



CCIE Professional Development

Routing TCP/IP

Volume I, Second Edition

A detailed examination of interior routing protocols

No. 1
Bestseller
Fully Updated

ciscopress.com

Jeff Doyle, CCIE® No. 1919
Jennifer Carroll, CCIE No. 1402

FREE SAMPLE CHAPTER



SHARE WITH OTHERS



CCIE Professional Development Routing TCP/IP Volume I

Second Edition

Jeff Doyle, CCIE No. 1919
Jennifer Carroll, CCIE No. 1402

Cisco Press

800 East 96th Street
Indianapolis, Indiana 46240 USA

CCIE Professional Development Routing TCP/IP

Volume I

Second Edition

Jeff Doyle, CCIE No. 1919, Jennifer Carroll, CCIE No. 1402

Copyright © 2006 Cisco Systems, Inc.

Published by:

Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

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

Printed in the United States of America 11 12 13 14 15

Eleventh Printing January 2012

Library of Congress Cataloging-in-Publication Number: 2004104363

ISBN: 1-58705-202-4

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.

Warning and Disclaimer

This book is designed to provide information about routing TCP/IP. 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.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales.

For more information please contact: **U.S. Corporate and Government Sales** 1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the U.S. 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 e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher	John Wait
Editor-in-Chief	John Kane
Executive Editor	Brett Bartow
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Jeff Brady
Production Manager	Patrick Kanouse
Development Editor	Andrew Cupp
Senior Project Editor	San Dee Phillips
Copy Editor	Interactive Composition Corporation
Technical Editors	Frank Knox, Steven Edward Moore, Rena Yang
Editorial Assistant	Tammi Barnett
Book and Cover Designer	Louisa Adair
Composition	Interactive Composition Corporation
Indexer	Tim Wright



Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the

Cisco.com Web site at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic
Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy
Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal
Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden
Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Copyright © 2003 Cisco Systems, Inc. All rights reserved. CCIP, CCSP, the Cisco Arrow logo, the Cisco *Powered* Network mark, the Cisco Systems Verified logo, Cisco Unity, Follow Me Browsing, FormShare, iQ Net Readiness Scorecard, Networking Academy, and ScriptShare are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, The Fastest Way to Increase Your Internet Quotient, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, EnterpriseSolve, EtherChannel, EtherSwitch, Fast Step, GigaStack, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, LightStream, MGX, MICA, the Networkers logo, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, SlideCast, SMARTnet, StrataView Plus, Stratm, SwitchProbe, TeleRouter, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this document or Web site are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0303R)

Printed in the USA

About the Authors

Jeff Doyle (CCIE No. 1919) specializes in IP routing protocols, MPLS, and IPv6. He has designed or assisted in the design of large-scale IP service provider networks throughout North America, Europe, Japan, Korea, and the People's Republic of China. Jeff has presented numerous corporate seminars, and has also spoken at NANOG, JANOG, APRICOT, and at IPv6 Forum conferences. Jeff holds a BA from Memphis State University, and studied Electrical Engineering at the University of New Mexico. Jeff lives in Denver, Colorado.

Jennifer Carroll (CCIE No. 1402) is an independent network consultant in Redmond, Washington. She has designed, implemented, and optimized many TCP/IP networks, and has developed and taught a variety of networking and internetworking courses on routing protocols and Cisco routers over the past 15 years. Jennifer can be contacted at jennifer.carroll@ieee.org.

About the Technical Reviewers

Frank Knox, Chief Technical Officer, has been with Skyline Computer for a little over six years. He is a dual CCIE (CCIE No. 3698: SNA/IP and Routing/Switching) as well as a CCSI. In addition to his CTO responsibilities, Frank teaches several advanced Cisco-related courses, including a one-week CCIE Lab Preparation Workshop. He is considered to be an expert in mainframe attached router technologies and in the technologies and issues associated with integrated networking (for example, SNA/IP and Voice/Data). He has more than 37 years of networking experience with IBM, GTE (Verizon) Directories, and Skyline Computer Corp. This experience includes field service, field support, product planning, management, and all facets of networking education. In addition, he developed and taught several courses for the University of Dallas Telecommunications MBA program. Frank also has an MS degree in Telecommunications from Pace University (4.0 GPA).

After working in various roles as an engineer within Cisco for the past 6.5 years, **Steven Edward Moore** transitioned to the IP Routing Protocol Scalability Team. There, his focus encompasses all aspects of extending network and protocol scalability: considering new features and optimizations to the protocol architecture, designing tests to measure current protocol scalability, working with customers to implement scaling functionality in their network, and participating in events such as Networkers to educate others on how to enhance their network's performance and scalability from the routing perspective.

Rena Yang is a software engineer at Cisco Systems. She has more than six years of experience implementing code in Cisco IOS. She currently works on IS-IS. Before this, she focused on IPv4, UDP, access lists, policy routing, and routing infrastructure. Rena holds a bachelor's of science and masters of engineering in computer science from MIT.

Dedications

I would like to dedicate this book to my wife, Sara, and my children, Anna, Carol, James, and Katherine.
—Jeff

I would like to dedicate this book to my husband, Mike, and sons, Mitchell and Jonathan. Their patience and support helped me complete this book.
—Jennifer

Acknowledgments

Many thanks to Brett Bartow, Chris Cleveland, Andrew Cupp, San Dee Phillips, and all of the staff of Cisco Press who made this book possible.

The technical editors, Steven Moore, Rena Yang and Frank Knox, did a fantastic job. We want to thank them for their outstanding advice and recommendations.

We want to thank Frank Knox, Carl Pike, Chris Tonini, and the rest of the employees of Skylabs networks. Skylabs' lab setup and access to the lab is easy to use and had everything we needed to complete all the configurations and case studies in this book.

Contents at a Glance

Foreword	xxii
Introduction	xxiii
Part I	Routing Basics 3
Chapter 1	TCP/IP Review 5
Chapter 2	IPv6 Overview 49
Chapter 3	Static Routing 81
Chapter 4	Dynamic Routing Protocols 131
Part II	Interior Routing Protocols 167
Chapter 5	Routing Information Protocol (RIP) 169
Chapter 6	RIPv2, RIPv6, and Classless Routing 201
Chapter 7	Enhanced Interior Gateway Routing Protocol (EIGRP) 251
Chapter 8	OSPFv2 335
Chapter 9	OSPFv3 477
Chapter 10	Integrated IS-IS 513
Part III	Route Control and Interoperability 629
Chapter 11	Route Redistribution 631
Chapter 12	Default Routes and On-Demand Routing 677
Chapter 13	Route Filtering 701
Chapter 14	Route Maps 731
Part IV	Appendixes 767
Appendix A	Tutorial: Working with Binary and Hex 769
Appendix B	Tutorial: Access Lists 775
Appendix C	CCIE Preparation Tips 803
Appendix D	Answers to Review Questions 809
Appendix E	Solutions to Configuration Exercises 827
Appendix F	Solutions to Troubleshooting Exercises 881
Index	887

Table of Contents

Foreword xxii

Introduction xxiii

Part I Routing Basics 3

Chapter 1 TCP/IP Review 5

TCP/IP Protocol Layers 5

IP Packet Header 7

IPv4 Addresses 16

First Octet Rule 18

Address Masks 20

Subnets and Subnet Masks 22

Designing Subnets 24

Breaking the Octet Boundary 25

Troubleshooting a Subnet Mask 29

Address Resolution Protocol (ARP) 30

Proxy ARP 35

Gratuitous ARP 36

Reverse ARP 36

Internet Control Message Protocol (ICMP) 37

Host-to-Host Layer 41

TCP 41

UDP 44

Looking Ahead 45

Summary Table: Chapter 1 Command Review 45

Recommended Reading 46

Review Questions 46

Configuration Exercises 47

Troubleshooting Exercises 47

Chapter 2 IPv6 Overview 49

IPv6 Addresses 50

Address Representation 50

IPv6 Address Types 52

Global Unicast Addresses 53

Identifying IPv6 Address Types	54
Local Unicast Addresses	55
Anycast Addresses	55
Multicast Addresses	56
Embedded IPv4 Addresses	58

IPv6 Packet Header Format 58

Extension Headers 60

ICMPv6 63

Neighbor Discovery Protocol 64

NDP Messages 65

Router Discovery 70

Address Autoconfiguration 71

Duplicate Address Detection 73

Neighbor Address Resolution 74

Privacy Addresses 75

Neighbor Unreachability Detection 76

Looking Ahead 77

Review Questions 77

Chapter 3 Static Routing 81

Route Table 81

Configuring Static Routes 84

Case Study: Simple IPv4 Static Routes 85

Case Study: Simple IPv6 Static Routes 90

Case Study: Summary Routes 94

Case Study: Alternative Routes 96

Case Study: Floating Static Routes 97

Case Study: IPv6 Floating Static Routes 100

Case Study: Load Sharing 102

Load Sharing and Cisco Express Forwarding 103

Per Destination Load Sharing and Fast Switching 105

Per Packet Load Sharing and Process Switching 105

Which Switching Method Will Be Used? 106

Case Study: Recursive Table Lookups 107

Troubleshooting Static Routes 109

Case Study: Tracing a Failed Route 109

Case Study: A Protocol Conflict 112

Case Study: A Replaced Router 116

Case Study: Tracing An IPv6 Failed Route 120

Looking Ahead 122

Summary Table: Chapter 3 Command Review	123
Review Questions	124
Configuration Exercises	124
Troubleshooting Exercises	126
Chapter 4 Dynamic Routing Protocols	131
Routing Protocol Basics	131
Path Determination	132
Metrics	133
Hop Count	134
Bandwidth	134
Load	135
Delay	135
Reliability	135
Cost	135
Convergence	135
Load Balancing	137
Distance Vector Routing Protocols	137
Common Characteristics	138
Periodic Updates	138
Neighbors	138
Broadcast Updates	138
Full Routing Table Updates	139
Routing by Rumor	139
Route Invalidation Timers	140
Split Horizon	141
Counting to Infinity	143
Triggered Updates	144
Holddown Timers	144
Asynchronous Updates	145
Link State Routing Protocols	146
Neighbors	147
Link State Flooding	147
Sequence Numbers	148
Aging	153
Link State Database	154
SPF Algorithm	156
Areas	160
Interior and Exterior Gateway Protocols	160
Static or Dynamic Routing?	162

Looking Ahead	164
Recommended Reading	164
Review Questions	164

Part II Interior Routing Protocols 167

Chapter 5 Routing Information Protocol (RIP) 169

Operation of RIP	170
RIP Timers and Stability Features	170
RIP Message Format	172
Request Message Types	174
Classful Routing	175
Classful Routing: Directly Connected Subnets	176
Classful Routing: Summarization at Boundary Routers	176
Classful Routing: Summary	178
Configuring RIP	178
Case Study: A Basic RIP Configuration	178
Case Study: Passive Interfaces	181
Case Study: Configuring Unicast Updates	182
Case Study: Discontiguous Subnets	184
Case Study: Manipulating RIP Metrics	187
Case Study: Minimizing the Impact of Updates	190
Troubleshooting RIP	193
Looking Ahead	193
Summary Table: Chapter 5 Command Review	193
Recommended Reading	194
Review Questions	194
Configuration Exercises	194
Troubleshooting Exercises	196

Chapter 6 RIPv2, RIPv6, and Classless Routing 201

Operation of RIPv2	201
RIPv2 Message Format	202
Compatibility with RIPv1	205
Classless Route Lookups	205
Classless Routing Protocols	206
Variable-Length Subnet Masking	206
Authentication	210

Operation of RIPng	212
Configuring IPv2	214
Case Study: A Basic IPv2 Configuration	215
Case Study: Compatibility with IPv1	215
Case Study: Using VLSM	218
Case Study: Discontiguous Subnets and Classless Routing	220
Case Study: Authentication	222
Configuring IPvng	224
Case Study: Basic IPvng Configuration	224
Case Study: IPvng Process Customization	232
Case Study: Metric Manipulation	233
Case Study: Route Summarization	235
Troubleshooting IPv2 and IPvng	236
Case Study: Misconfigured VLSM	236
Looking Ahead	242
Summary Table: Chapter 6 Command Review	242
Recommended Reading	244
Review Questions	244
Configuration Exercises	244
Troubleshooting Exercises	246
Chapter 7 Enhanced Interior Gateway Routing Protocol (EIGRP)	251
The Roots of EIGRP: An Overview of IGRP	251
Process Domains	252
IGRP Timers and Stability Features	254
IGRP Metrics	255
From IGRP to EIGRP	261
Operation of EIGRP	263
Protocol-Dependent Modules	263
Reliable Transport Protocol	264
Neighbor Discovery/Recovery	265
Diffusing Update Algorithm	266
DUAL: Preliminary Concepts	267
DUAL Finite State Machine	273
Diffusing Computation: Example 1	276
Diffusing Computation: Example 2	280

EIGRP Packet Formats	287
EIGRP Packet Header	288
General TLV Fields	289
IP-Specific TLV Fields	290
Address Aggregation	293
EIGRP and IPv6	297
Configuring EIGRP	297
Case Study: A Basic EIGRP Configuration	297
Case Study: Unequal-Cost Load Balancing	299
Case Study: Setting Maximum Paths	302
Case Study: Multiple EIGRP Processes	305
Case Study: Disabling Automatic Summarization	307
Case Study: Stub Routing	309
Case Study: Address Summarization	315
Authentication	316
Troubleshooting EIGRP	317
Case Study: A Missing Neighbor	318
Stuck-in-Active Neighbors	323
Looking Ahead	328
Summary Table: Chapter 7 Command Review	328
Review Questions	329
Configuration Exercises	330
Troubleshooting Exercises	332
Chapter 8 OSPFv2	335
Operation of OSPF	336
Neighbors and Adjacencies	337
Hello Protocol	338
Network Types	339
Designated Routers and Backup Designated Routers	340
OSPF Interfaces	344
OSPF Neighbors	350
Flooding	363
Areas	367
Router Types	369
Partitioned Areas	371
Virtual Links	373
Link-State Database	374
LSA Types	377
Stub Areas	386

Route Table	389
Destination Types	390
Path Types	391
Route Table Lookups	393
Authentication	393
OSPF over Demand Circuits	393
OSPF Packet Formats	395
Packet Header	396
Hello Packet	398
Database Description Packet	399
Link State Request Packet	400
Link State Update Packet	401
Link State Acknowledgment Packet	402
OSPF LSA Formats	402
LSA Header	403
Router LSA	403
Network LSA	406
Network and ASBR Summary LSAs	407
Autonomous System External LSA	408
NSSA External LSA	409
Options Field	410
Configuring OSPF	411
Case Study: A Basic OSPF Configuration	411
Case Study: Setting Router IDs with Loopback Interfaces	414
Case Study: Domain Name Service Lookups	417
Case Study: OSPF and Secondary Addresses	418
Case Study: Stub Areas	423
Case Study: Totally Stubby Areas	426
Case Study: Not-So-Stubby Areas	428
Case Study: Address Summarization	435
Case Study: Filtering Between Areas	441
Case Study: Authentication	441
Case Study: Virtual Links	444
Case Study: OSPF on NBMA Networks	446
Case Study: OSPF over Demand Circuits	454
Troubleshooting OSPF	455
Case Study: An Isolated Area	460
Case Study: Misconfigured Summarization	463
Looking Ahead	466
Summary Table: Chapter 8 Command Review	466

Recommended Reading	469
Review Questions	469
Configuration Exercises	470
Troubleshooting Exercises	472

Chapter 9 OSPFv3 477

Operation of OSPFv3	477
OSPFv3 Differences from OSPFv2	478
OSPFv3 Messages	479
An Overview of OSPFv3 LSAs	481
OSPFv3 LSA Formats	484
The Router LSA	484
Network LSA	486
Inter-Area Prefix LSA	486
Inter-Area Router LSA	487
AS-External LSA	488
Link LSA	489
Intra-Area Prefix LSA	491
Options Field	493
Configuring OSPFv3	494
Case Study: A Basic OSPFv3 Configuration	494
Case Study: Stub Areas	497
Case Study: Multiple Instances on a Link	499
Case Study: OSPFv3 on NBMA Networks	500
Troubleshooting OSPFv3	506
Case Study: Frame-Relay Mapping	506
Looking Ahead	509
Summary Table: Chapter 9 Command Review	509
Recommended Reading	510
Review Questions	510
Configuration Exercises	511

Chapter 10 Integrated IS-IS 513

Operation of Integrated IS-IS	515
IS-IS Areas	516
Network Entity Titles	519
IS-IS Functional Organization	521
Subnetwork Dependent Functions	521
Subnetwork Independent Functions	526

IS-IS PDU Formats	534
TLV Fields	536
IS-IS Hello PDU Format	539
IS-IS Link State PDU Format	546
IS-IS Sequence Numbers PDU Format	552
Extensions to IS-IS	554
3-Way Handshaking	555
Domain-Wide Prefix Distribution	556
Wide Metrics	558
Routing IPv6 with IS-IS	560
Dynamic Hostname Exchange	562
Multiple Topologies	562
Mesh Groups	566
Flooding Delays	571
Improving SPF Efficiency	573
Configuring Integrated IS-IS	574
Case Study: A Basic IPv4 Integrated IS-IS Configuration	574
Case Study: Changing the Router Types	579
Case Study: An Area Migration	581
Case Study: Route Summarization	585
Case Study: Authentication	587
Case Study: A Basic Integrated IS-IS Configuration for IPv6	591
Case Study: Transition to Multiple Topology Mode	596
Case Study: Route Leaking Between Levels	598
Case Study: Multiple L1 Areas Active on A Router	604
Troubleshooting Integrated IS-IS	606
Troubleshooting IS-IS Adjacencies	607
Troubleshooting the IS-IS Link-State Database	610
Case Study: Integrated IS-IS on NBMA Networks	615
Looking Ahead	619
Summary Table: Chapter 10 Command Review	619
Review Questions	623
Configuration Exercises	624
Troubleshooting Exercises	626

Part III Route Control and Interoperability 629

Chapter 11 Route Redistribution 631

Principles of Redistribution	632
Metrics	633

Administrative Distances	633
Redistributing from Classless to Classful Protocols	639
Configuring Redistribution	642
Case Study: Redistributing IGRP and RIP	645
Case Study: Redistributing EIGRP and OSPF	646
Case Study: Redistribution and Route Summarization	651
Case Study: Redistributing OSPFv3 and RIPng	657
Case Study: Redistributing IS-IS and RIP/RIPng	663
Case Study: Redistributing Static Routes	669
Looking Ahead	672
Summary Table: Chapter 11 Command Review	672
Review Questions	673
Configuration Exercises	673
Troubleshooting Exercises	674
Chapter 12 Default Routes and On-Demand Routing	677
Fundamentals of Default Routes	677
Fundamentals of On-Demand Routing	680
Configuring Default Routes and ODR	681
Case Study: Static Default Routes	682
Case Study: The Default-Network Command	687
Case Study: The Default-Information Originate Command	691
Case Study: Configuring On-Demand Routing	697
Looking Ahead	698
Summary Table: Chapter 12 Command Review	698
Review Questions	699
Chapter 13 Route Filtering	701
Configuring Route Filters	702
Case Study: Filtering Specific Routes	703
Case Study: Route Filtering and Redistribution	708
Case Study: A Protocol Migration	711
Case Study: Multiple Redistribution Points	716
Case Study: Using Administrative Distances to Set Router Preferences	722
Looking Ahead	724
Summary Table: Chapter 13 Command Review	724

Configuration Exercises 724

Troubleshooting Exercises 727

Chapter 14 Route Maps 731

Basic Uses of Route Maps 731

Configuring Route Maps 734

Case Study: Policy Routing 737

Case Study: Policy Routing and Quality of Service Routing 744

Case Study: Route Maps and Redistribution 747

Case Study: Route Tagging 752

Case Study: Filtering Tagged Routes Out of OSPF Route Table 757

Case Study: IPv6 Redistribution with Route Maps 759

Looking Ahead 761

Summary Table: Chapter 14 Command Review 761

Review Questions 763

Configuration Exercises 763

Troubleshooting Exercise 764

Part IV Appendixes 767

Appendix A Tutorial: Working with Binary and Hex 769

Appendix B Tutorial: Access Lists 775

Appendix C CCIE Preparation Tips 803

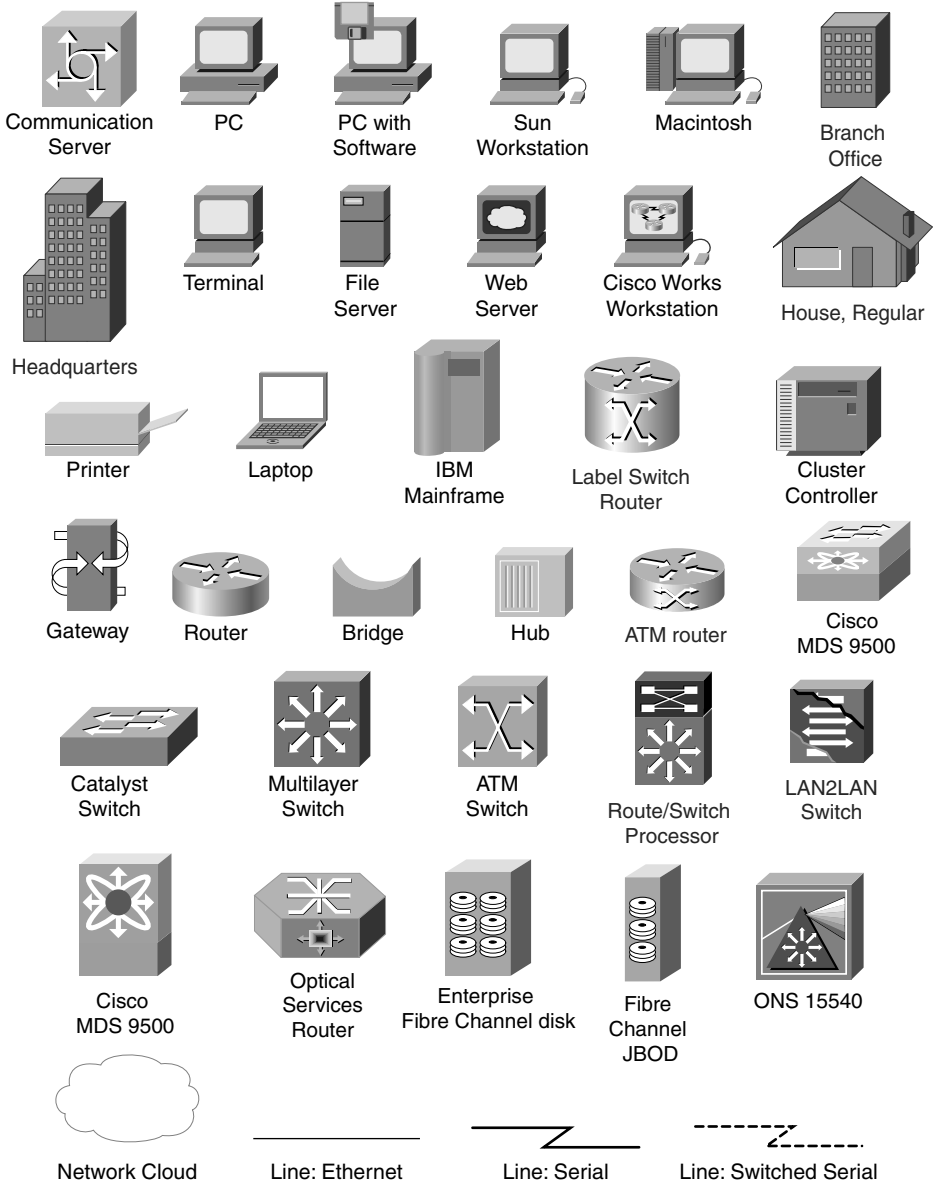
Appendix D Answers to Review Questions 809

Appendix E Solutions to Configuration Exercises 827

Appendix F Solutions to Troubleshooting Exercises 881

Index 887

Icons Used in This Book



Command Syntax Conventions

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

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

Foreword

In 1976, when I saw my first Arpanet IMP at Digital Equipment Corporation, networks as we know them today were in their infancy. SNA, XNS, and DECnet were under early development, and packet switching versus circuit switching was the hot topic of the day. Those of us involved in the design of the switching and routing algorithms were dealing with routers (although we didn't call them that) that had 64 kilobytes of memory, data link of 56 kilobits were considered blindingly fast, and networks with 256 nodes were big enough that if you were the salesman who sold those 256 computers, you would retire fabulously wealthy.

Thirty years is a long time, and today the individual networks that make up the Internet contain thousands or tens of thousands of nodes, while the Internet as a whole contains hundreds of millions of computers. Most striking in the evolution over this human generation is that the foundations of the Internet laid down in the TCP/IP protocol suite have survived mostly intact through four or more generations of computing architectures, three complete generations of operating system technology, and an increase of five orders of magnitude in transmission speeds.

Yet, we still treat routing in packet-switched networks as a black art. Why is that?

First, designing robust, scalable distributed algorithms is hard. Despite our best intentions to make them simple, complexity creeps in to deal with the inevitable special cases, optimizations, peculiar topologies, and link technologies one encounters. Because a “fork lift upgrade” of an entire network is rarely feasible, we have multiple generations of technology present simultaneously, and we must maintain backward-compatibility with essentially no disruption to deployed services. As policies governing the routing of packets become more sophisticated, our ability to devise automated discovery and configuration procedures gets overwhelmed, and we fall back on manual configuration and performance tuning techniques. Finally, as the environment in which these networks are operated has evolved from a cooperative one where trust was implicit to one in which the network is subject to both inside and outside attack, designing and deploying routing systems that can be made secure has become an urgent priority.

Routing TCP/IP tackles this black art comprehensively. The present Volume 1 covers all the needed fundamentals of TCP/IP networks and gives you all the tools needed to understand how routing is accomplished within a single administrative region of the Internet. Straightforward ideas of packet-switched routing are presented first in the chapters on addressing and static routing. The most popular IGP—RIP, EGRP, OSPF, and ISIS—are covered in depth. Advanced topics in route redistribution, route filtering, and policy routing round out Volume 1.

This second edition also adds essential material on IPv6 as well as bringing all the material up to date with examples and configurations for the latest releases of Cisco IOS.

For anyone wanting a comprehensive understanding of how routing in TCP/IP networks really works, from the design principles of routing algorithms, to the evolution of addressing schemes, to the practical aspects of designing and configuring the routing of large autonomous systems, this is the book for you.

David Oran
Cisco Fellow

Introduction

Routing is an essential element of all but the smallest data communications networks. At one level, routing and the configuration of routers are quite simple. But as networks grow in size and complexity, routing issues can become at once both large and subtle. Perversely, perhaps, we are grateful for the difficult problems large-scale routing can present—as network systems consultants, these problems are our bread and butter. Without them, the phrase “You want fries with that?” could be an unfortunate part of our daily vocabulary.

Cisco Certified Internetwork Experts are widely recognized for their ability to design, troubleshoot, and manage large networks. This recognition comes from the fact that you cannot become a CCIE by attending a few classes and then regurgitating some memorized facts onto a written test. A CCIE has proven expertise in an intense, famously difficult hands-on lab exam.

Objectives

This book is the first of two volumes that focuses on TCP/IP routing issues. Early in the writing of the first edition, Kim Lew, former Cisco Systems program manager, said, “Our objective is to make CCIEs, not to make people who can pass the CCIE lab.” We entirely agree with that statement and have used it as a guiding principle throughout the writing of this book. Although the book includes many case studies and exercises to help you prepare for the CCIE lab, my primary objective is to increase your understanding of IP routing—both on a generic level and as it is implemented on Cisco routers.

Audience

The audience for this book is any network designer, administrator, or engineer who needs a full understanding of the interior routing protocols of TCP/IP. Although the practical aspects of the book focus on the Cisco IOS, the information is applicable to any routing platform.

The book is not only for readers who plan to become CCIEs, but for people who wish to advance their knowledge of TCP/IP routing. These readers will fall into one of three categories:

- The “beginners” who have some basic networking knowledge and wish to begin a deep study of networking.
- The intermediate-level networking professionals who have experience with routers, Cisco or otherwise, and plan to advance that experience to the expert level.
- The highly experienced networking experts. These individuals have extensive hands-on expertise with Cisco routers and are ready to take the CCIE lab; however, they want a structured review and series of exercises for verification and validation.

CCIE Professional Development: Routing TCP/IP, Volume I focuses primarily on intermediate-level networking professionals while offering to beginners a structured outline of fundamental information and to experts the required challenges to hone their skills.

Changes from First Edition

There are several factors influencing the changes contained in this second edition. The first factor is the CCIE itself. When I (Jeff) wrote the first edition of this book, the CCIE—specifically what is now

called the Routing and Switching specialty of the CCIE—was the only certification Cisco Systems offered. Now, there is a series of certifications creating a path to the CCIE at the pinnacle. Moreover, the typical networking professional is more knowledgeable than in 1997. Given this, we have eliminated the first chapter of the original book, which covered such very basic concepts as the definition of bridges and routers and network addresses. (When was the last time you even saw a bridge in a network?)

The second factor influencing the changes in this edition is the changes in the Cisco Systems IOS. IGRP, which was frequently used when the first edition was written, is now a legacy protocol whose main significance is as the ancestor of EIGRP. Therefore the IGRP chapter of the first edition has been eliminated and IGRP is covered for historical perspective early in the EIGRP chapter. The IOS command suite itself has expanded to accommodate new functions and options; we have made every effort to include the commands and protocol extensions that did not exist in the late 1990s.

Lastly, a protocol that existed mostly only in proposal form in 1997—IPv6—is now in the early stages of worldwide deployment. You can expect to need a detailed knowledge of this protocol and the extensions to IP routing protocols that support it in the near future, if not already, so this second edition delves deeply into routing IPv6.

Other changes in this edition are semantic. For example, in the first edition, I (Jeff) made a point of differentiating between a “network” as a data link and an “internetwork” as a set of networks connected by routers. Although that terminology is certainly accurate, it is clumsy, and “internetwork” is seldom used these days. Instead, “network” usually refers to everything from a local link to worldwide autonomous systems operated by the likes of Level 3, NTT, and Sprint. We have attempted to bring the terminology in this edition up to modern, common usage.

Organization

The 14 chapters of the book are divided into three parts.

Part I, “Routing Basics,” examines the basics of IPv4 and IPv6, and the basics of routing. Although more advanced readers may wish to skip the first chapter, we recommend that they at least skim Chapter 3, “Static Routing,” and Chapter 4, “Dynamic Routing Protocols.” And, of course, if you are not yet familiar with IPv6, Chapter 2, “IPv6 Overview,” is a must-read.

Part II, “Interior Routing Protocols,” covers the IP Interior Gateway Protocols. Each protocol-specific chapter begins with a discussion of the theory, mechanics, and parameters of the protocol. This general overview is followed by case studies on configuring and troubleshooting the protocol using Cisco Systems’ IOS in various network topologies.

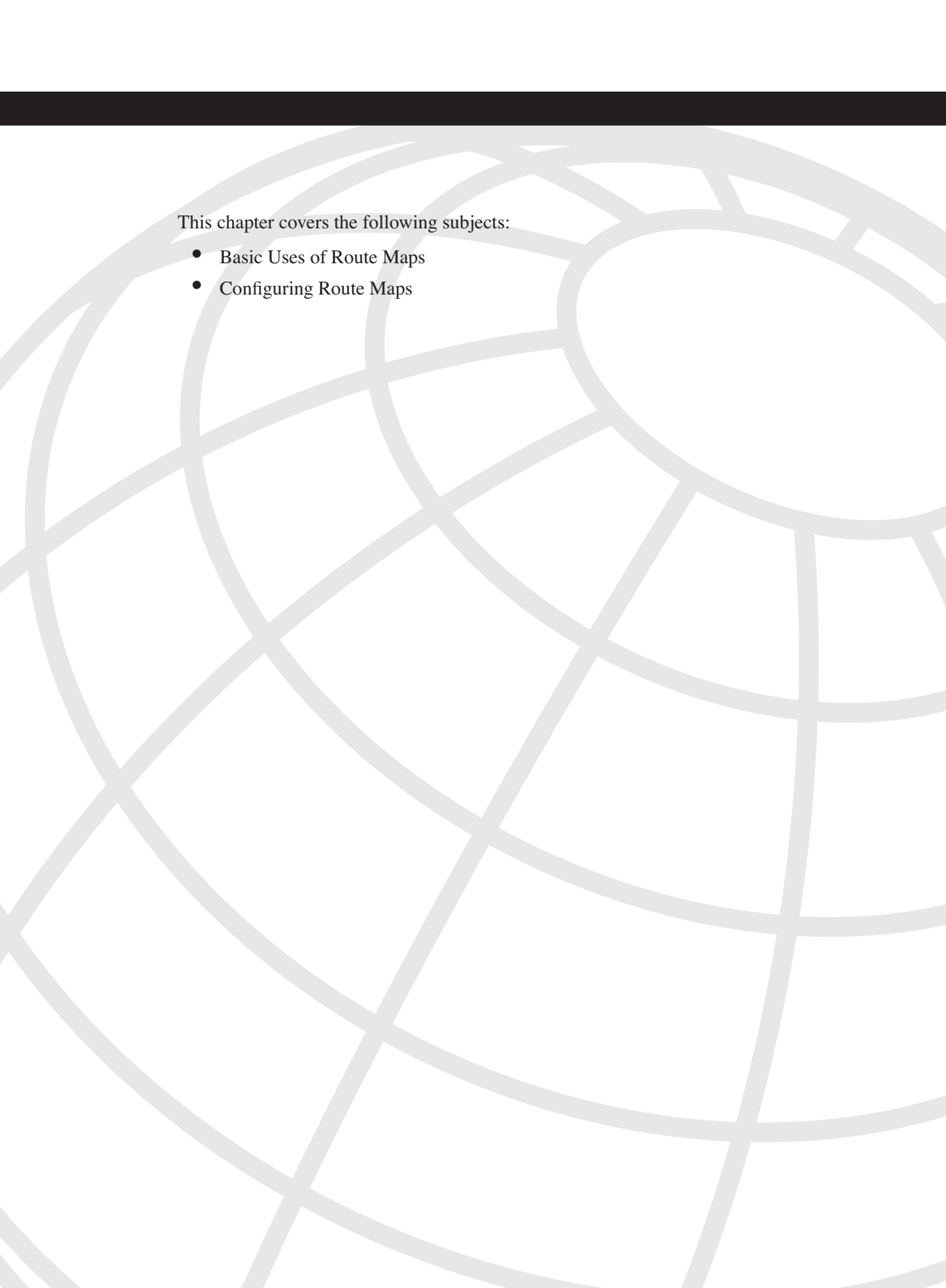
The Exterior Gateway Protocol, BGP, and topics such as multicast routing, Quality of Service, router security and management, and Network Address Translation, are covered in “Routing TCP/IP, Volume II.”

Part III, “Route Control and Interoperability,” examines the tools available for creating and managing interoperability with multiple IP routing protocols, and also such tools as default routes and route filtering. As such, the chapters of this last part provide an introduction to the tools necessary for building the complex routing policies introduced in Volume II. These chapters, like the ones in Part II, begin with concepts and conclude with case studies.

Book Features

Most chapters conclude with a set of review questions, configuration exercises, and troubleshooting exercises. The review questions focus on the theoretical aspects of the chapter topic, whereas the configuration and troubleshooting exercises address Cisco-specific aspects of the chapter topic.

Also at the end of each chapter is a table with a brief description of all important Cisco IOS commands used in that chapter. The conventions used to present these commands are the same conventions used in the IOS Command Reference and presented earlier in this introduction.



This chapter covers the following subjects:

- Basic Uses of Route Maps
- Configuring Route Maps

Route Maps

Route maps are similar to access lists; they both have criteria for matching the details of certain packets and an action of permitting or denying those packets. Unlike access lists, though, route maps can add to each “match” criterion a “set” criterion that actually changes the packet in a specified manner, or changes route information in a specified manner. Route maps also allow you more options for matching a given packet. Altogether, route maps are a powerful tool for creating customized routing policies in your network.

Basic Uses of Route Maps

Route maps can be used for both redistribution and for policy routing. Although redistribution has been covered extensively in previous chapters, this chapter introduces the topic of policy routing. And most commonly, they are an essential tool in large-scale Border Gateway Protocol (BGP) implementations. The use of route maps to implement BGP routing policies is outside the scope of this volume, and is covered in “Routing TCP/IP,” Volume II (CCIE Professional Development) (1-57870-089-2).

Policy routes are nothing more than sophisticated static routes. Whereas static routes forward a packet to a specified next hop based on the destination address of the packet, policy routes can forward a packet to a specified next hop based on the source of the packet or other fields in the packet header. Policy routes can also be linked to extended IP access lists so that routing might be based on such things as protocol types and port numbers. Like a static route, a policy route influences the routing only of the router on which it is configured.

Figure 14-1 shows an example of a typical policy routing application. AbnerNet is connected to two Internet service providers via router Dogpatch. AbnerNet’s corporate policy dictates that some users’ Internet traffic should be sent via ISP 1 and other users’ Internet traffic should be sent via ISP 2. If either ISP should become unavailable, the traffic normally using that provider would be sent to the other provider. A policy route at Dogpatch can distribute Internet traffic in accordance with local policy. The distribution of traffic might be based on subnet, specific user, or even user applications.

Figure 14-1 Policy routing allows traffic from AbnerNet to be routed to one of its two Internet service providers based on parameters such as source address, source/destination address combinations, packet size, or application-level ports.

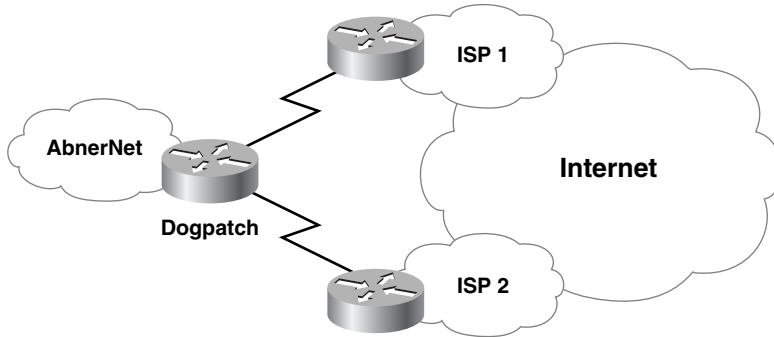
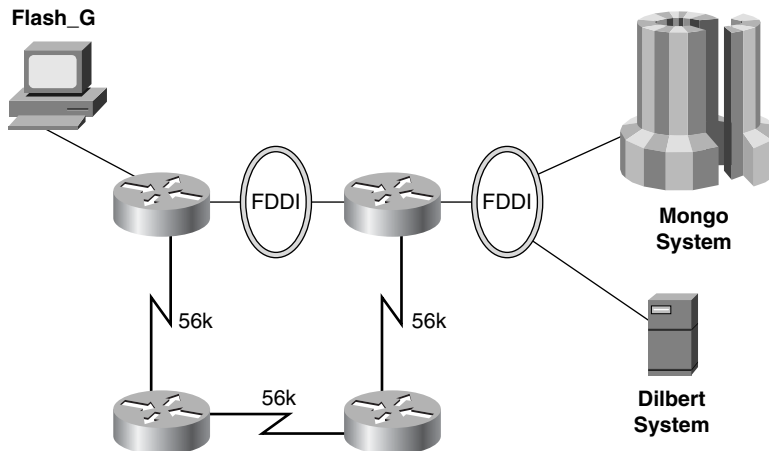


Figure 14-2 shows another use for policy routing. One of the systems on the right watches for invasion forces from the planet Mongo, while the other stores back copies of *Dilbert* comic strips. Policy routes can be configured to route the critical traffic from the Mongo System to Flash_G over the FDDI (Fiber Distributed Data Interface) link and to route the lower-priority Dilbert traffic over the 56K links—or vice versa.

Figure 14-2 Policy routing allows high-priority traffic from the Mongo System to be routed over the FDDI link while low-priority traffic from the Dilbert System is routed over the 56K links.



Tables 14-1 and 14-2 show the **match** and **set** commands that can be used with redistribution, and Tables 14-3 and 14-4 show the **match** and **set** commands that can be used with policy routing.

Table 14-1 *match commands that can be used with redistribution.*

Command	Description
match interface <i>type number</i> [... <i>type number</i>]	Matches routes that have their next hop out one of the interfaces specified
match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Matches routes that have a destination address specified by one of the access lists
match ip next-hop { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Matches routes that have a next-hop router address specified by one of the access lists
match ip route-source { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Matches routes that have been advertised by routers at the addresses specified in the access lists
match metric <i>metric-value</i>	Matches routes with the specified metric
match route-type { internal external [type-1 type-2] level-1 level-2 }	Matches OSPF, EIGRP, or IS-IS routes of the specified type
match tag <i>tag-value</i> [... <i>tag-value</i>]	Matches routes with the specified tags

Table 14-2 *set commands that can be used with redistribution.*

Command	Description
set level { level-1 level-2 level-1-2 stub-area backbone }	Sets the IS-IS level or the OSPF area into which a matched route is to be redistributed
set metric { <i>metric-value</i> <i>bandwidth delay reliability loading mtu</i> }	Sets the metric value for a matched route
set metric-type { internal external type-1 type-2 }	Sets the metric type for a matched route being redistributed into IS-IS or OSPF
set next-hop <i>next-hop</i>	Sets the next-hop router address for a matched route
set tag <i>tag-value</i>	Sets a tag value for a matched route

Table 14-3 *match commands that can be used with policy routing.*

Command	Description
match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Matches a packet with the characteristics specified in the standard or extended access lists
match length <i>min max</i>	Matches the Layer 3 length of a packet

Table 14-4 *set commands that can be used with policy routing.*

Command	Description
set default interface <i>type number</i> [... <i>type number</i>]	Sets the outgoing interface for matched packets when there is no explicit route to the destination
set interface <i>type number</i> [... <i>type number</i>]	Sets the outgoing interface for matched packets when there is an explicit route to the destination
set ip default next-hop <i>ip-address</i> [... <i>ip-address</i>]	Sets the next-hop router address for matched packets when there is no explicit route to the destination
set ip next-hop <i>ip-address</i> [... <i>ip-address</i>]	Sets the next-hop router address for matched packets when there is an explicit route to the destination
set ip precedence <i>precedence</i>	Sets the precedence bits in the Type of Service field of matched IP packets
set ip tos <i>type-of-service</i>	Sets the TOS bits in the Type of Service field of matched packets

Configuring Route Maps

Like access lists (see Appendix B, “Tutorial: Access Lists”), route maps by themselves affect nothing; they must be “called” by some command. The command will most likely be either a policy routing command or a redistribution command. Policy routing will send packets to the route map, whereas redistribution will send routes to the route map. The case studies in this section demonstrate the use of route maps for both redistribution and policy routing.

Route maps are identified by a name. For example, the route map in Example 14-1 is named Hagar.

Example 14-1 *A route map named Hagar is defined in this configuration.*

```
route-map Hagar permit 10
match ip address 110
set metric 100
```

Each route map statement has a “permit” or “deny” action and a sequence number. This route map shows a permit action and a sequence number of 10. These settings are the defaults—that is, if no action or sequence number is specified when the route map is configured, the route map will default to a permit and a sequence number of 10.

The sequence number allows the identification and editing of multiple statements. Consider the configuration steps in Example 14-2.

Example 14-2 *Route map Hagar is modified in this configuration.*

```
route-map Hagar 20
  match ip address 111
  set metric 50
route-map Hagar 15
  match ip address 112
  set metric 80
```

Here, a second and third set of route map statements, each with their own **match** and **set** statements, have been added to route map Hagar. Notice that a sequence number of 20 was configured first and then a sequence number of 15. In the final configuration, the IOS has placed statement 15 before 20 even though it was entered later, as shown in Example 14-3.¹

Example 14-3 *IOS places the commands in sequential order.*

```
route-map Hagar permit 10
  match ip address 110
  set metric 100
!
route-map Hagar permit 15
  match ip address 112
  set metric 80
!
route-map Hagar permit 20
  match ip address 111
  set metric 50
```

The sequence numbers also allow for the elimination of individual statements. For example, the statement

```
Linus(config)#no route-map Hagar 15
```

deletes statement 15 and leaves the other statements intact, as shown in Example 14-4.

Example 14-4 *Route-map Hagar after the match/set statements associated with sequence 15 have been removed.*

```
route-map Hagar permit 10
  match ip address 110
  set metric 100
!
route-map Hagar permit 20
  match ip address 111
  set metric 50
```

Be careful when editing route maps. In this example, if **no route-map Hagar** had been typed, without specifying a sequence number, the entire route map would have been deleted. Likewise, if no sequence numbers had been specified when the additional **match** and **set** statements were added, they would have simply changed statement 10.²

¹ Notice also that no action was specified in the configuration steps, so the default **permit** appears in the final configuration.

² Depending upon your IOS version, trying to change a match or set route-map statement without specifying a sequence number might result in an IOS message “% Please specify the entry by its sequence;” rather than assuming that sequence number 10 is to be changed.

A packet or route is passed sequentially through route map statements. If a match is made, any **set** statements are executed and the permit or deny action is executed. As with access lists, processing stops when a match is made and the specified action is executed; the route or packet is not passed to subsequent statements. Consider the route map in Example 14-5.

Example 14-5 *Route-map Sluggo.*

```
route-map Sluggo permit 10
  match ip route-source 1
  set next-hop 192.168.1.5
!
route-map Sluggo permit 20
  match ip route-source 2
  set next-hop 192.168.1.10
!
route-map Sluggo permit 30
  match ip route-source 3
  set next-hop 192.168.1.15
```

If a route does not match statement 10, it will be passed to statement 20. If a match is made at statement 20, the **set** command will be executed and the route will be permitted. The matched route will not be passed on to statement 30.

The behavior of a “deny” action depends on whether the route map is being used for policy routing or for redistribution. If a route map is being used for redistribution and a route matches a statement with a deny action, the route will not be redistributed. If the route map is being used for policy routing and a packet matches a statement with a deny action, the packet is not policy routed but is passed back to the normal routing process for forwarding.

Again as with access lists, there must be a default action for the route map to take in the event that a route or packet passes through every statement without a match. An implicit deny exists at the end of every route map. Routes that pass through a redistribution route map without a match are not redistributed, and packets that pass through a policy route map without a match are sent to the normal routing process.

If no **match** statement is configured under a route map statement, the default action is to match everything.

Each map statement might have multiple **match** and **set** statements, as shown in Example 14-6.

Example 14-6 *Route map Garfield contains multiple match and set statements for the map statement with sequence number 10.*

```
route-map Garfield permit 10
  match ip route-source 15
  match interface Serial0
  set metric-type type-1
  set next-hop 10.1.2.3
```

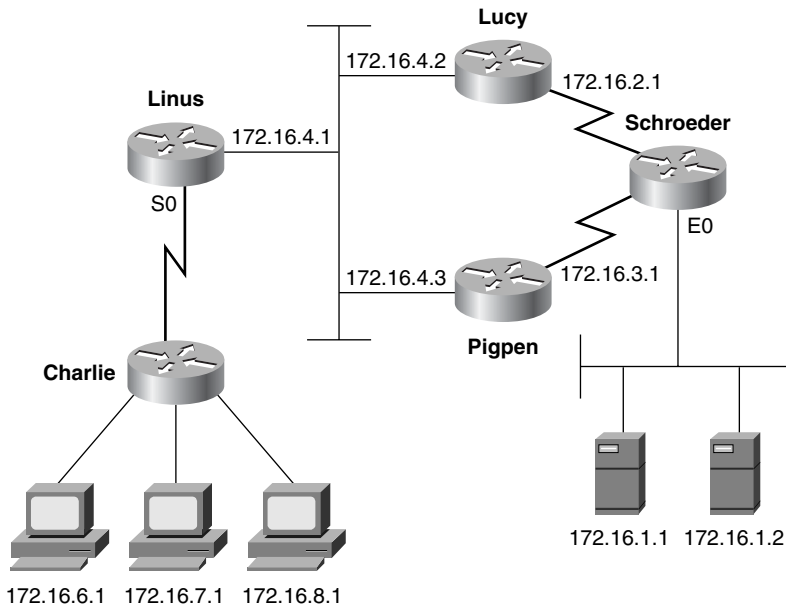
In a case such as this, every **match** statement must be matched for the **set** statements to be executed.

Case Study: Policy Routing

Policy routing is defined with the command **ip policy route-map**. The command is configured on an interface and affects incoming packets only.

Suppose a policy were to be implemented on Linus in Figure 14-3 such that traffic from subnet 172.16.6.0/24 would be forwarded to Lucy and traffic from subnet 172.16.7.0/24 would be forwarded to Pigpen. Linus's configuration is displayed in Example 14-7.

Figure 14-3 Policy routes can be configured at Linus to route some packets through Lucy and other packets through Pigpen.



Example 14-7 Linus's policy routing configuration.

```
interface Serial0
 ip address 172.16.5.1 255.255.255.0
 ip policy route-map Sally
!
access-list 1 permit 172.16.6.0 0.0.0.255
access-list 2 permit 172.16.7.0 0.0.0.255
!
route-map Sally permit 10
 match ip address 1
 set ip next-hop 172.16.4.2
!
route-map Sally permit 15
 match ip address 2
 set ip next-hop 172.16.4.3
```

The policy routing command on S0 sends incoming packets to route map Sally. Statement 10 of route map Sally uses access list 1 to identify source addresses from subnet 172.16.6.0/24. If a match is made, the packet is forwarded to Lucy, whose next-hop interface address is 172.16.4.2. If no match is made, the packet is sent to statement 15. That statement uses access list 2 to match source addresses from subnet 172.16.7.0/24. If a match is made at that statement, the packet is forwarded to Pigpen (172.16.4.3). Any packets that do not match statement 15, such as packets sourced from subnet 172.16.8.0/24, are routed normally. Example 14-8 shows the results of the policy route.³

Example 14-8 *The policy route configured on Linus's S0 interface routes packets from subnet 172.16.6.0/24 to Lucy (172.16.4.2) and routes packets from subnet 172.16.7.0/24 to Pigpen (172.16.4.3). Packets from subnet 172.16.8.0/24, which do not match the policy route, are routed normally (load balancing between Lucy and Pigpen).*

```

Linus#debug ip packet 5
IP packet debugging is on for access list 5
Linus#
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.6.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.8.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.3, len 60, forward

```

Suppose Lucy's Ethernet interface fails. Linus would still attempt to forward packets from 172.16.6.0 to Lucy's IP address. Because a match is made in the route map, even if the specified next-hop interface were down, no other matches would be attempted nor would the packet be routed normally. To force Linus to verify the availability of the next-hop address before attempting to forward the packet, use the command **set ip next-hop verify-availability**. Linus will search its CDP neighbors table to verify that the next-hop address is listed. If it is not, the policy is rejected and the packet is forwarded normally. Example 14-9 shows the output of the commands **debug ip policy** and **debug arp** while Lucy's Ethernet interface is down. Packets are matched and policy routed, even while the router is attempting to ARP for the next-hop address, without receiving an ARP reply. The packets are dropped.

Example 14-10 shows the output of the **debug ip policy** command with the addition of the **set ip next-hop verify-availability** command in the IP policy configuration. The example shows Linus's operation with Lucy's Ethernet interface still down. The packets are matched by the policy, but the policy is rejected because the next-hop address is no longer in Linus's CDP table. Since the policy is rejected, the packets are routed using the normal method.

³ Note that the **debug ip packet** command references an access list 5. This access list permits only the subnets connected to router Charlie so that uninteresting traffic is not displayed by the debug function.

Example 14-9 `debug ip policy` and `debug arp` on *Linus* shows packets are attempting to be routed according to the configured policy, even if the next hop specified by the policy is unavailable.

```

Linus#debug arp
ARP packet debugging is on
Linus#debug ip policy
Policy routing debugging is on
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, policy match
IP: route map Sally, item 10, permit
IP: s=172.16.6.1 (Serial0), d=172.16.2.1 (Ethernet0), len 100, policy routed
IP: Serial0 to Ethernet0 172.16.4.2
IP ARP: sent req src 172.16.4.1 0004.c150.e700,
           dst 172.16.4.2 0000.0000.0000 Ethernet0
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, policy match
IP: route map Sally, item 10, permit
IP: s=172.16.6.1 (Serial0), d=172.16.2.1 (Ethernet0), len 100, policy routed
IP: Serial0 to Ethernet0 172.16.4.2
IP ARP: sent req src 172.16.4.1 0004.c150.e700,
           dst 172.16.4.2 0000.0000.0000 Ethernet0
IP ARP: creating incomplete entry for IP address: 172.16.4.2 interface Ethernet0
IP ARP: sent req src 172.16.4.1 0004.c150.e700,
           dst 172.16.4.2 0000.0000.0000 Ethernet0
IP ARP throttled out the ARP Request for 172.16.4.2

```

Example 14-10 `debug ip policy` on *Linus* with `set ip next-hop verify-availability` configured, shows the policy is rejected (the next hop is not in *Linus*'s CDP neighbor table) and packets are routed normally (not policy routed).

```

Linus#debug ip policy
Policy routing debugging is on

IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy rejected - normal forwarding
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy rejected - normal forwarding

```

Example 14-11 shows the output of the same debug command, `debug ip policy`, after Lucy's Ethernet interface is backed up. The output shows that the packets are successfully policy routed.

Example 14-11 `debug ip policy` on *Linus* with `set ip next-hop verify-availability` configured. The packets are policy routed when the next hop is verified.

```

Linus#
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, g=172.16.4.2, len 100, FIB policy routed
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, len 100, FIB policy match
IP: s=172.16.6.1 (Serial0), d=172.16.2.1, g=172.16.4.2, len 100, FIB policy routed

```

Standard IP access lists are used when policy routing by source address only. To route by both source and destination, an extended IP access list is used. The configuration in Example 14-12 causes packets from any subnet to host 172.16.1.1 to be forwarded to Lucy, whereas packets from host 172.16.7.1 to host 172.16.1.2 are forwarded to Pigpen. All other packets are routed normally.

Example 14-12 *Policy route maps can reference extended IP access-lists to specify a source and destination address pair to match, as shown in this configuration.*

```
interface Serial0
ip address 172.16.5.1 255.255.255.0
ip policy route-map Sally
!
access-list 101 permit ip any host 172.16.1.1
access-list 102 permit ip host 172.16.7.1 host 172.16.1.2
!
route-map Sally permit 10
match ip address 101
set ip next-hop 172.16.4.2
!
route-map Sally permit 15
match ip address 102
set ip next-hop 172.16.4.3
```

Route map Sally is again used, except the **match** statements now reference access lists 101 and 102. Example 14-13 shows the results.

Example 14-13 *Packets from any host to host 172.16.1.1 match statement 10 of route map Sally and are forwarded to Lucy. Packets from host 172.16.7.1 to host 172.16.1.2 are forwarded to Pigpen. Packets from another address on subnet 172.16.7.0/24 to host 172.16.1.2 are not matched by Sally and are routed normally.*

```
Linus#debug ip packet 5
IP packet debugging is on for access list 5
Linus#
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.1 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.1 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.2, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.3, len 60, forward
IP: s=172.16.7.254 (Serial0), d=172.16.1.2 (Ethernet0), g=172.16.4.2, len 60, forward
```

Next, suppose your policy states that FTP traffic from the servers on subnet 172.16.1.0/24 should be forwarded to Lucy and that Telnet traffic from the same servers should be forwarded to Pigpen. This plan allows the bulk FTP traffic and the bursty, interactive Telnet traffic to be segregated on the two serial links from Schroeder. Schroeder will have the configuration in Example 14-14.

Example 14-14 Schroeder's policy route configuration forwarding FTP and Telnet traffic.

```
interface Ethernet0
ip address 172.16.1.4 255.255.255.0
ip policy route-map Rerun
!
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp any
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp-data any
access-list 106 permit tcp 172.16.1.0 0.0.0.255 eq telnet any
!
route-map Rerun permit 10
match ip address 105
set ip next-hop 172.16.2.1
!
route-map Rerun permit 20
match ip address 106
set ip next-hop 172.16.3.1
```

Access lists 105 and 106 are examining not only the source and destination addresses, but also the source port. In Example 14-15, the **detail** option is used with **debug ip packet** to allow observation of the packet types being forwarded by Schroeder. An access list 10 limits the displayed packets to those from 172.16.1.1 to 172.16.6.1.

Example 14-15 FTP packets (TCP ports 20 and 21) are being forwarded to Lucy, whereas Telnet packets (TCP port 23) with the same source and destination addresses are forwarded to Pigpen. Echo Reply packets (ICMP type 0), which do not find a match in the policy route, are routed normally.

```
Schroeder#debug ip packet detail 10
IP packet debugging is on (detailed) for access list 10
Schroeder#
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1064, forward
TCP src=20, dst=1047, seq=3702770065, ack=591246297, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 64, forward
TCP src=21, dst=1046, seq=3662108731, ack=591205663, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
TCP src=20, dst=1047, seq=3702771089, ack=591246297, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
TCP src=23, dst=1048, seq=3734385279, ack=591277873, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 52, forward
TCP src=23, dst=1048, seq=3734385279, ack=591277873, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
TCP src=23, dst=1048, seq=3734385291, ack=591277876, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
```

The purpose of segregating bulk and interactive traffic, as demonstrated in the last example, is so that the small packets characteristic of interactive traffic do not become delayed by the

large packets characteristic of bulk traffic. The problem with the approach in this example is that if many types of traffic must be segregated, the access lists identifying the traffic by destination port might become prohibitively large.

If the objective is to segregate small packets from large packets, the length of the packet can be matched, as shown in Example 14-16.

Example 14-16 *Schroeder's policy route configuration forwards traffic based on packet length.*

```
interface Ethernet0
ip address 172.16.1.4 255.255.255.0 ip policy route-map Woodstock !
route-map Woodstock permit 20
match length 1000 1600
set ip next-hop 172.16.2.1
!
route-map Woodstock permit 30
match length 0 400
set ip next-hop 172.16.3.1
```

Here the **match length** statement specifies a minimum and a maximum packet size. Statement 20 of the route map causes all packets between 1000 and 1600 octets in length to be routed across the serial link to Lucy. Statement 30 causes all packets up to 400 octets in length to be routed across the serial link to Pigpen. Packets between 400 and 1000 octets are routed normally.

Example 14-17 shows the results of the new route map. Again there are FTP, Telnet, and Echo Reply packets from 172.16.1.2 to 172.16.6.1, but now the packets are routed according to their size instead of their addresses and ports.

Example 14-17 *Packets of 1000 octets or larger are routed to Lucy, whereas packets of 400 octets or less are routed to Pigpen. Any packets between 400 and 1000 octets are routed normally.*

```
Schroeder#debug ip packet detail 10
IP packet debugging is on (detailed) for access list 10
Schroeder#
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
TCP src=20, dst=1063, seq=1528444161, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
TCP src=20, dst=1063, seq=1528442725, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial0), g=172.16.2.1, len 1476, forward
TCP src=20, dst=1063, seq=1528444161, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 840, forward
TCP src=20, dst=1063, seq=1528445597, ack=601956937, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
TCP src=21, dst=1062, seq=1469372904, ack=601897901, win=14329 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 54, forward
TCP src=21, dst=1062, seq=1469372904, ack=601897901, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
TCP src=21, dst=1062, seq=1469372918, ack=601897901, win=14335 ACK FIN
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 44, forward
TCP src=23, dst=1064, seq=1712116521, ack=602140570, win=14335 ACK SYN
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 43, forward
TCP src=23, dst=1064, seq=1712116522, ack=602140570, win=14335 ACK PSH
```

Example 14-17 *Packets of 1000 octets or larger are routed to Lucy, whereas packets of 400 octets or less are routed to Pigpen. Any packets between 400 and 1000 octets are routed normally. (Continued)*

```
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 40, forward
TCP src=23, dst=1064, seq=1712116525, ack=602140573, win=14332 ACK
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 52, forward
TCP src=23, dst=1064, seq=1712116525, ack=602140573, win=14335 ACK PSH
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
IP: s=172.16.1.2 (Ethernet0), d=172.16.6.1 (Serial1), g=172.16.3.1, len 60, forward
ICMP type=0, code=0
```

The policy routes demonstrated so far affect packets entering the router from a particular interface. But what about packets generated by the router itself? These can also be policy routed, with the command **ip local policy route-map**. Unlike the **ip policy route-map** command, which is configured on an interface, this command is configured globally on the router.

To apply the previously demonstrated policy to packets generated by Schroeder, the configuration is as displayed in Example 14-18.

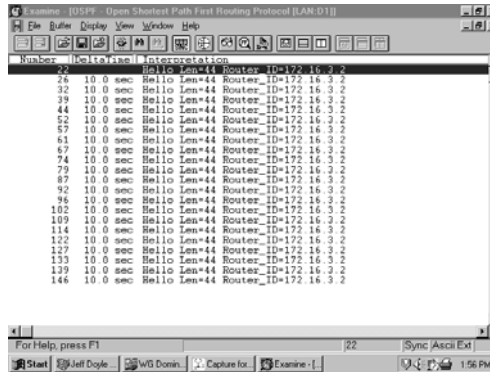
Example 14-18 *Route policy configuration for packets generated by Schroeder.*

```
interface Ethernet0
 ip address 172.16.1.4 255.255.255.0
 ip policy route-map Woodstock
 !
 ip local policy route-map Woodstock
 !
 access-list 120 permit ip any 172.16.1.0 0.0.0.255
 access-list 120 permit ospf any any
 !
 route-map Woodstock permit 10
 match ip address 120
 !
 route-map Woodstock permit 20
 match length 1000 1600
 set ip next-hop 172.16.2.1
 !
 route-map Woodstock permit 30
 match length 0 400
 set ip next-hop 172.16.3.1
```

Of particular interest is statement 10. This statement does not have a **set** statement, but merely permits packets that match access list 120. Access list 120, in turn, permits all packets destined for subnet 172.16.1.0/24 and all OSPF packets. Without the first line of the access list, some packets originated by Schroeder and destined for subnet 172.16.1.0/24 would be forwarded to the wrong interface by statement 20 or 30. Figure 14-4 shows why the second line of the access list is necessary. The length of Schroeder's OSPF Hellos is 44 octets. If statement 10 were not included, the OSPF Hellos would all match statement 30 and be

forwarded to Pigpen, breaking the adjacency between Lucy and Schroeder. By matching statement 10, the OSPF packets are permitted with no changes and are forwarded normally.

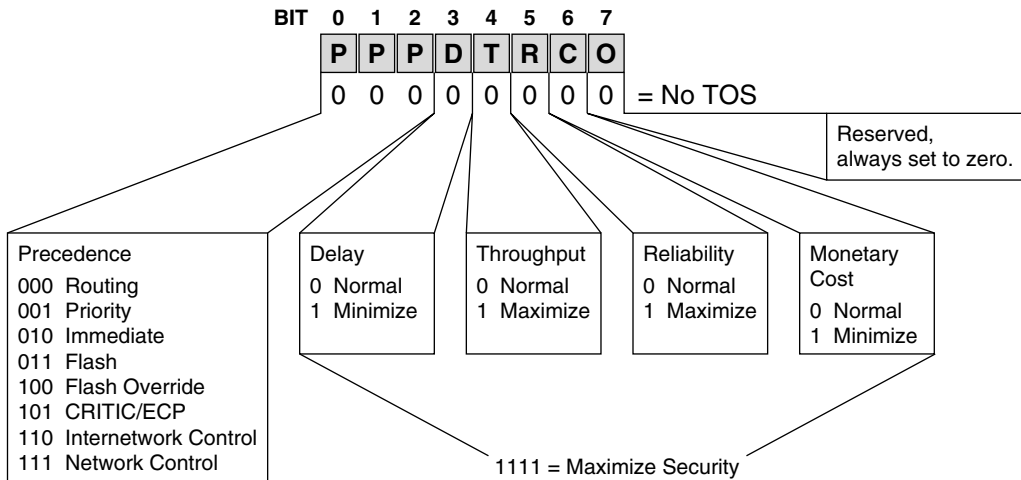
Figure 14-4 *The length of the OSPF Hello packets is seen in this analyzer capture.*



Case Study: Policy Routing and Quality of Service Routing

Although quality of service (QoS) routing is outside the scope of this volume, it must be noted here that policy routing can be an integral part of QoS. Policy routing in conjunction with QoS is done by setting the Precedence or the Type of Service (TOS) bits of the TOS field in the IP headers of packets as they enter a router’s interface. Figure 14-5 shows the bits of the TOS field. Although the TOS bits are seldom used in modern networks, the Precedence bits have found new life in QoS applications. The TOS bits are used to influence the path a router selects for a packet, whereas the Precedence bits are used to prioritize packets within a router.

Figure 14-5 *The Precedence and TOS bits of the Type of Service field of the IP header.*



The Precedence bits are set by using the **set ip Precedence** statement within a route map. The Precedence might be set by specifying the decimal equivalent of the three Precedence bits or by using keywords. Table 14-5 shows the decimal numbers and the keywords that can be used.

Table 14-5 *Precedence values and keywords used with the set ip precedence command.*

Bits	Number	Keyword
000	0	routine
001	1	priority
010	2	immediate
011	3	flash
100	4	flash-override
101	5	critical
110	6	internet
111	7	network

The TOS bits are set by using the **set ip tos** statement. Like the Precedence statement, the argument of the statement might be a number or a keyword, as shown in Table 14-6. Unlike Precedence, you might use a combination of TOS values. For example, specifying a TOS value of 12 (1100b) means *minimum delay* and *maximum throughput*. Only a single keyword can be used, so to set a combination of TOS values, a number must be specified.

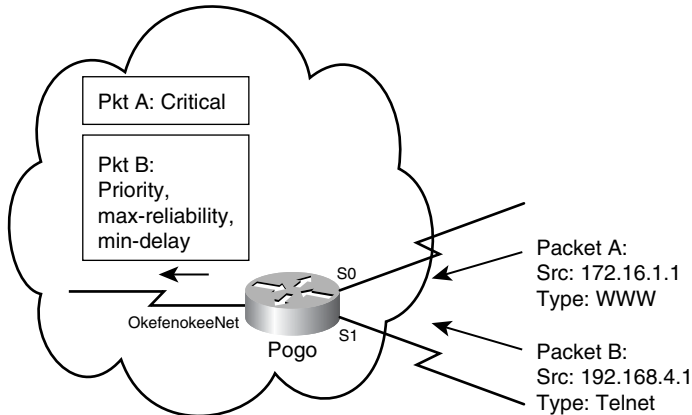
Table 14-6 *TOS values and keywords used with the set ip tos command.*

Bits	Number (0–15)	Keyword
0000	0	normal
0001	1	min-monetary-cost
0010	2	max-reliability
0100	4	max-throughput
1000	8	min-delay

Figure 14-6 shows an example of how policy routes can be used for QoS routing.

Here, router Pogo is at the “edge” of the Internet OkefenokeeNet. By configuring policy routes on Pogo’s serial links, the Precedence or TOS bits of incoming packets can be changed so that IP traffic is divided into several traffic classes. See Example 14-19 for instance.

Figure 14-6 Policy routes can be used to set the Precedence or TOS bits of packets entering a network. The routers within the network can then make QoS decisions based on the setting of these bits.



Example 14-19 Pogo's configuration sets Precedence and TOS bits.

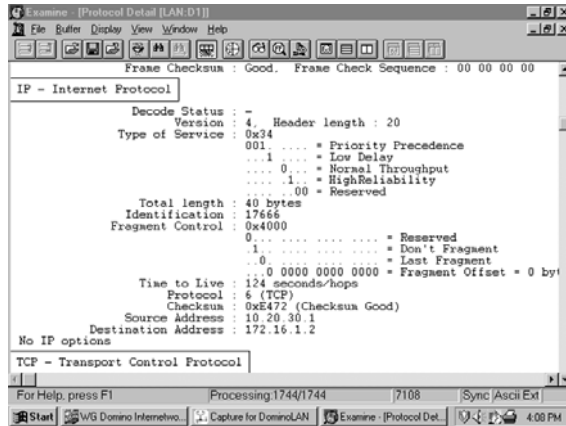
```
interface Serial0
ip address 10.1.18.67 255.255.255.252
ip policy route-map Albert
!
interface Serial1
ip address 10.34.16.83 255.255.255.252
ip policy route-map Albert
!
access-list 1 permit 172.16.0.0 0.0.255.255
access-list 110 permit tcp any eq www any
!
route-map Albert permit 10
match ip address 1 110
set ip precedence critical
!
route-map Albert permit 20
set ip tos 10
set ip precedence priority
```

Statement 10 says that if packets match both access lists 1 and 110, the Precedence will be set to *critical*. Notice that statement 20 has no **match** statement. This statement will match any packets that haven't been matched by statement 10. There are also two **set** statements under statement 20. These statements will set the TOS bits to *minimum delay* and *maximum reliability* and will set the Precedence bits to *priority*. Figure 14-7 shows a capture of a packet from somewhere inside OkefenokeeNet, which has been modified by the route map at Pogo.

After the Precedence or TOS bits have been set in packets entering the network, the routers within the Internet can make QoS decisions based in part or wholly on the class of service these bits define. For example, priority, custom, or weighted fair queuing might be configured

to prioritize traffic according to the Precedence or TOS bits. In some implementations, Precedence can be used with congestion avoidance mechanisms such as Weighted Random Early Detection (WRED). Or a crude Class-of-Service routing can be implemented by configuring access lists that permit or deny packets across certain links based on the setting of their Precedence or TOS bits.

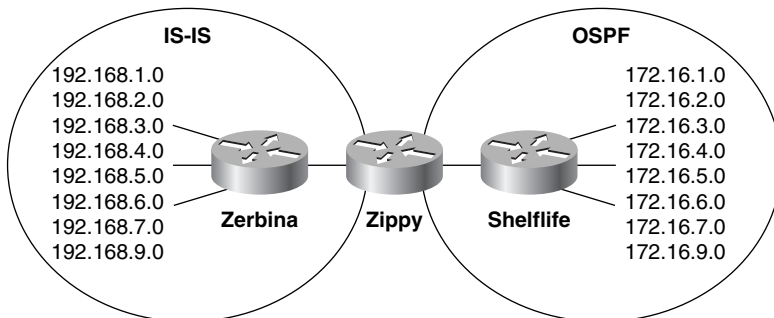
Figure 14-7 Pogo's policy route has set the Precedence bits of this packet to priority (001b) and the TOS bits to minimum delay and maximum reliability (1010b).



Case Study: Route Maps and Redistribution

A route map can be used with redistribution for both IPv4 and IPv6 by adding a call to the route map in the **redistribute** command. Figure 14-8 shows a network in which IPv4 IS-IS and OSPF routes are being mutually redistributed at router Zippy. Of the network and subnet addresses listed in the illustration, only the ones whose third octet is odd-numbered are to be redistributed.

Figure 14-8 The OSPF and IS-IS routes are being mutually redistributed. Route maps can be used with the **redistribute** command as simple route filters, or they can be used to modify characteristics of the redistributed routes.



Zippy's configuration is displayed in Example 14-20.

Example 14-20 *Zippy is configured to redistribute only addresses with an odd numbered third octet.*

```

router ospf 1
 redistribute isis level-1 metric 20 subnets route-map Griffy
 network 172.16.10.2 0.0.0.0 area 5
!
router isis
 redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
 net 47.0001.1234.5678.9056.00
!
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.6.0
access-list 2 permit 172.16.1.0
access-list 2 permit 172.16.3.0
access-list 2 permit 172.16.5.0
access-list 2 permit 172.16.7.0
access-list 2 permit 172.16.9.0
!
route-map Griffy deny 10
 match ip address 1
!
route-map Griffy permit 20
!
route-map Toad permit 10
 match ip address 2

```

Route maps Griffy and Toad perform the same functions, but with different logic. Griffy uses negative logic, identifying the routes that should not be redistributed, and Toad uses positive logic, identifying the routes that should be redistributed.

Statement 10 of Griffy denies any routes that are permitted by access list 1 (the addresses with an even third octet). Because the addresses with odd-numbered third octets do not find a match at statement 10, they are passed to statement 20. Statement 20 has no **match** statement, so the default is to match everything. Statement 20 has a permit action, so the odd routes are permitted. The result is shown in Example 14-21.

Example 14-21 *The only destinations within the IS-IS domain that are contained in Shelflife's route table are those with an odd-numbered third octet.*

```

Shelflife#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
Gateway of last resort is not set
O E2 192.168.9.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0
O E2 192.168.1.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0
O E2 192.168.3.0 [110/20] via 172.16.10.2, 00:24:46, Ethernet0

```

Example 14-21 *The only destinations within the IS-IS domain that are contained in Shelflife's route table are those with an odd-numbered third octet. (Continued)*

```

O E2 192.168.5.0 [110/20] via 172.16.10.2, 00:24:47, Ethernet0
O E2 192.168.7.0 [110/20] via 172.16.10.2, 00:24:47, Ethernet0
    172.16.0.0 255.255.255.0 is subnetted, 9 subnets
C     172.16.9.0 is directly connected, Serial0
C     172.16.10.0 is directly connected, Ethernet0
O     172.16.4.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.5.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.6.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.7.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.1.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.2.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
O     172.16.3.0 [110/159] via 172.16.9.2, 14:05:33, Serial0
Shelflife#

```

Route map Toad has a single statement that permits routes that have been permitted by access list 2 (addresses with an odd third octet). The addresses with an even third octet do not find a match at access list 2. The default route map statement when redistributing is to deny all routes, so the addresses that are not matched by access list 2 are not redistributed. Example 14-22 shows the results of route map Toad.

Example 14-22 *The only destinations within the OSPF domain that are contained in Zerbina's route table are those with an odd-numbered third octet.*

```

Zerbina#show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set
C    192.168.9.0/24 is directly connected, Serial0
C    192.168.10.0/24 is directly connected, Ethernet0
i L1 192.168.1.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.2.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.3.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.4.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.5.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.6.0/24 [115/15] via 192.168.9.2, Serial0
i L1 192.168.7.0/24 [115/15] via 192.168.9.2, Serial0
    172.16.0.0/24 is subnetted, 5 subnets
i L2   172.16.9.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.5.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.7.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.1.0 [115/35] via 192.168.10.2, Ethernet0
i L2   172.16.3.0 [115/35] via 192.168.10.2, Ethernet0
Zerbina#

```

Other configurations will achieve the same ends. For instance, route map Toad will have the same effect with the access list in Example 14-23.

Example 14-23 *An alternate configuration for route-map Toad on Zippy.*

```
access-list 2 deny 172.16.2.0
access-list 2 deny 172.16.4.0
access-list 2 deny 172.16.6.0
access-list 2 permit any
```

Although route maps work fine as simple route filters, their strength lies in their ability to change the routes in various ways. Consider the configuration in Example 14-24 of Zippy in Figure 14-8.

Example 14-24 *Zippy's route-map configuration sets the metric type, metric and level of certain redistributed routes.*

```
router ospf 1
 redistribute isis level-1 metric 20 subnets route-map Griffy
 network 172.16.10.2 0.0.0.0 area 5
!
router isis
 redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
 net 47.0001.1234.5678.9056.00
!
ip classless
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.6.0
access-list 2 permit 172.16.9.0
access-list 2 permit 172.16.5.0
access-list 2 permit 172.16.7.0
access-list 2 permit 172.16.1.0
access-list 2 permit 172.16.3.0
!
route-map Griffy permit 10
 match ip address 1
 set metric-type type-1
!
route-map Griffy permit 20
!
route-map Toad permit 10
 match ip address 2
 set metric 15
 set level level-1
!
route-map Toad permit 20
```

Statement 10 of route map Griffy permits routes to the addresses in access list 1 and redistributes them into OSPF as type 1 external routes. Statement 20 permits all other routes, which will be redistributed with the default external type 2. Example 14-25 shows the results.

Example 14-25 *The routes to destinations in the IS-IS domain are E1 if the third octet of the address is even and E2 if the third octet is odd.*

```
Shelflife#show ip route
Codes:C -connected,S -static,I -IGRP,R -RIP,M -mobile,B -BGP
       D -EIGRP,EX -EIGRP external,O -OSPF,IA -OSPF inter area
       E1 -OSPF external type 1,E2 -OSPF external type 2,E -EGP
       i -IS-IS,L1 -IS-IS level-1,L2 -IS-IS level-2,*-candidate default
Gateway of last resort is not set
O E2 192.168.9.0 [110/20 ] via 172.16.10.2,,00:13:43,Ethernet0
O E2 192.168.1.0 [110/20 ] via 172.16.10.2,,00:13:43,Ethernet0
O E1 192.168.2.0 [110/30 ] via 172.16.10.2,,00:13:43,Ethernet0
O E2 192.168.3.0 [110/20 ] via 172.16.10.2,,00:13:44,Ethernet0
O E1 192.168.4.0 [110/30 ] via 172.16.10.2,,00:13:44,Ethernet0
O E2 192.168.5.0 [110/20 ] via 172.16.10.2,,00:13:44,Ethernet0
O E1 192.168.6.0 [110/30 ] via 172.16.10.2,,00:13:44,Ethernet0
O E2 192.168.7.0 [110/20 ] via 172.16.10.2,,00:13:44,Ethernet0
172.16.0.0 255.255.255.0 is subnetted,9 subnets
C      172.16.9.0 is directly connected,Serial0
C      172.16.10.0 is directly connected,Ethernet0
O      172.16.4.0 [110/159 ] via 172.16.9.2,,15:49:29,Serial0
O      172.16.5.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
O      172.16.6.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
O      172.16.7.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
O      172.16.1.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
O      172.16.2.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
O      172.16.3.0 [110/159 ] via 172.16.9.2,,15:49:30,Serial0
Shelflife#
```

Statement 10 of route map Toad permits routes to addresses in access list 2 and redistributes them into IS-IS as level 1 routes. Their metric is set to 15. Statement 20 permits all other routes, which will be redistributed as level 2 and with a metric of 25, as specified by the **redistribute** command under the IS-IS configuration (see Example 14-26).

Example 14-26 *The routes to destinations in the OSPF domain are L2 if the third octet of the address is even and L1 if the third octet is odd. The “odds” are redistributed with a metric of 15, and the “evens” are redistributed with a metric of 25 (10 is added for the hop from Zippy to Zerbina).*

```
Zerbina#show ip route
Codes:C -connected,S -static,I -IGRP,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,E -EGP
       i -IS-IS,L1 -IS-IS level-1,L2 -IS-IS level-2,*-candidate default
       U -per-user static route,o -ODR
Gateway of last resort is not set
C      192.168.9.0/24 is directly connected,Serial0
C      192.168.10.0/24 is directly connected,Ethernet0
i L1 192.168.1.0/24 [115/15 ] via 192.168.9.2,,Serial0
i L1 192.168.2.0/24 [115/15 ] via 192.168.9.2,,Serial0
i L1 192.168.3.0/24 [115/15 ] via 192.168.9.2,,Serial0
```

continues

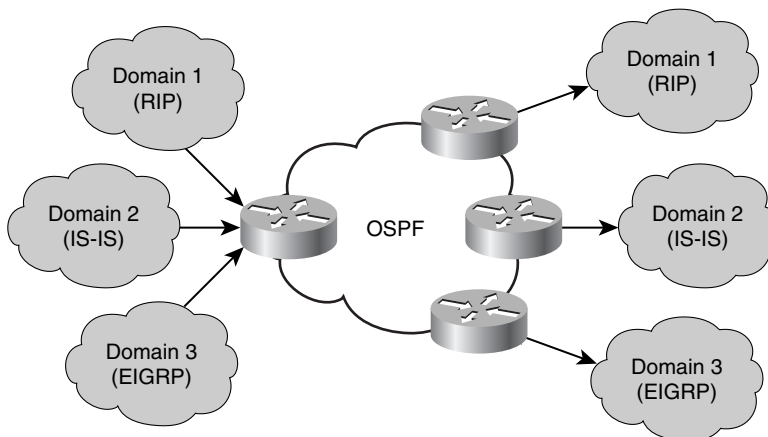
Example 14-26 *The routes to destinations in the OSPF domain are L2 if the third octet of the address is even and L1 if the third octet is odd. The “odds” are redistributed with a metric of 15, and the “evens” are redistributed with a metric of 25 (10 is added for the hop from Zippy to Zerbina). (Continued)*

```
i L1 192.168.4.0/24 [115/15 ] via 192.168.9.2,,Serial0
i L1 192.168.5.0/24 [115/15 ] via 192.168.9.2,,Serial0
i L1 192.168.6.0/24 [115/15 ] via 192.168.9.2,,Serial0
i L1 192.168.7.0/24 [115/15 ] via 192.168.9.2,,Serial0
    172.16.0.0/24 is subnetted,8 subnets
i L1   172.16.9.0 [115/25 ] via 192.168.10.2,,Ethernet0
i L2   172.16.4.0 [115/35 ] via 192.168.10.2,,Ethernet0
i L1   172.16.5.0 [115/25 ] via 192.168.10.2,,Ethernet0
i L2   172.16.6.0 [115/35 ] via 192.168.10.2,,Ethernet0
i L1   172.16.7.0 [115/25 ] via 192.168.10.2,,Ethernet0
i L1   172.16.1.0 [115/25 ] via 192.168.10.2,,Ethernet0
i L2   172.16.2.0 [115/35 ] via 192.168.10.2,,Ethernet0
i L1   172.16.3.0 [115/25 ] via 192.168.10.2,,Ethernet0
Zerbina#
```

Case Study: Route Tagging

Figure 14-9 shows a situation in which routes from several routing domains, each running a separate routing protocol, are being redistributed into a single transit domain running OSPF. On the other side of the OSPF cloud, the routes must be redistributed back into their respective domains. Route filters can be used at the egress points from the OSPF cloud into each domain to permit only the routes that belong to that domain. However, if each domain has many routes or if the routes within the domain change frequently, the route filters can become difficult to manage.

Figure 14-9 *Routes from each of the three domains on the left are redistributed into a transit network running OSPF. On the right, the routes for each domain must be redistributed back into their original domains.*

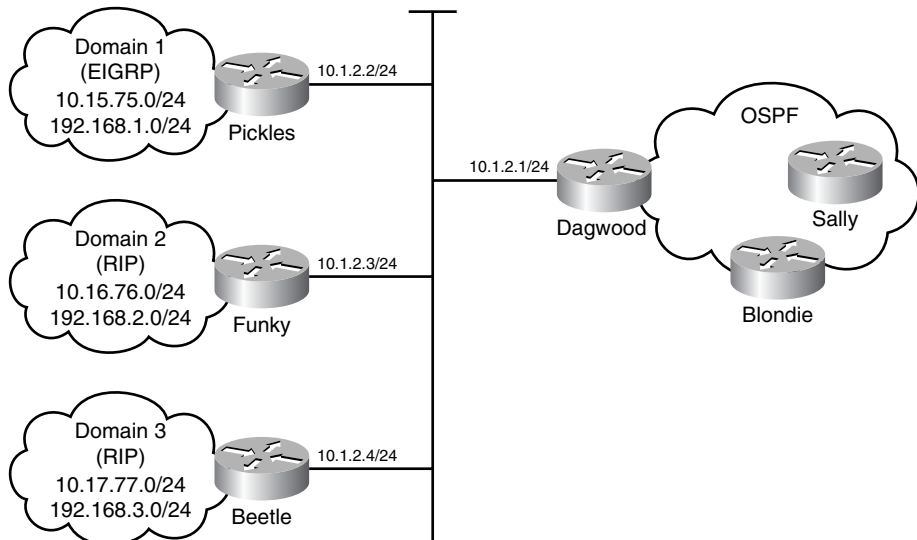


Another way of handling this problem is to tag the routes at their ingress points to the OSPF transit domain with a tag that is unique to each domain. At the egress points, the routes can be redistributed by their tags instead of by specific addresses. The routing protocol of the transit network does not necessarily use the tag, but merely conveys it to and from its external networks. RIPv2, EIGRP, Integrated IS-IS, and OSPF all support route tags. BGP also supports route tags. Tags are not supported by RIPv1. A case study in this section shows how a transit network running OSPF can use the route tags.

A re-examination of the packet formats in Chapter 6, “RIPv2, RIPv2, and Classless Routing;” Chapter 7, “Enhanced Interior Gateway Routing Protocol (EIGRP);” Chapter 8, “OSPFv2;” and Chapter 10, “Integrated IS-IS” show that RIPv2 messages support 16-bit tags, IS-IS Inter-Domain Routing Protocol Information TLVs support 16-bit tags, and EIGRP external route TLVs and OSPF type 5 LSAs support 32-bit tags. These tags are expressed as decimal numbers, so tags carried by RIPv2 and IS-IS will be between 0 and 65,535, and tags carried by EIGRP and OSPF will be between 0 and 4,294,967,295.

In Figure 14-10, router Dagwood is accepting routes from three different routing domains and redistributing them into a domain running OSPF. The objective here is to tag the routes from each domain so that their source domain might be identified within the OSPF cloud. Routes from domain 1 will have a tag of 1, domain 2 will have a tag of 2, and so on.

Figure 14-10 *Dagwood is configured to tag the routes from each of the three routing domains as they are redistributed into OSPF.*



Dagwood's configuration is displayed in Example 14-27.

Example 14-27 *Dagwood is configured to tag routes as they are redistributed into OSPF from RIP and EIGRP.*

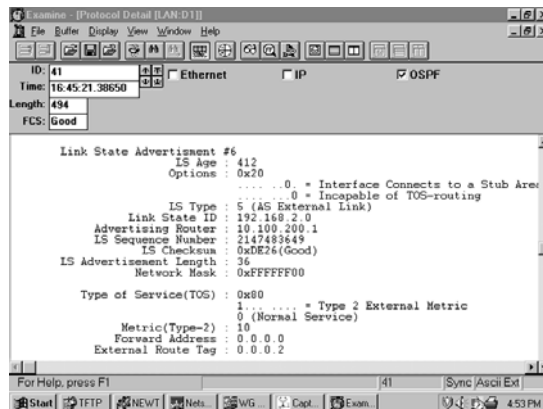
```

router ospf 1
 redistribute eigrp 1 metric 10 subnets tag 1
 redistribute rip metric 10 subnets route-map Dithers
 network 10.100.200.1 0.0.0.0 area 0
!
router rip
 network 10.0.0.0
!
router eigrp 1
 network 10.0.0.0
!
access-list 1 permit 10.1.2.3
access-list 2 permit 10.1.2.4
!
route-map Dithers permit 10
 match ip route-source 1
 set tag 2
!
route-map Dithers permit 20
 match ip route-source 2
 set tag 3

```

First, notice the **redistribute eigrp** command under OSPF. Dagwood is accepting routes from only one EIGRP domain, so the tag can be set to 1 directly on the **redistribute** command. However, routes are being learned from two RIP domains. Here a route map is needed. Route map Dithers sets the tag of the RIP routes to either 2 or 3, depending on whether the route was learned from Funky (10.1.2.3) or Beetle (10.1.2.4). Figure 14-11 shows an LSA advertising one of the RIP-learned routes, with the route tag set to 2.

Figure 14-11 *This type 5 LSA is advertising network 192.168.2.0, which is in domain 2, within the OSPF domain. The route tag is shown on the last line.*



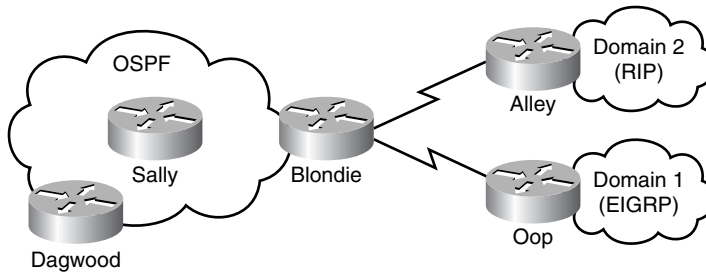
The route tags can also be observed in the OSPF link state database (see Example 14-28).

Example 14-28 *The OSPF link state database indicates the tags that were set for each of the external routes by Dagwood's redistribution processes.*

```
Blondie#show ip ospf database
      OSPF Router with ID (10.100.200.2) (Process ID 1)
      Router Link States (Area 0)
Link ID      ADV Router    Age      Seq#          Checksum Link count
10.100.200.3 10.100.200.3  671     0x80000003   0x00A137  4
10.100.200.2 10.100.200.2  39      0x80000002   0x6FF5    3
10.100.200.1 10.100.200.1  40      0x80000033   0x33E1    3
      Net Link States (Area 0)
Link ID      ADV Router    Age      Seq#          Checksum
10.100.200.1 10.100.200.1  40      0x80000001   0xB0A7
      AS External Link States
Link ID      ADV Router    Age      Seq#          Checksum Tag
192.168.2.0 10.100.200.1  641     0x80000028   0x904D    2
10.17.77.0  10.100.200.1  642     0x80000028   0xC817    3
192.168.3.0 10.100.200.1  642     0x80000028   0x9744    3
10.15.75.0  10.100.200.1  642     0x80000028   0xD213    1
10.1.2.0    10.100.200.1  642     0x80000028   0xA19B    1
10.16.76.0 10.100.200.1  642     0x80000028   0xCD15    2
192.168.1.0 10.100.200.1  644     0x80000028   0x8956    1
10.100.200.0 10.100.200.1  644     0x80000028   0x6EA4    1
Blondie#
```

In Figure 14-12, Blondie must redistribute only domain 2 routes to Alley and only domain 1 routes to Oop. Because the routes were tagged as they entered the OSPF transit domain, this is easily done. Blondie's configuration is shown in Example 14-29.

Figure 14-12 *Blondie is using route maps to redistribute routes according to their route tag.*



Example 14-29 *Blondie is configured to use route maps to redistribute routes with tag 1 to EIGRP and tag 2 to RIP.*

```
router ospf 1
  network 10.100.200.2 0.0.0.0 area 0
!
router rip
  redistribute ospf 1 match external 2 route-map Daisy
  passive-interface Ethernet0
  passive-interface Serial1
```

continues

Example 14-29 *Blondie is configured to use route maps to redistribute routes with tag 1 to EIGRP and tag 2 to RIP. (Continued)*

```

network 10.0.0.0
default-metric 5
!
router eigrp 1
 redistribute ospf 1 match external 2 route-map Herb
 passive-interface Ethernet0
 passive-interface Serial0
 network 10.0.0.0
 default-metric 10000 1000 255 1 1500
!
route-map Daisy permit 10
 match tag 2
!
route-map Herb permit 10
 match tag 1

```

Example 14-30 shows the resulting routes at Alley and Oop. One drawback to the use of route tags to filter routes is that there is no way to filter routes by interface. For example, if Blondie had to send routes to both domain 2 and domain 3, which both run RIP, route maps could not be configured to send some routes to one RIP process and other routes to another RIP process. The routes would have to be filtered by address with **distribute-list** commands.

Example 14-30 *The route tables of Alley and Oop in Figure 14-12 show the results of the redistribution configuration at Blondie.*

```

Alley#show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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, 5 subnets, 3 masks
C    10.1.3.0/24 is directly connected, Serial0
R    10.1.5.4/30 [120/1] via 10.1.3.1, 00:00:25, Serial0
R    10.1.4.0/24 [120/1] via 10.1.3.1, 00:00:25, Serial0
R    10.16.76.0/24 [120/5] via 10.1.3.1, 00:00:25, Serial0
R    10.100.200.2/32 [120/1] via 10.1.3.1, 00:00:25, Serial0
R    192.168.2.0/24 [120/5] via 10.1.3.1, 00:00:25, Serial0
Alley#

Oop#show ip route
Codes: C - connected, S - static, I - IGRP, 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

```

Example 14-30 *The route tables of Alley and Oop in Figure 14-12 show the results of the redistribution configuration at Blondie. (Continued)*

```

E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, 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, 6 subnets, 3 masks
D       10.1.3.0/24 [90/2681856] via 10.1.4.1, 00:21:36, Serial0
D EX    10.1.2.0/24 [170/2425856] via 10.1.4.1, 00:08:22, Serial0
D       10.1.5.4/30 [90/2681856] via 10.1.4.1, 00:22:40, Serial0
C       10.1.4.0/24 is directly connected, Serial0
D EX    10.15.75.0/24 [170/2425856] via 10.1.4.1, 00:04:56, Serial0
D       10.100.200.2/32 [90/2297856] via 10.1.4.1, 00:22:40, Serial0
D EX   192.168.1.0/24 [170/2425856] via 10.1.4.1, 00:04:56, Serial0
Oop#

```

Case Study: Filtering Tagged Routes Out of OSPF Route Table

The network running OSPF in Figure 14-10 and Figure 14-12 is a transit network. If devices within that transit area do not need to send packets to any of the other domains, there is no need to maintain the addresses of those domains in the OSPF route tables. The tags that were added to the routes, along with distribute lists and route maps, can be applied to the OSPF routers to prevent the addresses from being added to the route tables, while not affecting the entries in the link-state database.

Sally, a router wholly within the OSPF domain, has been modified with the configuration in Example 14-31.

Example 14-31 *Sally's configuration uses tags to filter routes from the OSPF route table.*

```

router ospf 1
 network 10.100.200.1 0.0.0.0 area 0
 network 10.1.5.0 0.0.0.255 area 0
 distribute-list route-map Charlie in
 !
 route-map Charlie deny 10
  match tag 1 2 3
 !
 route-map Charlie permit 20

```

The route map Charlie denies addresses that are marked with tags 1, 2 or 3. These addresses are omitted from the IP route table by the **distribute-list in** command. Any other address is added to the route table, permitted by the route map sequence 20. The distribute list is

not applied to Blondie and Dagwood, the edge routers that are performing redistribution. If an address is not in the route table, even if it is in the OSPF LSA database, it will not be redistributed into another routing protocol. Example 14-32 shows Sally's IP route table and OSPF LSA database.

Example 14-32 *The OSPF addresses marked with tags are filtered out of the IP route table. The addresses still exist in the OSPF LSA database.*

```
Sally#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, 5 subnets, 2 masks
C       10.1.5.4/30 is directly connected, Serial0/0.2
C       10.1.5.0/30 is directly connected, Serial0/0.1
O       10.100.200.2/32 [110/65] via 10.1.5.6, 00:17:24, Serial0/0.2
C       10.100.200.3/32 is directly connected, Loopback0
O       10.100.200.1/32 [110/65] via 10.1.5.1, 00:17:24, Serial0/0.1
```

```
Sally#show ip ospf database

        OSPF Router with ID (10.100.200.3) (Process ID 1)

                Router Link States (Area 0)

Link ID        ADV Router    Age           Seq#           Checksum Link count
10.100.200.1   10.100.200.1 1183         0x80000004    0x003756 3
10.100.200.2   10.100.200.2 1181         0x80000004    0x00E1A1 3
10.100.200.3   10.100.200.3 1177         0x8000000A    0x00933E 4

                Type-5 AS External Link States

Link ID        ADV Router    Age           Seq#           Checksum Tag
10.1.2.0       10.100.200.1 94           0x80000003    0x00EB76 1
10.15.75.0     10.100.200.1 603          0x80000002    0x001FEC 1
10.16.76.0     10.100.200.1 346          0x80000002    0x001AEE 2
10.16.77.0     10.100.200.1 353          0x80000002    0x001FEE 2
192.168.1.0    10.100.200.1 603          0x80000002    0x00D530 1
192.168.2.0    10.100.200.1 346          0x80000002    0x00DC27 2
192.168.3.0    10.100.200.1 353          0x80000002    0x00E427 2
Sally#
```

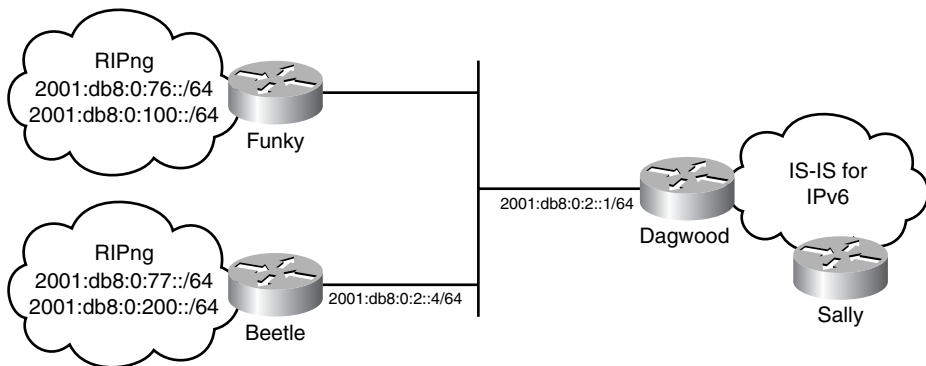
None of the tagged routes, tag 1, 2, or 3, exist in Sally's IP route table.

Case Study: IPv6 Redistribution with Route Maps

IPv6 routing protocols also support redistribution of routes using route maps. The configuration is almost identical to IPv4.

Figure 14-13 shows the addition of IPv6 addresses and routing protocols to the network shown in Figure 14-10. Funky and Beetle are each running RIPng. Dagwood is redistributing IPv6 prefixes between RIPng and IS-IS. Only IPv6 prefixes from Beetle are redistributed into IS-IS, and prefix 2001:db8:0:77::/64 is set with a metric of 10 while prefix 2001:db8:0:200::/64 has a metric of 100.

Figure 14-13 IPv6 has been added to the network in Figure 14-10. IPv6 prefixes are redistributed between RIPng and IS-IS.



Dagwood's configuration is displayed in Example 14-33.

Example 14-33 Dagwood's IPv6 configuration.

```
Interface ethernet 0/0
  ipv6 address 2001:db8:0:2::1/64
  ipv6 rip domain3 enable
!
interface serial 0/0.1 point-to-point
  ipv6 address 2001:db8:0:5::1/64
  ipv6 router isis
!
ipv6 router rip domain3
!
router isis
  net 00.0001.0000.5678.ef01.00
  Address-family ipv6
    Redistribute rip domain3 route-map Beetlefilter
!
route-map Beetlefilter permit 10
  match ipv6 route-source prefix-list 1
```

continues

Example 14-33 *Dagwood's IPv6 configuration. (Continued)*

```

match ipv6 address prefix-list 3
set metric 10
!
route-map Beetlefilter permit 20
match ipv6 route-source prefix-list 1
match ipv6 address prefix-list 2
set metric 100
!
ipv6 prefix-list 1 permit 2001:db8:0:2::4/128
ipv6 prefix-list 2 permit 2001:db8:0:200::/64
ipv6 prefix-list 3 permit 2001:db8:0:77::/64

```

IPv6 prefix-lists are used to match the source of route information or to match specific addresses for redistribution. Statement 10 of Beetlefilter specifies that the route source must equal Beetle's IPv6 address and the prefix must equal 2001:db8:0:77::/64 for the metric to be set to 10. If the route source or the prefix does not match, statement 20 is executed. If the source is Beetle and the prefix is 2001:db8:0:200::/64, the metric is set to 100. If the source is not Beetle, or the prefix is not one of the ones specified, the route is not redistributed.

Sally's IS-IS database (Example 14-34) shows the redistributed prefixes.

Example 14-34 *The redistributed routes are seen in Sally's IS-IS database.*

```

Sally#show isis database detail Dagwood.00.00

IS-IS Level-1 LSP Dagwood.00-00
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Dagwood.00-00        0x00000004  0xCA8B        938            0/0/0
  Area Address: 00.0001
  NLPID:            0x8E
  Hostname: Dagwood
  IPv6 Address: 2001:DB8:0:5::1
  Metric: 10        IPv6 2001:DB8:0:5::/64
  Metric: 10        IS Sally.00

IS-IS Level-2 LSP Dagwood.00-00
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
Dagwood.00-00        0x00000008  0x28D7        1089           0/0/0
  Area Address: 00.0001
  NLPID:            0x8E
  Hostname: Dagwood
  IPv6 Address: 2001:DB8:0:5::1
  Metric: 10        IS Sally.00
  Metric: 10        IPv6 2001:DB8:0:5::/64
  Metric: 10        IPv6 2001:DB8:0:77::/64
  Metric: 100       IPv6 2001:DB8:0:200::/64
Sally#

```


Looking Ahead

This chapter concludes this book's in-depth look at routing TCP/IP with respect to interior gateway protocols. If you are preparing to become a CCIE, you certainly will want to know this book's topics thoroughly before taking the exam. Use the end-of-chapter questions and exercises to test your level of understanding and preparedness. And if you haven't studied routing TCP/IP with respect to Exterior Gateway Protocols, doing so is a next logical step in your preparation.

Summary Table: Chapter 14 Command Review

Command	Description
access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>]	Defines a line of a standard IP access list
access-list <i>access-list-number</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i> [precedence <i>precedence</i>] [tos <i>tos</i>] [log]	Defines a line of an extended IP access list
ip local policy route-map <i>map-tag</i>	Defines a policy route for packets originated by the router itself
ip policy route-map <i>map-tag</i>	Defines a policy route for packets transiting the router
match interface <i>type number</i> [<i>...type number</i>]	Matches routes that have their next hop out one of the interfaces specified
match ip address { <i>access-list-number</i> <i>name</i> } [<i>...access-list-number</i> <i>name</i>]	Matches routes that have a destination address specified by one of the access lists
match ip next-hop { <i>access-list-number</i> <i>name</i> } [<i>...access-list-number</i> <i>name</i>]	Matches routes that have a next-hop router address specified by one of the access lists
match ip route-source { <i>access-list-number</i> <i>name</i> } [<i>...access-list-number</i> <i>name</i>]	Matches routes that have been advertised by routers at the addresses specified in the access lists
match ipv6 address prefix-list { <i>name</i> }	Matches routes that have a destination address specified by the prefix-list
match ipv6 next-hop prefix-list { <i>name</i> }	Matches routes that have a next-hop router address specified by the prefix-lists
match ipv6 route-source prefix-list { <i>name</i> }	Matches routes that have been advertised by routers at the addresses specified in the prefix-list
match length <i>min max</i>	Matches the level 3 length of a packet

continues

(Continued)

Command	Description
match metric <i>metric-value</i>	Matches routes with the specified metric
match route-type { internal external type-1 type-2 } level-1 level-2 }	Matches OSPF, EIGRP, or IS-IS routes of the specified type
match tag <i>tag-value</i> [... <i>tag-value</i>]	Matches routes with the specified tags
ipv6 prefix-list <i>list-name</i> [seq <i>seq-number</i>] { deny <i>ipv6-prefix/prefix-length</i> permit <i>ipv6-prefix/prefix-length</i> description <i>text</i> } [ge <i>ge-value</i>] [le <i>le-value</i>]	Defines an IPv6 prefix list
redistribute <i>protocol</i> [<i>process-id</i>] { level-1 level-1-2 level-2 } [metric <i>metric-value</i>] [metric-type <i>type-value</i>] [match { internal external 1 external 2 }] [tag <i>tag-value</i>] [route-map <i>map-tag</i>] [weight <i>weight</i>] [subnets]	Configures redistribution into a routing protocol and specifies the source of the redistributed routes
set level { level-1 level-2 level-1-2 stub-area backbone }	Sets the IS-IS level or the OSPF area into which a matched route is to be redistributed
set default interface <i>type number</i> [... <i>type number</i>]	Sets the outgoing interface for matched packets when there is no explicit route to the destination
set interface <i>type number</i> [... <i>type number</i>]	Sets the outgoing interface for matched packets when there is an explicit route to the destination
set ip default next-hop <i>ip-address</i> [... <i>ip-address</i>]	Sets the next-hop router address for matched packets when there is no explicit route to the destination
set ip next-hop <i>ip-address</i> [... <i>ip-address</i>]	Sets the next-hop router address for matched packets when there is an explicit route to the destination
set ip precedence <i>precedence</i>	Sets the precedence bits in the Type of Service field of matched IP packets
set ip tos <i>type-of-service</i>	Sets the TOS bits in the Type of Service field of matched packets
set metric { <i>metric-value</i> <i>bandwidth</i> <i>delay</i> <i>reliability</i> <i>loading</i> <i>mtu</i> }	Sets the metric value for a matched route
set metric-type { internal external type-1 type-2 }	Sets the metric type for a matched route being redistributed into IS-IS or OSPF
set next-hop <i>next-hop</i>	Sets the next-hop router address for a matched route
set tag <i>tag-value</i>	Sets a tag value for a matched route

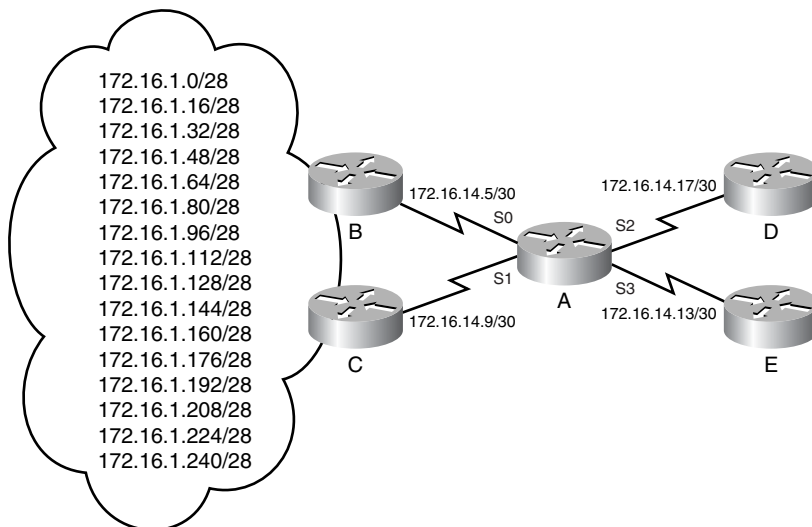
Review Questions

- 1 How are route maps similar to access lists? How are they different?
- 2 What are policy routes?
- 3 What are route tags?
- 4 In what way do route tags affect routing protocols?

Configuration Exercises

- 1 Configure policy routes for Router A in Figure 14-14 that forward packets from subnets 172.16.1.0/28 through 172.16.1.112/28 to Router D and forward packets from subnets 172.16.1.128/28 through 172.16.1.240/28 to Router E.

Figure 14-14 *The network for Configuration Exercises 1 through 3.*



- 2 Configure policy routes for Router A in Figure 14-14 so that packets from subnets 172.16.1.64/28 through 172.16.1.112/28 are forwarded to Router D if they are received from Router C. If packets from the same subnets are received from Router B, forward them to Router E. All other packets should be forwarded normally.
- 3 Configure policy routes for Router A in Figure 14-14 that will forward any packets destined for subnets 172.16.1.0/28 through 172.16.1.240/28, sourced from an SMTP port, to Router C. Route any other UDP packets destined for the same subnets to Router B. No other packets should be forwarded to Routers C or B by either the policy routes or the normal routing protocol.

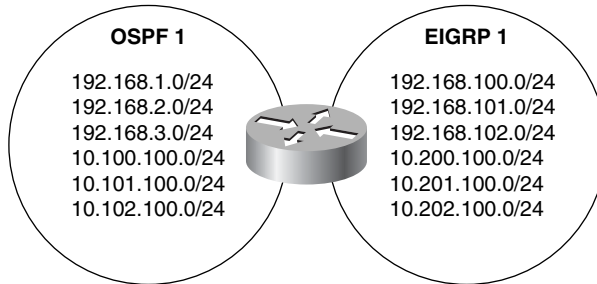
- 4 The OSPF and EIGRP configuration for the router in Figure 14-15 is

```

router eigrp 1
network 192.168.100.0
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 16

```

Figure 14-15 *The router for Configuration Exercises 4 and 5.*



Configure the router to redistribute internal EIGRP routes into OSPF as E1 routes with a metric of 10 and to redistribute external EIGRP routes into OSPF as E2 routes with a metric of 50. Of the networks and subnets shown in the EIGRP domain, all should be redistributed except 10.201.100.0/24.

- 5 Configure the router in Figure 14-15 to redistribute internal OSPF routes into EIGRP with a lower delay than external OSPF routes. Allow only the three Class C networks shown in the OSPF domain to be redistributed.

Troubleshooting Exercise

- 1 Given the following configuration:

```

interface TokenRing1
ip address 192.168.15.254 255.255.255.0
ip policy route-map Ex1
!
access-list 1 permit 192.168.0.0 0.0.255.255
access-list 101 permit host 192.168.10.5 any eq telnet
!
route-map Ex1 permit 5
match ip address 1
set ip next-hop 192.168.16.254

```

```
!  
route-map Ex1 permit 10  
match ip address 101  
set ip next-hop 192.168.17.254
```

The intention is to policy route all packets whose source address prefix is 192.168.0.0 to 192.168.255.255. The exception is that packets originating from the Telnet port of host 192.168.10.5 should be forwarded to 192.168.17.254. There are two errors in this configuration that are preventing the policy route from working correctly. What are they?





Numerics

- 3-way handshaking, 555
- 90-second timer (IGRP), 254

A

- A class IP addresses, 18–19**
 - address masks, 20
 - subnetting, 24
- ABRs (Area Border Routers), 369, 516**
- access-class command, 794**
- access-list compiled command, 789**
- Aceves, J.J. Garcia Luna, 137**
- Acknowledgment Number field (TCP), 43**
- acknowledgments, 364**
 - delayed, 365
 - direct, 365
- ACLs (access control lists)**
 - calling, 792–793
 - access-class command, 794*
 - categories of, 775
 - comments, adding, 783–784
 - dynamic, 786
 - editing, 780
 - effect on switching, 800
 - effect on system performance, 799
 - extended IP, 785, 789
 - effect on system performance, 788*
 - example of, 787–788*
 - features, 786–787*
 - sequence numbers, 786*
 - filter layers, numbering, 778, 780
 - fragments, handling, 787
 - ICMP, 791–792
 - implicit deny any filter, 777
 - keywords, 796–797
 - named, 780, 797
 - placement of, 799
 - reflexive, 794
 - example, 795–796*
 - parameters, 794*
 - sequence number, 781
 - sequentiality, 778
 - standard IP, 782–784
 - Boolean AND/OR functions, 784*
 - inverse masks, 785*
 - TCP, 790
 - Turbo ACLs, 789
 - UDP, 791
- acquiring hands-on experience, 805**
- active state, 273**
 - SIA, 274
- adding comments to ACLs, 783–784**
- address aggregation, 293–296**
 - CIDR, 296–297
- Address Family Identifier field (RIP), 173**
- Address Family Identifier field (RIPv2), 202**
- address masks, 20–21. *See also* subnet masks**
- Address Resolution Protocol. *See* ARP**
- address spaces, designing lollipop-shaped, 152**
- address summarization**
 - EIGRP, configuring, 315–316
 - OSPF
 - configuring, 435–440*
 - troubleshooting, 464–466*
- addresses, IP, 16**
 - internetworks, 18
 - masks, 20–21
 - reverse ARP, 36
 - RIP, 173
 - subnet mask, 24
 - subnets, 24
 - secondary RIP, 187
- adjacencies, 147, 267, 336–337**
 - building process, 358–362
 - formation of, 351
 - IS-IS, 517, 522–523
 - troubleshooting, 607–610*
 - troubleshooting, 456–458
- adjusting RIPng metrics, 233–235**
- administrative distance, 98, 175**
 - effect on redistribution, 633–639
 - setting route preferences, 722–723
- advertising default networks, 690**
- aging process (LSAs), 153–154**

alternative routes, creating for static routes,
96–97

always keyword (default-information originate command), 694

AND operator, 21

Boolean AND function, 784–785

anycast addresses, 55–56

application layer (TCP/IP), 6

Area ID, 368

areas, 160, 367

Area ID, 368

autonomous systems, 160–162

backbone, 369

IS-IS, 516–517

L1 routers, 517

maximum size of, 369

multiple L1 areas, IS-IS configuration, 604–606

NSSAs, 387–389

configuring, 428–434

partitioned areas, 371

path types, 391

stub areas, 386

configuring, 423–426

totally stubby areas, 387

configuring, 426–428

traffic types, 369

troubleshooting, 460–463

ARP (Address Resolution Protocol), 32–34, 40

gratuitous ARP, 36

packets

Hardware Address Length field, 33

Hardware Type field, 33

Operation field, 33–34

Protocol Address Length field, 33

proxy ARP, 35–36

reverse ARP, 36

arp command, 45, 123

arp timeout command, 45

ASBR Summary LSAs, 382

ASBRs (Autonomous System Border Routers),
370

AS-External LSAs, 488–489

assigning metrics to redistributed routes, 644

asynchronous updates, 145–146

ATT (Attached bit), 518

Attempt state (OSPF neighbors), 354

authentication

OSPF, 393

configuring, 441–444

IS-IS, *configuring, 587–591*

RIPng, 213

RIPv2, 210–212

configuring, 222–224

Authentication Information TLV, 544

auto-cost reference-bandwidth command, 345

automatic summarization, disabling on

EIGRP, 307–309

Autonomous System External LSAs, 383,
408–409

autonomous systems, 160–162

B

B class IP addresses, 18–19

address masks, 20

subnetting, 24

backbone, 369

virtual links, 373–374

backbone routers, 370

bandwidth, 134–135, 255–256

IGRP metric, *modifying, 257*

reference bandwidth, 345

BDRs (Backup Designated Routers),

election process, 342–344

best path determination, convergence, 135–137

binary numbering system, 769–770

converting to hex, 772–773

converting to decimal, 771

bitcount format, 24

bits, IP packet flags, 11

Boolean operators

AND, 784–785

OR, 784

boundaries, 174

boundary routers, 177–178

bounded updates, 262

broadcast networks, 339

IPv6 static routes, configuring, 94

broadcast updates, 138

building process for OSPF adjacencies, 358–362

bytes, 770

C

C class IP addresses, 18–19

- address masks, 20
- octet boundaries, 26
- subnetting, 24–25

calculating

- host addresses for subnets, 27–28
- hosts per subnet requirements, 25
- IGRP default metric, 260–261
- place values, 769

calling ACLs, 792–793

- access-class command, 794

case studies

- IPv6 redistribution with route maps, 759–760
- policy routing, 737–740
 - and QoS, 744–747
 - and redistribution, 747–751
 - router configuration, 743
 - standard IP access lists, 740–742
- protocol migration, 711, 714–716
 - routing loops, 712–714
- RIP configuration, 178–180
- route filtering, 708–709
 - filtering by routing process, 710
- route tagging, 752–758

categories of ACLs, 775

CCIE exam, preparing for, 803–804

- certification path, 805
- exam day, 807
- hands-on experience, 805
- study materials, 805–807
- training classes, 804

CDP (Cisco Discovery Protocol), 90, 680–681

- enabling, 697

cdp enable command, 698

cdp run command, 698

CEF (Cisco Express Forwarding), 104

- per destination load sharing, 103

Cerf, Vint, 5, 49

certification path for CCIE exam, 805

characteristics of distance vector protocols, 138

- broadcast updates, 138
- full routing table updates, 139
- neighbors, 138

Checksum field

- LSAs, 366
- TCP, 44

CIDR (Classless Interdomain Routing), 296–297

circular sequence number space, 150–151

Cisco IOS Software, switching method determination, 106

classes of IP addresses, 18

classful routing, 23

- RIP, 169, 175–178
 - replies, 170
 - requests, 170
 - subnetting, 176
 - summarization, 177

classless route lookups, 205–206, 682

classless routing, 24, 206

- redistribution into classful domains, 639–640
 - unmatching masks, 641–642

clear arp-cache command, 34, 45

clear ip accounting command, 801

CLNP (Connectionless Network Protocol), 513

columns in route tables, 82

command field (RIPv2), 202

commands, 46, 123

- access class, 794
- access-list compiled, 789
- arp, 45, 123
- arp timeout, 45
- auto-cost reference-bandwidth, 345
- cdp enable, 698
- cdp run, 698
- clear arp-cache, 34, 45
- clear ip accounting, 801
- config-router, 179
- debug arp, 32
- debug iegrp neighbors, 321
- debug ip icmp, 45
- debug ip packet, 123
- debug ip policy, 739
- debug ip rip, 180, 193
- debug ipv6 packet, 123
- default metric, 644
- default-information originate, 691–693
 - always keyword, 694
 - for IPv6 addresses, 695–697
- distance, 711–712, 715, 719–722
- distribute-list, 703, 706, 717–718

- eigrp stub, 310
- ip access-group, 792–793
- ip accounting access-violations, 801
- ip address, 45, 193
- ip cef, 123
- ip classless, 682, 698
- ip default network, 687–690
- ip default-network, 698
- ip load-sharing per packet, 104
- ip netmask-format, 24, 46
- ip ospf cost, 389
- ip ospf dead-interval, 338
- ip ospf priority, 346
- ip ospf transmit-delay, 346
- ip policy route-map, 743
- ip policy-route map, 737
- ip proxy-arp, 46, 123
- ip rip triggered, 190
- ip route, 123
- ip route cache, 123
- ip router isis, 574
- ipv6 cef, 123
- ipv6 nd prefix, 71
- ipv6 route command, 123
- ipv6 unicast-routing, 123
- isis isp-interval, 572
- lsp-gen-interval, 572
- maximum-paths, 393
- metric weights, 268
- metric-style wide, 560
- neighbor, 193
- network, 178, 194
- network area, 412–413
- no ip proxy-arp, 36
- no ip redirects, 40
- offset-list, 187, 194
- output-delay, 193–194
- passive-interface, 182, 194
- redistribute connected, 665
- redistribute eigrp, 754
- router, 179
- router odr, 697–698
- router rip, 178, 194
- set ip next-hop verify-availability, 738
- show arp, 34
- show cdp, 90
- show cdp neighbor detail, 123

- show clns is-neighbors, 608
- show clns is-neighbors detail, 597
- show interface, 256
- show ip access list, 800
- show ip accounting access-violations, 801
- show ip cef, 123
- show ip eigrp topology, 268
- show ip ospf database, 374
- show ip ospf database nssa-external, 409
- show ip ospf interface, 344
- show ip ospf neighbor, 351–352
- show ip route, 83, 123
- show ipv6 cef, 123
- show ipv6 detail, 123
- show ipv6 interface, 123
- show ipv6 rip, 228
- show ipv6 route, 123
- show isis database, 529
- summary address, 652
- timers active time, 275
- timers basic, 194
- trace, 12
- variance, 300
- which route, 534

comparing

- OSPFv2 and OSPFv3, 478–479
- RIPng and RIPv2, 213

compatibility of RIPv2 and RIPv1, 205

compatibility switches, 205, 215–217

config-router command, 179

configuring, 187–190

- ACLs, keywords, 796–797
- default routes, 681–682
 - in IPv6, 695–697*
 - ip default-network command, 687–690*
- dual-protocol routing, 711–716
- EIGRP, 297–299
 - address summarization, 315–316*
 - authentication, 316–317*
 - maximum paths for load sharing, 302–304*
 - multiple processes, 305–306*
 - stub routing, 309–315*
 - unequal-cost load balancing, 299–302*
- extended IP ACLs, 785–789
 - sequence number, 786*
- floating static routes, 97–100
- ICMP ACLs, 791–792

- IPv6 floating static routes, 100–101
- IS-IS
 - authentication*, 587–591
 - IPv4*, 574–579
 - IPv6*, 591–595
 - multiple area addresses*, 581–584
 - multiple LI areas*, 604–606
 - multiple topology mode*, 596–598
 - route leaking*, 598–603
 - route summarization*, 585, 587
 - routers*, 579–581
- load sharing, 102
 - CEF*, 103–104
- named ACLs, 797
- ODR, 697
- OSPF, 411–413
 - address summarization*, 435–440
 - authentication*, 441–444
 - Domain Name Service lookups*, 417–418
 - LSA filtering*, 441
 - network area command*, 412–413
 - NSSAs*, 428–434
 - on NBMA networks*, 446–454
 - over demand circuits*, 454–455
 - Router IDs*, 415–417
 - secondary addresses*, 418–422
 - stub areas*, 423–426
 - totally stubby areas*, 426–428
 - virtual links*, 444–445
- OSPFv3, 494–497
 - multiple instances on a link*, 499
 - on NBMA networks*, 500–505
 - stub area*, 497–498
- policy routing, 737–740
 - and QoS*, 744–747
 - and redistribution*, 747–751
 - global router configuration*, 743
 - standard IP access lists*, 740–742
- redistribution, 642
 - EIGRP into OSPF*, 646–651
 - IGRP into RIP*, 645–646
 - metrics*, 643–644
- reflexive ACLs, 794
 - example*, 795–796
 - parameters*, 794
- RIP, 178–180
 - discontiguous subnets*, 184–187
 - metrics*, 187–190
 - passive interfaces*, 181–182
 - unicast updates*, 182–184
- RIPng, 224–232
 - global parameters*, 232–233
- RIPv2, 214
 - authentication*, 222–224
- route maps, 734–736
- route tagging, 754–758
- standard IP ACLs, 782–784
 - Boolean AND/OR functions*, 784
 - inverse masks*, 785
- static routes, 84
 - IPv4*, 85–89
 - IPv6*, 90–94
- TCP ACLs, 790
- UDP ACLs, 791
- virtual links, 374
- connection-oriented services, TCP, 41–42**
- control protocols, ICMPv6 header format, 63–64**
- convergence, 135–137**
 - after router reboot in redistributed protocol environment, 635–637
- converting**
 - binary to hex, 772–773
 - decimal to binary, 771
 - routing protocols, 711–716
- CoS (Class of Service), 9**
- cost metric, 135, 155, 345, 389**
- counting to infinity, 142–144**
- cryptographic checksum, MD5, 211**
- CSNPs (Complete SNPs), 527, 552**

D

data links

- address masks, 21
- IP addresses
 - masks*, 20–21
 - subnets*, 24
- TCP/IP, 6

data structure of OSPF neighbors, 351

- components, 345–348, 352–353

- Database Description packets (OSPF), 399–400**
- data-link identifiers, ARP, 30**
- DC-bit, 395**
- DD (Database Description) packets, 358**
- debug arp command, 32**
- debug eigrp neighbors command, 321**
- debug ip icmp command, 45**
- debug ip packet command, 123**
- debug ip policy command, 739**
- debug ip rip command, 180, 193**
- debug ipv6 packet command, 123**
- debugging route table entries, 96**
- decimal numbering system, 769**
 - converting to binary, 771
 - converting to hexadecimal, 772–773
 - IP addresses, 17
- decision points for OSPF adjacency formation, 357**
- decision process (IS-IS), 531–534**
- default metrics, IGRP**
 - calculating, 260–261
 - modifying, 257
- default networks, 253**
 - advertising, 690
- default routes, 677–679, 683**
 - classless route lookups, 682
 - configuring, 681–682
 - IPv4, 695–697*
 - in IPv6, 695*
 - ip default-network command, 687–690*
 - filtering, 706
 - flapping, 694
 - gateway of last resort, 683
 - in hub-and-spoke topologies, 678
 - large-scale use of, 678
 - redistribution, 693
 - risk of suboptimal routing, 679
 - static, 682
 - “don’t care” bits, 683*
 - redistribution, 684–686*
- default IGRP timers, modifying, 255**
- default-information originate command, 691–693**
 - always keyword, 694
 - for IPv6 addresses, 695–697
- default-metric command, 644**
- delay metric, 135, 255–257**
 - modifying, 257
- delayed acknowledgments, 365**
- demand circuits, 393**
 - OSPF, configuring, 454–455
 - presumption of reachability, 394
- depletion of IPv4 addresses, 49**
- designated routers (IS-IS), 523–526**
- designing**
 - lollipop-shaped address spaces, 152
 - subnets, 24–25
- Destination Address field (IPv6 packets), 60**
- destination ports, TCP, 43**
- devices, boundary routers, 177–178**
- DF bit, 10**
- DiffServ (Differentiated Services framework), 9**
- diffusing computations, 262**
 - feasible successor selection process, 273–287
- digits, 769**
- Dijkstra, E.W., 266**
- Dijkstra’s algorithm, 156–159**
- direct acknowledgments, 366**
- directly connected subnets, 176**
- disabling**
 - automatic summarization on EIGRP, 307–309
 - split horizon, 722
 - summarization on RIPv2, 221
- discontiguous subnets, 85, 178**
 - configuring, 184–187
 - RI, 185
 - summarizing RIPv2 routes across, 220–222
- displaying**
 - IGRP delay, 256
 - IPv6 addresses, 90
 - IS-IS link state database, 529–530
 - IS-IS neighbor table, 523
 - RIPng processes, 228
- distance command, 711–712, 715, 719–722**
- distance vector routing protocols, 137, 261**
 - asynchronous updates, 145–146
 - characteristics of, 138
 - broadcast updates, 138*
 - full routing table updates, 139*
 - neighbors, 138*
 - periodic updates, 138*

counting to infinity, 143–144

holddown timers, 144–145

RIP

configuring, 178–180

metrics, configuring, 187–190

route filtering, 702

route invalidation timers, 140–141

routing by rumor, 139–140

split horizon, 141

triggered updates, 144

distributed database protocols. See link-state protocols

distribute-list command, 703, 706, 717–718

domain authentication, IS-IS configuration, 590–591

Domain Name Service lookups, OSPF configuration, 417–418

domains

OSPF, areas, 368

IS-IS, domain-wide prefix distribution extension, 556–557

Don't Fragment (DF) bit, 10

dotted-decimal format, 17, 24

Down state (OSPF neighbors), 353

DRs (Designated Routers), 341

DRothers, 344

election process, 342–344

DSCP (Differentiated Services Code Point), 9

DUAL (Diffusing Update Algorithm), 266–267

feasible successors, selection process, 270–287

input events, 273

dual-protocol routing, configuring, 711–716

Duplicate Address Detection, 73–74

dynamic ACLs, 786

Dynamic Hostname Exchange, 562

dynamic routing protocols, RIP. See RIP (Routing Information Protocol)

E

echo packets, tracing, 112

Echo Request/Reply messages (ICMP), 39–40

ECN (Explicit Congestion Notification), 10

editing

ACLs, 780

route maps, 735

EGPs (Exterior Gateway Protocols), 162

EIGRP (Enhanced Interior Gateway Routing Protocol), 263

address aggregation, 293–296

CIDR, 296–297

address summarization, configuring, 315–316

adjacencies, 267

authentication, configuring, 316–317

automatic summarization, disabling, 307–309

configuring, 297–299

diffusing computations, 262

DUAL, 266–267

feasible successors, selection process, 270–284, 287

input events, 273

load sharing, configuring maximum paths, 302–304

metrics, 270

delay, 268

FC, 267

FD, 267

feasible successors, 267

variance, 300

missing neighbors, troubleshooting, 318–323

multiple processes, configuring, 305–306

neighbor discovery/recovery, 265–266

neighbor table, 266

packets, 264

header format, 288–289

Hello, 265

sequence numbers, 264

TLVs, 289–293

protocol-dependent modules, 264

Q Count, 266

redistribution into OSPF, 656

configuring, 646–651

RTO, 265–266

RTP, 264

SRTT, 265

stub routing, configuring, 309–315

stuck-in-active neighbors, troubleshooting, 323, 325–327

summarization, 654–655

timers, mulicast flow timer, 265

topological table, contents of, 268

troubleshooting, 317

unequal-cost load balancing, configuring, 299–302

updates, 262

eigrp stub command, 310

election process for DRs/BDRs, 342–344

electrical protocols, 6

embedded IPv4 addresses, 58

enabling

CDP, 697

classless route lookups, 206

per packet load sharing, 104

process switching, 105

encapsulation (SNAP), ARP, 34

environments, 24

EPHOS (European Procurement Handbook for Open Systems), 513

equal-cost load balancing, 102, 175

error correction in IP packets, 13

ES (End System), 515

establishing adjacencies

decision points, 357

OSPF, 351

Ethernet, 184

examples

of diffusing computations, 276–284, 287

of extended IP ACLs, 787–788

of reflexive ACLs, 795–796

exams, CCIE

certification path, 805

preparing for, 803–807

training classes, 804

Exchange state (OSPF neighbors), 354

expiration timer, RIP, 171

explicit acknowledgments, 359, 364

ExStart state (OSPF neighbors), 354

extended IP ACLs, 785, 789

effect on system performance, 788

example of, 787–788

features, 786–787

placement of, 799

sequence numbers, 786

extended metrics, 596

Extended Ping utility, 10

extension headers, 61–63

functions performed, 62

extensions to IS-IS

3-way handshaking, 555

domain-wide prefix distribution, 556–557

Dynamic Hostname Exchange, 562

IPv6 routing, 560

mesh groups, 566–571

Multi Topology routing, 562–564
wide metrics, 558

exterior routes, 253

External Attributes LSAs, 383–385

external routes, 532

summarization, 435

external traffic, 369

F

failed routes, tracing, 109–112

IPv6, 120–122

fast switching, per destination load sharing, 105

FC (feasible condition), 267

FD (feasible distance), 267, 270

feasible successors, 267

diffusing computations, 273–284, 287

selection process, 270–276

features

of extended IP ACLs, 786–787

of OSPF, 335

of TCP ACLs, 790

fields

of IS-IS Hello PDUs, 541

of IS-IS LSPs, 546–547

of RIPng messages, 214

of TCP header, 42–44

filter layers, numbering, 778, 780

filtering

implicit deny any, 777

IPv6 routes, 704–705

prefix lists, 798

sequentiality, 778

first octet rule, 18–20

Flags field (TCP), 43

flapping routes, 694

flash updates. See triggered updates

Fletcher algorithm, 366

floating static routes, 97–100

IPv6, 100–101

flooding, 147

LSAs, 363

acknowledgments, 364–366

checksum, 366

- sequence numbers, 148–149, 366
 - circular sequence number space, 150–151*
 - linear sequence number space, 149–150*
 - lollipop-shaped sequence number space, 152–153*

flooding delays (IS-IS), 571–573

Flow Label field (IPv6 packets), 59

Floyd, Sally, 145

flush timer (RIP), 171

format

- of IPv6 multicast addresses, 57
- of OSPFv3 LSAs, 484
 - AS-External LSAs, 488–489*
 - Inter-Area Prefix LSAs, 486–487*
 - Inter-Area Router LSAs, 487*
 - Intra-Area Prefix LSAs, 491–493*
 - Link LSAs, 489–491*
 - Network LSAs, 486*
 - Router LSAs, 484–486*
- of RA messages, 66
- of Redirect messages, 69
- of RIPng messages, 213–214
- of RIPv2 messages, 201–204

formation of OSPF adjacencies, 351

- decision points, 357

Fragment Offset field (IP packets), 11

fragmenting IP packets, 10

fragments, processing through ACLs, 787

Frame-Relay mapping, OSPFv3, 506–509

full routing table updates, 139

Full state (OSPF neighbors), 354

functional protocols (TCP/IP), 6

G

garbage collection (RIP), 171

gateway of last resort, 683

general TLV fields, 289

global parameters for RIPng configuration, 232–233

global unicast addresses, 53

GOSIP (Government Open Systems

Interconnection Profile), 513

gratuitous ARP, 36

Group Membership LSAs, 385

group pacing, 377

H

Hardware Address Length field (ARP packets), 33

Hardware Type field (ARP packets), 33

Header Checksum field (IP packets), 13

Header Length field

- IP packets, 8
- TCP packets, 43

headers

- EIGRP packet format, 288–289
- IP packets, 7–8
 - flags, 10*
 - Fragment Offset field, 11*
 - Header Checksum field, 13*
 - identifiers, 10*
 - Options field, 14–16*
 - options field, 14*
 - size, 8*
 - TOS, 8*
 - total length, 10*
 - TTL field, 12–13*
- IS-IS PDU fields, 535–536
- TCP, 42–43
- UDP, 44–45

Hedrick, Charles, 169

Hello packets

- EIGRP, 264
 - hold time, 265*
- OSPF, 338–339, 398–399

Hello PDUs, 539

- Area Addresses TLV, 542–543
- Authentication Information TLV, 544
- fields, 541
- Intermediate System Neighbors TLV, 543
- IP Interface Address TLV, 545
- Padding TLV, 543
- Protocols Supported TLV, 545

Hello protocol, 147, 338

HelloInterval, 338

hexadecimal numbering system,**769–772**

converting to decimal, 772–773

IP address format, 24

hexadecimal IPv6 representation, 51

hippity bit, 528**holddown**

RIP, 172

IGRP, 255

holddown timers, 144–145**hop count metric, 134, 170–173****Hop Limit field (IPv6 packets), 60****host addresses, subnet mask octet boundaries, 26****hosts**

IP addresses, 16–17

quantity per subnet requirement, calculating, 25

host-to-host layer (TCP/IP)

TCP, 41–44

UDP, 44–45

Host-to-Host Layer Protocol, 13**“hot potato” routing, 392****hub-and-spoke topologies**

ODR, 680

stub routers, 678

Huitema, Christian, 514

ICMP (Internet Control Message Protocol),**37–38**

ACLs, 791–792

echo packets, tracing, 112

Echo Request/Reply messages, 39–40

redirect packets, 40

Router Advertisement packets, 40

Router Selection packets, 40

ICMPv6 header format, 63–64**identifying IPv6 address types, 54****IGPs (Interior Gateway Protocols), 161****IGRP (Interior Gateway Routing Protocol), 253**

default timers, modifying, 255

delay, 256–257

holddown, 255

load, 259

metrics, 255

modifying, 256–257

process domains, 252

redistribution into RIP, configuring, 645–646

reliability, 258–259

similarities to RIP, 251

timers

*90-second, 254**sleeptime, 255***Implicit Acknowledgments, 359****implicit acknowledgments, 364****implicit deny any filter, 777****improving SPF efficiency on IS-IS networks,****573–574****InfTransDelay, 366****Init state (OSPF neighbors), 354****input events, 273**

for OSPF interface state machine, 350

for OSPF neighbors, 356–357

inter-area paths, 391**Inter-Area Prefix LSAs, 486–487****Inter-Area Router LSAs, 487****inter-area summarization, 435****inter-area traffic, 369****Inter-Domain Routing Protocol****Information TLV, 551****Interface ID, 53****interface state machine (OSPF), 348–349**

input events, 350

interfaces, OSPF, 344

data structure, 344–348

input events, 350

states, 348–349

interface-specific summarization, 653**interior routes, 252****Intermediate System Neighbors TLV, 549****internal routers, 369****internal routes, 532****Internet layer (TCP/IP), 6****internetwork IP addresses, 17–18****intra-area paths, 391****Intra-Area Prefix LSAs, 491–493****intra-area traffic, 369****Intradomain Routing Protocol Discriminator, 535****invalid timers**

IGRP, 254

RIP, 171

invalid updates, sources of, 210

- inverse masks, 784–785
- ip access-group command, 792–793**
- ip accounting access-violations command, 801**
- ip address command, 45, 193**
- IP Address field**
 - RIP, 173
 - RIPv2, 203
- IP addresses, 16–17**
 - bitcount format, 24
 - dotted decimal format, 24
 - dotted-decimal format, 17
 - first octet rule, 18–20
 - hexadecimal format, 24
 - internetworks, 17
 - masks, 20–21
 - prefix lists, 798
 - reverse ARP, 36
 - RIP, 173
 - subnet masks, troubleshooting, 29
 - subnets, 22–24, 27–28
 - designing, 24–26*
 - summary routes, 94
- IP packets, 7**
 - flags, 10
 - Fragment Offset field, 11
 - Header Checksum field, 13
 - header length, 8
 - identifiers, 10
 - Options field, 14–16
 - Protocol field, 13
 - TOS (Type of Service), 8
 - total length, 10
 - TTL (Time to Live) field, 12
 - versions of, 8
- ip cef command, 123**
- ip classless command, 682, 698**
- ip default-network command, 687–690, 698**
- IP External Reachability TLV, 551**
 - U/D bit, 556
- IP External Routes TLV, 291–293**
- IP Interface Address TLV, 545**
- IP Internal Reachability TLV, 550–551**
 - U/D bit, 556
- IP Internal Routes TLV, 290–291**
- ip load-sharing, 123**
- ip load-sharing command, 123**
- ip load-sharing per-packet command, 104**
- ip netmask-format, 24**
- ip netmask-format command, 24, 46**
- ip ospf cost command, 389**
- ip ospf dead-interval command, 338**
- ip ospf priority command, 346**
- ip ospf transmit-delay command, 346**
- ip policy route-map command, 737, 743**
- ip proxy-arp command, 46, 123**
- ip redirects command, 46**
- ip rip triggered command, 190**
- ip route command, 123**
- ip route-cache command, 123**
- ip router isis command, 574**
- IP-specific TLV fields, 290**
- IPv4, 8**
 - configuring on IS-IS, 574–579
 - depletion of, 49
 - static routes, configuring, 85–89
- IPv6, 8**
 - address representation, 50–51
 - anycast addresses, 55–56
 - configuring on IS-IS, 591–595
 - default-information originate command, default
 - route configuration, 695–697
 - double colon representation, 51
 - embedded IPv4 addresses, 58
 - extension headers, 61–63
 - failed routes, tracing, 120–122
 - floating static routes, 100–101
 - global unicast addresses, 53
 - identifying address types, 54
 - local unicast addresses, 55
 - modifying static entries on replaced routers,
 - 116–120
 - multicast addresses, 56–57
 - NDP, 64–65
 - address autoconfiguration, 71–73*
 - Duplicate Address Detection, 73–74*
 - messages, 65*
 - Neighbor Address Resolution, 74–75*
 - Neighbor Unreachability Detection, 76–77*
 - privacy addresses, 75–76*
 - router discovery, 70–71*
 - need for, 50
 - packet header format, 58–60

- prefixes, 52
- redistribution with route tags, 759–760
- RIPng, 212
 - configuring*, 224–232
 - next-hop route entries*, 214
 - route summarization*, 235
- route filtering, 704–705
- routing with IS-IS, 560
- static routes, *configuring*, 90–94
- unspecified addresses, 52

ipv6 cef command, 123**ipv6 nd prefix command, 71****IPv6 Prefix field (RIPng messages), 214****ipv6 route command, 123****ipv6 unicast-routing command, 123****IS (intermediate system), 515****IS-IS (intermediate System-to-Intermediate System), 513**

- adjacencies, 517
 - troubleshooting*, 607–610
- areas, 516
- authentication, *configuring*, 587–591
- extensions
 - 3-way handshaking*, 555
 - domain-wide prefix distribution*, 556–557
 - Dynamic Hostname Exchange*, 562
 - IPv6 routing*, 560
 - mesh groups*, 566–571
 - Multi Topology routing*, 562–564
 - wide metrics*, 558
- flooding delay, 571–572
 - controlling*, 572–573
- IPv4, *configuring*, 574–579
- IPv6, *configuring*, 591–595
- layered network architecture, 521
 - subnetwork dependent functions*, 521–526
 - subnetwork independent functions*, 526–534
- link-state database
 - displaying*, 529–530
 - troubleshooting*, 607, 610–614
- LSPs, 546
 - fields*, 546–547
 - Inter-Domain Routing Protocol Information TLV*, 551

- Intermediate System Neighbors TLV*, 549
- IP External Reachability TLV*, 551
- IP Internal Reachability TLV*, 550–551
- TLVs*, 547–548

- multiple area addresses, *configuring*, 581–584
- multiple L1 areas, *configuring*, 604–606
- multiple topology mode, *configuring*, 596–598
- neighbor table, *displaying*, 523
- NETs, 519

- format*, 519–520

- selecting*, 520

- on NBMA networks, *troubleshooting*, 615–619
- PDU

- format*, 534

- header fields*, 535–536

- Hello PDU format*, 539

- Hello PDUs*, 541–545

- IS-IS LSPs*, 546–551

- IS-IS Sequence Number PDU format*, 552–553

- TLV fields*, 536–539

- redistribution into RIP/RIPng, 663–669

- route leaking, *configuring*, 598–603

- route summarization, *configuring*, 585–587

- router types, *configuring*, 579–581

- similarities with OSPF, 514

- SNPs, 527

- SPF efficiency, *improving*, 573–574

- System ID, 519

- troubleshooting*, 606

- virtual link support, 518

isis lsp-interval command, 572**IS-IS Sequence Number PDUs, 552**

- LSP Entries TLV, 553

J–K

Jacobson, Van, 145**Kahn, Bob, 5, 49**

- key chains, *configuring for RIPv2 authentication*, 223**

- keywords for ACL configuration, 796–797**

L

- L1 LSPs, TLVs, 547**
- L1 routers, 517**
- L1/L2 routers, 517**
- L2 LSPs, TLVs, 548**
- L2 routers, 517**
- LAN Hellos, 540**
- layered network architecture, IS-IS, 521–534**
- Level 3 Routing, 162**
- linear sequence number space, 149–150**
- link costs, 155**
- Link LSAs, 489, 491**
- Link State Acknowledgment packets (OSPF), 402**
- link state refresh process, 376**
- Link State Request packets (OSPF), 400**
- Link State Update packets (OSPF), 401**
- link-local unicast addresses, 55**
- links, OSPF, 344**
- link-state advertisements, see LSAs**
- link-state database, 147, 154–156, 336, 363, 374**
 - IS-IS
 - displaying, 529–530*
 - troubleshooting, 607, 610–614*
 - LSA group pacing, 377
- link-state routing protocols, 146, 262**
 - areas, 160–162
 - Dijkstra’s algorithm, 156–157
 - router adaptation, 157–159*
 - flooding, 147
 - sequence numbers, 148–153*
 - IS-IS. *See* IS-IS
 - LSAs, 146–147
 - aging process, 153–154*
 - OSPF. *See* OSPF
 - neighbor discovery, 147
 - route filtering, 702
- LIRs (Local Internet Registries), 54**
- load (IGRP), 259**
- load balancing, 137, 175**
 - unequal-cost, EIGRP configuration, 299–302
- load sharing**
 - CEF, 103–104
 - configuring, 102
 - maximum paths, EIGRP configuration, 302–304
 - per destination, 103–105, 301
 - per packet
 - enabling, 104*
 - process switching, 105–106*
- Loading state (OSPF neighbors), 354**
- local computations, 273**
- local unicast addresses, 55**
- lollipop-shaped sequence number space, 152–153**
- lookups**
 - classless, 682
 - recursive, 107–109
- loopback interfaces, 337, 340**
- LSAs (link-state advertisements), 146–147, 336, 402, 702**
 - acknowledgments, 359
 - during flooding process, 364–366*
 - aging process, 153–154
 - ASBR Summary LSAs, 382
 - Autonomous System External LSAs, 408–409
 - External Attributes LSAs, 385
 - filtering, OSPF configuration, 441
 - flooding, 147, 363, 366
 - checksum, 366*
 - sequence numbers, 148–150, 152–153*
 - Group Membership LSAs, 385
 - group pacing, 377
 - header format, 403
 - Network LSAs, 379–381, 406
 - Network Summary LSAs, 381–382, 407–408
 - NSSA External LSAs, 385, 409
 - Opaque LSAs, 385
 - Options field, 410
 - OSPFv3, 481–483
 - format, 484*
 - Inter-Area AS-External LSAs, 488–489*
 - Inter-Area Prefix LSAs, 486–487*
 - Inter-Area Router LSAs, 487*
 - Intra-Area Prefix LSAs, 491–493*
 - Link LSAs, 489–491*
 - Network LSAs, 486*
 - Router LSAs, 484–486*
 - Router LSAs, 378–379, 403–406
 - sequence numbers, 367
 - types of, 377
- LSP Entries TLV, 553**
- lsp-gen-interval command, 572**
- LSPs**
 - ATT, 518
 - flooding delays, 571–573
 - controlling, 572–573

M

- MAC addresses, ARP, 31
- MAC-to-EUI64 conversion, 72–73
- masks, 176. *See also* subnet masks
 - octet boundaries, 25–27
- MaxAge timer, link state refresh, 376
- maximum-paths command, 393
- MD5, 211
- mechanical protocols (TCP/IP), 6
- medium internetworks, 18
- medium networks, 18
- mesh groups, 566–571
- messages
 - ICMP, Echo Request/Reply, 39–40
 - NDP, 65
 - OSPFv3, 479–480
 - RIP, 172–175
 - RIPng, 213–214
 - RIPv2, 201–204
- Metric field
 - RIP, 173
 - RIPng, 214
- metric weights command, 268
- metrics, 84, 133. *See also* administrative distance
 - and redistribution, 633
 - bandwidth, 134–135
 - cost, 135, 155, 345, 389
 - delay, 135
 - EIGRP, 270
 - delay, 268
 - FC, 267
 - FD, 267, 270
 - feasible successors, 267
 - variance, 300
 - hop-count, 134
 - IGRP, 255
 - default metric, calculating, 260–261
 - load, 259
 - modifying, 256
 - reliability, 259
 - redistribution, configuring, 643–644
 - reliability, 135
 - RIP, 170, 187–190
 - RIPng, adjusting, 233–235
 - triggered updates, 144

- metric-style wide command, 560
- MF bit (More Fragments), 11
- microflows, 59
- migrating to new routing protocols, 711–716
- minimizing impact of updates on RIP, 190–193
- minimumLSPTransmissionInterval, 528
- misconfigured OSPF summarization,
 - troubleshooting, 463–466
- missing EIGRP neighbors, troubleshooting,
 - 318–323
- missing route table entries, 83
- modifying
 - cost metric, 345
 - IGRP default metrics, 256–257
 - IGRP default timers, 255
 - RouterDeadInterval, 338
- More Fragments (MF) bit, 11
- MOSPF (Multicast OSPF), 385
- MT (Multi Topology) routing, 562–564
- MTU (Maximum Transmission Unit), 10
 - IP packets, 10
- multiaccess, 446
- multicast addresses, 56–57
- multicast flow timer, 265
- multiple area addresses, configuring on IS-IS,
 - 581–584
- multiple EIGRP processes, configuring, 305–306
- multiple L1 areas, IS-IS configuration, 604–606
- multiple topology mode, IS-IS configuration,
 - 596–598
- mutual redistribution, 657, 701
 - at multiple redistribution points, 716–722

N

- NA (Neighbor Advertisement) messages, 65
 - format, 68
- named ACLs, 780, 797
- NAT (Network Address Translation), 49
- NBMA networks, 340
 - IS-IS, troubleshooting, 615–619
 - OSPF, configuring, 446–454, 500–505
- NDP (Neighbor Discovery Protocol), 64–65
 - address autoconfiguration, MAC-to-EUI64
 - conversion, 71–73
 - Duplicate Address Detection, 73–74
 - messages, 65–66

- Neighbor Address Resolution, 74–75
- neighbor discovery, 70–71
- Neighbor Unreachability Detection, 76–77
- privacy addresses, 75–76

Neighbor Address Resolution, 74–75

neighbor command, 193

neighbor state machine (OSPF), 353–357

neighbor table, 337

- EIGRP, 266

Neighbor Unreachability Detection, 76–77

neighbors, 138

- adjacencies

- formation of, 351*

- troubleshooting, 456–458*

- discovery/recovery

- adjacencies, 147*

- EIGRP, 265–266*

- EIGRP, adjacencies, 267

- IS-IS, 522–523

- OSPF, 350

- adjacency building process, 358–362*

- data structure, 351–353*

- input events, 356–357*

NETs (Network Entity Titles)

- format, 519

- selecting, 520

network area command, 412–413

network command, 178, 194

Network LSAs, 379, 406, 486

Network Summary LSAs, 382, 407–408

Next Header field (IPv6 packets), 60

Next Hop field (RIPv2), 203

next-hop addresses on replaced routers,

- troubleshooting, 116–120*

next-hop route entries (RIPng), 214

NLA (Next-Level Aggregator), 53

no ip proxy-arp command, 36

no ip redirects command, 40

nonperiodic updates, 262

not-so-stubby areas, 387–389

NPDUs (Network Protocol Data Units), 515

NS (Neighbor Solicitation) messages, 65

NS messages, format, 67

NSSA External LSAs, 385, 409

NSSAs (not-so-stubby areas), 388–389

- configuring, 428–434

null interface, 656

numbering ACL filter layers, 778–780

numbering systems

- binary, 769–770

- hexadecimal, 769–773

- place values, 769

O

octets, 770–773

- binary, 770

- boundary subnet masks, 25–27

- hexadecimal, 771–772

- subnetting, 26

ODR (On-Demand Routing)

- CDP, 680–681

- configuring, 697

- redistribution, 697

offset lists, RIP, 189

offset-list command, 187, 194

OL (Overload) bit, 528

opaque LSAs, 385

operation of OSPFv3, 477

Operation field (ARP packets), 33–34

operators

- Boolean AND function, 21, 784–785

- Boolean OR function, 784

optical protocols, 6

Options field

- IP packets, 14–16

- LSAs, 410

- TCP, 44

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

OSPFv3

- ABRs, 369

- address summarization, configuring, 435–440

- adjacencies, 336–337

- formation of, decision points, 357*

- troubleshooting, 456–458*

- areas, 367

- Area ID, 368*

- backbone, 369*

- maximum size of, 369*

- not-so-stubby areas, 387–389*

- partitioned areas, 371*

- path types, 391*

- stub areas, 386*

- totally stubby areas, 387*

- traffic types, 369*

- area-wide problems, troubleshooting, 459
- ASBRs, 370
- authentication, 393
 - configuring, 441–444
- backbone routers, 370
- BDRs, 342
 - election process, 342–344
- broadcast networks, 339
- configuring, 411–413
 - on NBMA networks, 446–454
 - over demand circuits, 454–455
- cost metric, modifying, 345
- DD packets, 358
- Domain Name Service lookups, configuring, 417–418
- DRs, 341
 - election process, 342–344
- features of, 335
- Hello packets, 338–339
- Hello protocol, 338
- interface state machine, 348–349
 - input events, 350
- interfaces, data structure, 344–348
- internal routers, 369
- isolated areas, troubleshooting, 460–463
- link state database, 374
 - LSA group pacing, 377
- link state refresh process, 376
- link-state database, 336, 363
- links, 344
- loopback interfaces, 337
- LSAs, 402
 - ASBR Summary LSAs, 382
 - Autonomous System External LSAs, 408–409
 - External Attributes LSAs, 385
 - filtering, 441
 - flooding process, 363
 - Group Membership LSAs, 385
 - group pacing, 377
 - header format, 403
 - Network LSAs, 379–381, 406
 - Network Summary LSAs, 381–382, 407–408
 - NSSA External LSAs, 385, 409
 - Opaque LSAs, 385
 - Options field, 410
 - Router LSAs, 378–379, 403–406
 - sequence numbers, 367
 - types of, 377
- metrics, cost, 345, 389
- misconfigured summarization, troubleshooting, 463–466
- MOSPF, 385
- NBMA networks, 340
- neighbor table, 337
- neighbors, 350
 - data structure, 351–353
 - input events, 356–357
 - state machine, 353–357
- NSSAs, 388–389
 - configuring, 428–434
- over demand circuits, 393
- oversubscription, 394
- packets, 395–396
 - Database Description packets, 399–400
 - header format, 396–398
 - Hello packets, 398–399
 - Link State Acknowledgment packets, 402
 - Link State Request packets, 400
 - Link State Update packets, 401
- point-to-multipoint networks, 340
- point-to-point networks, 339
- PollInterval, 351
- presumption of reachability, 394
- pseudonodes, 341
- redistribution into EIGRP, configuring, 646–651
- reference bandwidth, 345
- refresh process versus IS-IS Update process, 526
- route table entries, lookups, 393
- Router IDs, 337
 - configuring, 415–417
- RouterDeadInterval, 338
- routing table entries
 - destination types, 390–391
 - path types, 391–392
- secondary addresses, configuring, 418–422
- similarities with IS-IS, 514
- stub areas, configuring, 423–426
- stub networks, 340
- totally stubby areas, configuring, 426, 428
- transit networks, 340

troubleshooting, 455–456
 virtual links, 340, 373
 configuring, 374, 444–445

OSPFv3, 477

configuring, 494–497
 Frame-Relay mapping, 506–509
 LSAs, 481–483
 AS-External LSAs, 488–489
 format, 484
 Inter-Area Prefix LSAs, 486–487
 Inter-Area Router LSAs, 487
 Intra-Area Prefix LSAs, 491–493
 Link LSAs, 489–491
 Network LSAs, 486
 Router LSAs, 484, 486
 messages, 479–480
 multiple instances on a link, *configuring*, 499
 on NBMA networks, *configuring*, 500–505
 operation, 477
 redistribution into RIPng, 657–663
 stub areas *configuring*, 497–498
 troubleshooting, 506
 versus OSPFv2, 478–479

out keyword, 706

output-delay command, 193–194

oversubscription, 394

P

packet filtering routers, 800

packets, 34

ARP, 30–34
 Hardware Address Length field, 33
 Hardware Type field, 33
 Operation field, 33–34
 Protocol Address Length field, 33
 EIGRP
 header format, 288–289
 Hello, 265
 sequence numbers, 264
 TLVs, 289–293
 fragments, processing through ACLs, 787
 Hello (OSPF), 338–339
 ICMP, 37–39
 Echo Reply, 39
 Echo Request, 39

redirect, 40
Router Advertisement, 40
Router Selection, 40

IP, 7

flags, 10
Fragment Offset field, 11
Header Checksum field, 13
header length, 8
identifiers, 10
Options field, 14–16
Protocol field, 13
TOS (Type of Service), 8
total length, 10
TTL field, 12

IPv6, 58

Destination Address field, 60
Flow Label field, 59
Hop Limit field, 60
Next Header field, 60
Payload Length field, 60
Source Address field, 60
Traffic Class field, 59

LSAs, 402

acknowledgment during flooding process,
 364–366
ASBR Summary LSAs, 382
AS-External LSAs, 488–489
Autonomous System External LSAs,
 408–409
External Attributes LSAs, 385
Group Membership LSAs, 385
header format, 403
Inter-Area Prefix LSAs, 486–487
Inter-Area Router LSAs, 487
Intra-Area Prefix LSAs, 491–493
Link LSAs, 489–491
Network LSAs, 379–381, 406, 486
Network Summary LSAs, 381–382,
 407–408
NSSA External LSAs, 385, 409
Opaque LSAs, 385
Options field, 410
OSPFv3, 481–484
Router LSAs, 378–379, 403–406,
 484–486
sequence numbers, 367
types of, 377

- OSPF, 395–396
 - Database Description*, 358
 - Database Description packets*, 399–400
 - header format*, 396–398
 - Hello packets*, 398–399
 - Link State Acknowledgment packets*, 402
 - Link State Request packets*, 400
 - Link State Update packets*, 401
- PDU. *See* PDU (Protocol Data Units)
- Padding TLV, 543**
- parameters**
 - of extended IP ACLs, 786–787
 - of reflexive ACLs, 794
- Partial Route Calculation, 573**
- partial updates, 262**
- partitioned areas, 371**
- passive interfaces, configuring, 181–182**
- passive state, 273**
- passive-interface command, 182, 194**
- path types (OSPF), 391–392**
- Payload Length field (IPv6 packets), 60**
- P-bit, 388**
- PDU (Protocol Data Units), 515**
 - format, 534
 - header fields, 535–536
 - Hello PDUs
 - Area Addresses TLV*, 542–543
 - Authentication Information TLV*, 544
 - fields*, 541
 - format*, 539
 - Intermediate System Neighbors TLV*, 543
 - IP Interface Address TLV*, 545
 - Padding TLV*, 543
 - Protocols Supported TLV*, 545
 - IS-IS LSPs, 546
 - fields*, 546–547
 - Inter-Domain Routing Protocol Information TLV*, 551
 - Intermediate System Neighbors TLV*, 549
 - IP External Reachability TLV*, 551
 - IP Internal Reachability TLV*, 550–551
 - TLVs*, 547–548
 - IS-IS Sequence Number PDU format, 552
 - LSP Entries TLV*, 553
 - TLV fields, 536–539
- per destination load sharing, 105, 301**
 - CEF, 103–104
- per packet load sharing**
 - enabling, 104
 - process switching, 105–106
- periodic updates, 138**
- Perlman, Radia, 152, 514**
- PHB (Per-Hop Behavior), 9**
- physical layer (TCP/IP), 6**
- place values of numbering systems, 769**
- placement of ACLs, 799**
- point-to-multipoint networks, 340**
- point-to-point connections, TCP, 41**
- point-to-point Hellos, 540**
- point-to-point networks, 339**
- policy routing, 731**
 - and QoS, 744–747
 - and redistribution, 747–751
 - case study, 737–40
 - match commands, 733
 - set commands, 734
 - standard IP access lists, case study, 740–743
- PollInterval, 351**
- port numbers, 42**
- ports**
 - sockets, 43
 - TCP, 43
- Prefix Length field (RIPng messages), 214**
- prefix lists, 798**
- preparing for CCIE exam, 803–804**
 - certification path, 805
 - exam day, 807
 - hands-on experience, 805
 - study materials, 805–807
 - training classes, 804
- presumption of reachability, 394**
- privacy addresses, 75–76**
- procedural protocols, 6**
- process domains, 252**
- process switching, 105–106**
- promiscuous ARP, 35–36**
- Protocol Address Length field (ARP packets), 33**
- protocol conflicts, troubleshooting, 112, 115–116**
- Protocol field, 13**
- protocol migration, case study, 711–716**
- protocol-dependent modules, 264**
- Protocols Supported TLV, 545**
- proxy ARP, 35–36**
 - resolving protocol conflicts, 115

pseudonodes, 341
 PSNPs (Partial SNPs), 527–552

Q-R

Q Count, 266

QoS and policy routing, 744–747

queries, EIGRP, 264

query origin flags, 275

RA (Router Advertisement) messages, 65–66

RARP (Reverse ARP), 36

recursive table lookups, 107–109

Redirect messages, 65

format, 69

redirects (ICMP), 40

redistribute connected command, 665

redistribute eigrp command, 754

redistribution

administrative distances, 633–639

and policy routing, 747–751

and summarization, 652–657

case study, 708–710

classless to classful protocols, 639–640

unmatching masks, 641–642

configuring, 642

conflicting metrics, 633

convergence after router reboot, 635–637

EIGRP into OSPF, 656

configuring, 646–651

IGRP into RIP, configuring, 645–646

IPv6 addresses with route maps, 759–760

IS-IS into RIP/RIPng, 663–669

match commands, 733

metrics, configuring, 643–644

mutual redistribution, 701

at multiple redistribution points, 716–722

of default routes, 693

of default static routes, 684–686

of ODR-discovered routes, 697

of static routes, 669–671

into dynamic routing

protocols, 632

of static routes into OSPF, 631

OSPFv3 into RIPng, 657–663

route feedback, 701

route tagging, 752–758

tagged routes, filtering, 757

routing loops, 636–639

set commands, 733

SIN routing, 632

reference bandwidth, 345

reflexive ACLs, 794

example, 795–796

parameters, 794

reliability as IGRP metric, 135, 258–259

reliable multicast, 264

remarks, adding to ACLs, 783–784

replaced routers, troubleshooting next-hop addresses, 116–120

replies

EIGRP, 264

gratuitous ARP, 36

ICMP, 39

RIP, 170

representation of IPv6 addresses, 50

double colon, 51

prefixes, 52

Request messages (RIP), 170, 174–175

requests

ARP, 30

EIGRP, 264

gratuitous ARP, 36

ICMP, 39

RIP, 170

Reserved field, TCP, 43

Response messages, RIP, 170

reverse ARP, 36

reverse routes, 141

RFCs, ICMP-related, 37

RIB (routing information base). *See route tables*

RIP (Routing Information Protocol), 169, 172.

See also RIPng; RIPv1; RIPv2

classful routing, 175–178

subnetting, 176

summarization, 177

configuring, 178–180

discontiguous subnets, 184–187

metrics, 187

equal-cost load balancing, 175

- hop counts, 170
- messages, 172–173
- metrics, configuring, 187–190
- offset lists, 189
- passive interfaces, configuring, 181–182
- redistribution into IGRP, configuring, 645–646
- redistribution into IS-IS, 663–669
- replies, 170
- Request messages, 174–175
- requests, 170
- silent hosts, 172
- similarities to IGRP, 251
- timers, 170–172
- triggered updates, 172
- troubleshooting, 193
- updates
 - minimizing impact of, 190–193*
 - unicast, configuring, 182–184*

RIP_JITTER variable, 170**RIPng (Routing Information Protocol next generation), 212**

- authentication, 213
- configuring, 224–232
- global parameters, configuring, 232–233
- message format, 213–214
- metrics, adjusting, 233–235
- misconfigured VLSM, troubleshooting, 236–242
- next-hop route entries, 214
- processes, displaying, 228
- redistribution
 - into IS-IS, 663–669*
 - into OSPFv3, 657–663*
- route summarization, 235

RIPv1, compatibility with RIPv2, 205**RIPv2, 201**

- authentication, 210–212
 - configuring, 222–224*
- compatibility switches, 215–217
- compatibility with RIPv1, 205
- configuring, 214
- messages, 202–204
- misconfigured VLSM, troubleshooting, 236–242
- similarity to RIPng, 213
- summarization, 220–222
- VLSM, 218–220

RIRs (Regional Internet Registries), 54**route feedback, 701****route filtering, 701**

- case study, 708–709
- filtering by routing process, case study, 710
- specific routes, filtering, 703–708
- tagged routes, filtering, 757

route flapping, 135**route invalidation timers, 140–141****route leaking, 556, 601**

- IS-IS configuration, 598–603

route lookups, classless, 205–206**route maps, 731**

- configuring, 734, 736
- editing, 735
- redistribution of IPv6 addresses, 759–760
- sequence number, configuring, 734–735

route redistribution. *See* redistribution**route summarization. *See* summarization****route tables, 81**

- administrative distance, 98
- columns, 82
- entries
 - debugging, 96*
 - of static routes, 89*
- floating static routes, IPv6, 100–101
- metrics, 84
- missing entries, 83
- static routes
 - configuring, 84–94*
 - creating alternative routes, 96–97*
 - floating static routes, 97–100*
 - load sharing, configuring, 102–104*
 - recursive table lookups, 107–109*
 - summarizing, 95*
 - troubleshooting, 109–112*

Route Tag field (RIPng messages), 214**Route Tag field (RIPv2), 202****route tagging, 722, 752–758****Router Advertisement packets (ICMP), 40****router command, 179****Router IDs, 337**

- configuring, 415–417

Router LSAs, 378, 403–406, 484–486**router odr command, 697–698****router preferences, setting with administrative distance, 722–723**

router rip command, 178, 194

Router Selection packets (ICMP), 40

RouterDeadInterval, 338

routers

boundary routers, 177–178

configuring on IS-IS, 579–581

IS-IS

designated routers, 523–524, 526

hold time, 522

System ID, 519

“routing by rumor,” 137–140

routing domains versus process domains, 252

Routing Information Protocol. *See* RIP

routing loops

counting to infinity, 143–144

in redistributed protocol environments, 636–639

troubleshooting, 114

routing policies, 701

routing protocols

classless, 206

*redistribution into classful domains,
639–642*

distance vector, 137

asynchronous updates, 145–146

characteristics of, 138–139

counting to infinity, 143–144

holddown timers, 144–145

route invalidation timers, 140–141

routing by rumor, 139–140

split horizon, 141

triggered updates, 144

link state, 146

areas, 160

autonomous systems, 160–162

Dijkstra’s algorithm, 156–159

flooding, 147–153

LSAs, 146–147

neighbor discovery, 147

metrics, 133

bandwidth, 134–135

cost, 135

delay, 135

hop count, 134

reliability, 135

selecting, 162–164

routing table entries (OSPF)

destination types, 390–391

lookups, 393

path types, 391–392

RS (Router Solicitation) messages, 65–67

RTO (retransmission timeout), 265–266

RTP (Reliable Transport Protocol), 264

S

Scholten, C.S., 266

secondary addresses

OSPF configuration, 418–422

RIP, 187

security, authentication

configuring on EIGRP, 316–317

configuring on IS-IS, 587–591

OSPF configuration, 393, 441–444

RIPv2, 210–212, 222–224

selecting

NETs, 520

routing protocols, 162–164

Sequence Number field (TCP), 43

**sequence numbers, 148–149, 366–367, 734–735,
781, 786**

circular sequence number space, 150

EIGRP, 264

linear sequence number space, 149–150

lollipop-shaped sequence number space,
152–153

TCP, 43

**sequentiality of access list filters, importance of,
778**

set ip next-hop verify-availability command, 738

**shortest path first protocols. *See* link-state
routing protocols**

show arp command, 34

show cdp command, 90

show cdp neighbor detail command, 123

show clns is-neighbors command, 608

show clns is-neighbors detail command, 597

show interface command, 256

show ip access-list command, 800

**show ip accounting access-violations
command, 801**

show ip cef command, 123

show ip eigrp topology command, 268

- show ip ospf interface command, 344**
- show ip ospf database command, 374**
- show ip ospf database nssa-external command, 409**
- show ip ospf neighbor command, 351–352**
- show ip route command, 83, 123**
- show ipv6 cef command, 123**
- show ipv6 cef detail command, 123**
- show ipv6 interface command, 123**
- show ipv6 rip command, 228**
- show ipv6 route command, 123**
- show isis database command, 529**
- SIA (stuck-in-active), 274**
 - troubleshooting, 324–327
- silent hosts, 172**
- Simple Internet Protocol (SIP), 8**
- SIN (ships in the night) routing, 632**
- SIP (Simple Internet Protocol), 8**
- site-local unicast addresses, 55**
- sizing IP packet headers, 8**
- sleeptime, IGRP, 255**
- small internetworks, 18**
- SNAP (subnet access protocol), 681**
 - ARP encapsulation, 34
- SNPA (Subnetwork Point of Attachment), 515**
- sockets, 43**
- solicited-node multicast addresses, 73**
- Source Address field (IPv6 packets), 60**
- source ports, TCP, 43**
- sources of invalid updates, 210**
- SPF efficiency, improving on IS-IS networks, 573–574**
- split horizon, 141**
 - disabling, 722
 - resolving routing loops in redistributed environments, 637–639
 - with poisoned reverse, 142, 172
- spoofing attacks, 799**
- SRTT (smooth round-trip time), 265**
- standard IP ACLs, 782–784**
 - Boolean AND/OR functions, 784
 - inverse masks, 785
 - placement of, 799
 - policy routing configuration, 740–742
- static default routes, 682**
 - “don’t care” bits, 683
- static routes**
 - administrative distance, 98
 - configuring, 84
 - failed routes, tracing, 109–112
 - floating static routes, creating, 97–100
 - IPv4, configuring, 85–89
 - IPv6
 - configuring, 90–94
 - floating static routes, 100–101
 - troubleshooting on replaced routers, 116–120
 - load sharing
 - CEF, 103–104
 - configuring, 102
 - per destination, 103
 - per packet, 105–106
 - policy routes, 731
 - policy routing
 - match commands, 733
 - set commands, 734
 - protocol conflicts, resolving, 112, 115–116
 - recursive table lookups, 107–109
 - redistribution, 669–671
 - redistribution into dynamic routing
 - protocols, 632
 - redistribution into OSPF, 631
 - summarizing, 95
 - troubleshooting, 109
- static routing**
 - creating alternative routes, 96–97
 - per destination load sharing, 105
- stub areas, 386, 497–498**
 - configuring, 423–426
- stub networks, 340**
- stub routers, 678**
 - configuring on EIGRP, 309–315
 - ODR, 680
 - unreachable destinations, 678
- stuck-in-active EIGRP neighbors, troubleshooting, 323–327**
- study materials for CCIE exam, 805–807**
- subinterfaces, 208**
- subnet, 177**
- subnet hiding, 177**
- Subnet ID field, 53**
- Subnet Mask field (RIPv2), 203**
- subnet masks, 22–24**
 - octet boundaries, 25–27
 - troubleshooting, 29
 - unmatching, advertising, 641–642
 - VLSM, 206–209

subnets, 22–24, 27–28, 180
 bitcount format, 24
 classful routing protocols, 23
 designing, 24–25
 discontinuous, 85, 178
 summarizing RIPv2 routes across,
 220–222
 dotted decimal format, 24
 hexadecimal format, 24
 IP addresses, 24
 number of required hosts, calculating, 25
 RIP, classful routing, 176

subnetting, VLSM
 and RIPv2 routing, 218–220
 troubleshooting, 236–242

subnetwork dependent functions of IS-IS layered network architecture, 521
 Decision process, 531–534
 designated routers, 523–524, 526
 neighbors, 522–523
 Update process, 526–530

Subnetwork Independent sublayer (IS-IS), 521

sub-subnetting, 208

summarization, 677
 address aggregation, 293–296
 CIDR, 296–297
 EIGRP, 655
 in redistributed environments, 652–657
 interface-specific, 653
 null interfaces, 657
 of RIPv2 routes, disabling, 221
 OSPF
 configuring, 435–440
 troubleshooting, 463–466
 RIP, 177
 RIPvng, 235

summary addresses, 677

summary routes, 94

summary-address command, 652

supernetting, 95

System ID (IS-IS), 519

system routes, 252

T

TCP (Transmission Control Protocol), 41–44
 ACLs, 790
 header information, 42

TCP/IP (Transmission Control Protocol/Internet Protocol), 5
 application layer, 6
 ARP, 33–34
 proxy ARP, 36
 data link layer, 6
 host-to-host layer, 6, 41
 TCP, 41–42, 44
 UDP, 44–45
 ICMP Echo Requests/Replies, 39
 Internet layer, 6
 physical layer, 6

three-way handshaking, 555

timers
 EIGRP, multicast flow timer, 265
 holddown timers, 144–145
 IGRP, 254–255
 OSPF, group pacing timer, 377
 RIP, 170–172
 expiration timer, 171
 route invalidation, 140–141

timers active time command, 275

timers basic command, 194

timing jitter, 145

TLA (Top Level Aggregator), 53

TLV (Type/Length/Value) fields, 70
 general fields, 289
 IP External Routes TLV, 291–293
 IP Internal Reachability, U/D bit, 556
 IP Internal Routes TLV, 290–291
 IP-specific fields, 290
 of IS-IS PDUs, 536–539
 Area Addresses TLVs, 542–543
 Authentication Information TLV, 544
 Intermediate System Neighbors TLVs, 543
 IP Interface Address TLV, 545
 Padding TLV, 543
 Protocols Supported TLV, 545

- of IS-IS LSPs, 547–548
 - Inter-Domain Routing Protocol Information*, 551
 - Intermediate System Neighbors*, 549
 - IP External Reachability*, 551
 - IP Internal Reachability*, 550–551
- topological table, contents of**, 268
- totally stubby areas**, 387
 - configuring, 426, 428
- trace command**, 12
- tracing failed routes**, 109–112
 - IPv6, 120–122
- traffic, load balancing**, 137
- Traffic Class field (IPv6 packets)**, 59
- training classes for CCIE exam**, 804
- transit networks**, 340
- transition technologies**, 58
- triggered updates**, 144
 - RIP, 172
- troubleshooting**
 - EIGRP, 317
 - missing neighbors*, 318–323
 - stuck-in-active neighbors*, 323–327
 - IS-IS, 606
 - adjacencies*, 607–610
 - link-state database*, 607, 610–614
 - on NBMA networks*, 615–619
 - OSPF, 455–456
 - adjacencies*, 456–458
 - area-wide problems*, 459
 - isolated areas*, 460–463
 - misconfigured summarization*, 463–466
 - OSPFv3, Frame-Relay mapping, 506–509
 - RIP, 193
 - RIPng, misconfigured VLSM, 236–242
 - RIPv2, misconfigured VLSM, 236–242
 - static routes, 109
 - failed routes*, 109–112
 - protocol conflicts*, 112, 115–116
 - subnet masks, 29
- TTL (Time to Live) field**, 12
- Type 1 external paths**, 392
- Type 2 external paths**, 392
- types of LSAs**, 377
 - ASBR Summary LSAs, 382
 - External Attributes LSAs, 385
 - Group Membership LSAs, 385
 - Network LSAs, 379–381

- Network Summary LSAs, 381–382
- NSSA External LSAs, 385
- Opaque LSAs, 385
- Router LSAs, 378–379
- types of traffic in OSPF areas**, 369

U

- U/D (Up/Down) bit**, 556
- UDP (User Datagram Protocol)**
 - ACLs, 791
 - header information, 44–45
- unequal-cost load balancing, EIGRP configuration**, 299–302
- unequal-cost load sharing**, 102
- unicast addresses**, 174
 - global addresses, 53
 - updates, configuring, 182–184
- unreachable networks**, 144
- unspecified addresses**, 52
- Update process (IS-IS)**, 526–530
- update timers, RIP**, 172
- updates**
 - EIGRP, 264
 - invalid, sources of, 210
 - minimizing impact of, 190–193
 - nonperiodic, 262
 - RIP, 172
- uptime**, 266
- Urgent Pointer field, TCP**, 44
- User Datagram Protocol. See UDP**

V

- variables, RIP_JITTER**, 170
- variance command**, 300
- verifying EIGRP stub router configuration**, 311
- Version field**
 - RIP, 173
 - RIPv2, 202
- versions of IP**, 8
- viewing**
 - IS-IS link-state database, 529–530
 - IS-IS neighbor table, 523
- virtual links**, 340, 373

configuring, 374

IS-IS support, 518

OSPF, configuring, 444–445

**VLSM (variable-length subnet masking),
206–209**

and RIPv2 routing, 218–220

troubleshooting, 236–242

W-X-Y-Z

which-route command, 534

wide metrics, 558

Window Size field, TCP header, 43

windowing, TCP, 42