C. OSPFv3 supports multiple OSPF instances per interface, whereas OSPFv2 does not. Also, each version requires a different set of requirements be met before becoming neighbors, most notably that OSPFv3 does not require neighboring OSPFv3 routers to be in the same subnet.

6. C. Because the question states that no IPv4 configuration exists, OSPFv3 cannot derive a 32-bit OSPF router ID. Before OSPFv3 works, R1 needs to define an OSPF router ID (using the OSPF **router-id** command).

7. A. The **redistribute** command does not have a **subnets** option for IPv6 because OSPFv3 has no concept of IPv6 classful networks nor their subnets. As configured, the **redistribute** command redistributes only EIGRP-learned routes. If the **include-connected** parameter had been included, the connected route for 2000:0:0:1::/64 would have also been redistributed. Local routes are never redistributed.

8. B. The only configurable item after the interface that does not first list a keyword is the administrative distance parameter.

Chapter 18

1. C. Native IPv6 makes the most sense when the IPv6 deployment is pervasive, with traffic loads heavy or at least steady. Point-to-point tunnels work best when IPv6 is needed in only a subset of sites but also when the traffic should be somewhat regular or with higher volume. Multipoint tunnels also work well when IPv6 is needed in a subset of sites, but it is more appropriate when the traffic is more occasional and lower volume. Finally, NAT-PT is useful when an IPv4-only host needs to communicate with an IPv6-only host.

2. D. Native IPv6 makes the most sense when the IPv6 deployment is pervasive, with traffic loads heavy or at least steady.. Point-to-point tunnels work best when IPv6 is needed in only a subset of sites, but also when the traffic should be somewhat regular or with higher volume. Multipoint tunnels also work well when IPv6 is needed in a subset of sites, but it is more appropriate when the traffic is more occasional and lower volume. Finally, NAT-PT is useful when an IPv4-only host needs to communicate with an IPv6-only host.

3. E. Dual stacks means that the host runs both IPv4 and IPv6. A host must run both if the host is to send packets of each protocol. The host may use a multipoint tunnel, but the other three answers list features applicable to routers, but not hosts.

4. B. A manually configured tunnel explicitly defines the destination IPv4 address, as does a GRE IPv6 tunnel. The other two types listed in the answer do not. Additionally, a manually configured tunnel uses mode **ipv6ip**, whereas GRE tunnels use mode **gre**.

5. C. IOS defaults to use GRE encapsulation mode on tunnel interfaces, so the **tunnel mode** command is not required. The mode will default to GRE. The **tunnel source** and **tunnel destination** commands are required.

6. B and D. The two point-to-point tunneling methods—manually configured tunnel and GRE—support IPv6 IGPs. The two multipoint tunneling methods—automatic 6to4 and ISATAP—do not.

7. A. An automatic 6to4 tunnel does not use a **tunnel destination** command on the tunnel interface; ISATAP tunnels also do not use this command. However, automatic 6to4 tunnels use IPv6 addresses that begin 2002::/16 and have the tunnel's source IPv4 address imbedded as the second and third octets of the IPv6 address, whereas ISATAP tunnels do not. Because C0A5:101 hex equals 192.168.1.1 in dotted decimal, this configuration represents the almost completed configuration for an automatic 6to4 tunnel.

8. A and D. ISATAP uses a modified EUI-64 format, which adds the IPv4 address, in hex, into quartets 7 and 8. Automatic 6to4 tunnels use address range 2002::/16, with the next two quartets (second and third quartets) used to store the hex version of an IPv4 address.

9. D. The combination of the configured eui-64 parameter on the **ipv6 address** command, and the tunnel mode of **isatap**, tells the router to use modified EUI-64 rules. These rules start with the configured 64 bit prefix (2000::/64 in this case), adding 0000:5EFE as the fifth and sixth quartets. The last two quartets are taken from the **tunnel source** command's referenced IPv4 address. In this case, 192.168.1.1 converts to C0A8:0101, making the last answer correct.

Chapter 19

1. C. IPsec tunnels make for more secure communications, including encryption and authentication. However, it does not support IGP communications across the tunnel.

2. A and C. An IPsec tunnel would be useful to allow the packet to pass over the Internet and into the Enterprise. The GRE tunnel would only be needed if an IGP is also needed, and for this design, an IGP is not required. Instead, a floating static default route would work fine, with the static route sending traffic over the IPsec tunnel but only when the private leased line fails.

3. D. The ATM details, like VPI/VCI, will be configured under the ATM interface. PPP (including CHAP) and Layer 3 details will be configured under the dialer interface.

4. C. The NAT configuration acts only on packets permitted by a referenced ACL. As a result, the ACL can permit packets destined for the Internet, performing NAT on those packets. The ACL also denies packets going to the Enterprise, meaning that the router does not apply NAT to those packets.

5. B. As for the correct answer, the process of routing a packet out a GRE tunnel interface triggers the GRE encapsulation action. As for the incorrect answers: There is no **tunnel gre** *acl* command. There is no IPsec tunnel interface. Finally, one answer refers to logic that would describe a router's logic when determining whether to encapsulate a packet into an IPsec tunnel.

APPENDIX B

Conversion Tables

This appendix lists two conversion tables for reference when studying:

- Hex-to-decimal
- Decimal-to-binary

Use these tables for learning; however, such tables will not be available on the exam.

Table B-1 *Hex-to-Decimal Conversion Table*

Hex	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Table B-2 *Binary-to-Decimal Conversion Table*

Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value
0	00000000	32	00100000	64	01000000	96	01100000
1	00000001	33	00100001	65	01000001	97	01100001
2	00000010	34	00100010	66	01000010	98	01100010
3	00000011	35	00100011	67	01000011	99	01100011
4	00000100	36	00100100	68	01000100	100	01100100
5	00000101	37	00100101	69	01000101	101	01100101
6	00000110	38	00100110	70	01000110	102	01100110
7	00000111	39	00100111	71	01000111	103	01100111
8	00001000	40	00101000	72	01001000	104	01101000
9	00001001	41	00101001	73	01001001	105	01101001
10	00001010	42	00101010	74	01001010	106	01101010
11	00001011	43	00101011	75	01001011	107	01101011
12	00001100	44	00101100	76	01001100	108	01101100
13	00001101	45	00101101	77	01001101	109	01101101
14	00001110	46	00101110	78	01001110	110	01101110
15	00001111	47	00101111	79	01001111	111	01101111
16	00010000	48	00110000	80	01010000	112	01110000
17	00010001	49	00110001	81	01010001	113	01110001
18	00010010	50	00110010	82	01010010	114	01110010
19	00010011	51	00110011	83	01010011	115	01110011
20	00010100	52	00110100	84	01010100	116	01110100
21	00010101	53	00110101	85	01010101	117	01110101
22	00010110	54	00110110	86	01010110	118	01110110
23	00010111	55	00110111	87	01010111	119	01110111
24	00011000	56	00111000	88	01011000	120	01111000
25	00011001	57	00111001	89	01011001	121	01111001
26	00011010	58	00111010	90	01011010	122	01111010
27	00011011	59	00111011	91	01011011	123	01111011
28	00011100	60	00111100	92	01011100	124	01111100
29	00011101	61	00111101	93	01011101	125	01111101
30	00011110	62	00111110	94	01011110	126	01111110
31	00011111	63	00111111	95	01011111	127	01111111

Table B-2 *Binary-to-Decimal Conversion Table*

Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value	Decimal Value	Binary Value
128	10000000	160	10100000	192	11000000	224	11100000
129	10000001	161	10100001	193	11000001	225	11100001
130	10000010	162	10100010	194	11000010	226	11100010
131	10000011	163	10100011	195	11000011	227	11100011
132	10000100	164	10100100	196	11000100	228	11100100
133	10000101	165	10100101	197	11000101	229	11100101
134	10000110	166	10100110	198	11000110	230	11100110
135	10000111	167	10100111	199	11000111	231	11100111
136	10001000	168	10101000	200	11001000	232	11101000
137	10001001	169	10101001	201	11001001	233	11101001
138	10001010	170	10101010	202	11001010	234	11101010
139	10001011	171	10101011	203	11001011	235	11101011
140	10001100	172	10101100	204	11001100	236	11101100
141	10001101	173	10101101	205	11001101	237	11101101
142	10001110	174	10101110	206	11001110	238	11101110
143	10001111	175	10101111	207	11001111	239	11101111
144	10010000	176	10110000	208	11010000	240	11110000
145	10010001	177	10110001	209	11010001	241	11110001
146	10010010	178	10110010	210	11010010	242	11110010
147	10010011	179	10110011	211	11010011	243	11110011
148	10010100	180	10110100	212	11010100	244	11110100
149	10010101	181	10110101	213	11010101	245	11110101
150	10010110	182	10110110	214	11010110	246	11110110
151	10010111	183	10110111	215	11010111	247	11110111
152	10011000	184	10111000	216	11011000	248	11111000
153	10011001	185	10111001	217	11011001	249	11111001
154	10011010	186	10111010	218	11011010	250	11111010
155	10011011	187	10111011	219	11011011	251	11111011
156	10011100	188	10111100	220	11011100	252	11111100
157	10011101	189	10111101	221	11011101	253	11111101
158	10011110	190	10111110	222	11011110	254	11111110
159	10011111	191	10111111	223	11011111	255	11111111

APPENDIX C

Route Exam Updates

Over time, reader feedback allows Cisco Press to gauge which topics give our readers the most problems when taking the exams. To assist readers with those topics, the authors created new materials clarifying and expanding upon those troublesome exam topics. As mentioned in the introduction, the additional content about the exam is contained in a PDF document on this book's companion website, at http://www.ciscopress.com/title/9781587202537.

This appendix is intended to provide you with updated information if Cisco makes minor modifications to the exam upon which this book is based. When Cisco releases an entirely new exam, the changes are usually too extensive to provide in a simple update appendix. In those cases, you might need to consult the new edition of the book for the updated content.

This appendix attempts to fill the void that occurs with any print book. In particular, this appendix does the following:

- Mentions technical items that might not have been mentioned elsewhere in the book
- Covers new topics if Cisco adds new content to the exam over time
- Provides a way to get up-to-the-minute current information about content for the exam

Always Get the Latest at the Companion Website

You are reading the version of this appendix that was available when your book was printed. However, given that the main purpose of this appendix is to be a living, changing document, it is important that you look for the latest version online at the book's companion website. To do so:

Step 1 Browse to http://www.ciscopress.com/title/9781587202537.

Step 2 Select the Appendix option under the More Information box.

Step 3 Download the latest "Appendix C" document.

Note Note that the downloaded document has a version number. Comparing the version of this print Appendix C (Version 1.0) with the latest online version of this appendix, you should do the following:

- Same version—Ignore the PDF that you downloaded from the companion website.
- Website has a later version—Ignore this Appendix C in your book and read only the latest version that you downloaded from the companion website.

Technical Content

The current version of this appendix does not contain any additional technical coverage.

This page intentionally left blank

Index

A

B

C

D

E

F

G

H

I

J - K - L

M

N

O

P

Q - R

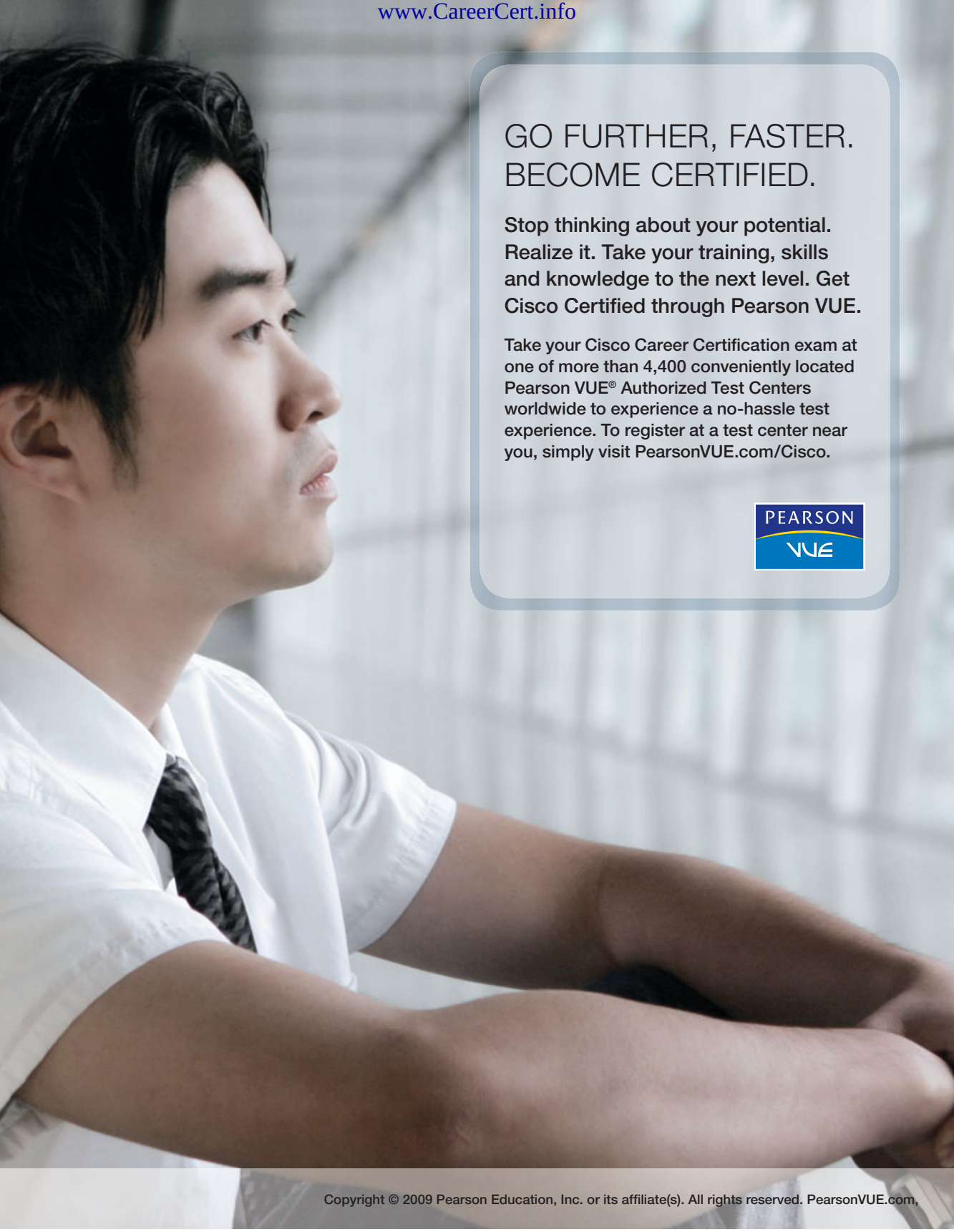
S

T

U

V

W - X - Y - Z



FREE Online Edition

Your purchase of **CCNP ROUTE 642-902 Official Certification Guide** includes access to a free online edition for 45 days through the Safari Books Online subscription service. Nearly every Cisco Press book is available online through Safari Books Online, along with more than 5,000 other technical books and videos from publishers such as Addison-Wesley Professional, Exam Cram, IBM Press, O'Reilly, Prentice Hall, Que, and Sams.

SAFARI BOOKS ONLINE allows you to search for a specific answer, cut and paste code, download chapters, and stay current with emerging technologies.

Activate your FREE Online Edition at www.informit.com/safarifree

- **STEP 1:** Enter the coupon code: CMNJZAA.
- **STEP 2:** New Safari users, complete the brief registration form. Safari subscribers, just log in.

If you have difficulty registering on Safari or accessing the online edition, please e-mail customer-service@safaribooksonline.com

Safari Books Online

Addison Wesley · Adobe Press · Alpha · Cisco Press · FT Press · IBM Press · lynda.com · Microsoft Press · New Riders · O'Reilly · Peachpit Press · Prentice Hall · Que · Redbooks · Sams · SAS Publishing · Sun Microsystems · Wharton School Publishing · Wiley

APPENDIX D

Memory Tables

Chapter 2

Table 2-2 *Key EIGRP Verification Commands*

Command	Key Information
show ip eigrp interfaces	
show ip protocols	
show ip eigrp neighbors	
show ip eigrp topology	
show ip route	

Table 2-3 *EIGRP Feature Summary*

Feature	Description
Transport	
Metric	
Hello interval	
Hold timer	
Update destination address	
Full or partial updates	
Authentication	

VLSM/classless
Route Tags
Next-hop field
Manual route summarization
Automatic Summarization
Multiprotocol

Table 2-4 *Neighbor Requirements for EIGRP and OSPF*

Requirement	EIGRP	OSPF
The routers must be able to send/receive IP packets to one another.	Yes	Yes
Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
Must not be passive on the connected interface.		
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.		
Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.		
Must pass neighbor authentication (if configured).		
Must be in same area.	N/A	Yes
IP MTU must match.	No	Yes
K-values (used in metric calculation) must match.		
Router IDs must be unique.		

Chapter 3

Table 3-3 *Parameters on the* ***eigrp stub*** *Command*

Option	This router is allowed to...
Connected	
Summary	
Static	
Redistributed	
Receive-only	

Chapter 4

Table 4-3 *LE and GE Parameters on IP Prefix Lis, and the Implied Range of Prefix Lengths*

Prefix List Parameter	Range of Prefix Length
Neither	
Both **ge** and **le**	
Only **le**	
Only **ge**	

Chapter 5

Table 5-2 *Commonly Used OSPF Terms*

Term	Definition
Link state database	
Shortest Path First (SPF)	
Link State Update (LSU)	
Link State Advertisement (LSA)	
Area	
Area Border Router (ABR)	
Backbone router	
Internal routers	
Designated Router (DR)	
Backup Designated Router (BDR)	

Table 5-3 *Most Commonly Used OSPF* ***show*** *Commands*

Command	Key Information
show ip ospf interface brief	
show ip protocols	
show ip ospf neighbors	
show ip ospf database	
show ip route	

Table 5-4 *OSPF Feature Summary*

Feature	Description
Transport	
Metric	
Hello interval	
Dead interval	
Update destination address	
Full or partial updates	
Authentication	
VLSM/classless	
Route Tags	
Next-hop field	
Manual route summarization	

Table 5-5 *Neighbor Requirements for EIGRP and OSPF*

Requirement	OSPF	EIGRP
Interfaces' primary IP addresses must be in same subnet.		
Must not be passive on the connected interface.		
Must be in same area.		
Hello interval/timer plus either the Hold (EIGRP) or Dead (OSPF) timer must match.		
Router IDs must be unique.		
IP MTU must match.		
Must pass neighbor authentication (if configured).		
K-values (used in metric calculation) must match.		
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.		

Table 5-8 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Default Hello Interval	Dynamic Discovery of Neighbors?	More Than Two Routers Allowed in the Subnet?
Broadcast				
Point-to-point[1]				
Loopback				
Nonbroadcast[2] (NBMA)	Yes	30	No	Yes
Point-to-multi-point	No	30	Yes	Yes
Point-to-multi-point nonbroad-cast	No	30	No	Yes

[1]Default on Frame Relay point-to-point subinterfaces.

[2]Default on Frame Relay physical and multipoint subinterfaces.

Chapter 6

Table 6-2 *OSPF LSA Types*

LSA Type	Common Name	Description
1		
2		
3		
4	ASBR Summary	Like a type 3 LSA, except it advertises a host route used to reach an ASBR.
5	AS External	Created by ASBRs for external routes injected into OSPF.
6	Group Membership	Defined for MOSPF; not supported by Cisco IOS.
7	NSSA External	Created by ASBRs inside an NSSA area, instead of a type 5 LSA.
8	External Attributes	Not implemented in Cisco routers.
9–11	Opaque	Used as generic LSAs to allow for easy future extension of OSPF; for example, type 10 has been adapted for MPLS traffic engineering.

Table 6-3 *Facts about LSA Types 1, 2, and 3*

LSA Type (Number)	LSA Type (Name)	This Type Represents	Display Using show ip ospf database ***keyword...***	LSID Is Equal To	Created By
1					
2					
3					

Chapter 7

Table 7-3 *Stub Area Configuration Options*

Action	Configuration Steps
Stubby	
Totally stubby	
Set the metric of the default route	

Table 7-4 *OSPF Stubby Area Types*

Area Type	ABRs flood Type 5 External LSAs into the area?	ABRs flood Type 3 Summary LSAs into the area?	Allows redistribution of external LSAs into the stubby area?
Stub			
Totally stubby			
NSSA			
Totally NSSA			

Chapter 8

Table 8-2 *Configuring OSPF Authentication on Virtual Links*

Type (Name)	Type (Number)	Command Syntax for Virtual Links
none		
clear text		
MD5		

Table 8-3 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Dynamic Discovery of Neighbors?	Default Hello Interval	Cisco Proprietary?
broadcast				Yes
nonbroadcast				No
point-to-multipoint				Yes
point-to-multipoint nonbroadcast				No

Chapter 9

Table 9-3 *Methods of Setting EIGRP Metrics When Redistributing into EIGRP*

Function	Command
Setting the default for all **redistribute** commands	
Setting the component metrics applied to all routes redistributed by a single **redistribute** command	
Setting different component metrics to different routes from a single route source	

Table 9-5 *Summary of Metric Values When Redistributing into OSPF*

Function	Command or Metric Values
Default if no metric configuration exists	
Setting the default for all redistribute commands	
Setting the metric for one route source	
Setting different metrics for routes learned from a single source	

Chapter 10

Table 10-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	
Static	
EIGRP summary route	
eBGP	
EIGRP (internal)	
IGRP	
OSPF	
IS-IS	
RIP	
On-Demand Routing (ODR)	
EIGRP (external)	
iBGP	
Unreachable	

Chapter 12

Table 12-2 *Private IP Address Reference*

Number of Classful Networks	Range of Classful Networks	Prefix for Entire Range
(1) Class A:		
(16) Class B:		
(256) Class C:		

Table 12-5 *16-Bit ASN Assignment Categories from IANA*

Value or Range	Purpose
0	
1 through 64,495	
64,496 through 65,511	
64,512 through 65,534	
65,535	

Chapter 13

Table 13-2 *BGP Neighbor States*

State	Typical Reasons
Idle	
Connect	
Active	
Opensent	
Openconfirm	
Established	

Table 13-3 *BGP Message Types*

Message	Purpose	Similarity with EIGRP
Open		
Keepalive		
Update		
Notification		

Table 13-4 *Verification Commands for eBGP-Learned Routes*

Verification Step	Command
List possible default routes.	
List possible routes, per prefix.	
List routes learned from one neighbor, before any inbound filtering is applied.	
List routes learned from a specific neighbor that passed any inbound filters.	
Lists routes advertised to a neighbor after applying outbound filtering.	
List the number of prefixes learned per neighbor.	

Chapter 14

Table 14-3 *BGP clear Command Options*

Command	Hard or Soft	One or All Neighbors	Direction (in or out)
clear ip bgp *			
clear ip bgp *neighbor-id*			
clear ip bgp *neighbor-id* out			
clear ip bgp *neighbor-id* soft out			
clear ip bgp *neighbor-id* in			
clear ip bgp *neighbor-id* soft in			
clear ip bgp * soft			
clear ip bgp *neighbor-id* soft			

Chapter 15

Table 15-2 *BGP Path Attributes That Affect the BGP Best Path Algorithm*

PA	Description	Enterprise Route Direction (Typical)
NEXT_HOP		
Weight[1]		
Local Preference (LOCAL_PREF)		
AS_PATH (length)		
ORIGIN		
Multi Exit Discriminator (MED)		

[1]Weight is not a BGP PA; it is a Cisco-proprietary feature that acts somewhat like a PA.

Table 15-3 *BGP Decision Process Plus Mnemonic: N WLLA OMNI*

Step	Mnemonic letter	Short Phrase	Which Is Better?
0	N		
1	W		
2	L		
3	L		
4	A		
5	O		
6	M		
7	N		
8	I		

Table 15-4 *Key Features of Administrative Weight*

Feature	Description
Is it a PA?	
Purpose	
Scope	
Range	0 through 65,535 ($2^{16} - 1$)
Which is best?	
Default	
Defining a new default	Not supported
Configuration	

Table 15-5 *Key Features of Local_Pref*

Feature	Description
PA?	
Purpose	
Scope	
Range	0 through 4,294,967,295 ($2^{32} - 1$)
Which is best?	
Default	
Changing the default	Using the **bgp default local-preference <0-4294967295>** BGP sub-command
Configuration	

Table 15-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	
Static	
EIGRP summary route	
eBGP	
EIGRP (internal)	
IGRP	
OSPF	
IS-IS	
RIP	
On-Demand Routing (ODR)	
EIGRP (external)	
iBGP	
Unreachable	

Table 15-7 *Key Features of MED*

Feature	Description
Is it a PA?	
Purpose	
Scope	
Range	0 through 4,294,967,295 ($2^{32} - 1$).
Which is best?	
Default	
Configuration	

Chapter 16

Table 16-4 *Example IPv6 Prefixes and Their Meanings*[1]

Term	Assignment	Example from Chapter 16
Registry prefix		
ISP prefix		
Site prefix or global routing prefix		
Subnet prefix		

Although an RIR can assign a prefix to an ISP, an RIR may also assign a prefix to other Internet registries, which might subdivide and assign additional prefixes, until eventually an ISP and then their customers are assigned some unique prefix.

Table 16-5 *Summary of IPv6 Address Assignment for Global Unicast Addresses*

Method	Dynamic or Static	Prefix and length learned from...	Host learned from...	Default router learned from...	DNS addresses learned from...
Stateful DHCP					
Stateless autoconfig					
static configuration					
Static config with EUI-64					

Table 16-6 *Details of the RS/RA Process*

Message	RS	RA
Multicast destination		
Meaning of Multicast address		

Table 16-7 *Comparing Stateless and Stateful DHCPv6 Services*

Feature	Stateful DHCP	Stateless DHCP
Remembers IPv6 address (state information) of clients that make requests		
Assigns IPv6 address to client		
Supplies useful information, such as DNS server IP addresses		
Most useful in conjunction with stateless autoconfiguration		

Table 16-9 *Common Link-Local Multicast Addresses*

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet		
Unique local	Unicast packets inside one organization		
Link local	Packets sent in the local subnet		
Site local	Deprecated; originally meant to be used like private IPv4 addresses	FECO::/10	FEC, FED, FEE, FEF
Unspecified	An address used when a host has no usable IPv6 address		
Loopback	Used for software testing, like IPv4's 127.0.0.1		

IPv6 RFCs define the FE80::/10 prefix, which technically means that the first three hex digits could be FE8, FE9, FEA, or FEB. However, bit positions 11-64 of link local addresses should be 0, so in practice, link local addresses should always begin with FE80.

Table 16-10 *Common Multicast Addresses*

Purpose	IPv6 Address	IPv4 Equivalent
All IPv6 nodes on the link		
All IPv6 routers on the link		
OSPF messages		
RIP-2 messages		
EIGRP messages		
DHCP relay agents (routers that forward to the DHCP server)	FF02:1:2	N/A
DHCP servers (site scope)	FF05::1:3	N/A
All NTP servers (site scope)	FF05::101	N/A

Table 16-11 *Router IOS IPv6 Configuration Command Reference*

Command	Description
ipv6 address *address/length*	
ipv6 address *prefix/length* **eui-64**	
ipv6 address autoconfig	
ipv6 address dhcp	
ipv6 unnumbered *interface-type number*	
ipv6 enable	
ipv6 address *address* **link-local**	
ipv6 address *address/length* **anycast**	

Chapter 17

Table 17-3 *Comparing RIP-2 to RIPng*

Feature	RIP-2	RIPng
Advertises routes for...		
RIP messages use these Layer 3/4 protocols		
UDP Port		
Use Distance Vector		
Default Administrative distance		
Supports VLSM		
Can perform automatic summarization		
Uses Split Horizon		
Uses Poison Reverse		
30 second periodic full updates		
Uses triggered updates		
Uses Hop Count metric		
Metric meaning infinity		
Supports route tags		
Multicast Update destination		
Authentication		

Table 17-4 *Comparing Verification Commands:* **show ip** *and* **show ipv6**

Function	IPv4	IPv6
All routes		
All RIP learned routes		
Details on the routes for a specific prefix		
Interfaces on which RIP is enabled		
List of routing information sources		
Debug that displays sent and received Updates		

Table 17-5 *Comparing EIGRP for IPv4 and IPv6*

Feature	EIGRP for IPv4	EIGRP for IPv6
Advertises routes for...		
Layer 3 protocol for EIGRP messages		
Layer 3 header protocol type		
UDP Port		
Uses Successor, Feasible Successor logic		
Uses Dual		
Supports VLSM		
Can perform automatic summarization		
Uses triggered updates		
Uses composite metric, default using bandwidth and delay		
Metric meaning infinity		
Supports route tags		
Multicast Update destination		
Authentication		

Table 17-6 *Comparing EIGRP Verification Commands:* **show ip** *and* **show ipv6...**

Function	show ip...	show ipv6...
All routes		
All EIGRP learned routes		
Details on the routes for a specific prefix		
Interfaces on which EIGRP is enabled, plus metric weights, variance, redistribution, max-paths, admin distance		
List of routing information sources		
Hello interval		
EIGRP database		
Debug that displays sent and received Updates		

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
Advertises routes for...		
OSPF messages use this Layer 3 protocol		
IP Protocol Type		
Uses Link State logic		
Supports VLSM		
Process to choose RID, compared to OSPFv2		
LSA flooding and aging compared to OSPFv2		
Area structure compared to OSPFv2		
Packet types and uses compared to OSPFv3 (Table 6-4)		
LSA flooding and aging compared to OSPFv2		

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
RID based on highest up/up loopback IPv4 address, or highest other IPv4 interface address?		
32-bit LSID		
Uses interface cost metric, derived from interface bandwidth		
Metric meaning infinity		
Supports route tags		
Elects DR based on highest priority, then highest RID		
Periodic reflooding every...		
Multicast–all SPF routers		
Multicast–All Designated routers		
Authentication		
Neighbor checks compared to OSPFv2 (table 5-5)		
Multiple instances per interface		

Table 17-8 *Comparing OSPF Verification Commands:* **show ip** *and* **show ipv6...**

Function	show ipv4...	show ipv6...
All OSPF-learned routes		
Router ID, Timers, ABR, SPF statistics		
List of routing information sources		
Interfaces assigned to each area		
OSPF interfaces–costs, state, area, number of neighbors		
Detailed information about OSPF interfaces		
Displays summary of OSPF database		

Chapter 18

Table 18-3 *Comparing Manual and GRE IPv6-over-IP Tunnels*

	Manual Tunnels	GRE
RFC	4213	2784
Tunnel mode command		
Passenger MTU default	1480	1476
Supports IPv6 IGPs?		
Forwards IPv6 multicasts?		
Uses static configuration of tunnel destination?		
Supports multiple passenger protocols?		
Link local based on...		

Table 18-4 *Comparing IPv6 Multipoint Tunnels*

	Automatic 6to4	ISATAP
Defined by RFC or Cisco?	3056	4214
Uses a reserved IPv6 address prefix.		
Supports the use of global unicast addresses?		
Quartets holding the IPv4 destination address.		
End-user host addresses embed the IPv4 destination?		
Tunnel endpoints IPv6 addresses encode IPv4 destination.		
Uses modified EUI-64 to form tunnel IPv6 addresses?		

This page intentionally left blank

APPENDIX E

Memory Tables Answer Key

Chapter 2

Table 2-2 *Key EIGRP Verification Commands*

Command	Key Information
show ip eigrp interfaces	Lists the working interfaces on which EIGRP is enabled (based on the **network** commands); it omits passive interfaces.
show ip protocols	Lists the contents of the **network** configuration commands for each routing process, and a list of neighbor IP addresses.
show ip eigrp neighbors	Lists known neighbors; does not list neighbors for which some mismatched parameter is preventing a valid EIGP neighbor relationship.
show ip eigrp topology	Lists all successor and feasible successor routes known to this router. It does not list all known topology details. (See Chapter 3 for more detail on successors and feasible successors.)
show ip route	Lists the contents of the IP routing table, listing EIGRP-learned routes with a code of D on the left side of the output.

Table 2-3 *EIGRP Feature Summary*

Feature	Description
Transport	IP, protocol type 88 (does not use UDP or TCP).
Metric	Based on constrained bandwidth and cumulative delay by default, and optionally load and reliability.
Hello interval	Interval at which a router sends EIGRP Hello messages on an interface.

Table 2-3 *EIGRP Feature Summary*

Feature	Description
Hold timer	Timer used to determine when a neighboring router has failed, based on a router not receiving any EIGRP messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.10, with retransmissions being sent to each neighbor's unicast IP address. Can also be sent to the neighbor's unicast IP address.
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used.
Authentication	Supports MD5 authentication only.
VLSM/classless	EIGRP includes the mask with each route, also allowing it to support discontiguous networks and VLSM.
Route Tags	Allows EIGRP to tag routes as they are redistributed into EIGRP.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Manual route summarization	Allows route summarization at any point in the EIGRP network.
Automatic Summarization	EIGRP supports, and defaults to use, automatic route summarization at classful network boundaries.
Multiprotocol	Supports the advertisement of IPX and AppleTalk routes, and IP Version 6, which is discussed in Chapter 17 of this book.

Table 2-4 *Neighbor Requirements for EIGRP and OSPF*

Requirement	EIGRP	OSPF
The routers must be able to send/receive IP packets to one another.	Yes	Yes
Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
Must not be passive on the connected interface.	Yes	Yes
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.	Yes	No
Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	No	Yes
Must pass neighbor authentication (if configured).	Yes	Yes
Must be in same area.	N/A	Yes
IP MTU must match.	No	Yes
K-values (used in metric calculation) must match.	Yes	N/A
Router IDs must be unique.	No[1]	Yes

Chapter 3

Table 3-3 *Parameters on the **eigrp stub** Command*

Option	This router is allowed to...
Connected	Advertise connected routes but only for interfaces matched with a network command.
Summary	Advertise auto-summarized or statically configured summary routes.
Static	Advertises static routes, assuming the **redistribute static** command is configured.
Redistributed	Advertises redistributed routes, assuming redistribution is configured.
Receive-only	Does not advertise any routes. This option cannot be used with any other option.

Chapter 4

Table 4-3 *LE and GE Parameters on IP Prefix Lis, and the Implied Range of Prefix Lengths*

Prefix List Parameter	Range of Prefix Length
Neither	*conf-length must = route-length*
Both **ge and le**	*ge-value* <= *route-length* <= *le-value*
Only **le**	*conf-length* <= *route-length* <= *le-value*
Only **ge**	*ge-value* <= *route-length* <= 32

Chapter 5

Table 5-2 *Commonly Used OSPF Terms*

Term	Definition
Link state database	The data structure held by an OSPF router for the purpose of storing topology data.
Shortest Path First (SPF)	The name of the algorithm OSPF uses to analyze the LSDB. The analysis determines the best (lowest cost) route for each prefix/length.
Link State Update (LSU)	The name of the OSPF packet that holds the detailed topology information, specifically LSAs
Link State Advertisement (LSA)	The name of a class of OSPF data structures that hold topology information. LSAs are held in memory in the LSDB and communicated over the network in LSU messages.
Area	A contiguous grouping of routers and router interfaces. Routers in an area strive to learn all topology information about the area, but they do not learn topology information about areas to which they do not connect.

Area Border Router (ABR)	A router that has interfaces connected to at least two different OSPF areas, including the backbone area. ABRs hold topology data for each area, and calculate routes for each area, and advertise about those routes between areas.
Backbone router	Any router that has at least one interface connected to the backbone area.
Internal routers	A router that has interfaces connected to only one area, making the router completely internal to that one area.
Designated Router (DR)	On multiaccess data links like LANs, an OSPF router elected by the routers on that data link to perform special functions. These functions include the generation of LSAs representing the subnet, and playing a key role in the database exchange process.
Backup Designated Router (BDR)	A router on a multiaccess data link that monitors the DR and becomes prepared to take over for the DR, should the DR fail.

Table 5-3 *Most Commonly Used OSPF* ***show*** *Commands*

Command	Key Information
show ip ospf interface brief	Lists the interfaces on which OSPF is enabled (based on the **network** commands); it omits passive interfaces.
show ip protocols	Lists the contents of the **network** configuration commands for each routing process, and a list of enabled but passive interfaces.
show ip ospf neighbors	Lists known neighbors, including neighbor state; does not list neighbors for which some mismatched parameter is preventing a valid OSPF neighbor relationship.
show ip ospf database	Lists all LSAs for all connected areas. (See Chapter 6 for more detail on the LSA types seen in the database.)
show ip route	Lists the contents of the IP routing table, listing OSPF-learned routes with a code of O on the left side of the output.

Table 5-4 *OSPF Feature Summary*

Feature	Description
Transport	IP, protocol type 89 (does not use UDP or TCP).
Metric	Based on cumulative cost of all outgoing interfaces in a route. The interface cost defaults to a function of interface bandwidth but can be set explicitly.
Hello interval	Interval at which a router sends OSPF Hello messages on an interface.
Dead interval	Timer used to determine when a neighboring router has failed, based on a router not receiving any OSPF messages, including Hellos, in this timer period.
Update destination address	Normally sent to 224.0.0.5 (All SPF Routers) and 225.0.0.6 (All Designated Routers).
Full or partial updates	Full updates are used when new neighbors are discovered; otherwise, partial updates are used.
Authentication	Supports MD5 and clear-text authentication.
VLSM/classless	OSPF includes the mask with each route, also allowing it to support discontiguous networks and VLSM.
Route Tags	Allows OSPF to tag routes as they are redistributed into OSPF.
Next-hop field	Supports the advertisement of routes with a different next-hop router than the advertising router.
Manual route summarization	Allows route summarization at ABR routers only.

Table 5-5 *Neighbor Requirements for EIGRP and OSPF*

Requirement	OSPF	EIGRP
Interfaces' primary IP addresses must be in same subnet.	Yes	Yes
Must not be passive on the connected interface.	Yes	Yes
Must be in same area.	Yes	N/A
Hello interval/timer, plus either the Hold (EIGRP) or Dead (OSPF) timer, must match.	Yes	No
Router IDs must be unique.	Yes	No
IP MTU must match.	Yes[1]	No
Must pass neighbor authentication (if configured).	Yes	Yes
K-values (used in metric calculation) must match.	N/A	Yes
Must use the same ASN (EIGRP) or process-ID (OSPF) on the **router** configuration command.	No	Yes

Table 5-8 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Default Hello Interval	Dynamic Discovery of Neighbors?	More Than Two Routers Allowed in the Subnet?
Broadcast	Yes	10	Yes	Yes
Point-to-point[1]	No	10	Yes	No
Loopback	No	—	—	No
Nonbroadcast[2] (NBMA)	Yes	30	No	Yes
Point-to-multipoint	No	30	Yes	Yes
Point-to-multipoint nonbroadcast	No	30	No	Yes

[1]Default on Frame Relay point-to-point subinterfaces.

[2]Default on Frame Relay physical and multipoint subinterfaces.

Chapter 6

Table 6-2 *OSPF LSA Types*

LSA Type	Common Name	Description
1	Router	Each router creates its own Type 1 LSA to represent itself for each area to which it connects. The LSDB for one area contains one Type 1 LSA per router per area, listing the RID and all interface IP addresses on that router that are in that area. Represents stub networks as well.
2	Network	One per transit network. Created by the DR on the subnet, and represents the subnet and the router interfaces connected to the subnet.
3	Net Summary	Created by ABRs to represent subnets listed in one area's type 1 and 2 LSAs when being advertised into another area. Defines the links (subnets) in the origin area, and cost, but no topology data.
4	ASBR Summary	Like a type 3 LSA, except it advertises a host route used to reach an ASBR.
5	AS External	Created by ASBRs for external routes injected into OSPF.
6	Group Membership	Defined for MOSPF; not supported by Cisco IOS.
7	NSSA External	Created by ASBRs inside an NSSA area, instead of a type 5 LSA.
8	External Attributes	Not implemented in Cisco routers.
9–11	Opaque	Used as generic LSAs to allow for easy future extension of OSPF; for example, type 10 has been adapted for MPLS traffic engineering.

Table 6-3 *Facts about LSA Types 1, 2, and 3*

LSA Type (Number)	LSA Type (Name)	This Type Represents	Display Using show ip ospf database ***keyword...***	LSID Is Equal To	Created By
1	Router	A router	**router**	RID of router	Each router creates its own
2	Network	A subnet in which a DR exists	**network**	DR's IP address in the subnet	The DR in that subnet
3	Summary	Subnet in another area	**summary**	Subnet number	An ABR

Chapter 7

Table 7-3 *Stub Area Configuration Options*

Action	Configuration Steps
Stubby	**Configure area** *area-id* **stub on each router in the area**
Totally stubby	**Configure area** *area-id* **stub no-summary command on the ABRs** **Configure area** *area-id* **stub, without the no-summary keyword, on all other routers in the area**
Set the metric of the default route	**Configure area** *area-id* **default-metric** *metric* **on an ABR; can differ from ABR to ABR. Default value is 1.**

Table 7-4 *OSPF Stubby Area Types*

Area Type	ABRs flood Type 5 External LSAs into the area?	ABRs flood Type 3 Summary LSAs into the area?	Allows redistribution of external LSAs into the stubby area?
Stub	No	Yes	No
Totally stubby	No	No	No
NSSA	No	Yes	Yes
Totally NSSA	No	No	Yes

Chapter 8

Table 8-2 *Configuring OSPF Authentication on Virtual Links*

Type (Name)	Type (Number)	Command Syntax for Virtual Links
none	0	**area** num **virtual-link** router-id **authentication null**
clear text	1	**area** num **virtual-link** router-id **authentication authentication-key** key-value
MD5	2	**area** num **virtual-link** router-id **authentication message-digest message-digest-key** key-num **md5** key-value

Table 8-3 *OSPF Network Types*

Interface Type	Uses DR/BDR?	Dynamic Discovery of Neighbors?	Default Hello Interval	Cisco Proprietary?
broadcast	Yes	Yes	10	Yes
nonbroadcast	Yes	No	30	No
point-to-multipoint	No	Yes	30	Yes
point-to-multipoint nonbroadcast	No	No	30	No

Chapter 9

Table 9-3 *Methods of Setting EIGRP Metrics When Redistributing into EIGRP*

Function	Command
Setting the default for all redistribute commands	The **default-metric** *bw delay reliability load mtu* EIGRP subcommand
Setting the component metrics applied to all routes redistributed by a single **redistribute** command	The **metric** *bw delay reliability load mtu* parameters on the **redistribute** command
Setting different component metrics to different routes from a single route source	Use the **route-map** parameter on the **redistribute** command, matching routes and setting metric components.

Table 9-5 *Summary of Metric Values When Redistributing into OSPF*

Function	Command or Metric Values
Default if no metric configuration exists	Cost 1 for routes learned from BGP. If redistributed from another OSPF process, use the source route's OSPF cost. Cost 20 for all other route sources.
Setting the default for all **redistribute** commands	The **default-metric** cost OSPF subcommand.
Setting the metric for one route source	The metric cost parameters on the **redistribute** command.
Setting different metrics for routes learned from a single source	Use the **route-map** parameter on the **redistribute** command.

Chapter 10

Table 10-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
On-Demand Routing (ODR)	160
EIGRP (external)	170
iBGP	200
Unreachable	255

Chapter 12

Table 12-2 *Private IP Address Reference*

Number of Classful Networks	Range of Classful Networks	Prefix for Entire Range
(1) Class A:	10.0.0.0	10.0.0.0/8
(16) Class B:	172.16.0.0 through 172.31.0.0	172.16.0.0/12
(256) Class C:	192.168.0.0 through 192.168.255.0	192.168.0.0/16

Table 12-5 *16-Bit ASN Assignment Categories from IANA*

Value or Range	Purpose
0	Reserved
1 through 64,495	Assignable by IANA for public use
64,496 through 65,511	Reserved for use in documentation
64,512 through 65,534	Private use
65,535	Reserved

Chapter 13

Table 13-2 *BGP Neighbor States*

State	Typical Reasons
Idle	The BGP process is either administratively down or awaiting the next retry attempt.
Connect	The BGP process is waiting for the TCP connection to be completed. You cannot determine from this state information whether the TCP connection can complete.
Active	The TCP connection has been completed, but no BGP messages have been sent to the peer yet.
Opensent	The TCP connection exists, and a BGP Open message has been sent to the peer, but the matching Open message has not yet been received from the other router.
Openconfirm	An Open message has been both sent to and received from the other router. The next step is to receive a BGP Keepalive message (to confirm all neighbor-related parameters matched) or BGP Notification message (to learn there is some mismatch in neighbor parameters).
Established	All neighbor parameters match, the neighbor relationship works, and the peers can now exchange Update messages.

Table 13-3 *BGP Message Types*

Message	Purpose	Similarity with EIGRP
Open	Used to establish a neighbor relationship and exchange basic parameters, including ASN and MD5 authentication values.	Hello
Keepalive	Sent on a periodic basis to maintain the neighbor relationship. The lack of receipt of a Keepalive message within the negotiated Hold timer causes BGP to bring down the neighbor connection.	Hello
Update	Used to exchange PAs and the associated prefix/length (NLRI) that use those attributes.	Update
Notification	Used to signal a BGP error; typically results in a reset to the neighbor relationship.	No direct equivalent

Table 13-4 *Verification Commands for eBGP-Learned Routes*

Verification Step	Command
List possible default routes.	**show ip bgp 0.0.0.0 0.0.0.0**
List possible routes, per prefix.	**show ip bgp** *prefix [subnet-mask]*
List routes learned from one neighbor, before any inbound filtering is applied.	**show ip bgp neighbors** *ip-address* **received-routes**
List routes learned from a specific neighbor that passed any inbound filters.	**show ip bgp neighbors** *ip-address* **routes**
Lists routes advertised to a neighbor after applying outbound filtering.	**show ip bgp neighbors** *ip-address* **advertised-routes**
List the number of prefixes learned per neighbor.	**show ip bgp summary**

Chapter 14

Table 14-3 *BGP* **clear** *Command Options*

Command	Hard or Soft	One or All Neighbors	Direction (in or out)
clear ip bgp *	Hard	all	both
clear ip bgp *neighbor-id*	Hard	one	both
clear ip bgp *neighbor-id* **out**	Soft	one	out
clear ip bgp *neighbor-id* **soft out**	Soft	one	out
clear ip bgp *neighbor-id* **in**	Soft	one	in
clear ip bgp *neighbor-id* **soft in**	Soft	one	in
clear ip bgp * **soft**	Soft	all	both
clear ip bgp *neighbor-id* **soft**	Soft	one	both

Chapter 15

Table 15-2 *BGP Path Attributes That Affect the BGP Best Path Algorithm*

PA	Description	Enterprise Route Direction (Typical)
NEXT_HOP	Lists the next-hop IP address used to reach a prefix.	N/A
Weight[1]	A numeric value, range 0 through $2^{16} - 1$, set by a router when receiving Updates, influencing that one router's route for a prefix. Not advertised to any BGP peers.	Outbound
Local Preference (LOCAL_PREF)	A numeric value, range 0 through $2^{32} - 1$, set and communicated throughout a single AS for the purpose of influencing the choice of best route for all routers in that AS.	Outbound
AS_PATH (length)	The number of ASNs in the AS_Path PA.	Outbound, Inbound
ORIGIN	Value implying the route was injected into BGP; I (IGP), E (EGP), or ? (incomplete information).	Outbound
Multi Exit Discriminator (MED)	Set and advertised by routers in one AS, impacting the BGP decision of routers in the other AS. Smaller is better.	Inbound

[1] Weight is not a BGP PA; it is a Cisco-proprietary feature that acts somewhat like a PA.

Table 15-3 *BGP Decision Process Plus Mnemonic: N WLLA OMNI*

Step	Mnemonic letter	Short Phrase	Which Is Better?
0	N	Next hop: reachable?	If no route to reach Next_Hop, router cannot use this route.
1	W	Weight	Bigger.
2	L	LOCAL_PREF	Bigger.
3	L	Locally injected routes	Locally injected is better than iBGP/eBGP learned.
4	A	AS_PATH length	Smaller.
5	O	ORIGIN	Prefer I over E. Prefer E over ?
6	M	MED	Smaller.
7	N	Neighbor Type	Prefer eBGP over iBGP.
8	I	IGP metric to Next_Hop	Smaller.

Table 15-4 *Key Features of Administrative Weight*

Feature	Description
Is it a PA?	No; Cisco proprietary feature
Purpose	Identifies a single router's best route
Scope	Set on inbound route Updates; influences only that one router's choice
Range	0 through 65,535 ($2^{16} - 1$)
Which is best?	Bigger values are better
Default	0 for learned routes, 32,768 for locally injected routes
Defining a new default	Not supported
Configuration	**neighbor route-map** (per prefix) **neighbor weight** (all routes learned from this neighbor)

Table 15-5 *Key Features of Local_Pref*

Feature	Description
PA?	Yes
Purpose	Identifies the best exit point from the AS to reach a given prefix
Scope	Throughout the AS in which it was set; not advertised to eBGP peers
Range	0 through 4,294,967,295 ($2^{32} - 1$)
Which is best?	Higher values are better
Default	100
Changing the default	Using the **bgp default local-preference <0-4294967295>** BGP subcommand
Configuration	Via **neighbor route-map** command; **in** option is required for updates from an **eBGP** peer

Table 15-6 *Default Administrative Distances*

Route Type	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
On-Demand Routing (ODR)	160

EIGRP (external)	170
iBGP	200
Unreachable	255

Table 15-7 *Key Features of MED*

Feature	Description
Is it a PA?	Yes.
Purpose	Allows an AS to tell a neighboring AS the best way to forward packets into the first AS.
Scope	Advertised by one AS into another, propagated inside the AS, but not sent to any other anonymous systems.
Range	0 through 4,294,967,295 ($2^{32} - 1$).
Which is best?	Smaller is better.
Default	0
Configuration	Via **neighbor route-map out** command, using the **set metric** command inside the route map.

Chapter 16

Table 16-4 *Example IPv6 Prefixes and Their Meanings*

Term	Assignment	Example from Chapter 16
Registry prefix	By IANA to an RIR	2340::/12
ISP prefix	By an RIR to an ISP[1]	2340:1111/32
Site prefix or global routing prefix	By an ISP or registry to a customer (site)	2340:1111:AAAA/48
Subnet prefix	By an Enterprise engineer for each individual link	2340:1111:AAAA:0001/64

[1] Although an RIR can assign a prefix to an ISP, an RIR may also assign a prefix to other Internet registries, which might subdivide and assign additional prefixes, until eventually an ISP and then their customers are assigned some unique prefix.

Table 16-5 *Summary of IPv6 Address Assignment for Global Unicast Addresses*

Method	Dynamic or Static	Prefix and length learned from...	Host learned from...	Default router learned from...	DNS addresses learned from...
Stateful DHCP	Dynamic	DHCP Server	DHCP Server	Router, using NDP	(Stateful) DHCP Server
Stateless autoconfig	Dynamic	Router, using NDP	Derived from MAC	Router, using NDP	Stateless DHCP
static configuration	Static	Local config	Local config	Router, using NDP	Stateless DHCP
Static config with EUI-64	Static	Local config	Derived from MAC	Router, using NDP	Stateless DHCP

Table 16-6 *Details of the RS/RA Process*

Message	RS	RA
Multicast destination	FF02::2	FF02::1
Meaning of Multicast address	All routers on this link	All IPv6 nodes on this link

Table 16-7 *Comparing Stateless and Stateful DHCPv6 Services*

Feature	Stateful DHCP	Stateless DHCP
Remembers IPv6 address (state information) of clients that make requests	Yes	No
Assigns IPv6 address to client	Yes	No
Supplies useful information, such as DNS server IP addresses	Yes	Yes
Most useful in conjunction with stateless autoconfiguration	No	Yes

Table 16-9 *Common Link-Local Multicast Addresses*

Type of Address	Purpose	Prefix	Easily Seen Hex Prefix(es)
Global unicast	Unicast packets sent through the public Internet	2000::/3	2 or 3
Unique local	Unicast packets inside one organization	FD00::/8	FD
Link local	Packets sent in the local subnet	FE80::/10	FE8
Site local	Deprecated; originally meant to be used like private IPv4 addresses	FECO::/10	FEC, FED, FEE, FEF
Unspecified	An address used when a host has no usable IPv6 address	::/128	N/A
Loopback	Used for software testing, like IPv4's 127.0.0.1	::1/128	N/A

IPv6 RFCs define the FE80::/10 prefix, which technically means that the first three hex digits could be FE8, FE9, FEA, or FEB. However, bit positions 11-64 of link local addresses should be 0, so in practice, link local addresses should always begin with FE80.

Table 16-10 *Common Multicast Addresses*

Purpose	IPv6 Address	IPv4 Equivalent
All IPv6 nodes on the link	FF02::1	subnet broadcast address
All IPv6 routers on the link	FF02::2	N/A
OSPF messages	FF02::5, FF02::6	224.0.0.5, 224.0.0.6
RIP-2 messages	FF02::9	224.0.0.9
EIGRP messages	FF02::A	224.0.0.10
DHCP relay agents (routers that forward to the DHCP server)	FF02:1:2	N/A
DHCP servers (site scope)	FF05::1:3	N/A
All NTP servers (site scope)	FF05::101	N/A

Table 16-11 *Router IOS IPv6 Configuration Command Reference*

Command	Description
ipv6 address *address/length*	Static configuration of the entire IPv6 unicast address.
ipv6 address *prefix/length* **eui-64**	Static configuration of the first 64 address bits; the router derives the last 64 bits with EUI-64.
ipv6 address autoconfig	Router uses stateless autoconfig to find address.
ipv6 address dhcp	Router uses stateful DHCP to find address.
ipv6 unnumbered *interface-type number*	Uses the same IPv6 unicast address as the referenced interface.
ipv6 enable	Results in only a link local address.
ipv6 address *address* **link-local**	Overrides the automatically created link local address. The configured value must conform to the FE80::/10 prefix.
ipv6 address *address/length* **anycast**	Designates that the unicast address is an anycast.

Chapter 17

Table 17-3 *Comparing RIP-2 to RIPng*

Feature	RIP-2	RIPng
Advertises routes for...	IPv4	IPv6
RIP messages use these Layer 3/4 protocols	IPv4, UDP	IPv6, UDP
UDP Port	520	521
Use Distance Vector	Yes	Yes
Default Administrative distance	120	120
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses Split Horizon	Yes	Yes
Uses Poison Reverse	Yes	Yes
30 second periodic full updates	Yes	Yes
Uses triggered updates	Yes	Yes
Uses Hop Count metric	Yes	Yes
Metric meaning infinity	16	16
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.9	FF02::9
Authentication	RIP-specific	uses IPv6 AH/ESP

Table 17-4 *Comparing Verification Commands:* **show ip** *and* **show ipv6**

Function	IPv4	IPv6
All routes	**... route**	**... route**
All RIP learned routes	**... route rip**	**... route rip**
Details on the routes for a specific prefix	**... route** *subnet mask*	**... route** *prefix/length*
Interfaces on which RIP is enabled	**... protocols**	**... protocols**
List of routing information sources	**... protocols**	**... rip next-hops**
Debug that displays sent and received Updates	**debug ip rip**	**debug ipv6 rip**

Table 17-5 *Comparing EIGRP for IPv4 and IPv6*

Feature	EIGRP for IPv4	EIGRP for IPv6
Advertises routes for...	IPv4	IPv6
Layer 3 protocol for EIGRP messages	IPv4	IPv6
Layer 3 header protocol type	88	88
UDP Port	N/A	N/A
Uses Successor, Feasible Successor logic	Yes	Yes
Uses Dual	Yes	Yes
Supports VLSM	Yes	Yes
Can perform automatic summarization	Yes	N/A
Uses triggered updates	Yes	Yes
Uses composite metric, default using bandwidth and delay	Yes	Yes
Metric meaning infinity	$2^{32} - 1$	$2^{32} - 1$
Supports route tags	Yes	Yes
Multicast Update destination	224.0.0.10	FF02::10
Authentication	EIGRP-specific	Uses IPv6 AH/ESP

Table 17-6 *Comparing EIGRP Verification Commands:* **show ip** *and* **show ipv6...**

Function	show ip...	show ipv6...
All routes	... **route**	... **route**
All EIGRP learned routes	... **route eigrp**	... **route eigrp**
Details on the routes for a specific prefix	... **route** *subnet mask*	... **route** *prefix/length*
Interfaces on which EIGRP is enabled, plus metric weights, variance, redistribution, max-paths, admin distance	... **protocols**	... **protocols**
List of routing information sources	... **protocols** ... **eigrp neighbors**	... **eigrp neighbors**
Hello interval	... **eigrp interfaces detail**	... **eigrp interfaces detail**
EIGRP database	... **eigrp topology [all-links]**	... **eigrp topology [all-links]**
Debug that displays sent and received Updates	**debug ip eigrp notifications**	**debug ipv6 eigrp notifications**

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
Advertises routes for...	IPv4	IPv6
OSPF messages use this Layer 3 protocol	IPv4	IPv6
IP Protocol Type	89	89
Uses Link State logic	Yes	yes
Supports VLSM	Yes	Yes
Process to choose RID, compared to OSPFv2	Same	Same
LSA flooding and aging compared to OSPFv2	Same	Same
Area structure compared to OSPFv2	Same	Same
Packet types and uses compared to OSPFv3 (Table 6-4)	Same	Same
LSA flooding and aging compared to OSPFv2	Same	Same
RID based on highest up/up loopback IPv4 address, or highest other IPv4 interface address?	Yes	yes
32-bit LSID	Yes	yes
Uses interface cost metric, derived from interface bandwidth	Yes	Yes
Metric meaning infinity	$2^{16} - 1$	$2^{16} - 1$
Supports route tags	Yes	Yes
Elects DR based on highest priority, then highest RID	Yes	Yes
Periodic reflooding every...	30 minutes	30 minutes
Multicast–all SPF routers	224.0.0.5	FF02::5
Multicast–All Designated routers	224.0.0.6	FF02::6

Table 17-7 *Comparing OSPFv2 and OSPFv3*

Feature	OSPFv2	OSPFv3
Authentication	OSPF-specific	Uses IPv6 AH/ESP
Neighbor checks compared to OSPFv2 (table 5-5)	Same	Same, except no "same subnet" check
Multiple instances per interface	No	Yes

Table 17-8 *Comparing OSPF Verification Commands:* **show ip** *and* **show ipv6...**

Function	show ipv4...	show ipv6...
All OSPF-learned routes	**... route ospf**	**... route ospf**
Router ID, Timers, ABR, SPF statistics	**... ospf**	**... ospf**
List of routing information sources	**... protocols** **... ospf neighbor**	**... ospf neighbor**
Interfaces assigned to each area	**... protocols** **... ospf interface brief**	**... protocols** **... ospf interface brief**
OSPF interfaces–costs, state, area, number of neighbors	**... interface brief**	**... interface brief**
Detailed information about OSPF interfaces	**... ospf interface**	**... ospf interface**
Displays summary of OSPF database	**... ospf database**	**... ospf database**

Chapter 18

Table 18-3 *Comparing Manual and GRE IPv6-over-IP Tunnels*

	Manual Tunnels	GRE
RFC	4213	2784
Tunnel mode command	tunnel mode ipv6ip	tunnel mode gre ip
Passenger MTU default	1480	1476
Supports IPv6 IGPs?	Yes	Yes
Forwards IPv6 multicasts?	Yes	Yes
Supports multiple passenger protocols?	No	Yes
Link local based on...	FE80::/96, plus 32 bits from tunnel source IPv4 address	IPv6 EUI-64, using lowest numbered interface's MAC address

Table 18-4 *Comparing IPv6 Multipoint Tunnels*

	Automatic 6to4	ISATAP
Defined by RFC or Cisco?	3056	4214
Uses a reserved IPv6 address prefix.	Yes (2002::/16)	No
Supports the use of global unicast addresses?	Yes	Yes
Quartets holding the IPv4 destination address.	2/3	7/8
End-user host addresses embed the IPv4 destination?	Sometimes	No
Tunnel endpoints IPv6 addresses encode IPv4 destination.	Sometimes	Yes
Uses modified EUI-64 to form tunnel IPv6 addresses?	No	Yes

This page intentionally left blank

APPENDIX F

Completed Planning Practice Tables

Chapter 2

Table 2-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Improve EIGRP convergence.	Tune EIGRP Hold and Hello timers so that neighbor failures are recognized more quickly.
Implement EIGRP on each router so that neighborships are formed (2).	Discover neighbors using multicasts as a result of matching an interface with the EIGRP **network** command, in **router eigrp** configuration mode. Allow only specific neighbors on an interface by configuring the **neighbor** command, in **router eigrp** configuration mode.
Limit neighborship formation on interfaces matched with an EIGRP network command (3).	Use EIGRP authentication to allow only neighbors with the correct keys. Prevent all neighborships on an interface by making the interface passive. Allow only specific neighbors on an interface by configuring static neighbor.

Table 2-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
What happens on a router interface on which an **EIGRP network** command matches the interface? (2)	EIGRP attempts to discover EIGRP neighbors by sending and receiving multicast EIGRP Hellos. EIGRP advertises about the subnet on the connected interface.
What configuration settings prevent EIGRP neighbor discovery on an EIGRP-enabled interface? (2)	Static configuration of at least one neighbor on that interface. Configuring the interface as passive.
What configuration settings prevent any neighborships on an EIGRP-enabled interface?	Configuring the interface as passive.
What settings do potential neighbors check before becoming EIGRP neighbors? (5)	Whether the neighbor's IP address is in the same primary subnet as the local router. EIGRP authentication failure. ASN in **router eigrp** *asn* commands must match. The interfaces cannot be passive. The configured K-values must match.
What settings that you might think would impact EIGRP neighbor relationships actually do not prevent neighborship? (3)	Mismatched Hello and Hold Timer settings. Duplicate Router IDs. IP MTU mismatch.
What issues typically arise when the design calls for the use of EIGRP authentication key chains with lifetime settings? (2)	The time of various neighbors should be synchronized, typically using NTP. You must ensure that the lowest time-valid sending key on one router is the same key-string as any of the time-valid receive keys on the other router.

Table 2-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling EIGRP on interfaces	**router eigrp** *autonomous-system* **network** *network-number* [*wildcard-mask*]
Setting Hello and Hold Timers	**ip hello-interval eigrp** *as-number timer-value* **ip hold-time eigrp** *as-number timer-value*
EIGRP authentication	**ip authentication key-chain eigrp** *asn chain-name* **ip authentication mode eigrp** *asn* md5 **key chain** *name* **key** *integer-number* **key-string** *text* **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
Passive interfaces	**passive-interface** *type number* **passive-interface default** **no passive-interface** *type number*
Static EIGRP neighbors	**neighbor** *a.b.c.d interface*
K-values	**metric weights 0** *k1 k2 k3 k4 k5*
EIGRP router ID	**eigrp router-id** *a.b.c.d*

Table 2-8 *Verification Plan Memory Drill*

Information Needed	Command
Which routes have been added to the IP routing table by EIGRP	**show ip route eigrp**
All routes in a router's routing table.	**show ip route**
The specific route for a single destination address or subnet.	**show ip route** *ip-address* [*mask*]
A list of all (both static and dynamically discovered) EIGRP neighbors.	**show ip eigrp neighbors** **show ip eigrp neighbors detail**
Notation of whether a neighbor was dynamically discovered or statically configured.	**show ip eigrp neighbors detail**
Lists statistics regarding the numbers of EIGRP messages sent and received by a router.	**show ip eigrp traffic**
List interfaces on which EIGRP has been enabled (by virtue of the EIGRP **network** command).	**show ip eigrp interfaces** **show ip eigrp interfaces detail**
List the number of EIGRP peers known via a particular interface.	**show ip eigrp interfaces** **show ip eigrp interfaces detail** **show ip eigrp interfaces** *type number*
The elapsed time since a neighborship was formed.	**show ip protocols** **show ip eigrp neighbors** [**detail**]
The parameters of any EIGRP **network** commands.	**show ip protocols**
The configured Hello timer for an interface.	**show ip eigrp interfaces detail** [*type number]*
The configured Hold Timer for an interface.	none
The current actual Hold Timer for a neighbor.	**show ip eigrp neighbor** [**detail**]
A router's EIGRP ASN.	**show ip protocols** **show ip eigrp traffic** **show ip eigrp accounting**

Table 2-8 *Verification Plan Memory Drill*

Information Needed	Command
A list of EIGRP passive interfaces.	**show ip protocols**
A list of nonpassive EIGRP interfaces.	**show ip eigrp interfaces** [**detail**]
The currently used EIGRP authentication key, when sending EIGRP packets.	**show key chain**
The currently used EIGRP authentication key, when receiving EIGRP packets.	**show key chain**
Lists EIGRP K-values.	**show ip protocols**
Lists traffic statistics about EIGRP.	**show ip eigrp traffic**
A router's EIGRP Router ID.	**show ip eigrp topology** **show ip eigrp accounting**

Chapter 3

Table 3-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Limit consumption of IP subnets in Frame Relay WAN design.	Use multipoint subinterfaces, with more than two routers sharing the same WAN subnet.
In a relatively slow Frame Relay WAN, protect against consuming too much bandwidth with overhead traffic.	Use the EIGRP WAN bandwidth control feature to limit the amount of bandwidth consumed by EIGRP.
Plan to change bandwidth from 1X CIR to 2X CIR on all Frame Relay subinterfaces.	Adjust metrics with delay as well, to ensure the correct best routes are chosen plus that backup routes are feasible successors where possible.
Plan to set bandwidth to values other than actual interface speeds to manipulate EIGRP metrics.	Ask whether the design could use delay instead.

Table 3-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A goal of ensuring all remote routers' secondary EIGRP route does not require Queries.	Tune metrics using delay or offset lists such that secondary routes are FS routes.
What tools can we use to meet the design goal of fast convergence? (four items)	Tune metrics so that feasible successor routes exist. Make appropriate routers EIGRP stubs. Use unequal cost multipath to add multiple routes to routing table. Use route summarization to limit query scope.

Table 3-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A Frame Relay multipoint interface, with 20 PVCs attached, has a configuration for 10% of bandwidth to be used for EIGRP. How much is allocated per PVC?	IOS first divides the subinterface bandwidth by 20 (number of PVCs) and then takes 10% (per the configuration).
Could any planned topologies have EIGRP Split Horizon issues?	LANs and point-to-point topologies should be fine. Multiaccess Layer 2 WANs, particularly when a full mesh is not used, may cause problems. The typical case exists with Frame Relay, multipoint or physical serial interfaces, with a partial mesh of PVCs.
A configuration lists the **no ip split-horizon** command–when would that matter?	This command influences RIP's use of Split Horizon, not EIGRP's, so consider the routing protocol in use.
The plan calls for setting all EIGRP K-values to 1–what negative effect could this have on routes in the IP routing table?	Route flapping.
The configuration uses offset lists. Will that impact the calculation of FD and/or RD?	Both.

Table 3-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
The plan lists a sample configuration migrating an interface from **delay 20** to **delay 200**. How much will the metric go up?	The **delay** interface subcommand and the metric formula both use a unit of tens-of microseconds. In this case, the delay is 180 more, and then multiplied by 256, for a total of 4608. (Author's note: Don't worry if your answer wasn't as detailed in this case.)
The plan shows the use of the **variance 4** command. What must be configured to add other routes to a routing table? (two items)	Check that the number of maximum-paths is high enough for all the routes you want to include. Configure metrics such that the alternative routes are feasible successors.

Table 3-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling EIGRP on interfaces	**router eigrp** *autonomous-system* **network** *network-number* [*wildcard-mask*]
Enabling or disabling Split Horizon for EIGRP	**[no] ip split-horizon eigrp** *asn*
Setting the Bandwidth consumed by EIGRP on an interface	**ip bandwidth-percent eigrp** *asn percent*
Setting an interface's logical bandwidth	**bandwidth** *value*
Setting an interface's logical delay	**delay** *value*
K-values	**metric weights** *tos k1 k2 k3 k4 k5*
Configuring an EIGRP offset list that matches a prefix	1) Create an IP ACL to match routes (various; considered prerequisite). 2) In EIGRP configuration mode, configure: **offset-list** {*access-list-number* | *access-list-name*} {**in** | **out**} *offset [interface-type interface-number]*

Table 3-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configuring an EIGRP offset list that matches a prefix and prefix length	The same as the previous row of the table, except that you create an extended IP ACL that matches the prefix with the ACL source IP address parameter, and the mask with the destination IP address field.
Configuring unequal cost load balancing	**maximum-paths** *value* **variance** *value* Tune metrics to ensure feasible successor routes
Configure an EIGRP stub router	**eigrp stub [[connected] [summary] [static] [redistributed]]	[receive-only]]**

Table 3-8 *Verification Plan Memory Drill*

Information Needed	Command
The composite metric values for all EIGRP prefixes.	**show ip eigrp topology** *prefix/length*
Display EIGRP Split Horizon settings.	**show running-config**
Calculate the maximum bandwidth EIGRP will consume on a physical or point-to-point subinterface.	**show interfaces**–to find the interface bandwidth **show running-config**–to find the EIGRP bandwidth percentage
Calculate the maximum bandwidth EIGRP will consume per PVC on a multipoint Frame Relay subinterface.	**show interfaces**–to find the interface bandwidth **show running-config**–to find the EIGRP bandwidth percentage **show frame-relay pvc interface** *number type*–to find the number of active PVCs associated with the interface Calculate (interface bandwidth/# pvcs) * percentage
Display the increase in FD after implementing an EIGRP offset list.	**show ip route** **show ip eigrp topology** **show ip eigrp topology** *prefix/length*

Table 3-8 *Verification Plan Memory Drill*

Information Needed	Command
Display the increase in RD after implementing an EIGRP offset list.	**show ip eigrp topology** **show ip eigrp topology** *prefix/length*
Display interface bandwidth and delay settings.	**show interfaces** *[type number]* **show ip eigrp topology** **show ip eigrp topology** *prefix/length*
Lists EIGRP K-values.	**show ip protocols**
Find the number of successor and feasible successor routes.	**show ip eigrp topology** **show ip eigrp topology** *prefix/length*
Find all routes, including non-successors.	**show ip eigrp topology all-links** **show ip eigrp topology all-links** *prefix/length*
Determine if the local router is a stub router.	**show running-config** **show ip protocols**
Determine if a neighboring router is a stub.	**show ip eigrp neighbors detail**
Find the current setting of variance and maximum-paths.	**show ip protocols**
Display messages each time EIGRP suppresses a prefix advertisement due to Split Horizon.	**debug eigrp packet**

Chapter 4

Table 4-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Prevent the edge routers in sites for one division of the company from knowing routes for subnets in another division.	Use EIGRP route filtering (distribution lists).
The design shows the use of class B networks 172.16.0.0, 172.17.0.0, and 172.18.0.0 throughout the Enterprise, with a goal to reduce routing table sizes when possible. (3 options)	Use manual route summarization. Use auto-summarization if appropriate. Use route filtering.
R1 and R2 will advertise the same summary route; ensure that R1 is the preferred EIGRP path for that summary.	Tune EIGRP metrics so that all R2's metrics for the subordinate routes are higher than the metric of R1's best subordinate route.
Always ensure that the shortest path is taken with each route.	Avoid the use of summary routes.

Table 4-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
The plan calls for filtering 10.10.10.0/26 and 10.10.12.0/26, but not 10.10.11.0/24. What tools can be used?	EIGRP distribution lists, with prefix-list or route-map that refers to a prefix-list for matching. An ACL can also be used, matching each prefix explicitly.
A plan lists a configuration of an EIGRP distribution list, referring to route-map *one*. The route-maps clauses use only the deny option. However, it refers to prefix lists that use some deny and permit actions. Can any routes pass through the filter?	No. When a distribution-list refers to a route-map, the route-map determines the route filtering action. With only deny clauses, all routes will be filtered.

Table 4-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A plan lists a configuration of an EIGRP distribution list, with a route-map with two clauses, each with a permit action. Both clauses refer to a different prefix list, each of which has some permit and deny actions. Can any routes pass through the filter? Will all routes pass through?	Some routes may pass through, and some may be filtered. When a distribution-list refers to a route-map, the route-map determines the route filtering action. With the permit clauses, any routes matched with a permit action by the respective prefix lists will be passed. However, route-maps also have a final implicit deny clause, which filters all routes not matched by the other route map clauses.
The plan shows extensive use of class C private networks inside a large Enterprise. What effect might auto-summary have?	Auto-summary will cause EIGRP to advertise a summary for a class C network when advertising out an interface in a different class C network, resulting in many summary routes.
The plan shows a sample configuration of the **ip summary-address eigrp 1 10.10.0.0 255.255.252.0** command on Router R1. What routes should I see on R1? What will their administrative distance be?	R1 will list 10.10.0.0/18 as a summary route, AD 5, with outgoing interface null0, if at least one subordinate route exists. R1 will also have routes for all the subordinate subnets in the range. Other routers will just see a summary route, with the same EIGRP AD (90) as for other internal routes.

Table 4-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Command/Notes
Filtering EIGRP routes using numbered ACLs	**access-list {1-99} {permit | deny}** *subnet-number wildcard-mask* **router eigrp** *asn* **distribute-list** *acl-number {in | out} [interface-type number]*
Filtering EIGRP routes using prefix lists	**ip prefix-list** [**seq** *sequence-no*] *list-name* [**seq** seq-value] {**deny | permit** *prefix/prefix-length*}[**ge** *ge-value*] [**le** *le-value*] **router eigrp** *asn* **distribute-list prefix** *prefix-list-name* {**in | out**} [*interface-type number*]
Enabling filtering EIGRP routes using route-maps	(Create route map) **router eigrp** *asn* **distribute-list route-map** *route-map-name* {**in | out**} [*interface-type number*]
Commands to create a route-map clause and match based on standard numbered, standard named, and prefix-list	**route-map** *list-name* **[deny | permit]** [*sequence-number*] **match ip address** *1-99* **match ip address** *std-ACL-name* **match ip address prefix-list** *listname*
Configuring a summary route	in interface mode: **ip summary-address eigrp** *asn prefix subnet-mask* [*admin-distance*]
Enable/disable auto-summary	(EIGRP configuration mode) [**no**] **auto-summary**
Configure a default route using **ip default-network**	**ip default-network** *net-id*

Table 4-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Command/Notes
Configure a default route using static routes	**ip route 0.0.0.0 0.0.0.0** *outgoing-interface*
network 0 0.0.0., or	configure redistribution of static routes

Table 4-8 *Verification Plan Memory Drill*

Information Needed	Command
Display access lists.	**show access-lists**
Display prefix lists.	**show ip prefix-list**
Display route maps.	**show route-maps**
Verify whether a filter worked by displaying all known routes in a range of addresses.	**show ip route** *prefix/length* **longer-prefixes**
Display a summary IP route.	**show ip route**
On summarizing router, display EIGRP topology info on a summary route.	**show ip eigrp topology** *prefix/length*
On summarizing router, display IP routes for a summary route and it subordinate routes.	**show ip route** *prefix* mask **longer-prefixes**
On summarizing router, display the administrative distance of the null route.	**show ip route** *prefix mask*
Display the current auto-summary setting.	**show ip protocols**
Determine if a prefix in the EIGRP topology table has been flagged as a candidate default route.	**show ip eigrp topology** *prefix/length* Look for the exterior flag setting.
Determine if an IP route has been flagged as a candidate default route.	**show ip route** Look for the asterisk beside the route.
Display a router's preferred default route.	**show ip route** Look for the gateway of last resort setting.

Chapter 5

Table 5-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Improve OSPF convergence.	Tune OSPF Hello and Dead intervals so that neighbor failures are recognized more quickly.
Implement OSPF on each router so that neighborships are formed (2).	Discover neighbors using multicasts as a result of matching an interface with the OSPF **network** command, in **router ospf** configuration mode. Instead of the OSPF **network** command, use the **ip ospf** *process-id* **area** *area-id* interface subcommand to enable OSPF on an interface.
Limit neighborship formation on OSPF-enabled interfaces (2).	Use OSPF authentication to allow only neighbors with the correct keys. Prevent all neighborships on an interface by making the interface passive.
The design shows branch routers with WAN interfaces in area 0 and LAN interfaces in different areas for each branch. What LSDB information do you expect to see in the branch routers?	The branch routers, traditionally less expensive, less powerful, with less memory, will need to hold the area 0 LSDB, which may become large given the design.

Table 5-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
What happens on a router interface on which an OSPF **network** command matches the interface? (2)	OSPF attempts to discover OSPF neighbors by sending and receiving multicast OSPF Hellos. OSPF advertises about the subnet on the connected interface.
What configuration settings prevent OSPF neighbor discovery on an OSPF-enabled interface?	Configuring the interface as passive.

Table 5-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
What settings do potential neighbors check before becoming OSPF neighbors? (7)	Whether the neighbor's IP address is in the same primary subnet as the local router. OSPF authentication failure. The interfaces cannot be passive. Must be in the same area. Hello and Dead intervals must match. Unique RIDs. IP MTUs must match.
What settings that you might think would impact OSPF neighbor relationships actually do not prevent neighborship?	Mismatched process-id's on the **router ospf** commands.
A design shows one main site and 100 branches, with OSPF, and MPLS VPNs. How many OSPF neighborships over the WAN do you expect to see on the central site router?	One. Each router becomes neighbors with the provider edge (PE) router inside the MPLS VPN service.
A design shows one main site and 100 branches, with one Frame Relay PVC between the main site and each branch. How many OSPF neighborships over the WAN do you expect to see on the central site router?	100. The central site router forms a neighborship with each branch router.
A design shows six routers connected to the same VLAN and subnet. How many OSPF fully adjacent neighborships over this subnet do you expect each router to have?	The DR and BDR will be fully adjacent with each other and with all four of the other routers. The other four routers will be fully adjacent with only two routers: the DR and BDR.
A design shows one main site and 100 branches, each connected with a VPWS service. The configuration shows that the central site router uses a separate VLAN subinterface to connect to each branch, but the branch routers do not have a VLAN connecting to other branches. How many OSPF fully adjacent neighborships over the WAN do you expect to see on the central site router?	100. The central site's subinterface for each VLAN acts like a separate logical interface. A DR and BDR will be used, but in this design, 100 such instances exist, and the central site will become fully adjacent with all 100 branches.

Table 5-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Enabling OSPF on interfaces–traditional method	**router ospf** *process-id* **network** *network-number wildcard-mask* **area** *area-id*
Enabling OSPF on interfaces–using interface subcommands	**router ospf** *process-id* **interface** *type number* **ip ospf** *process-id* **area** *area-id*
Setting Hello and Dead intervals	**ip ospf hello-interval** *timer-value* **ip ospf dead-interval** *timer-value*
MD5 authentication, with no **router** subcommands	**interface** *type number* **ip ospf authentication message-digest** **ip ospf message-digest-key** *key-number* **md5** *key-value*
MD5 authentication, with **router** subcommands	**router ospf** *process-id* **area** *area-number* **authentication message-digest** **interface** *type number* **ip ospf message-digest-key** *key-number* **md5** *key-value*
Passive interfaces, with router subcommands	**passive-interface** *type number* **passive-interface default** **no passive-interface** *type number*
OSPF router ID	**router-id** *a.b.c.d*

Table 5-8 *Verification Plan Memory Drill*

Information Needed	Command
Which routes have been added to the IP routing table by OSPF?	**show ip route ospf**
All routes in a router's routing table.	**show ip route**
The specific route for a single destination address or subnet.	**show ip route** *ip-address* [*mask*]
A list of all (both static and dynamically discovered) OSPF neighbors.	**show ip ospf neighbor** **show ip ospf neighbor detail**
List interfaces on which OSPF has been enabled.	**show ip ospf interface** **show ip ospf interface brief** **show ip protocols** (if enabled with the **ip ospf area** interface subcommand)
List the number of OSPF neighbors and fully adjacent neighbors known via a particular interface.	**show ip ospf interface** **show ip ospf interface brief** **show ip ospf interface** *type number*
The elapsed time since a neighborship was formed.	**show ip protocols** **show ip ospf neighbor [detail]**
The configured Hello timer for an interface.	**show ip ospf interface [***type number* **]**
The configured Dead Interval timer for an interface.	**show ip ospf interface [***type number* **]**
The current actual dead timer for a neighbor.	**show ip ospf neighbor [detail]**
A router's RID.	**show ip ospf** **show ip ospf database** **show ip ospf statistics**
A list of OSPF passive interfaces.	**show ip protocols**
The currently used OSPF authentication type.	**show ip ospf interface [***type number* **]**
The smallest-numbered OSPF authentication key number.	**show ip ospf interface [***type number* **]**
Lists traffic statistics about OSPF.	**show ip ospf statistics**

Chapter 6

Table 6-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design sets specific limits to the number of Type 1 and 2 LSAs in each area. Describe how to predict the number of each type of LSA.	Add 1 Type 1 per internal router in that area. Add 1 Type 1 per ABR. Add 1 Type 2 per subnet in which a DR should be elected, and for which two such routers exist in that subnet.
How could you tune OSPF metrics to favor 10 Gbps links over 1 Gbps and 1 Gig over 100 Mbps (2)?	Configure all routers with a **auto-cost-reference-bandwidth** command, in **router ospf** configuration mode, of at least 10,000. Manually configure OSPF interface costs with the **ip ospf cost** *cost* interface subcommand.
The design shows one physical path from ABR1 to core subnet 1 inside area 0, and one longer area 1 path to the same subnet. What can be done to ensure both paths can be used?	Nothing–ABR1, like all ABRs, ignore Type 3 LSAs (like the LSA for subnet 1) learned in a nonbackbone area (like area 1).

Table 6-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
What conditions must be true for a router to create/flood a Type 2 LSA? (2)	At least two routers must be neighbors and have elected a DR. The router creating the LSA must be the DR.
The plan shows Frame Relay with all point-to-point subinterfaces. By default, will a DR/BDR be elected?	This is the case where, on Frame Relay, the default point-to-point OSPF network type works fine. No DR/BDR will be elected.

The plan shows a reference bandwidth change planned for all routers with high speed links, but not all other routers. What is the impact? (2)	This plan breaks the recommendation to use the same value throughout the network. Potential impacts 1. The setting on one router changes only that router's OSPF costs. 2. It may result in poor route choices.
The plan shows many different WAN links speeds, but with the interface bandwidths not matching the actual speed. All OSPF cost changes are made explicitly with the **ip ospf cost** interface subcommand. Do the incorrect bandwidths cause any OSPF problems?	No problem for OSPF, which uses the interface cost per **ip ospf cost** if it is configured.

Table 6-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Tune metrics by changing the formula for calculating OSPF cost based on interface bandwidth.	**router ospf** *process-id* **auto-cost reference-bandwidth** ref-bw
Tune metrics by changing interface bandwidth.	**interface** *type number* **bandwidth** *bandwidth*
Change metrics by setting cost directly.	**interface** *type number* **ip ospf cost** *cost*
Set the number of equal-cost OSPF routes allowed in a router's routing table.	**router ospf** *process-id* **maximum-paths** *number*
Influence the choice of DR on a LAN. (2)	Configure **ip ospf priority** *value* on the interface. Set the OSPF router ID, using either the **router-id** value router subcommand, creating a loopback interface with a high IP address, or another interface with a high IP address.

Table 6-8 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display a summary of the OSPF database.	**show ip ospf database**
Display all Type 1 router LSAs known to a router.	**show ip ospf database router**
Display the details of a particular Type 1 router LSA.	**show ip ospf database router** *lsid*
Display all Type 2 network LSAs known to a router.	**show ip ospf database network**
Display the details of a particular Type 2 router LSA.	**show ip ospf database network** *lsid*
Display all Type 3 summary LSAs known to a router.	**show ip ospf database summary**
Display the details of a particular Type 3 router LSA.	**show ip ospf database summary** *lsid*
Display a list of OSPF-enabled interfaces on a router.	**show ip ospf interface** **show ip ospf interface brief** **show ip ospf interface** *type number* **show ip protocols**
Determine on which interfaces a router has formed at least one OSPF neighborship.	**show ip ospf interface** **show ip ospf interface brief** **show ip ospf interface** *type number*
Determine the number of fully adjacent neighbors on an interface.	**show ip ospf interface** **show ip ospf interface brief** **show ip ospf interface** *type number* **show ip ospf neighbor** **show ip ospf neighbor detail**
Determine which transit networks connect to a Type 1 LSA.	**show ip ospf database router [***lsid]*
Determine the router that created and flooded a Type 3 LSA.	**show ip ospf database** **show ip ospf database summary**

Table 6-8 *Verification Plan Memory Drill*

Information Needed	Command(s)
Determine the router that created and flooded a Type 2 LSA.	**show ip ospf database** **show ip ospf database network**
Determine the router that created and flooded a Type 1 LSA.	**show ip ospf database** **show ip ospf database router**
Display the IP address of the current DR and BDR on a LAN.	**show ip ospf neighbor [detail]** **show ip ospf interface** *type number*
Display the OSPF interface cost (metric).	**show ip ospf database router** **show ip ospf interface [brief]**
Display all OSPF-learned routes.	**show ip route ospf**
Display statistics about the number of SPF algorithm runs.	**show ip ospf [statistics]**

Chapter 7

Table 7-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
When using OSPF, prevent the routers in sites for one division of the company from knowing IP routes for subnets in another division. (3)	Type 3 LSA filtering on ABRs. Type 5 LSA Filtering on ASBRs. Filtering Routes added by OSPF to the IP routing table.
The design shows an Enterprise that uses only OSPF. It lists a goal of keeping the LSDBs and routing tables in each area small. (3)	Use manual route summarization on ABRs. Use Type 3 LSA filtering. Use stub areas.
The design lists a goal of extremely small LSDBs and IP routing tables on branch office routers–which stub area types work best? (2)	Totally stubby areas. Totally NSSA areas.
The design calls for the flooding of a domain-wide default route to draw traffic toward the Internet-connected routers.	Use the **default-information originate** command on the Internet routers.

Table 7-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a design with area 0, with different ABRs connecting area 0 to areas 1, 2, and 3. The configurations show Type 3 LSA filtering into the nonbackbone areas but not in the opposite direction. Could this configuration filter subnets in area 1 from being seen in area 2?	Type 3 LSA filtering only filters subnets whose Type 3 LSAs would be created by that ABR. So, if the area 1–area 0 ABR created an LSA for an area 1 subnet, flooding that LSA into area 0, then the Area 0–area 2 ABR would not attempt to filter that subnet with Type 3 LSA filtering.
The design shows the configuration of Type 3 LSA filtering on an internal router in area 1. Could the filter have any effect?	No. The filtering only has effect on ABRs, for Type 3 LSAs created on that ABR.
The plan shows the configuration of the **area range** command on an ABR. What is the metric for the summary route? And in what conditions will the ABR advertise the summary?	The metric, if not listed in with the cost parameter on the **area range** command, is the lowest-cost among all subordinate routes. The ABR advertises only the summary if at least one subordinate subnet exists as an intra-area route.
The plan shows the configuration of the **area 1 stub** command for an area mostly located on the west coast of the USA. The company just bought another company whose sites are also on the west coast. What issues exist if you add links from the acquired company into area 1?	As a stubby area, the area will not allow the redistribution of external routes. The acquired company's routes may at least initially need to be redistributed into OSPF.
The plan shows the configuration of the **default-information originate always** command on the one router to which the Internet links connect. What happens to default route when Internet link fails? What happens to packets destined for the Internet during this time?	This command makes the router always advertise the default, even if the links fail and that router's default route pointing toward the Internet fails. As such, all packets destined outside the Enterprise will still pass through the Enterprise to this router and then be discarded.

Table 7-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Filter Type 3 LSAs from being sent into an area.	(Create an IP prefix list) **router ospf** *process-id* **area** *area-number* **filter-list prefix** *list-name* **{in | out}**
Filter the OSPF routes calculated on one router from being added to that one router's routing table.	(Create an IP prefix list) **router ospf** *process-id* **distribute-list prefix** *list-name* **in**
Configure route summarization on ABRs.	**router ospf** *process-id* **area** *area-id* **range** *ip-address mask* [**cost** *cost*]
Configure route summarization on ASBRs.	**router ospf** *process-id* **summary-address** {*ip-address mask* | *prefix mask*}
Configure the OSPF domain-wide advertisement of a default route.	**router ospf** *process-id* **default-information originate [always] [cost** *metric*] **[metric-type** *type*]
Configure stubby or totally stubby areas.	**router ospf** *process-id* **area** *area-number* **stub** (stubby areas and totally stubby areas on non-ABRs) **area** *area-number* **stub no-summary** (totally stubby areas, on ABRs only) **area** *area-num* **default-cost** *cost* (optional)
Configure NSSA or totally NSSA areas.	**router ospf** *process-id* **area** *area-number* **nssa** (NSSA areas and totally NSSA areas on non-ABRs) **area** *area-number* **nssa no-summary** (totally NSSA areas, on ABRs only) **area** *area-num* **default-cost** *cost* (optional)

Table 7-8 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display all IP routes for subnets in a range, regardless of prefix length.	**show ip route** *subnet mask* **longer-prefixes**
Display the contents of an IP prefix list.	**show ip prefix-list** *[name]*
Display details of all Type 3 LSAs known to a router.	**show ip ospf database summary**
Display details of all Type 5 external LSAs known to a router.	**show ip ospf database external**
Display the metric advertised in a summary route created by the **area range** command.	**show ip ospf database summary** *[lsid]*
Display the metric advertised in a summary route created by the **summary-address** command.	**show ip ospf database external** *[lsid]*
Discover whether a router resides in a stubby area, and if so, which kind.	**show ip ospf**
Confirm stubby area concepts by looking at the numbers of Type 3 and Type 5 LSAs known to a router.	**show ip ospf database database-summary**

Chapter 8

Table 8-4 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A merger design plan shows two companies with OSPF backbone areas. How can the two areas 0 be connected? (2)	Physical links between routers in the two areas 0. A virtual link.
The subnetting design over Frame Relay calls for a set of 20 routers to use a single WAN subnet. What Frame Relay interface types make sense?	The configuration can use either the physical serial interface or a multipoint subinterface.
The design shows 20 routers connected to Frame Relay, in a partial mesh, with one subnet. What OSPF network type would avoid the use of DR/BDR?	Point-to-multipoint and point-to-multipoint nonbroadcast.
The design shows 20 routers connected to Frame Relay, in a full mesh, with one subnet. What OSPF network type would allow dynamic neighbor discovery?	Point-to-multipoint and broadcast.

Table 8-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The subnetting design over Frame Relay calls for a set of 20 routers to use a single WAN subnet. What three general issues exist that requires extra configuration?	If the network type does not allow neighbor discovery, the configuration will need **neighbor** commands. If the network type causes the election of a DR, use priority to influence who can become DR/BDR. If a partial mesh exists, creating static mapping for all routers in the subnet.

Table 8-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a virtual link between R1 and R2, through area 99, with the connection using R1's Fa0/0 interface. The configuration shows MD5 authentication configuration as interface subcommands on Fa0/0. Will those commands apply to the virtual link?	No. Virtual link authentication must be configured on the **area virtual-link** command.
The plan shows a sample OSPF configuration with five routers, Frame Relay connected, with one subnet, each using multipoint subinterfaces. A full mesh exists. The configuration shows network type point-to-multipoint nonbroadcast. What OSPF configuration is needed beyond the enabling of OSPF on the interfaces?	The network type requires the definition of neighbors. It does not elect a DR, so no priority values need be configured. With a full mesh, no additional static mappings need to be configured.
Using the same scenario as the previous row but with network type nonbroadcast, and a partial mesh, answer the same questions.	The network type requires the definition of neighbors. It also causes an election of a DR, so the priority values should be configured. Additional static mapping should be configured for the router pairs that do not have a PVC between the two routers.

Table 8-6 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Create a virtual link through transit area X.	**router ospf** *process-id* **area** *X* **virtual-link** *neighbor-RID*
Configure MD5 authentication on a virtual link.	**area** *num* **virtual-link** *router-id* **authentication message-digest** **area** *num* **virtual-link** *router-id* **message-digest-key** *key-num* **md5** *key-value*
Configure clear-text authentication on a virtual link.	**area** *num* **virtual-link** *router-id* **authentication** **area** *num* **virtual-link** *router-id* **authentication-key** *key-value*

Table 8-6 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure null authentication on a virtual link.	**area** *num* **virtual-link** *router-id* **authentication null**
Configure the OSPF network type on a Frame Relay multipoint interface.	**interface** *type number* **ip ospf network {non-broadcast | broadcast | point-to-point | point-to-point non-broadcast}**
Statically configure OSPF neighbors.	**router ospf** *process-id* **neighbor** *interface-IP* **[cost** *cost***] [priority** *priority***]**
Influence OSPF priority of the local router.	**interface** *type number* **ip ospf priority** *priority*
Influence OSPF priority of a neighboring router.	**router ospf** *process-id* **neighbor** *interface-IP* **[priority** *priority***]**
Add static Frame Relay mappings.	**frame-relay map ip** *neighbor-IP DLCI* **[broadcast]**

Table 8-7 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display the name and status of a virtual link.	**show ip ospf virtual-links** **show ip ospf neighbor [detail]**
Display the RID of a neighbor on a virtual link.	**show ip ospf neighbor** **show ip ospf virtual-links**
Display the OSPF authentication type and youngest key for MD5 authentication on a Virtual Link.	**show ip ospf virtual-links**
Display the OSPF network type for an interface.	**show ip ospf interface [***type number***]**
Display a router's own OSPF priority.	**show ip ospf interface [***type number***]**
Display the mapping of neighbor IP address and Frame Relay DLCI.	**show frame-relay map**

Table 8-7 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display neighbors known on an interface.	**show ip ospf neighbor** **show ip ospf interface** [*type number*]

Chapter 9

Table 9-6 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A design shows Router R1 as being connected to both an EIGRP and OSPF routing domain, with all external EIGRP routes using a particular set of component EIGRP metrics. How can these metrics be set? (3)	Set the metrics on the **redistribute** command. Use the **default-metric** command. Set the metric inside a route-map referenced by the **redistribute** command.
A design shows Router R1 as being connected to two different EIGRP domains, with redistribution planned. Can the design cause the routers to calculate metrics based on both the metric assigned when redistributing and the internal EIGRP topology?	No special action is required; this behavior occurs for all routes redistributed into EIGRP.
The same design as in the previous row is shown, except describe whether or not the design can cause the routers to calculate metrics based solely on the metric components assigned when redistributing.	The behavior is not supported by EIGRP.
A design shows Router R1 as being connected to two different OSPF domains, with redistribution planned, and all routes calculated by including internal and external OSPF distance.	Routes must be distributed as E1 routes, using **redistribute... metric-type 1**.
The same design as in the previous row is shown, except that all external route metrics are based solely on external metrics.	Routes must be distributed as E2 routes, using **redistribute... metric-type 2**, or by omitting the **metric-type** keyword on the **redistribute** command.

Table 9-7 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A design shows Router R1 as being connected to both an EIGRP and OSPF routing domain. What default metrics will be used by the **redistribute** command for each routing protocol, if not set in R1's configuration?	EIGRP–no default metrics OSPF–20
A plan shows redistribution between two EIGRP domains. What must be done to use the source route's original component metrics?	Nothing when redistributing from EIGRP into EIGRP. The default action, if the metric is not set by any other means, uses the source route's metric components.
A plan shows redistribution between two OSPF domains. What must be done to use the source route's original component metrics?	Nothing when redistributing from OSPF into OSPF. The default action, if the metric is not set by any other means, uses the source route's metric.
The plan shows the **redistribute eigrp 2** command to redistribute from EIGRP 2 into OSPF. What other optional parameters are required to ensure redistribution of 10.1.1.0/24 from EIGRP?	10.1.1.0/24 is a subnet of a classful network, and redistribution into OSPF takes only classful networks if the (optional) **subnets** keyword is omitted on the **redistribute** command.
R1 has two connected interfaces in the EIGRP 2 domain and knows dozens of EIGRP routes. The plan shows the **redistribute eigrp 2 subnets** under an OSPF process. What else must be done to redistribute the two connected subnets inside the EIGRP domain?	Nothing–the **redistribute** command takes routes learned by the source routing protocol, plus connected routes for interfaces enabled by that protocol.

Table 9-8 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configuring redistribution into EIGRP from OSPF. (List all parameters you can recall.)	**router eigrp** *asn* **redistribute** *protocol* [*process-id* | *as-number*] [**metric** *bw delay reliability load mtu*] [**match** {**internal** | **nssa-external** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *name*]
Configuring redistribution into EIGRP from OSPF. (List all parameters you can recall.)	**router ospf** *process-id* **redistribute** *protocol* [*process-id* | *as-number*] [**metric** *metric-value*] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2** | **nssa-external**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
Setting default metrics for all **redistribute** commands, redistributing into EIGRP.	**router eigrp** *asn* **default-metric** *bw delay reliability loss mtu*
Setting default metrics for all redistribute commands, redistributing into OSPF.	**router ospf** *process-id* **default-metric** *cost*

Table 9-9 *Verification Plan Memory Drill*

Information Needed	Command(s)
Display a brief version of the EIGRP topology table, listing external routes.	**show ip eigrp topology**
Display the EIGRP topology table, including notations identifying external routes.	**show ip eigrp topology** *prefix/length*
For external EIGRP routes, display the source of the route, external metric, and IP address of the router that redistributed the route.	**show ip eigrp topology** *prefix/length*
Identify external EIGRP-learned IP routes.	**show ip route** **show ip route eigrp**
Display a brief version of the OSPF topology table, listing Type 5 External LSAs.	**show ip ospf topology**
Display all OSPF Type 4 LSAs.	**show ip ospf topology asbr-summary**
Display all OSPF Type 5 LSAs.	**show ip ospf topology external**
Display all OSPF Type 7 LSAs.	**show ip ospf topology nssa-external**
Display the external route type for an OSPF external route.	**show ip ospf database external** **show ip route ospf**
Display OSPF cost for each interface, briefly.	**show ip ospf interface brief**
On an internal router, display any same-area ABRs' costs to reach any ASBRs.	**show ip ospf database asbr-summary**
On an internal router, display that router's best cost to reach an ASBR.	**show ip ospf border-routers**
Display the metric for all currently best external OSPF routes.	**show ip route** **show ip route ospf**

Chapter 10

Table 10-8 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Filter routes when redistributing. (2)	Refer to a route-map on the **redistribute** command. Use a **distribute-list** command that refers to the routing source.
Set different metrics for different routes redistributed from one routing source.	Use the **redistribute... route-map** option.
Set some OSPF routes as E1, and some as E2, when redistributed from one routing source.	Use the **redistribute... route-map** option, with the **set metric-type** command.
The design shows multiple redistribution points with two routing domains, with a need to prevent domain loops. (3)	Set high metrics when redistributing. Set administrative distance (AD) on redistributing routers so that internal routes are better than other routing protocol's external routes. Set and filter on route tags.
The design shows multiple redistribution points with more than two routing domains, and a need to prevent domain loops. (2)	Set per-route administrative distance (AD) on redistributing routers. Set and filter on route tags.

Table 10-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
A design shows an OSPF and EIGRP routing domain, with multiple redistributing routers, with no obvious configuration to prevent routing domain loops. What default AD values exist, and do they prevent any problems?	EIGRP: internal 90, external 170. OSPF: internal 110, external 110. Domain loops are prevented because EIGRP's internal 90 is less than OSPF's external 110, and OSPF's internal 110 is less than EIGRP's external 170.
The same question as the previous row, except with RIP and OSPF domains.	RIP: 120 (all). OSPF: internal 110, external 110. Domain loops are not prevented because RIP's 120 AD is not less than OSPF's external 110.

Table 10-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answer
The same question as the previous row, except with RIP and EIGRP domains.	RIP: 120 (all). EIGRP: internal 90, external 170. Domain loops are prevented because EIGRP's internal 90 is less than RIP's 120, and RIP's 120 is less than EIGRP's external 170.
A plan shows redistribution between EIGRP and OSPF on two routers. The configuration for OSPF on one router lists **redistribute eigrp 1 subnets** and **distribute-list 1 out**. Will this configuration attempt to filter routes? Is a route-map option required to filter when redistributing?	The configuration is incomplete for filtering. If the **distribute-list 1 out eigrp 1** command was used (for instance), referring to the routing source, the redistributed routes would be filtered. This is an alternative to filtering by using the **route-map** option on the **redistribute** command.
A partially complete plan shows three different routing domains, with multiple redistribution points between each pair of routing domains. The configuration shows large ACLs matching various subnets and setting AD per-route using the **distance** command. What alternative method might be easier to maintain as the network changes?	Using route tags does not require matching on subnets, which could reduce the amount of configuration changes required over time.

Table 10-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Filtering routes on redistribution from OSPF into EIGRP, allowing only routes permitted by ACL 1 (2 methods).	**router eigrp** *asn* **redistribute ospf 2 metric 1000 10 255 1 1500 route-map fred** **route-map fred permit 10** **match ip address 1** or... **router eigrp** *asn* **redistribute ospf 2** **distribute-list 1 out ospf 2**

Table 10-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Filtering routes on redistribution from EIGRP into OSPF, allowing only routes permitted by prefix list barney (2 methods).	**router ospf** *process* **redistribute eigrp 1 subnets route-map fred** **route-map fred permit 10** **match ip prefix-list barney** or... **router ospf** *process* **redistribute eigrp 1 subnets** **distribute-list prefix barney out eigrp 1**
Configuring the route map that will set metric components to 1000, 200, 255, 1, and 150, for routes permitted by ACL 1, and filter all other routes.	**route-map fred permit 10** **match ip address 1** **set metric 1000 200 255 1 1500**
Set OSPF's administrative distance for all internal routes to 110, and all external routes to 180.	**router ospf** *process-id* **distance ospf external 180** (intra-area and interarea AD will default to 110)
Set EIGRP's administrative distance for routes learned from neighbor 1.1.1.1 to 190, only for subnets between 10.1.0.0 – 10.1.255.255.	**router eigrp** *asn* **distance 190 1.1.1.1 0.0.0.0 list 1** **access-list 1 permit 10.1.0.0 0.0.255.255**

Table 10-11 *Verification Plan Memory Drill*

Information Needed	Command
Confirm that OSPF routes were redistributed from the IP routing table into that same router's EIGRP topology table.	**show ip eigrp topology**
Display the number of matches in an ACL used for redistribution filtering.	**show ip access-lists** *[number-or-name]* **show access-lists** *[number-or-name]*
Display the number of matches in an IP prefix list used for redistribution filtering.	**show ip prefix-list detail** *[name]*

Table 10-11 *Verification Plan Memory Drill*

Information Needed	Command
Display the configuration of a route map.	**show route-map** *[name]*
Display the component metrics of a route redistributed into EIGRP.	**show ip eigrp topology** *prefix/length*
Confirm the absence or presence of a route that could have been redistributed from OSPF into EIGRP.	**show ip eigrp topology** *prefix/length* **show ip eigrp topology**
Confirm the absence or presence of a route that could have been redistributed from EIGRP into OSPF.	**show ip ospf topology** *prefix/length* **show ip ospf topology**
Display an IP route's administrative distance.	**show ip route** *[subnet]*
Display the administrative distance settings for EIGRP.	**show ip protocols**
Display the administrative distance settings for OSPF.	**show ip protocols**

Chapter 11

Table 11-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design calls for traffic destined for one server in subnet 10.1.1.0/24 to be sent over a different route than the IGP-learned route for 10.1.1.0/24.	Use Policy-Based Routing. Configure a static route for packets sent to that single server.
Same requirement as the previous row, except that only a subset of the source hosts should have their packets take a different route than the IGP-learned route.	Use PBR.
The design requires that a static route be used, but only when a particular database server is reachable.	Use IP SLA with object tracking for the static route.

Table 11-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A plan lists two PBR route maps–one that uses the **default** keyword in its **set** command, and the other that does not. What is the fundamental difference?	The route map with the **default** keyword will cause IOS to attempt to route the packet as normal first, and if no nondefault route is matched, then use the route in the **set** command. Without the **default** keyword, IOS tries the PBR route first.
A plan shows a route map enabled for policy routing, and the route map matches some packets with a deny route-map clause. What does IOS do with those packets?	IOS does not route these packets with PBR and allows the packets through the normal IOS packet forwarding logic. The packets are not filtered.
The plan document shows a PBR route map with the command **set ip dscp ef**. Does PBR support marking? And can it mark DSCP?	PBR can mark the IP Precedence bits and the entire ToS byte, but it cannot mark DSCP. Class-based marking is preferred for QoS marking today.
The plan shows an IP SLA operation number 5, with a static route configured with the **track 5** parameters. What issues might exist with the linkages between these commands?	To track the state of an IP SLA operation, the **ip route** command must refer to a tracking object number, which in turn refers to the IP SLA operation number.
The IP SLA configuration shows an IP SLA operation that uses ICMP Echo, with the destination IP address of a server. What must be done on the server to support this operation?	Nothing–the server naturally responds to the ICMP Echo.
Same scenario as the previous row, except the destination address is on a router.	Nothing–the router also naturally responds to an ICMP Echo.
Same scenario as the previous row, except the operation generates RTP packets to measure voice jitter.	The remote router needs to be configured as an IP SLA responder.

Table 11-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure the matching logic in a PBR route map (two options).	*route-map name* [*number*] **permit** **match ip address** {*acl-number* | *acl-name*} or **match length** *min max*
Configure the next-hop IP address in a PBR route map (two options).	*route-map name* [*number*] **permit** **set ip next-hop** ip-address [... *ip-address*] or **set ip default next-hop** *ip-address* [... *ip-address*]
Configure the outgoing interface in a PBR route map (two options).	*route-map name* [*number*] **permit** **set interface** *type number* [... *type number*] or **set default interface** *type number* [... *type number*]
Enable PBR on an interface.	**interface** *type number* **ip policy route-map** *route-map-name*
Enable PBR for packets created by the router.	**ip local policy route-map** *route-map-name* (global command)

Table 11-8 *Verification Plan Memory Drill*

Information Needed	Command
List interfaces on which PBR is enabled and the route-map used.	**show ip policy**
Display the configuration of a route map.	**show route-map**
Generate debug messages for each packet that matches PBR.	**debug ip policy**
Display the configuration of an SLA operation.	**show ip sla configuration**
Show the measurements from an SLA operation.	**show ip sla statistics**
Display the status of a tracking object.	**show track**

Chapter 12

Table 12-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
A design shows a single router connected to the Internet as part of a single-homed Internet design. It lists sections for Enterprise routing toward the Internet router(s) and another section for choosing routes into the Internet. List the reasonable options.	For Enterprise routers toward the Internet-facing router in the Enterprise: ■ Internet router to inject/flood a default route with the IGP. Can be based on a static default or BGP-learned default. For routes on the Internet-facing router, use either ■ Static default. ■ BGP-learned default.
Use the same criteria as the previous item in this table, except the single Enterprise router connected to the Internet now has two links to the same ISP (dual homed).	Same as previous table row for Enterprise toward Internet router. For routes on the Internet-facing router,, use ■ Static defaults. ■ BGP-learned defaults. ■ Accept partial or full tables and choose the best path for each destination.
Use the same criteria as the previous item, except use two routers with one link each to the same ISP (dual homed).	For Enterprise routers toward the Internet-facing routers in the Enterprise, flood default routes with the IGP from each Internet-facing router. Same three options as in previous row of this table for routes toward the Internet. However, if the design requires a choice of some paths (routes) as better than others with BGP, use iBGP between Internet-connected routers, and possibly with other Enterprise routers.
Same criteria as previous row, but with a single multihomed with two routers.	Same as previous row.

Table 12-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a single router in a dual-homed Internet design, with the router using BGP over each link to that same ISP. What criteria would impact your choice of accepting only default routes, or partial updates, or full updates, using BGP in this case? (3)	Defaults: If the goal were to use one path as primary, but to use the other path if the first failed Partial: If the ISP can identify each link as a better link by setting PAs, but for a smaller set of prefixes If each link can be considered better for the majority of BGP routes
The plan shows four Enterprise routers with BGP configuration, with two of those routers with links to two different ISPs. Which connections are eBGP? iBGP?	The BGP neighborships between the four Enterprise routers would be iBGP. Any neighborships with ISP routers would be eBGP.

Table 12-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configuring multiple static default routes, each with different administrative distance settings.	**ip route** *subnet mask next-hop-ip ad-value* (Configure two different routes, each with a different AD value.)

Chapter 13

Table 13-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The plan shows the use of public prefix 200.1.1.0/26 by an Enterprise. What methods should I consider adding to my implementation plan for advertising that prefix to my ISPs using BGP? (2)	Use the BGP **network** command. Redistribute from the IGP into BGP.

Table 13-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows Enterprise Router R1, with two parallel Layer 3 paths to ISP Router R2, with a need for BGP. What options exist for high availability eBGP peering? (2) Which is better?	1. Use a loopback interface as update source, and configure eBGP multihop with a single BGP peer. 2. Use interface IP addresses for BGP peering, but with two neighbor relationships with the same neighboring router. The first option reduces the amount of overhead, while giving the same higher availability.
The implementation plan shows an Enterprise router with an eBGP connection to an ISP router, using a loopback interface as Update source. What other feature must be configured to make the eBGP connection work?	eBGP multihop.
Router R1 connects via eBGP to Router I1 at ISP1. R1 has routes for 130.1.1.0/24 and 130.1.2.0/24 in its routing table. The design claims the company uses 130.1.0.0/21 as its public range. What methods can be used to advertise one route for the entire range to the eBGP peer? (2)	1. Cause a route to be created on R1, via static config or IGP route summarization, for 130.1.0.0/21, combined with the BGP **network 130.1.0.0 mask 255.255.248.0** command. 2. Redistribute from the IGP into BGP, and then configure route summarization with the **summary-only** keyword to advertise 130.1.0.0/21 without advertising the subordinate routes.

Table 13-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an eBGP connection as follows: local AS 1, remote AS 2, remote router uses 1.1.1.1 for BGP peering, with 1.1.1.1 being an IP address on a common link between the routers.	**router bgp 1** **neighbor 1.1.1.1 remote-as 2**

Table 13-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an eBGP connection as follows: local AS 1, remote AS 2, local uses loopback1 (1.1.1.1), remote uses loopback2 (2.2.2.2).	**router bgp 1** **neighbor 2.2.2.2 remote-as 2** **neighbor 2.2.2.2 update-source loopback1** **neighbor 2.2.2.2 ebgp-multihop**
Add to the previous row the configuration to use MD5 authentication, key "barney."	**router bgp 1** **neighbor 2.2.2.2 password barney**
Administratively disable the neighbor configured in the previous two items in this table	**router bgp 1** **neighbor 2.2.2.2 shutdown**
Re-enable the neighbor that was disabled in the previous row of this table.	**router bgp 1** **no neighbor 2.2.2.2 shutdown**
Cause the advertisement of IGP-learned prefix 130.1.1.0/24 to the neighbor configured in this table, without redistribution.	**router bgp 1** **network 130.1.1.0 mask 255.255.255.0**
Repeat the task in the previous row of this table, but this time with route redistribution, assuming OSPF process 1 is used for the IGP.	**router bgp 1** **redistribute ospf 1**

Table 13-8 *Verification Plan Memory Drill*

Information Needed	Commands
Display a single-line neighbor status for each iBGP neighbor.	**show ip bgp summary**
Display the number of prefixes learned from a neighbor. (List where the information is located.)	**show ip bgp [summary]** (Look for "State/PfxRcd" heading on the right.) **show ip bgp neighbors** *[neighbor-id]* (Look for "prefix activity.")
Display the number of prefixes advertised to a neighbor. (List where the information is located.)	**show ip bgp neighbors** *[neighbor-id]* (Look for "prefix activity.") **show ip bgp neighbors** *neighbor-id* **advertised-routes** (Look for "Total number of prefixes"–note that this command is not covered until Chapter 14.)

Table 13-8 *Verification Plan Memory Drill*

Information Needed	Commands
Display the local and neighbor ASN.	**show ip bgp summary** (Look for "Local AS Number" near the top.)
Display the number of eBGP hops allowed.	**show ip bgp neighbors** *[neighbor-id]*
List the current TCP ports used for BGP connections.	**show ip bgp nei** ghbors *[neighbor-id]* **show tcp brief**
List all prefixes in the BGP table.	**show ip bgp**
List all the best routes in the BGP table.	**show ip bgp**
Find the AS_Path for each BGP Table entry. (Describe how.)	**show ip bgp** (Look for "Path" heading on the right.)
Determine if a particular BGP table entry is iBGP-learned. (Describe how.)	**show ip bgp** (Look for code "i" on the left) **show ip bgp** *prefix/length* (Look for "internal.")
Display one-line entries for all BGP table entries with a given prefix/length, plus any subnets inside that ranges.	**show ip bgp** *prefix/length* **longer-prefixes**
List possible default routes.	**show ip bgp 0.0.0.0 0.0.0.0**
List possible routes per prefix.	**show ip bgp** *prefix [subnet-mask]*
List routes learned from one neighbor, which passed any inbound filters.	**show ip bgp neighbors** *ip-address* **routes**
List routes learned from one neighbor before any inbound filtering is applied.	**show ip bgp neighbors** *ip-address* **received-routes**
Display routes suppressed and added to the BGP table due to BGP route summarization (aggregation).	**show ip bgp**

Chapter 14

Table 14-4 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The plan shows a typical single-multihomed design with two routers connected to 2 ISPs. How will you ensure next-hop reachability? (2)	1. Configure the Enterprise routers with **next-hop-self**. 2. Ensure both Enterprise BGP routers have routes to reach all eBGP peers.
The plan shows the same design as the last item. The two Enterprise Internet-connected routers do not have a direct link between each other. What methods discussed in this chapter can be used to prevent packet loops in the Enterprise core? (2)	1. Run an iBGP mesh with all Enterprise core routers between the Internet-connected routers. 2. Redistribute BGP into your IGP, and enable synchronization.
The plan shows the same design as the previous items but with public range 200.1.1.0/24 being the only public address range used by the Enterprise. How can the Enterprise avoid becoming a transit AS?	Use filtering to advertise only prefix 200.1.1.0/24 to the ISPs.

Table 14-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan shows a typical single-multihomed design with two routers (R1 and R2) connected to 2 ISPs. Will R1 and R2 be BGP neighbors? Why?	R1 can use R2 as the best next-hop router to reach some destinations, and vice versa, but only if the two routers exchange BGP routes by becoming iBGP neighbors.
The plan shows the same design as the previous item. What configuration setting must be used to ensure the routers are iBGP rather than eBGP peers?	Each router's **neighbor remote-as** command refer to their own ASN, as configured in the **router bgp** command.

Table 14-5 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan calls for filtering all prefixes except the 200.1.1.0/24 public address range when advertising the any eBGP peers. Which **neighbor** command options exist for filtering based on the prefix/length? (3)	1. **neighbor prefix-list** 2. **neighbor distribute-list** 3. **neighbor route-map**

Table 14-6 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an iBGP peer.	**router bgp** *asn* **neighbor** *neighbor-ip* **remote-as** *asn*
Advertise the local router's Update source IP address as the next-hop address to iBGP peers.	**router bgp** *asn* **neighbor** *neighbor-ip* **next-hop-self**
Configure an iBGP mesh with peers 1.1.1.1, 2.2.2.2, 3.3.3.3.	**router bgp** *asn* **neighbor** *1.1.1.1* **remote-as** *same-asn* **neighbor** *2.2.2.2* **remote-as** *same-asn* **neighbor** *3.3.3.3* **remote-as** *same-asn*
Enable BGP synchronization.	**router bgp** *asn* **synchronization**
Configure filtering of routes sent to eBGP peer 9.9.9.9, using a prefix list to allow only 200.1.1.0/24.	**ip prefix-list** *plist-name* **permit 200.1.1.0/24** **router bgp** *asn* **neighbor** *9.9.9.9* prefix-list *plist-name* **out**
Configure filtering of routes sent to eBGP peer 9.9.9.9, using an ACL to allow only 200.1.1.0/24.	**access-list 101 permit ip host 200.1.1.0 host 255.255.255.0** **router bgp** *asn* **neighbor** *9.9.9.9* **distribute-list 101 out**

Table 14-7 *Verification Plan Memory Drill*

Information Needed	Commands
Display a single-line neighbor status for all iBGP neighbors.	**show ip bgp summary**
Determine if a particular BGP table entry is iBGP-learned.	**show ip bgp** (Look for code "i" on the left) **show ip bgp** *prefix/length* (Look for "internal")
Determine the next-hop IP address of an iBGP learned route .	**show ip bgp** **show ip bgp** *prefix/length*
Identify the neighbor from which a BGP route was learned.	**show ip bgp** *prefix/length* (Look for the "from *ip-address*" phrase)
Display one-line entries for all BGP table entries with a given prefix/length, plus any subnets inside that ranges.	**show ip bgp** *prefix/length* **longer-prefixes**
Display BGP routes learned from a neighbor, before being processed by an inbound filter.	**show ip bgp neighbors** *neighbor-ip* **received-routes**
The same as the previous item, but after applying the inbound filter.	**show ip bgp neighbors** *neighbor-ip* **routes**
Display BGP routes sent to a neighbor but after applying the outbound filter.	**show ip bgp neighbors** *neighbor-ip* **advertised-routes**
Display whether a neighbor can perform BGP route refresh.	**show ip bgp neighbors** *[neighbor-id]*

Chapter 15

Table 15-8 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
Influence outbound route from an Enterprise toward prefixes in the Internet (3).	Setting administrative Weight, Local Preference, and AS_Path Length (using AS_Path prepending)
Influence outbound route from an Enterprise toward prefixes in the Internet so that multiple Internet-connected Enterprise routers make the same choice based on the same information (2).	Setting Local Preference and AS_Path Length (using AS_Path prepending)
Influence inbound routes into the Enterprise from the neighboring AS (2).	Setting MED, AS_Path Length (using AS_Path prepending)

Table 15-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A plan shows two Enterprise routers, R1 and R2, connected to two different ISPs, with iBGP between R1 and R2. The plan shows R1 setting Weight for routes learned from an ISP. Will R2 react to those settings? Why or why not?	No. Weight is local to a single router and is not advertised to neighboring routers.
A plan shows two Enterprise routers, R1 and R2, connected to two different ISPs, with iBGP between R1 and R2. The plan shows R1 setting Local_Pref for routes learned from an ISP. Will R2 react to those settings? Why or why not?	Yes. Local_Pref is advertised to iBGP peers, so both R1 and R2 make the same choices based on the Local_Pref.
The plan calls for the use of BGP Weight, but the incomplete plan lists no configuration yet. What configuration alternatives exist? (2)	1. Setting weight for routes matched in a route map clause. 2. Setting weight for all routes learned from a neighbor with the **neighbor weight** command.

The plan calls for the use of BGP Local Preference, but the incomplete plan lists no configuration yet. What configuration alternatives exist?	Setting Local Preference for routes matched in a route map clause.
A plan shows two Enterprise routers, R1 and R2, connected to different ISPs. The plan calls for using MED to influence inbound routes. Which configuration options exist?	Setting MED for routes matched in a route map clause, either outbound from the Enterprise or inbound into the ISP.
A plan shows the use of BGP Weight, Local Preference, AS_Path prepending, and MED to influence the best path algorithm. Which of these can be set and advertised to eBGP peers?	AS_Paths can be prepended, and MED can be set, before being sent to eBGP neighbors in BGP Updates.

Table 15-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure a route map that sets weight.	**route-map** *name* **permit** **match...** **set weight** *value*
Enable a route map to set BGP weight.	**router bgp** *asn* **neighbor** *neighbor-ip* **route-map** *name* **in**
Enable a router to set BGP weight for all routes received from a neighbor.	**router bgp** *asn* **neighbor** *neighbor-ip* **weight** *value*
Configure a route map that sets BGP Local Preference.	**route-map** *name* **permit** **match...** **set local-preference** *value*
Enable a route map to set BGP Local Preference.	**router bgp** *asn* **neighbor** *neighbor-ip* **route-map** *name* **in**
Configure a route map that prepends ASNs to an AS_Path.	**route-map** *name* **permit** **match...** **set as-path prepend** *asn1 [asn2]...*

Enable a route map to perform AS_Path prepending.	**router bgp** *asn* **neighbor** *neighbor-ip* **route-map** *name* **[in | out]**
Configure a route map that sets MED.	**route-map** *name* **permit** **match...** **set metric** *value*
Enable a route map to set MED.	**router bgp** *asn* **neighbor** *neighbor-ip* **route-map** *name* **[in | out]**

Table 15-11 *Verification Plan Memory Drill*

Information Needed	Commands
Display the BGP table, including the chosen best path for each prefix. (State how to identify the best paths.)	**show ip bgp** (look for > as the 2nd character)
List 1 line per BGP route but for the prefixes within a range.	**show ip bgp** *prefix/length* **longer-prefixes**
Identify a BGP table entry's BGP Weight. (Specify where to find output.)	**show ip bgp** (Look for heading "Weight")
Identify a BGP table entry's BGP Local Preference. (Specify where to find output.)	**show ip bgp** (Look for heading "LocPrf") **show ip bgp** *prefix/length* (Look for "localpref")
Identify a BGP table entry's AS_Path length. (Specify where to find output.)	**show ip bgp** (Count the ASNs under heading "Path") **show ip bgp** *prefix/length* (Count the number of ASNs)

Table 15-11 *Verification Plan Memory Drill*

Information Needed	Commands
Identify a BGP table entry's MED. (Specify where to find output.) (4 methods)	**show ip bgp** (Look for heading "Metric") **show ip bgp** *prefix/length* (Look for "metric") **show ip route** (Look for 2nd number in square brackets) **show ip route** *prefix mask* (Look for "route metric")
Display routes received from a neighbor before being processed by an inbound filter.	**show ip bgp neighbors** *neighbor-ip* **received-routes**
The same as the previous item but after applying the inbound filter.	**show ip bgp neighbors** *neighbor-ip* **routes**
Display BGP routes sent to a neighbor but after applying the inbound filter.	**show ip bgp neighbors** *neighbor-ip* **advertised-routes**
Display BGP best paths that were not added to the IP routing table.	**show ip bgp rib-failures**

Chapter 16

Table 16-12 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
An IPv6 design suggests that all client hosts should dynamically learn their IPv6 addresses. Which tools can be used? (2)	Stateful DHCP Stateless autoconfig
A plan shows the use of stateless autoconfiguration. What functions should we expect the IPv6 DHCP server to perform?	To supply the DNSv6 server's IPv6 addresses

Table 16-13 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
An implementation plan states that router IPv6 address should be assigned as obvious values, using the lowest numbers in the range per each assigned prefix. What configuration methods could be used to configure these low address values?	Statically configure the entire address with the **ipv6 address** command. Configure the MAC address to a low number, and configure the address with the **ipv6 address eui-64** command.
A plan calls for the use of stateless autoconfig for client hosts. What must be configured on the routers to support this process?	Routers must respond to Router Solicitation messages with Router Advertisement (RA). To do so, a router must have IPv6 routing enabled and a unicast IPv6 address configured on the interface in which the RS is received.

Table 16-14 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure the full global unicast address on an interface.	**interface** *type number* **ipv6 address** *address/prefix-length*
Configure the unicast IPv6 prefix on an interface, and let the router add the interface ID.	**interface** *type number* **ipv6 address** *address/prefix-length* **eui-64**
Configure an interface to find its unicast IPv6 address using stateless autoconfig.	**interface** *type number* **ipv6 address autoconfig**
Configure an interface to enable IPv6 and use another interface's IPv6 address as needed.	**interface** *type number* **ipv6 unnumbered** *type number*
Enable IPv6 on an interface and do not configure a unicast IPv6 address.	**interface** *type number* **ipv6 enable**

Table 16-14 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure the link local address of an interface.	**interface** *type number* **ipv6 address** *address* **link-local**

Table 16-15 *Verification Plan Memory Drill*

Information Needed	Commands
All IPv6 routes	**show ipv6 route**
A single line per IPv6 address	**show ipv6 interface brief**
Detailed information about IPv6 on an interface, including multicast addresses	**show ipv6 interface** *[type number]*
The MAC address used by an interface	**show interfaces** *[type number]*
The MAC addresses of neighboring IPv6 hosts	**show ipv6 neighbors**
The information learned from another router in an RA message	**show ipv6 router**

Chapter 17

Table 17-9 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
A RIPng implementation plan lists two neighboring routers with unicast IPv6 addresses 2000::1/64 and 2001::2/64, respectively. Will this cause a neighborship issue?	No; RIPv6 does not use the concept of neighbors, but it also does not prevent routes from being exchanged.
Same issues as in the previous row, but the plan uses EIGRP for IPv6.	No; EIGRP for IPv6 does not require neighboring routers to be in the same IPv6 subnet.
A plan shows a planned config for a new router, with no IPv4 addresses, IPv6 addresses on all interfaces, and EIGRP for IPv6 configuration. What potential issues should you look for in the configuration? (3)	Look to confirm that the **ipv6 router eigrp** *asn* command has been added to each interface. Make sure the **no shutdown** command is listed under the EIGRP process. Make sure the router ID has been defined with the **eigrp router-id** router subcommand.
Same scenario as the previous row, but with OSPFv3.	Look to confirm that the **ipv6 router ospf** *process-id* command has been added to each interface. Make sure the router ID has been defined with the **router-id** router subcommand.
The plan shows an EIGRP for IPv6 and OSPFv3 domain with mutual redistribution. The configuration shows a **redistribute eigrp 1** command under the OSPF process. What kinds of routes should be redistributed? Which kinds will not?	The configuration redistributes EIGRP-learned routes. It will not redistribute: ■ link local addresses ■ local routes ■ connected routes

Table 17-10 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Assuming IPv6 routing and IPv6 addresses have already been configured, configure RIPng.	**ipv6 router rip** *process-name* **interface** *type number* **ipv6 rip** *process-name* **enable** (Repeat previous 2 commands for each interface)
Assuming IPv6 routing and IPv6 addresses have already been configured and no IPv4 addresses exist on the router, configure EIGRP for IPv6.	**ipv6 router eigrp** *asn* **eigrp router-id a.b.c.d** **no shutdown** **interface** *type number* **ipv6 eigrp** *asn* (Repeat previous 2 commands for each interface)
Assuming IPv6 routing and IPv6 addresses have already been configured and no IPv4 addresses exist on the router, configure OSPFv3.	**ipv6 router ospf** *process-id* **router-id a.b.c.d** **interface** *type number* **ipv6 ospf** *process-id* **area** *area-number* (Repeat previous 2 commands for each interface)
Configure RIPng to redistribute routes from OSPF process 1 including subnets, and connected interfaces.	**ipv6 router rip** *process-name* **redistribute ospf** *process* **include-connected**

Table 17-11 *Verification Plan Memory Drill*

Information Needed	Commands
All IPv6 routes	**show ipv6 route**
Details about a given IPv6 prefix	**show ipv6 route** *prefix/length*
All routes within a given range of IPv6 addresses	**show ipv6 route** *prefix/length* **longer-prefixes**
All RIP-learned IPv6 routes	**show ipv6 route rip**
All next-hop IPv6 addresses used by RIP routes	**show ipv6 rip next-hops**
The interfaces on which RIP is enabled	**show ipv6 protocols**
All EIGRP-learned IPv6 routes	**show ipv6 route eigrp**
All EIGRP neighbors	**show ipv6 eigrp neighbors**
Summary of the EIGRP topology table	**show ipv6 eigrp topology**
OSPF router ID and SPF statistics	**show ipv6 ospf**
List of OSPF neighbors	**show ipv6 ospf neighbor**
All OSPF-learned IPv6 routes	**show ipv6 route ospf**
Interfaces enabled for OSPF and their assigned areas	**show ipv6 protocols** **show ipv6 ospf interface brief**
OSPF costs per interface	**show ipv6 ospf interface brief** **show ipv6 ospf interface** *[type number]*
Summary of the OSPF database	**show ipv6 ospf database**

Chapter 18

Table 18-5 *Design Review*

Design Goal	Possible Implementation Choices Covered in This Chapter
The design states that an Enterprise needs IPv6 support for most LANs, with a regular high-volume of IPv6 traffic. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	Native IPv6
The design states that an Enterprise needs IPv6 support for a set small subset of LANs but that their traffic will be regular. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	Point-to-point tunnels
The design states that an Enterprise needs IPv6 support for a set small subset of LANs but that their traffic will be irregular and occasional. Would native IPv6, point-to-point tunnels, or multipoint tunnels seem most appropriate?	Multipoint tunnels
The plan calls for IPv6 tunneling so that new routers, when added to the tunnel, do not require additional configuration on existing routers. What type tunnel would you choose, and what IPv6 address ranges?	Automatic 6to4 tunnels with all IPv6 addresses from the 2002::/16 range

Table 18-6 *Notable Questions from This Chapter to Consider During an Implementation Plan Peer Review*

Question	Answers
The plan calls for the use of OSPFv3 along with the implementation of IPv6 tunnels. What tunnel types do you expect to find the sample configurations? (2)	Manually configured tunnels or GRE tunnels.
The planning diagrams show multipoint tunnels, with IPv6 addresses that imbed an IPv4 address in the last two quartets. What type of tunneling do you expect to see in the sample configurations?	ISATAP multipoint tunnels.
The plan lists a sample configuration with the command **tunnel mode ipv6ip** under a tunnel interface. What type of tunneling is used in this case?	Manually configured tunnels.
Same question as the previous row, but the command listed is **tunnel mode ipv6ip isatap**.	ISATAP tunnels.
Same question as the previous row, but the command listed is **tunnel mode gre ip**.	GRE.
Same question as the previous row, but the command listed is **tunnel mode ipv6ip 6to4**.	Automatic 6to4 tunnels.
A plan shows the use of a manually configured tunnel and an ISATAP tunnel. What tunnel subcommand would you expect to see for the point-to-point tunnel, but not the multipoint tunnel?	The **tunnel destination** command is not needed on multipoint tunnels.

Table 18-7 *Implementation Plan Configuration Memory Drill*

Feature	Configuration Commands/Notes
Configure an IPv6 manually configured tunnel using a loopback IPv4 address. Ignore IPv6 addressing and routing configuration.	**interface loopback loopback-***number* **ip address** *address mask* **interface tunnel** *number* **tunnel source loopback** *loopback-number* **tunnel destination** *ip-address* **tunnel mode ipv6ip**
Add IPv6 addressing and routing configuration to the previous row's list. Assume EIGRP for IPv6 ASN 1 is preconfigured.	**ipv6 unicast-routing** **interface tunnel** *number* **ipv6 address** *address/prefix-length* **ipv6 eigrp 1**
Configure an IPv6 GRE tunnel using a loopback IPv4 address. Ignore IPv6 addressing and routing configuration.	**interface loopback loopback-***number* **ip address** *address mask* **interface tunnel** *number* **tunnel source loopback** *loopback-number* **tunnel destination** *ip-address* **tunnel mode gre ip** (or just allow this setting to default)
Configure an IPv6 automatic 6to4 tunnel using a loopback IPv4 address. Include only IPv6 configuration required for the tunnel to pass IPv6 traffic. Assume all hosts use addresses in the 2002::/16 range.	**interface loopback loopback-***number* **ip address** *address mask* **interface tunnel** *number* **tunnel source loopback** *loopback-number* **tunnel mode ipv6ip 6to4** **ipv6 unicast-routing** **interface tunnel** *number* **ipv6 address** *address/prefix-length* **ipv6 route 2002::/16 tunnel** *number*
List steps to migrate from the automatic 6to4 tunnel from the previous row to a comparable ISATAP tunnel.	**interface tunnel** *number* **ipv6 address** *prefix*/**64 eui-64** **tunnel mode ipv6ip isatap** **no ipv6 route 2002::/16 tunnel** *number* **ipv6 route** *prefix/length next-hop*

Table 18-8 *Verification Plan Memory Drill*

Information Needed	Commands
Tunnel interface status for IPv6.	**show ipv6 interface brief** **show ipv6 interface tunnel** *number* **show interfaces tunnel** *number*
Tunnel interface's IPv6 address(es).	**show ipv6 interface brief** **show ipv6 interface tunnel** *number*
Connected routes related to the tunnel.	**show ipv6 route**
The tunnel source and destination IPv4 addresses.	**show interfaces tunnel** *number* **show running-config**
Test the tunnel to see if it can pass traffic.	**ping** **traceroute**

Chapter 19

Table 19-2 *Design Review*

Question	Answer
When a branch uses its broadband Internet connection to communicate into the rest of an Enterprise network, what benefits does an IPsec tunnel provide? (3)	Privacy (encryption) Authentication Delivering private packets over the public Internet
To make the basic broadband connection work, and to support flows from the branch office hosts to/from public websites, what features discussed in this chapter might the router need to configure? (5)	PPPoA (DSL) PPPoE (Cable) DHCP NAT Firewall services
What method allows a branch to statically route over the IPsec tunnel to the rest of the Enterprise, but only when routes through the private WAN connection fail?	Configuring floating static routes–static routes that have a higher/worse administrative distance than the IGP

Table 19-2 *Design Review*

Question	Answer
For what reasons might a network engineer consider also using a GRE tunnel when connecting a branch router, over the Internet, with the rest of the Enterprise network?	To run an IGP, which in turn allows the routers to perform wider ranging path control and load balancing
When configuring DSL using dialer interfaces, in what configuration mode are the ATM details configured? PPP details? Layer 3 details?	ATM–under the ATM interface PPP and Layer 3–under the dialer interface
A branch router has configured its one LAN interface as a NAT inside interface and its DSL dialer interface as an outside interface. What prevents packets in the IPsec tunnel from being NATted?	NAT also must match the packet with a permit action in an ACL; the ACL typically only matches packets that have not been processed through a tunnel
When configuring an IPsec tunnel on a branch router, identify the three main configuration components that link the interface to the matching logic that determines which packets the router processes into the tunnel.	The outgoing interface's **crytpo map** command, which names a crypto map The crypto map's **match address** command, which refers to an ACL The ACL, which permits packets that should fed into the tunnel

This page intentionally left blank

GLOSSARY

224.0.0.5 The All OSPF Routers multicast IP address, listened for by all OSPF routers.

224.0.0.6 The All OSPF DR Routers multicast IP address, listened for by DR and BDR routers.

2Way (OSPF) A neighbor state that signifies the other router has reached neighbor status, having passed the parameter check.

6to4 An IPv6/IPv4 tunneling method.

ABR *See area border router.*

access layer A Cisco network design term that refers to the devices that connect directly to the user. For LAN designs, the access layer consists of the switches connected to end user hosts. For WANs, the access layer consists mainly of routers at remote sites.

Ack (EIGRP) An EIGRP message that is used to acknowledge reliable EIGRP messages, namely Update, Query, and Reply messages. Acks messages do not require acknowledgement with an ACK message.

ACL Access control list. A list configured on a router to control packet flow through the router, such as to prevent packets with a certain IP address from leaving a particular interface on the router.

active (BGP state) A BGP neighbor state in which the TCP connection has successfully completed, but the BGP neighbors have not yet agreed to exchange path information.

active (EIGRP) A state for a route in an EIGRP topology table that indicates that the router is actively sending Query messages for this route, attempting to validate and learn the current best route to that subnet.

address block Refers to a set of consecutive IP addresses. Often, this term is used more generically than the terms subnet or CIDR block, all of which refer to a set of IP addresses.

adjacent (OSPF) Any OSPF neighbor for which the database flooding process has completed.

administrative distance In Cisco routers, a means for one router to choose between multiple routes to reach the same subnet when those routes are learned by different routing protocols. The lower the administrative distance, the more preferred the source of the routing information.

administrative weight A Cisco-proprietary BGP feature. The administrative weight can be assigned to each NLRI and path locally on a router, impacting the local router's choice of the best BGP routes. The value cannot be communicated to another router.

advertised distance *See reported distance*.

aggregate route Another term for summary route.

aggregator An optional transitive BGP path attribute that, for a summary route, lists the BGP RID and ASN of the router that created the summary.

All DR Multicast The multicast IP address 224.0.0.6, listened for by DR and BDR routers.

All SPF Routers multicast The multicast IP address 224.0.0.5, listened for by all OSPF routers.

anycast An IPv6 address type that is used by a number of hosts in a network that are providing the same service. Hosts accessing the service are routed to the nearest host in an anycast environment based routing protocol metrics.

area A grouping of routers and router interfaces, typically contiguous. Routers in an area strive to learn all topology information about the area, and do not learn topology information about areas to which they do not connect.

area border router (ABR) A router that has interfaces connected to at least two different OSPF areas, one of which must be the backbone area. ABRs hold topology data for each area, and calculate routes for each area, and advertise about those routes between areas.

ARP Address Resolution Protocol. Defined in RFC 826, a protocol used on LANs so that an IP host can discover the MAC address of another device that uses a particular IP address.

AS number (ASN) A number between 1 and 64,511 (public) and 64,512 and 65,535 (private) assigned to an AS for the purpose of proper BGP operation.

AS_PATH A BGP path attribute that lists ASNs through which the route has been advertised. The AS_PATH includes four types of segments: AS_SEQ, AS_SET, AS_CONFED_SEQ, and AS_CONFED_SET. Often, this term is used synonymously with AS_SEQ.

AS_PATH access list A Cisco IOS configuration tool, using the **ip as-path access-list** command that defines a list of statements that match the AS_PATH BGP path attribute using regular expressions.

AS_PATH length A calculation of the length of the AS_PATH PA, which includes 1 for each number in the AS_SEQ, 1 for an entire AS_SET segment, and possibly other considerations.

AS_PATH prepending This term has two BGP-related definitions. First, it is the normal process in which a router, before sending an Update to an eBGP peer, adds its local ASN to the beginning of the AS_PATH path attribute. Second, it is the routing policy of purposefully adding one or more ASNs to the beginning of a route's AS_PATH path attribute, typically to lengthen the AS_PATH and make the route less desirable in the BGP decision process.

AS_SEQUENCE A type of AS_PATH segment consisting of an ordered list of ASNs through which the route has been advertised.

AS_SET A type of AS_PATH segment consisting of an unordered list of ASNs consolidated from component subnets of a summary BGP route.

ASBR Autonomous System Border Router. A router using OSPF in which the router learns routes via another source, typically another routing protocol, exchanging routes that are external to OSPF with the OSPF domain.

ASBR Summary LSA *See Type 4 LSA.*

authentication With routing protocols, the process by which the router receiving a routing update determines if the routing update came from a trusted router.

auto summary A routing protocol feature in which a router that connects to more than one classful network advertises summarized routes for each entire classful network when sending updates out interfaces connected to other classful networks.

Automatic 6to4 tunnel A type of IPv6 multipoint tunnel that uses a reserved address range (2002::/16) and imbeds the IPv4 address in the second and third quartets of the IPv6 address.

autonomous system In BGP, a set of routers inside a single administrative authority, grouped together for the purpose of controlling routing policies for the routes advertised by that group to the Internet.

Autonomous System Border Router *See ASBR.*

Autonomous System Number (ASN) *See AS number.*

autosummarization A routing protocol feature in which a router that connects to more than one classful network advertises summarized routes for each entire classful network when sending updates out interfaces connected to other classful networks.

backbone area (OSPF) Area 0; the area to which all other OSPF areas must connect for OSPF to work.

backbone router Any OSPF router that has at least one interface connected to the backbone area.

backup designated router (BDR) In OSPF, a router that is prepared to take over the designated router.

balanced hybrid Refers to one of three general types of routing protocol algorithms. The other two are distance vector and link-state. EIGRP is the only routing protocol that Cisco classifies as using a balanced hybrid algorithm.

bandwidth 1) The rate at which bits are sent on an interface. 2) The Cisco IOS Software setting, per the **bandwidth** command, that tells IOS the speed of the interface.

BDR *See backup designated router.*

best path algorithm A set of rules by which BGP examines the details of multiple BGP routes for the same NLRI and chooses the single best BGP route to install in the local BGP table.

BGP *See Border Gateway Protocol.*

BGP decision process *See best path algorithm.*

BGP hard reset The process of restarting a BGP neighbor relationship by closing the TCP connection, causing both neighboring routers to remove all paths formerly learned from that neighbor from their respective BGP tables.

BGP peer Another name for a BGP neighbor. A BGP neighbor is another router running BGP with which the local router has formed a BGP neighbor relationship for the purpose of exchanging BGP Updates.

BGP peer group In BGP, a configuration construct in which multiple neighbors' parameters can be configured as a group, thereby reducing the length of the configuration. Additionally, BGP performs routing policy logic against only one set of Updates for the entire peer group, improving convergence time.

BGP soft reset The process of restarting a BGP neighbor relationship without closing the underlying TCP connection, instead resending full Updates to the neighbor, and asking for the neighbor to send a full Update again.

BGP synchronization In BGP, a feature in which BGP routes cannot be considered to be a best route to reach an NLRI unless that same prefix exists in the router's IP routing table as learned via some IGP.

BGP table A table inside a router that holds the path attributes and NLRI known by the BGP implementation on that router.

BGP Update A BGP message that includes withdrawn routes, path attributes, and NLRI.

BGP Weight A local Cisco-proprietary BGP setting that is not advertised to any peers. A larger value is considered to be better.

Border Gateway Protocol (BGP) An exterior routing protocol designed to exchange prefix information between different autonomous systems. The information includes a rich set of characteristics called path attributes, which in turn allows for great flexibility regarding routing choices.

cable A short term to refer to using Cable TV (CATV) to transmit data, typically for high-speed Internet connections.

Challenge Handshake Authentication Protocol (CHAP) A *security feature defined by PPP that allows either or both endpoints on a link to authenticate the other device as a particular authorized device.*

CHAP *See Challenge Handshake Authentication Protocol.*

CIDR *See Classless interdomain routing.*

CIDR notation *See prefix notation.*

Cisco Express Forwarding (CEF) An optimized Layer 3 forwarding path through a router or switch. CEF optimizes routing table lookup by creating a special, easily searched tree structure based on the contents of the IP routing table. The forwarding information is called

the Forwarding Information Base (FIB), and the cached adjacency information is called the adjacency table.

Cisco Lifecycle Services An approach to the implementation of Cisco technologies, as defined by Cisco.

classful IP addressing A convention for discussing and thinking about IP addresses by which Class A, B, and C default network prefixes (of 8, 16, and 24 bits, respectively) are considered.

classful network An IPv4 Class A, B, or C network. It is called a classful network because these networks are defined by the class rules for IPv4 addressing.

classful routing A type of logic for how a router uses a default route. When a default route exists, and the Class A, B, or C network for the destination IP address does not exist in the routing table, the default route is used. If any part of that classful network exists in the routing table, but the packet does not match any existing subnet of that classful network, the packet does not match the default route and thus is discarded.

classful routing protocol An inherent characteristic of a routing protocol. Specifically, the routing protocol does not send subnet masks in its routing updates. This requires the protocol to make assumptions about classful networks and makes it unable to support VLSM and manual route summarization.

classless addressing A concept in IPv4 addressing that defines a subnetted IP address as having two parts: a prefix (or subnet) and a host.

classless interdomain routing (CIDR) Defined in RFCs 1517–1520, a scheme to help reduce Internet routing table sizes by administratively allocating large blocks of consecutive classful IP network numbers to ISPs for use in different global geographies. CIDR results in large blocks of networks that can be summarized, or aggregated, into single routes.

classless IP addressing A convention for IP addresses in which Class A, B, and C default network prefixes (of 8, 16, and 24 bits, respectively) are ignored.

classless routing protocol An inherent characteristic of a routing protocol. Specifically, the routing protocol sends subnet masks in its routing updates, thereby removing any need to make assumptions about the addresses in a particular subnet or network. This allows the protocol to support VLSM and manual route summarization.

component route Refer to a route that is included in a larger summary route.

contiguous network In IPv4, a internetwork design in which packets forwarded between any two subnets of a single classful network only pass through the subnets of that classful network.

control plane In IP routing, refers to the building of IP routing tables by IP routing protocols.

convergence The time required for routing protocols to react to changes in the network, removing bad routes and adding new, better routes so that the current best routes are in all the routers' routing tables.

core layer A Cisco network design term that refers to the devices through which most traffic flows, typically located near the center of a network. Core devices need to forward packets/frames with low delay, in high volume, and either do little or no services with the packets, or do so without a degradation in speed or throughput.

CSU/DSU Channel service unit/data service unit. A device that connects a physical circuit installed by the telco to some CPE device, adapting between the voltages, current, framing, and connectors used on the circuit to the physical interface supported by the DTE.

data plane In IP routing, a term referring to a set of processes that forward packets through a router.

Database Description (DD) A type of OSPF packet used to exchange and acknowledge LSA headers. Sometimes called DBD.

data-link connection identifier A Frame Relay address used in Frame Relay headers to identify the Virtual Circuit.

data communications equipment From a physical layer perspective, the device providing the clocking on a WAN link, typically a CSU/DSU, is the DCE. From a packet-switching perspective, the service provider's switch, to which a router might connect, is considered the DCE.

DCE *See data communications equipment.*

DD *See Database Description.*

Dead Interval With OSPF, the timer used to determine when a neighboring router has failed, based on a router not receiving any OSPF messages, including Hellos, in this timer period. Also called the *Dead Timer*.

default network An IOS mechanism for determining a router's default route, by which the router is configured with a classful network number as the default network, and the router uses its route for that network as its default route.

default route A route that is used for forwarding packets when the packet does not match any more specific routes in the IP routing table.

delay A Cisco IOS Software setting, per the **delay** command, that defines to the router an estimate of the time that a packet is expected to spend trying to exit a router interface. The **delay** command uses a unit of tens-of-microseconds.

designated router (DR) On multiaccess data links such as LANs, an OSPF router elected by the routers on that data link to perform special functions. These functions include the generation of LSAs representing the subnet and playing a key role in the database exchange process.

DHCP *See Dynamic Host Configuration Protocol.*

Differentiated Services A set of QoS RFCs that redefines the IP header's ToS byte and suggests specific settings of the DSCP field and the implied QoS actions based on those settings.

Differentiated Services Code Point The first six bits of the DS field, used for QoS marking.

Diffie-Hellman Key Exchange A key exchange protocol in which two devices can generate a shared secure symmetric key over an insecure medium.

DiffServ *See Differentiated Services.*

Diffused Update Algorithm A convergence algorithm used in EIGRP that provides loop-free operation at every instant throughout a route computation. Allows routers involved in a topology change to synchronize at the same time, while not involving routers that are unaffected by the change. Also called *Diffusing Update Algorithm* in some references.

Digital Signal Level 0 Inside Telcos' original TDM hierarchy, the smallest unit of transmission at 64 kbps.

digital subscriber line (DSL) A Layer 1 technology used on the Telco local loop to transmit digital data signals, using frequencies more than 4000 Hz, over the same two-wire circuit as analog voice signals (which typically use frequencies less than 4000 Hz).

Dijkstra Alternative name for the SPF algorithm, named for its inventor, Edsger W. Dijkstra.

Dijkstra Shortest Path First (SPF) algorithm The name of the algorithm used by link-state routing protocols to analyze the LSDB and find the least-cost routes from that router to each subnet.

discontiguous network In IPv4, an internetwork design in which packets forwarded between two subnets of a single classful network must pass through the subnets of another classful network.

Discretionary Path Attribute Describe some BGP Path Attributes, specifically those attributes for which a router does not have to support the PA.

distance vector The logic behind the behavior of some interior routing protocols, such as RIP and IGRP, characterized by routers sending brief information about a subnet, and a metric (vector) describing how far away that subnet is. Distance vector routing algorithms call for each router to send its entire routing table in each periodic update, but only to its neighbors. Distance vector routing algorithms can be prone to routing loops but are computationally simpler than link-state routing algorithms. Also called *Bellman-Ford routing algorithm*.

distribute list A Cisco IOS configuration tool for routing protocols by which routing updates may be filtered.

distribution layer A Cisco design term that refers to the devices to which the access layer connects, with the distribution layer distributing packets among the many access devices.

DLCI *See data-link connection identifier.*

domain loop A term used in this book, but not necessarily used widely, to describe a routing loop that occurs between different IGP routing domains as a result of multiple route redistribution points between routing domains.

DR *See designated router.*

DR election (OSPF) The process by which neighboring OSPF routers examine their Hello messages and elect the DR. The decision is based on priority (highest), or RID (highest) if priority is a tie.

DROther The term to describe a router that is neither the DR nor the BDR on a subnet that elects a DR and BDR.

DS field The second byte of the IP header, formerly known as the ToS byte and redefined by DiffServ.

DSCP *See Differentiated Services Code Point.*

DUAL *See Diffused Update Algorithm.*

dual multihomed Refers to a particular type of design between an Enterprise and the Internet, in which more than one ISP is used, with more than one link to each ISP.

dual stacks In IPv6, a mode of operation in which a host or router runs both IPv4 and IPv6.

dual homed Refers to a particular type of design between an Enterprise and the Internet, in which only one ISP is used, but using two or more links to that ISP.

duplicate address detection (DAD) An IPv6 mechanism through which a host can determine if another active host on the same link is trying to use the same IPv6 address.

Dynamic Host Configuration Protocol A standard (RFC 2131) protocol by which a host can dynamically broadcast a request for a server to assign to it an IP address, along with other configuration settings, including a subnet mask and default gateway IP address.

E1 route (OSPF) An OSPF external route for which internal OSPF cost is added to the cost of the route as it was redistributed into OSPF.

E2 route (OSPF) An OSPF external route for which internal OSPF cost is not added to the cost of the route as it was redistributed into OSPF.

eBGP *See External BGP.*

eBGP multihop A BGP feature that defines the IP TTL field value in packets sent between two eBGP peers. This feature is required when using IP addresses other than the interface IP address on the link between peers.

EGP *See Exterior Gateway Protocol.*

EIGRP Enhanced Interior Gateway Routing Protocol. An advanced version of IGRP developed by Cisco. Provides superior convergence properties and operating efficiency and combines the advantages of link-state protocols with those of distance vector protocols.

EIGRP for IPv6 An interior routing protocol for IPv6 based on the original EIGRP protocol for IPv4.

EIGRP stub router A router running EIGRP that limits itself in several different ways for the purpose of limiting the EIGRP DUAL algorithm and reducing EIGRP Query scope.

Enterprise Edge A network design term referring to the routers at the distribution layer, connected to the WAN. Also called the *WAN edge*.

established A BGP neighbor state in which the BGP neighbors have stabilized and can exchange routing information using BGP Update messages.

Ethernet over MPLS (EoMPLS) The transport of Ethernet frames (mostly) transparently across an MPLS network.

EUI-64 A specification for the 64-bit interface ID in an IPv6 address, composed of the first half of a MAC address (with the seventh bit flipped), hex FFFE, and the last half of the MAC.

extended ping An IOS command in which the **ping** command accepts many other options besides just the destination IP address.

Exterior Gateway Protocol (EGP) A routing protocol that was designed to exchange routing information between different autonomous systems. EGP has been replaced by BGP and is no longer supported in IOS.

External BGP A term referring to how a router views a BGP peer relationship, in which the peer is in another AS.

External LSA In OSPF, an LSA that represents a subnet that OSPF learned from another (external) routing source, typically through route redistribution.

external route A characteristic of a route, as defined by a particular routing protocol, that means that the route was learned by that routing protocol through the route redistribution process.

External Type 1 *See E1 route.*

External Type 2 *See E2 route.*

FD *See feasible distance.*

feasibility condition With EIGRP, for a particular route, the case in which the reported distance is lower than the feasible distance.

feasible distance With EIGRP, the metric value for the lowest-metric route to a particular subnet.

feasible successor With EIGRP, a route that is not a successor route but that meets the feasibility condition; can be used when the successor route fails, without causing loops.

flash updates *See triggered updates.*

floating static route A static route configured with an administrative distance greater than a routing protocol on that same router, resulting in the static route floating into the routing table when the routing protocol's learned route fails.

flooding In OSPF, the process of exchanging LSA information throughout an area, by having a router send the LSAs to their neighbors who in turn send the LSAs to their neighbors, and so on.

forward route From one host's perspective, the route over which a packet travels from that host to some other host.

Frame Relay An international standard data-link protocol that defines the capabilities to create a frame-switched (packet-switched) service, allowing DTE devices (typically routers) to send data to many other devices using a single physical connection to the Frame Relay service.

Frame Relay Inverse ARP Defined in RFC 1293, this protocol enables a Frame Relay–attached device to react to a received LMI "PVC up" message by announcing its Layer 3 addresses to the device on the other end of the PVC.

Frame Relay mapping The information that correlates, or maps, a Frame Relay DLCI to the Layer 3 address of the DTE on the other end of the VC identified by the local DLCI.

full mesh A network design term often used with multiaccess network such as Frame Relay, referring to the case in which a direct communications path exists between every pair of devices in the design.

full SPF calculation An SPF calculation as a result of changes inside the same area as a router, for which the SPF run must examine the full LSDB.

Full State In OSPF, a neighbor state that implies that the two routers have exchanged the complete (full) contents of their respective LSDBs.

full update A routing protocol feature by which the routing update includes the entire set of routes, even if some or all the routes are unchanged.

fully adjacent (OSPF) Any OSPF neighbor for which the database flooding process directly between the two neighbors has completed. Note that not all neighbors directly exchange databases, so not all neighbors reach a full state.

gateway of last resort The notation in a Cisco IOS IP routing table that identifies the route used by that router as the default route.

Generic Routing Encapsulation A tunneling protocol that can be used to encapsulate many different protocol types, including IPv4, IPv6, IPsec, and others, to transport them across a network.

global routing prefix The first 48 bits of an IPv6 global address, used for efficient route aggregation.

global unicast address A type of unicast IPv6 address that has been allocated from a range of public globally unique IP addresses as registered through ICANN, its member agencies, and other registries or ISPs.

going active EIGRP jargon meaning that EIGRP has placed a route into active status.

Goodbye (EIGRP) An EIGRP message that is used by a router to notify its neighbors when the router is gracefully shutting down.

Graceful Restart (OSPF) As defined in RFC 3623, graceful restart allows for uninterrupted forwarding in the event that an OSPF router's OSPF routing process must restart. The router does this by first notifying the neighbor routers that the restart is about to occur; the neighbors must be RFC 3623-compliant and the restart must occur within the defined grace period.

Graceful shutdown EIGRP process of sending a goodbye message (actually held inside a Hello message) for the purpose of informing neighbors that the local EIGRP process is shutdown.

GRE *See Generic Routing Encapsulation*.

GRE tunnel A tunnel created using Generic Route Encapsulation. *See Generic Route Encapsulation*.

Hello (EIGRP) An EIGRP message that identifies neighbors, exchanges parameters, and is sent periodically as a keepalive function. Hellos do not require an Ack.

Hello (OSPF) A type of OSPF packet used to discover neighbors, check for parameter agreement, and monitor the health of another router.

Hello interval With OSPF and EIGRP, an interface timer that dictates how often the router should send Hello messages.

Hold timer With EIGRP, the timer used to determine when a neighboring router has failed, based on a router not receiving any EIGRP messages, including Hellos, in this timer period.

holddown A state into which a route is placed so that routers neither advertise the route nor accept advertisements about it for a specific length of time (the holddown period). Holddown is used to flush bad information about a route from all routers in the network. A route typically is placed in holddown when a link in that route fails.

iBGP Internal BGP. Refers to how a router views a BGP peer relationship, in which the peer is in the same AS.

iBGP Mesh A BGP design convention in which all BGP peers internal to a single AS have been directly peered so that all pairs of internal BGP routers are neighbors.

IEEE 802.1X An IEEE standard that, when used with EAP, provides user authentication before their connected switch port allows the device to fully use the LAN.

IGRP Interior Gateway Routing Protocol. An old, no-longer-supported Interior Gateway Protocol (IGP) developed by Cisco.

InARP *See Inverse ARP.*

infinity In the context of IP routing protocols, a finite metric value defined by the routing protocol that is used to represent an unusable route in a routing protocol update.

input event Any occurrence that could change a router's EIGRP topology table, including a received Update or Query, a failed interface, or the loss of a neighbor.

inside global address A NAT term referring to the IP address used for a host inside the trusted part of the network, but in packets as they traverse the global (untrusted) part of the network.

inside local address A NAT term referring to the IP address used for a host inside the trusted part of the network, but in packets as they traverse the local (trusted) part of the network.

interface ID Sixty-four bits at the end of an IPv6 global address, used to uniquely identify each host in a subnet.

interior gateway protocol (IGP) A routing protocol designed to be used to exchange routing information inside a single autonomous system.

Internal BGP (iBGP) A characteristic of a BGP neighbor relationship, specifically when the two routers are internal to the same BGP ASN.

internal routers An OSPF router that has interfaces connected to only one area, making the router completely internal to that one area.

Internet Assigned Numbers Authority (IANA) An organization that directs the assignment of IPv4 and IPv6 addresses worldwide.

Internet Service Provider (ISP) A company that provides Internet connectivity.

Inter-Switch Link (ISL) The Cisco-proprietary VLAN trunking protocol that predated 802.1Q by many years. ISL defines a 26-byte header that encapsulates the original Ethernet frame.

Invalid timer With RIP, a per-route timer that increases until the router receives a routing update that confirms the route is still valid, upon which the timer is reset to 0. If the updates cease, the Invalid timer will grow, until reaching the timer setting (default 180 seconds), after which the route is considered invalid.

Inverse ARP Defined in RFC 1293, this protocol enables a Frame Relay–attached device to react to a received LMI "PVC up" message by announcing its Layer 3 addresses to the device on the other end of the PVC.

inverse neighbor discovery An IPv6 feature on nonbroadcast multiaccess (NBMA) data links such as Frame Relay, providing the ability to learn a neighbor's Layer 3 address when the underlying Layer 2 address is known. The IPv6 equivalent of Frame Relay Inverse ARP.

IOS service level agreement (IOS SLA) An IOS feature that can be configured to generate packets, measure the delay, jitter, and simple working state of the measurement, and collect the data for reporting.

IP forwarding The process of forwarding packets through a router. Also called *IP routing*.

IP Precedence A three-bit field in the first three bits of the ToS byte in the IP header, used for QoS marking.

IP prefix list *See prefix list.*

IP routing The process of forwarding packets through a router. Also called *IP forwarding*.

IPsec Refers to the IP Security Protocols, which is an architecture for providing encryption and authentication services, typically when creating VPN services through an IP network.

IPsec tunnel A tunnel created using IPsec protocols.

IPv4 Version 4 of the IP protocol, which is the generally deployed version worldwide (at publication) and uses 32-bit IP addresses.

IPv6 Version 6 of the IP protocol, which uses 128-bit IP addresses.

ISATAP The Intra-site Automatic Tunnel Addressing Protocol that defines a protocol for creating dynamic multipoint IPv6 over IPv4 tunnels by imbedding the tunnel destination's IPv4 address in the last two quartets of the IPv6 address.

ISATAP tunnel A tunnel created using ISATAP. *See ISATAP.*

ISP prefix In IPv6, the prefix that describes an address block that has been assigned to an ISP by some Internet registry.

K-value EIGRP (and IGRP) allows for the use of bandwidth, load, delay, MTU, and link reliability; the K values refer to an integer constant that includes these five possible metric components. Only bandwidth and delay are used by default, to minimize recomputation of metrics for small changes in minor metric components.

keepalive A feature of many data-link protocols in which the router sends messages periodically to let the neighboring router know that the first router is still alive and well.

Keepalive (BGP) A BGP message sent to maintain an active neighbor relationship and maintain the underlying TCP connection when a router has no other BGP messages to send.

LAPF *See Link Access Procedure for Frame-Mode Bearer Services.*

leased line A transmission line reserved by a communications carrier for a customer's private use. A leased line is a type of dedicated line.

limiting query scope (EIGRP) An effort to reduce the query scope with EIGRP, using route summarization or EIGRP stub routers.

Link Access Procedure for Frame-Mode Bearer Services An ITU standard Frame Relay header, including the DLCI, DE, FECN, and BECN bits in the LAPF header, and a frame check in the LAPF trailer.

Link Control Protocol The portion of PPP focused on features that are unrelated to any specific Layer 3 protocol.

link local address A type of unicast IPv6 address that represents an interface on a single data link. Packets sent to a link local address cross only that particular link and are never forwarded to other subnets by a router. Used for communications that do not need to leave the local link, such as neighbor discovery.

Link state A classification of the underlying algorithm used in some routing protocols. Link-state protocols build a detailed database that lists links (subnets) and their state (up, down), from which the best routes can then be calculated.

Link State Acknowledgment A type of OSPF packet used to acknowledge LSU packets.

Link State Advertisement (LSA) The name of a class of OSPF data structures that hold topology information. LSAs are held in memory in the LSDB and communicated over the network in LSU messages.

Link State Database The data structure held by an OSPF router for the purpose of storing topology data.

Link State Identifier (LSID) A 32-bit number used to uniquely identify an OSPF LSA.

Link State Request A name of an OSPF packet that routers use to acknowledge the receipt of an LSU packet from another router.

Link State Update (LSU) The name of the OSPF packet that holds the detailed topology information, specifically LSAs.

Link State Database (LSDB) In OSPF, the data structure in RAM of a router that holds the various LSAs, with the collective LSAs representing the entire topology of the network.

Link state request An OSPF packet used to ask a neighboring router to send a particular LSA.

Link state routing protocol Any routing protocol that uses the concept of using the SPF algorithm with an LSDB to compute routes.

LMI *See Local Management Interface.*

Load A Cisco router interface statistic that measures the percentage link utilization, with the value represented as an integer between 0 to 255, and the percentage calculated as the listed number/255. EIGRP can use load as input to the EIGRP metric calculation.

Loading An OSPF neighbor state that occurs after the completion of database description messages, but while the database exchange using Link State Request and Link State Update packets continues.

local computation An EIGRP router's reaction to an input event, leading to the use of a feasible successor or going active on a route.

Local Management Interface (LMI) A Frame Relay protocol used between a DTE (router) and DCE (Frame Relay switch). LMI acts as a keepalive mechanism. The absence of LMI messages means that the other device has failed. It also tells the DTE about the existence of each VC and DLCI, along with its status.

Local Preference *See LOCAL_PREF*.

LOCAL_PREF A BGP path attribute that is communicated throughout a single AS to signify which route of multiple possible routes is the best route to be taken when leaving that AS. A larger value is considered to be better.

LSA *See Link-State Advertisement*.

LSA flooding The process of successive neighboring routers exchanging LSAs such that all routers have an identical LSDB for each area to which they are attached.

LSA type (OSPF) A definition that determines the data structure and information implied by a particular LSA.

LSAck *See Link-State Acknowledgment.*

LSDB *See Link-State Database.*

LSRefresh Link-State Refresh. An OSPF timer that determines how often the originating router should reflood an LSA, even if no changes have occurred to the LSA.

LSU *See Link State Update.*

LxPDU *See Layer x PDU.*

Management Information Base (MIB) The definitions for a particular set of data variables, with those definitions following the Structure of Management Information (SMI) specifications.

Mandatory PA A description of a BGP Path Attribute that means that all routers using BGP must support, understand, and react to that PA.

manually configured tunnel A type of IPV6-over-IPv5 point-to-point tunnel in which the tunnel source and destination is preconfigured.

Maximum Transmission Unit An IP variable that defines the largest size allowed in an IP packet, including the IP header. IP hosts must support an MTU of at least 576 bytes.

Measured Round-Trip Time A TCP variable used as the basis for a TCP sender's timer defining how long it should wait for a missing acknowledgment before resending the data.

Message Digest 5 Authentication With IP routing protocols, a method of applying a mathematical formula, with input including a private key, the message contents, and sometimes a shared text string, with the resulting digest being included with the message. The sender and the receiver perform the same math to allow authentication and to prove that no intermediate device changed the message contents.

metric With routing protocols, the measurement of favorability that determines which entry will be installed in a routing table if more than one router is advertising that exact network and mask with one routing protocol.

Metro Ethernet A general term for Ethernet-like WAN connectivity services, including VPWS and VPLS.

MIB *See Management Information Base.*

Multilink PPP A method of splitting, recombining, and sequencing datagrams across multiple point-to-point WAN links.

MLP *See Multilink PPP.*

MLS *See Multilayer Switching.*

Modified EUI-64 A variation on the EUI-64 method of completing the last 64 bits of an IPv6 address, specifically used for ISATAP tunnels. The last 64 bits (last four quartets) consist of 0000:5EFE, followed by the hex version of the tunnel destination's IPv4 address.

MRTT *See Measured Round-Trip Time.*

MTU Maximum transmission unit. The maximum packet size, in bytes, that a particular interface can handle.

MULTI_EXIT_DISC (MED) A BGP path attribute that allows routers in one AS to set a value and advertise it into a neighboring AS, impacting the decision process in that neighboring AS. A smaller value is considered better. Also called the BGP metric.

Multi Exit Discriminator *See MULTI_EXIT_DISC.*

multicast IP address range IP multicast address range from 224.0.0.0 through 239.255.255.255.

multicast IP address structure The first 4 bits of the first octet must be 1110. The last 28 bits are unstructured.

multicast MAC address A type of Ethernet MAC address meant to be used to send frames to a subset of the devices on a single broadcast domain. More specifically, as used with IPv4 multicast packets, a 48-bit address that is calculated from a Layer 3 multicast address by using 0x0100.5E as the multicast vendor code (OUI) for the first 24 bits, always binary 0 for the 25th bit, and copying the last 23 bits of the Layer 3 multicast address.

Multihomed A description of an Enterprise's connection to the Internet. This term refers to both single multihomed, which consists of one link each to two or more ISPs, and dual multihomed, with two or more links each to two or more ISPs.

Multilayer switching A process whereby a switch, when making a forwarding decision, uses not only Layer 2 logic but other OSI layer equivalents as well.

Multilink PPP A PPP feature used to load balance multiple parallel links at Layer 2 by fragmenting frames, sending one frame over each of the links in the bundle, and reassembling them at the receiving end of the link.

multipoint subinterface A configuration construct in a Cisco routers, typically with Frame Relay, in which one logical subinterface can be used to forward traffic to more than one remote router.

multipoint tunnel A type of tunnel in which more than one destination may be reached over a single tunnel.

NA *See Neighbor Advertisement.*

named access list An ACL that identifies the various statements in the ACL based on a name, rather than a number.

NAT *See Network Address Translation.*

NAT overload *See Port Address Translation (PAT).*

native VLAN The one VLAN on an 802.1Q trunk for which the endpoints do not add the 4-byte 802.1Q tag when transmitting frames in that VLAN.

NBMA *See nonbroadcast multiaccess (NBMA).*

NCP *See Network Control Protocol.*

ND *See Neighbor Discovery.*

neighbor In routing protocols, another router with which a router decides to exchange routing information.

Neighbor Advertisement (NA) In IPv6, the Neighbor Discovery message used by an IPv6 node to send information about itself to its neighbors.

Neighbor Discovery (ND) The protocol used in IPv6 for many functions, including those address autoconfiguration, duplicate address detection, router, neighbor, and prefix discovery, neighbor address resolution, and parameter discovery.

Neighbor Discovery Protocol (NDP) A longer name for IPv6 Neighbor Discovery. *See Neighbor Discovery*.

neighbor (EIGRP) With EIGRP, a router sharing the same primary subnet, with which Hellos are exchanged, parameters match, and with which routes can be exchanged.

neighbor (OSPF) Any other router, sharing a common data link, with which a router exchanges Hellos, and for which the parameters in the Hello pass the parameter-check process.

Neighbor Solicitation (NS) In IPv6, the Neighbor Discovery message used by an IPv6 node to request information about a neighbor or neighbors.

neighbor state A state variable kept by a router for each known neighbor or potential neighbor.

neighbor table For OSPF and EIGRP, a list of routers that have reached neighbor status.

Neighbor Type In BGP, either external BGP (eBGP), confederation eBGP, or internal BGP (iBGP). The term refers to a peer connection and whether the peers are in different ASs (eBGP), different confederation subanonomous systems (confederation eBGP), or in the same AS (iBGP).

Neighborship A shortened version of the phrase *neighbor relationship*.

Network Address Translation A mechanism for reducing the need for globally unique IPv4 addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space.

network address translation-protocol translation (NAT-PT) As defined in RFCs 2765 and 2766, a method of translating between IPv4 and IPv6 packets, which allows an IPv4-only host to communicate with an IPv6-only host.

Network Control Protocol The portions of PPP focused on features that are related to specific Layer 3 protocols.

Network layer reachability information A BGP term referring to an IP prefix and prefix length.

Network LSA An OSPFv2 Type 2 LSA. *See Type 2 LSA*.

network type (OSPF) A characteristic of OSPF interfaces that determines whether a DR election is attempted, whether neighbors must be statically configured, and the default Hello and Dead timer settings.

NEXT_HOP A BGP Path Attribute that lists the next-hop IP address used to reach an NLRI.

Next Hop field With a routing update, or routing table entry, the portion of a route that defines the next router to which a packet should be sent to reach the destination subnet. With routing protocols, the Next Hop field may define a router other than the router sending the routing update.

Next-hop self A BGP configuration setting that tells the local router to change the NEXT_HOP path attribute to refer to its own BGP Update Source when advertising routes to BGP neighbors.

NLPID Network Layer Protocol ID is a field in the RFC 2427 header that is used as a Protocol Type field to identify the type of Layer 3 packet encapsulated inside a Frame Relay frame.

NLRI *See Network layer reachability information.*

nonbackbone area Any OSPF area that is not the backbone area.

nonbroadcast multiaccess (NBMA) A characterization of a type of Layer 2 network in which more than two devices connect to the network, but the network does not allow broadcast frames to be sent to all devices on the network.

Notification–(BGP message) A BGP message used to inform BGP neighbors of a protocol error.

not-so-stubby area A type of OSPF stub area, which acts like other stub areas in that ABRs inject default routes into the area, but unlike non-NSSA stub areas in that external routes can be injected into the area.

NS *See Neighbor Solicitation.*

NSSA *See not-so-stubby area.*

NSSA External A reference to a Type 7 LSA. *See Type 7 LSA*.

object tracking An IOS feature in which IOS repeatedly checks the current state of some item so that other items can then act to a change in that state. For example, object tracking can track the state of IP SLA operations, with static routes and policy routes reacting to a change in the object tracking feature.

offset list A Cisco IOS configuration tool for RIP and EIGRP for which the list matches routes in routing updates and adds a defined value to the sent or received metric for the routes. The value added to the metric is the *offset*.

one-way redistribution The process of route redistribution in which one routing protocol redistributes routes into a second routing protocol, but the reverse redistribution is not configured.

Open A BGP message type used when the underlying TCP connection completes, for the purpose of exchanging parameter information to determine if the two routers are willing to become BGP neighbors.

OSPF *See Open Shortest Path First*.

optional nontransitive A characterization of a BGP path attribute in which BGP implementations are not required to support the attribute (optional), and for which if a router receives a route with such an attribute, the router should remove the attribute before advertising the route (nontransitive).

optional transitive A characterization of a BGP path attribute in which BGP implementations are not required to support the attribute (optional), and for which if a router receives a route with such an attribute, the router should forward the attribute unchanged (transitive).

ORIGIN A BGP path attribute that implies how the route was originally injected into some router's BGP table.

OSPF area A group of routers and links, identified by a 32-bit area number, whose detailed topology information OSPF shares among all routers in the group. Routers inside an area learn full detailed topology information about the area; this detailed information is not advertised outside the area.

OSPF network type A characteristic of OSPF interfaces that determines whether a DR election is attempted, whether neighbors must be statically configured, and the default Hello and Dead timer settings.

OSPF Version 3 An interior routing protocol created for IPv6 but based on OSPF Version 2, which was designed for IPv4.

Open Shortest Path First A popular link-state IGP that uses a link-state database and the Shortest Path First (SPF) algorithm to calculate the best routes to reach each known subnet.

Outside Global address A NAT term describing an IP address representing a host that resides outside the enterprise network, with the address being used in packets outside the enterprise network.

Outside Local address A NAT term describing an IP address representing a host that resides outside the enterprise network, with the address being used in packets inside the enterprise network.

overlapping subnets An (incorrect) IP subnet design condition in which one subnet's range of addresses includes addresses in the range of another subnet.

overloading Another term for Port Address Translation. *See PAT*.

packet switching A WAN service in which each DTE device connects to a telco using a single physical line, with the possibility of forwarding traffic to all other sites connected to the same service. The telco switch makes the forwarding decision based on an address in the packet header.

partial mesh A network topology in which more than two devices could physically communicate, but by choice, only a subset of the pairs of devices connected to the network are allowed to communicate directly.

partial SPF calculation An SPF calculation for which a router does not need to run SPF for any LSAs inside its area but instead runs a simple algorithm for changes to LSAs outside its own area.

partial update A routing protocol feature by which the routing update includes only routes that have changed, rather than include the entire set of routes.

passive (EIGRP) A state for a route in an EIGRP topology table that indicates that the router believes that the route is stable and that it is not currently looking for any new routes to that subnet.

Passive interface A routing protocol setting on an interface for which the router does not send Updates on the interface (RIP) or the router does not attempt to dynamically discover neighbors (EIGRP and OSPF), which indirectly prevents the EIGRP or OSPF router from sending Updates on the interface.

PAT *See Port Address Translation.*

path attribute Generally describes characteristics about BGP paths advertised in BGP Updates.

path control A general term, with several shades of meanings, that refers to any function that impacts how routers forward packets. These functions include routing protocols and any other feature that impacts the IP routing table, plus any feature that impacts the packet forwarding process.

peer group See *BGP peer group*.

periodic update With routing protocols, the concept that the routing protocol advertises routes in a routing update on a regular periodic basis. This is typical of distance vector routing protocols.

permanent virtual circuit (PVC) A preconfigured communications path between two Frame Relay DTEs, identified by a local DLCI on each Frame Relay access link, that provides the functional equivalent of a leased circuit but without a physical leased line for each VC.

permit An action taken with an ACL that implies that the packet is allowed to proceed through the router and be forwarded.

Point-to-Point Protocol (PPP) An Internet standard serial data-link protocol used on synchronous and asynchronous links that provides data-link framing, link negotiation, Layer 3 interface features, and other functions.

point-to-point tunnel A logical path between two devices created by encapsulating packets of one protocol (the passenger protocol) inside packets of another protocol (the transport protocol) specifically in cases where only two routers exist in the tunnel.

poison reverse With RIP, the advertisement of a poisoned route out an interface when that route was formerly not advertised out that interface due to split horizon rules.

poisoned route A route in a routing protocol's advertisement that lists a subnet with a special metric value, called an infinite metric, that designates the route as a failed route.

policy-based routing Cisco IOS router feature by which a route map determines how to forward a packet, typically based on information in the packet other than the destination IP address.

port (Multiple definitions) 1) In TCP and UDP, a number used to uniquely identify the application process that either sent (source port) or should receive (destination port) data. 2) In LAN switching, another term for switch interface.

Port Address Translation (PAT) A NAT term describing the process of multiplexing TCP and UDP flows, based on port numbers, to a small number of public IP addresses. Also called *NAT overloading*.

PPDIOO Prepare, Plan, Design, Implement, Operate, Optimize. The six phases of the Cisco Lifecycle Services approach.

PPP *See Point-to-Point Protocol*.

PPP over ATM (PPPoA) A convention often used as the data link protocol over DSL in which Asynchronous Transfer Mode (ATM) is used as the data link protocol, but with PPP encapsulated inside ATM. The combination gives the data link features of both ATM and PPP, in particular, the capability to forward the Layer 2 ATM cells to the DSLAM and the PPP authentication function of CHAP.

PPP over Ethernet (PPPoE) A convention often used as the data link protocol over cable in which Ethernet is used as the data link protocol but with PPP being encapsulated inside Ethernet. The combination gives the data link features of both Ethernet and PPP, in particular, the capability to forward the Layer 2 Ethernet frames to the correct router, plus PPP authentication function of CHAP.

Prefix (IPv4) Formally, a numeric value between 0 and 32 (inclusive) that defines the number of beginning bits in an IP address for which all IP addresses in the same group have the same value. Less formally, the subnet number when writing an address/mask combination using prefix notation.

Prefix (IPv6) A numeric value between 0 and 128 (inclusive) that defines the number of beginning bits in an IPv6 address for which all IP addresses in the same group have the same value.

prefix list A Cisco IOS configuration tool that you can use to match routing updates based on a base network address, a prefix, and a range of possible masks used inside the values defined by the base network address and prefix.

prefix notation A shorter way to write a subnet mask in which the number of binary 1s in the mask is simply written in decimal. For instance, /24 denotes the subnet mask with 24 binary 1 bits in the subnet mask. The number of bits of value binary 1 in the mask is considered to be the prefix.

priority (OSPF) An administrative setting included in Hellos that is the first criteria for electing a DR. The highest priority wins, with values from 1 to 255, with priority 0 meaning a router cannot become DR or BDR.

private address space An IP address in several Class A, B, and C networks that is set aside for use inside private organizations. These addresses, as defined in RFC 1918, are not routable through the Internet.

private addresses RFC 1918-defined IPv4 network numbers that are not assigned as public IP address ranges and are not routable on the Internet. Intended for use inside Enterprise networks.

private AS A BGP ASN whose value is between 64,512 and 65,535. These values are not assigned for use on the Internet and can be used for private purposes, typically either within confederations or by ISPs to hide the ASN used by some customers.

private ASN An Autonomous System Number (ASN) that falls inside the Private AS range.

private IP address *See private addresses*.

private IP network One of several classful IPv4 network numbers that will never be assigned for use in the Internet, meant for use inside a single enterprise.

private key A secret value used in public/private key encryption systems. Values encrypted with the public key can be decrypted with the private key and vice versa.

process switching A least optimized Layer 3 forwarding path through a router.

protocol data unit A generic term that refers to the data structure used by a layer in a layered network architecture when sending data.

protocol type A field in the IP header that identifies the type of header that follows the IP header, typically a Layer 4 header, such as TCP or UDP. ACLs can examine the protocol type to match packets with a particular value in this header field.

proxy ARP A router feature used when a router sees an ARP request searching for an IP host's MAC, when the router believes the IP host could not be on that LAN because the host is in another subnet. If the router has a route to reach the subnet where the ARP-determined host resides, the router replies to the ARP request with the router's MAC address.

public address space (IPv4) The nonreserved portions of the IPv4 unicast address space.

public ASN An ASN that fits below the private ASN range, specifically from 1 through 54,511.

public IP address *See public address space*.

public key A published value used in public/private key encryption systems. Values encrypted with the public key can be decrypted with the private key and vice versa.

PVC *See permanent virtual circuit.*

quartet A set of four hex digits listed in an IPv6 address. Each quartet is separated by a colon.

Query (EIGRP) An EIGRP message that asks neighboring routers to verify their route to a particular subnet. Query messages require an Ack.

query scope (EIGRP) The characterization of how far EIGRP Query messages flow away from the router that first notices a failed route and goes active for a particular subnet.

RA *See Router Advertisement.*

RD *See reported distance.*

redistribution The process on a router of taking the routes from the IP routing table, as learned by one routing protocol, and injecting routes for those same subnets into another routing protocol.

reference bandwidth In OSPF, the numerator in the calculation of interface cost. The formula is reference-bandwidth / interface-bandwidth.

Regional Internet Registry (RIR) The generic term for one of five current organizations responsible for assigning the public, globally unique IPv4 and IPv6 address space.

registry prefix In IPv6, the prefix that describes a block of public, globally unique IPv6 addresses assigned to a Regional Internet Registry by IANA.

regular area In OSPF, a nonbackbone area.

regular expression A list of interspersed alphanumeric literals and metacharacters used to apply complex matching logic to alphanumeric strings. Often used for matching AS_PATHs in Cisco routers.

reliability A Cisco router interface statistic that measures the percentage of packet loss, with the value represented as an integer between 0 to 255, and the percentage calculated as the listed number / 255. EIGRP can use reliability as input to the EIGRP metric calculation.

Reliable Transport Protocol A protocol used for reliable multicast and unicast transmissions. Used by EIGRP.

Reply (EIGRP) An EIGRP message that is used by neighbors to reply to a query. Reply messages require an Ack.

reported distance From one EIGRP router's perspective, the metric for a subnet as calculated on a neighboring router and reported in a routing update to the first router.

Retransmission Timeout With EIGRP, a timer started when a reliable (to be acknowledged) message is transmitted. For any neighbor(s) failing to respond in its RTO, the RTP protocol causes retransmission. RTO is calculated based on SRTT.

reverse route From one host's perspective, for packets sent back to this host from another host, the route over which the packet travels.

RIB Failure An event that occurs when the Routing Table Manager (RTM) attempts to add a route to the IP routing table, but a problem exists with the route that prevents RTM from adding the route.

RID *See router ID.*

RIP Routing Information Protocol. An Interior Gateway Protocol (IGP) that uses distance vector logic and router hop count as the metric. RIP version 1 (RIP-1) has become unpopular. RIP version 2 (RIP-2) provides more features, including support for VLSM.

RIP Next Generation An IPv6 Interior Routing Protocol based on RIP (for IPv4).

routable protocol *See routed protocol.*

route map A configuration tool in Cisco IOS that enables basic programming logic to be applied to a set of items. Often used for decisions about what routes to redistribute and for setting particular characteristics of those routes—for instance, metric values.

route poisoning The process of sending an infinite-metric route in routing updates when that route fails.

route redistribution The process of taking routes known through one routing protocol and advertising those routes with another routing protocol.

route summarization A consolidation of advertised addresses that causes a single summary route to be advertised.

Route Tag A field within a route entry in a routing update used to associate a generic number with the route. It is used when passing routes between routing protocols, allowing an intermediate routing protocol to pass information about a route that is not natively defined to that intermediate routing protocol. Frequently used for identifying certain routes for filtering by a downstream routing process.

routed protocol A Layer 3 protocol that defines a packet that can be routed, such as IPv4 and IPv6.

Router Advertisement (RA) In IPv6, a router advertisement message used by an IPv6 router to send information about itself to nodes and other routers connected to that router.

router ID (RID) In OSPF, a 32-bit number, written in dotted decimal, that uniquely identifies each router.

Router LSA Another name for an OSPF Type 1 LSA.

Router Solicitation (RS) An IPv6 message, part of the Neighbor Discovery Protocol (NDP), used by a host to request that the routers on the same data link announce their presence, IPv6 addresses, and all prefix/length combinations using a Router Advertisement (RA) message.

routing black hole A problem that occurs when an AS does not run BGP on all routers, with synchronization disabled. The routers running BGP might believe they have working routes to reach a prefix, and forward packets to internal routers that do not run BGP and do not have a route to reach the prefix.

Routing Information Base (RIB) A term referring to the IP routing table.

routing protocol A set of messages and processes with which routers can exchange information about routes to reach subnets in a particular network. Examples of routing protocols include Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).

Routing Table Manager A component of IOS that manages the process of adding IP routes to the IP routing table. RTM considers routes from all routing sources (static, connected, routing protocols) and chooses the best route to add for a given prefix/length.

RTP *See Reliable Transport Protocol.*

RTTMON MIB A MIB used by the IP SLA feature to collect data generated by IP SLA.

secondary IP address The second (or more) IP address configured on a router interface using the secondary keyword on the **ip address** command.

Secure Sockets Layer (SSL) A security protocol integrated into commonly used web browsers that provides encryption and authentication services between the browser and a website.

seed metric When redistributing routes, the metric set for routes injected into another routing protocol.

segment (multiple definitions) 1) In TCP, a term used to describe a TCP header and its encapsulated data (also called an L4PDU). 2) Also in TCP, the set of bytes formed when TCP breaks a large chunk of data given to it by the application layer into smaller pieces that fit into TCP segments. 3) In Ethernet, either a single Ethernet cable or a single collision domain (no matter how many cables are used).

sequence number (OSPF) In OSPF, a number assigned to each LSA, ranging from 0x80000001 and wrapping back around to 0x7FFFFFFF, which to determines which LSA is most recent.

Service-Oriented Network Architecture (SONA) A robust open framework for building Unified Communications products.

shared key A reference to a security key whose value is known by both the sender and receiver.

Shortest Path First (SPF) The name of the algorithm OSPF uses to analyze the LSDB. The analysis determines the best (lowest cost) route for each prefix/length.

SIA-query An EIGRP Hello specially used halfway through a router's active timer for a route in which a router queries the downstream neighbor to discover if that neighbor is still working.

single homed Refers to a particular type of design between an Enterprise and the Internet in which only one ISP is used with a single link to that ISP.

single multihomed Refers to a particular type of design between an Enterprise and the Internet in which more than one ISP is used with one link to each ISP.

site prefix In IPv6, the prefix that describes a public globally unique IPv6 address block that has been assigned to an end-user organization (for example, an Enterprise or government agency). The assignment typically is made by an ISP or Internet registry.

SLA Operation A configuration construct used by the IP SLA feature inside router IOS that defines a type of packet to be sent, plus a set of measurements to be made about the packet. (Did a reply occur? What delay occurred, jitter, and so on?)

SLSM Static-length subnet mask. The usage of the same subnet mask for all subnets of a single Class A, B, or C network.

Smoothed Round-Trip Time With EIGRP, a purposefully slowly changing measurement of round-trip time between neighbors from which the EIGRP RTO is calculated.

socket A three-tuple consisting of an IP address, port number, and transport layer protocol. TCP connections exist between a pair of sockets.

soft reconfiguration A BGP process by which a router reapplies routing policy configuration (route maps, filters, and the like) based on stored copies of sent and received BGP Updates.

solicited node multicast In IPv6, an address used in the neighbor discovery (ND) process. The format for these addresses is FF02::1:FF00:0000/104, and each IPv6 host must join the corresponding group for each of its unicast and anycast addresses.

SONA *See Service-Oriented Network Architecture.*

SPF calculation The process of running the SPF algorithm against the OSPF LSDB, with the result being the determination of the current best route(s) to each subnet.

split horizon Instead of advertising all routes out a particular interface, the routing protocol omits the routes whose outgoing interface field matches the interface out which the update would be sent.

SSL *See Secure Sockets Layer.*

standard access list A list of IOS global configuration commands that can match only a packet's source IP address for the purpose of deciding which packets to discard and which to allow through the router.

stateful autoconfiguration A method of obtaining an IPv6 address that uses DHCPv6. *See also stateless autoconfiguration.*

stateful DHCP A term used in IPv6 to contrast with stateless DHCP. Stateful DHCP keeps track of which clients have been assigned which IPv6 addresses (state information).

stateless autoconfiguration A method used by an IPv6 host to determine its own IP address, without DHCPv6, by using Neighbor Discovery Protocol (NDP) and the modified EUI-64 address format. *See also stateful autoconfiguration.*

stateless DHCP A term used in IPv6 to contrast with stateful DHCP. Stateless DHCP servers don't lease IPv6 addresses to clients. Instead, they supply other useful information, such as DNS server IP addresses, but with no need to track information about the clients (state information).

static default route A default route configured in IOS using the **ip route** command.

Static Length Subnet Masking A strategy for subnetting a classful network for which all masks/prefixes are the same value for all subnets of that one classful network.

stub area An OSPF area into which external (Type 5) LSAs are not introduced by its ABRs; instead, the ABRs originate and inject default routes into the area.

stub network (OSPF) A network/subnet to which only one OSPF router is connected.

stub router (EIGRP) A router that should not be used to forward packets between other routers. Other routers will not send Query messages to a stub router.

stub router (OSPF) A router that should either permanently or temporarily not be used as a transit router. Can wait a certain time after OSPF process starts, or after BGP notifies OSPF that BGP has converged, before ceasing to be a stub router.

stubby area The same as *stub area*. *See stub area*.

stuck-in-active The condition in which a route has been in an EIGRP active state for longer than the router's Active timer.

subinterface One of the virtual interfaces on a single physical interface.

subnet A subdivision of a Class A, B, or C network, as configured by a network administrator. Subnets allow a single Class A, B, or C network to be used and still allow for a large number of groups of IP addresses, as is required for efficient IP routing.

subnet broadcast address A single address in each subnet for which packets sent to this address will be broadcast to all hosts in the subnet. It is the highest numeric value in the range of IP addresses implied by a subnet number and prefix/mask.

subnet prefix In IPv6, a term for the prefix that is assigned to each data link, acting like a subnet in IPv4.

subnet zero When subnetting a Class A, B, or C address, the subnet for which all subnet bits are binary 0.

subordinate route A term used in this book to refer to routes whose address range sits inside a large range that is advertised as a summary route.

successor In EIGRP, the route to reach a subnet that has the best metric and should be placed in the IP routing table.

successor route With EIGRP, the route to each destination for which the metric is the lowest of all known routes to that network.

Summary LSA In OSPF, a Type 3 LSA. *See Type 3 LSA*.

summary route A route that is created to represent one or more smaller component routes, typically to reduce the size of routing and topology tables.

sync An abbreviation of synchronization; also the command that enables BGP synchronization. *See synchronization*.

synchronization In BGP, a feature in which BGP routes cannot be considered to be a best route to reach an NLRI unless that same prefix exists in the router's IP routing table as learned via some IGP.

synchronous The imposition of time ordering on a bit stream. Practically, a device tries to use the same speed as another device on the other end of a serial link. However, by examining transitions between voltage states on the link, the device can notice slight variations in the speed on each end and can adjust its speed accordingly.

Time-To-Live A field in the IP header that is decremented at each pass through a Layer 3 forwarding device.

topology database The structured data that describes the network topology to a routing protocol. Link-state and balanced hybrid routing protocols use topology tables, from which they build the entries in the routing table.

ToS byte *See Type of Service byte.*

totally NSSA area A type of OSPF NSSA area for which neither external (Type 5) LSAs are introduced, nor Type 3 summary LSAs; instead, the ABRs originate and inject default routes into the area. External routes can be injected into a totally NSSA area.

totally stubby area A type of OSPF stub area for which neither external (Type 5) LSAs are introduced, nor Type 3 summary LSAs; instead, the ABRs originate and inject default routes into the area. External routes cannot be injected into a totally stubby area.

tracking object A concept in IOS that analyzes different conditions on a router that results in the object's state either being up or down. IOS can then use different features, or not use different features, based on the current state of the tracking object. (In this book, tracking objects watch IP SLA operations and influence static routes and policy-based routing.)

transit area The area over which an OSPF virtual link's messages flow.

transit AS With BGP, an AS that receives packets from one neighboring AS and forwards the packet to yet another AS. An Enterprise typically does not want to be a transit AS.

transit network (OSPF) A network/subnet over which two or more OSPF routers have become neighbors, thereby able to forward packets from one router to another across that network.

transit router (OSPF) A router that is allowed to receive a packet from an OSPF router and then forward the packet to another OSPF router.

Transitive PA A description of a BGP PA, meaning that the PA can and should transit over multiple ASNs.

triggered updates A routing protocol feature for which the routing protocol sends routing updates immediately upon hearing about a changed route, even though it may normally only send updates on a regular update interval.

TTL *See Time-To-Live.*

tunnel A method of taking one packet and encapsulating it another packet so that the original encapsulated packet can be delivered across another network–in some cases across networks through which the original packet could not have been forwarded. The tunnel might simply provide for packet delivery, and it might add other services such as encryption and authentication.

tunnel interface In IOS, a software interface used as a configuration construct to configure a tunnel.

tunneling The process of using a tunnel. *See tunnel*.

two-way redistribution With route redistribution, the process of redistributing routes from one routing protocol into a second routing protocol and vice versa.

two-way state In OSPF, a neighbor state that implies that the router has exchanged Hellos with the neighbor and all required parameters match.

Type 1 LSA An OSPF LSA type that describes a router. It lists the router's OSPF ID, its interfaces, their states, and the Link State IDs of neighboring LSAs.

Type 2 LSA An OSPF LSA type that describes a multiaccess network on which a DR has been elected and for which at least one other router connects. The LSA represents the subnet. Also called a network LSA.

Type 3 LSA An OSPF LSA type that describes a subnet in another area. Also called a summary LSA.

Type 3 LSA Filtering The process of causing an ABR to not create and flood a Type 3 LSA into another area.

Type 4 Summary ASBR LSA An LSA type used to describe an ASBR and the cost to reach that ASBR for the purpose of allowing routers to determine the OSPF cost to reach an external subnet advertised as a Type 5 or Type 7 LSA. Also called an ASBR summary LSA.

Type 5 External LSA An LSA type that describes an external subnet as advertised into OSPF by an ASBR. Also called an external LSA.

Type 7 AS External LSA An LSA type that describes an external subnet as injected into an NSSA area.

Type of Service byte A 1-byte field in the IP header, originally defined by RFC 791 for QoS marking purposes.

U/L bit The second most significant bit in the most significant byte of an Ethernet MAC address, a value of binary 0 implies that the address is a Universally Administered Address (UAA) (also known as Burned-In Address [BIA]), and a value of binary 1 implies that the MAC address is a locally configured address.

unequal-cost load balancing A feature of EIGRP in which EIGRP includes multiple routes for the same prefix in the IP routing table but with IOS forwarding packets proportionally based on the calculated integer metric for each route.

unicast MAC address Ethernet MAC address that represents a single NIC or interface.

unique local address A type of IPv6 unicast address meant as a replacement for IPv4 private addresses.

Update (EIGRP) An EIGRP message that informs neighbors about routing information. Update messages require an Ack.

Update Source (BGP) In BGP, a reference to the IP address used as the source address of packets that hold BGP messages. The Update source can differ from neighbor to neighbor and is important in that a BGP router may set a route's NEXT_HOP PA to its Update Source IP address.

update timer The time interval that regulates how often a routing protocol sends its next periodic routing updates. Distance vector routing protocols send full routing updates every update interval.

Variable-Length Subnet Masking A strategy for subnetting a classful network for which masks/prefixes are different for some subnets of that one classful network.

variance An integer setting for EIGRP. Any FS route whose metric is less than this variance multiplier times the successor's metric is added to the routing table, within the restrictions of the **maximum-paths command**.

virtual circuit A logical concept that represents the path over which frames travel between DTEs. VCs are particularly useful when comparing Frame Relay to leased physical circuits.

virtual link With OSPF, the encapsulation of OSPF messages inside IP to a router with which no common subnet is shared for the purpose of either mending partitioned areas or providing a connection from some remote area to the backbone area.

Virtual Private LAN Service (VPLS) Ethernet-like service that provides connectivity between two or more endpoints, typically using Ethernet over MPLS (EoMPLS) technology.

virtual private network (VPN) A set of security protocols that, when implemented by two devices on either side of an unsecure network such as the Internet, can enable the devices to send data securely. VPNs provide privacy, device authentication, antireplay services, and data integrity services.

Virtual Private Wire Service (VPWS) Ethernet-like service that provides connectivity between exactly two endpoints, typically using Ethernet over MPLS (EoMPLS) technology.

VLSM *See Variable-Length Subnet Masking.*

VLSM Variable-Length Subnet Mask(ing). The ability to specify a different subnet mask for the same Class A, B, or C network number on different subnets. VLSM can help optimize available address space.

VoIP Voice over IP. The transport of voice traffic inside IP packets over an IP network.

VPN *See virtual private network.*

VPN client Software that resides on a PC, often a laptop, so that the host can implement the protocols required to be an endpoint of a VPN.

WAN Edge Same as Enterprise Edge. *See Enterprise Edge*.

weight A local Cisco-proprietary BGP setting that is not advertised to any peers. A larger value is considered to be better.

well-known discretionary A characterization of a BGP path attribute in which all BGP implementations must support and understand the attribute (well known), but BGP Updates can either include the attribute or not, depending on whether a related feature has been configured (discretionary).

well-known mandatory A characterization of a BGP path attribute in which all BGP implementations must support and understand the attribute (well known), and all BGP Updates must include the attribute (mandatory).

well-known PA *See well-known mandatory and well-known discretionary*.

zero subnet For every classful IPv4 network that is subnetted, the one subnet whose subnet number has all binary 0s in the subnet part of the number. In decimal, the 0 subnet can be easily identified because it is the same number as the classful network number.