Optimal Routing Design

Techniques for optimizing large-scale IP routing operation and managing network growth

Russ White, CCIE® No. 2653
Don Slice, CCIE No. 1929
Alvaro Retana, CCIE No. 1609

ciscopress.com

Optimal Routing Design

Russ White, CCIE No. 2635
Don Slice, CCIE No. 1929
Alvaro Retana, CCIE No. 1609

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

Optimal Routing Design

Russ White, Don Slice, Alvaro Retana

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

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing June 2005

Library of Congress Cataloging-in-Publication Number is on file.

ISBN: 1-58714-244-9

Warning and Disclaimer

This book is designed to provide information about scalable IP network design. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

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

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

Trademark Acknowledgments

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

Corporate and Government Sales

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 at 1-800-382-3419 or at corpsales@pearsontechgroup.com.

For sales outside the U.S., please contact International Sales at 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
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Jeff Brady
Executive Editor	Brett Bartow
Production Manager	Patrick Kanouse
Senior Development Editor	Christopher Cleveland
Senior Project Editor	Marc Fowler
Copy Editor	Karen A. Gill
Technical Editor(s)	Neil Lovering, Danny McPherson, Steven Moore
Team Coordinator	Tammi Barnett
Book and Cover Designer	Louisa Adair
Composition	Mark Shirar
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

Printed in the USA

About the Authors

Russ White, CCIE No. 2635, is a member of the Routing Deployment and Architecture team at Cisco Systems in RTP, North Carolina. He works in all areas of routing protocol design, routed network design, and routed network deployment. He is a regular speaker at Networkers, has coauthored several books on routing protocols and several IETF RFCs, and periodically contributes to network journals.

Don Slice , CCIE No. 1929, is a development engineer on the Distance Vector Routing Protocol Team, responsible for creating new features and resolving software defects with EIGRP and RIP. Previously, Don worked on the Routing Deployment and Architecture Team and Routing Protocol Escalation Team designing, implementing, and troubleshooting networks running all of the IP routing protocols.

Alvaro Retana, CCIE No. 1609, is a technical leader in the IP Routing Deployment and Architecture Team at Cisco, where he works first hand on advanced features in routing protocols. His current work includes topics such as BGP Security and Ad Hoc Networking.

About the Technical Reviewers

Neil Lovering , CCIE No. 1772, is a design consultant with Cisco Systems. He has been a network consultant for more than 10 years, and has worked on various routing, switching, dialup, security, and network design projects for many customers all over North America. Neil currently works with large systems integrators in the Washington DC area. When not at the keyboard or at a customer site, Neil enjoys spending time with his wife and two children in Virginia.

Danny McPherson is currently the director of architecture and development at Arbor Networks, Inc., and has extensive technical leadership in the telecommunications industry. He has more than 12 years of experience as a network architect for global Internet service providers such as Qwest and MCI, in addition to network equipment vendors such as Amber Networks.

Danny is a common contributor within the routing, operations, and Internet areas of the IETF and global network operations community. He has authored several Internet protocol standards, books, and other documents related to Internet routing protocols, network security, Internet addressing, and network operations. His most recent work, *Practical BGP*, was published in mid-2004. You can reach Danny at danny@tcb.net.

Steve Moore , CCIE No. 4927, is an engineer with the Cisco IP Routing Deployment and Scalability team, which is a part of the IOS Technologies division of Cisco Engineering. He is responsible for discovering, testing, validating, and assisting in the customer deployment of new ideas relating to the scalability of routing protocols. He works closely with customers and with development, support, testing, and consulting groups within Cisco. A part of Steve's job is to educate; he does so by working with customers directly, writing whitepapers, and speaking at various Networkers conferences. Within the nine years that Steve has worked at Cisco, he has become known for his experience with routing protocols, WAN technologies, and optical networking.

Dedications

Russ White: I would like to thank my wife, Lori, for putting up with a little more computer time than usual when I'm writing. For Bekah and Hannah—we can build those monkey bars now! I also thank God for my family and their support, including my mother and stepfather, and my wife's family, for their support through the years.

Don Slice : I would like to thank my wife, Pam, and my daughters Jessica, Amy, and Heather, for being understanding when I've been distracted by the work required to write this book. I also praise God for sustaining me in this effort and providing me with the time and ability to put these thoughts down on paper.

Acknowledgments

We'd like to thank our technical editors, Steve Moore, John Cavanaugh, Danny McPherson, and Neil Lovering, for their perseverance; your help improved this book immeasurably. Thanks to Brett Bartow, who puts up with our missed deadlines, and with Chris Cleveland, who actually tries to make sense out of our various ramblings and make them fit for printing.

Thanks to all the managing and marketing people at Cisco Press, who make these books possible.

This page intentionally left blank

Contents at a Glance

Foreword xvii
Introduction xviii

Part I **Network Design Overview 3**
Chapter 1 Network Design Goals and Techniques 5
Chapter 2 Applying the Fundamentals 35
Part II **Interior Gateway Protocols 73**
Chapter 3 EIGRP Network Design 75
Chapter 4 OSPF Network Design 143
Chapter 5 IS-IS Network Design 189
Part III **Advanced Network Design 223**
Chapter 6 BGP Cores and Network Scalability 225
Chapter 7 High Availability and Fast Convergence 261
Chapter 8 Routing Protocol Security 309
Chapter 9 Virtual Private Networks 353
Part IV **Appendixes 381**
Appendix A EIGRP for IP Basics of Operation 383
Appendix B OSPF Basics of Operation 399
Appendix C Integrated IS-IS Basics of Operation 411
Appendix D Border Gateway Protocol 4 Basics of Operation 421
Appendix E IP Network Design Checklist 435
Appendix F Answers to Review Questions 441
Appendix G Which Routing Protocol? 457
Index 471

Contents

Foreword xvii
Introduction xviii

Part I Network Design Overview 3

Chapter 1 Network Design Goals and Techniques 5
- Goals for Network Design 5
- Reliability 6
  - Packet Delivery Reliability 6
  - Packet Delivery Times 8
  - Delay and Jitter Budgets 9
  - The Impact of Network Design on Delay and Jitter Budgets 9
- Reliability and Resiliency 10
  - Defining Network Failure 12
  - Network Recovery Time 13
- Manageability 13
  - Day-to-Day Operational Maintenance 14
    - Taking a Network Baseline 14
    - Network Documentation 16
  - Emergency Management 18
- Scalability 20
  - Redundancy 21
    - How Redundancy Increases Resiliency 21
    - Statistical Analysis 23
    - How Redundancy Can Increase Management Complexity 25
    - How Redundancy Can Reduce Scalability 26
- Layering 27
  - Hiding Information 28
    - Hiding Topology Information 28
    - Hiding Reachability Information 30
  - Separate Functionality 32
- Summary 32
- Review Questions 33

Chapter 2 Applying the Fundamentals 35
- Hierarchical Design 35
  - Abstraction Through Layering 35
  - Horizontal Layers in a Network 36
  - Layer Functions 38
    - Forwarding Traffic 38

Aggregation of Routing Information 39
Definition and Implementation of Routing Policies 41
User Attachment 42
Controlling Traffic Admittance into the Network 42
Network Hierarchies 43
Two-Layer Hierarchy 43
Three-Layer Hierarchy 44
Determining How Many Layers to Use in Network Design 45
Hiding Layers Within Layers 46
Creating Layers 47
Creating Choke Points 48
Separating Complexity from Complexity 49
Addressing and Summarization 50
Assigning Addresses in a Network 50
Working Around Addressing 54
Leaking More Specifics 54
Smaller Summary Blocks 56
Change the Logical Layout 57
Summary Issues 58
Summarization Black Holes 58
Summary Suboptimal Routing 59
Summary Metrics 60
Redistribution 62
Alternatives to IGP to IGP Redistribution 62
Single Point of Redistribution 64
Multiple Points of Redistribution 66
Filters 67
Tags 69
Review Questions 71

Part II Interior Gateway Protocols 73

Chapter 3 EIGRP Network Design 75

Deploying EIGRP on a Large-Scale Three-Layer Hierarchical Network 75
Analyzing the Network Core for Summarization 77
Summarizing from the Core to the Distribution Layer 77
Summarizing into the Core at Its Edge 78
Analyzing the Network Distribution Layer for Summarization 80
Summarizing Toward the Network Core 80
Summarizing Toward the Remote Sites 83
Analyzing Routing in the Network Access Layer 83
Single-Homed Sites 84
Dual-Homed Remotes 85

Analyzing Use of the Stub Feature in Access Routers 87
Analyzing Routes to External Connections 90
Analyzing Routes to the Common Services Area 91
Analyzing Routes to Dial-In Clients 94
Host Routes 94
Bandwidth Issues 97
Deploying EIGRP on a Two-Layer Hierarchical Network 97
Summarization in the Core 98
Summarization in the Aggregation Layer 98
Summary of EIGRP Network Design 98
New Features in EIGRP 99
Third-Party Next Hop 99
NBMA Hub-and-Spoke Network 99
Redistributed Next Hop 102
Enhanced Route Map Support 104
Before Enhanced Route Map Support 104
Route Map Enhancements 106
Enhanced EIGRP Active Process 110
Case Study: Summarization Methods 114
IP Summary Addresses 114
Distribute Lists 116
Case Study: Controlling Query Propagation 116
Case Study: A Plethora of Topology Table Entries 118
Case Study: Troubleshooting EIGRP Neighbor Relationships 120
EIGRP Neighbor Relationships: Common Problem 1 120
EIGRP Neighbor Relationships: Common Problem 2 122
Case Study: Troubleshooting SIA Routes 124
Case Study: Redistribution 129
Using Distribute Lists to Prevent Redistribution Routing Loops 130
Using Route Maps to Prevent Redistribution Routing Loops 130
Using Prefix Lists to Prevent Redistribution Routing Loops 131
Setting the Administrative Distance to Troubleshoot Redistribution Routing Loops 131
Using External Flags to Prevent Redistribution Routing Loops 132
Case Study: Retransmissions and SIA 134
The Hold Timer 134
SIA Timer 135
Interaction Between the Hold Timer and the SIA Timer 135
Case Study: Multiple EIGRP Autonomous Systems 136

Review Questions 139

Chapter 4 OSPF Network Design 143

Summarization and Aggregation 143
Deploying OSPF on a Three-Layer Hierarchy 146
The Core Routers as ABRs 146
The Distribution Layer Routers as ABRs 148
Mixing ABR Locations 150
Deploying OSPF on a Two-Layer Hierarchy 152
Reducing Flooding Through Stub Areas 153
Stub Areas 155
Totally Stubby Areas 156
Not-So-Stubby Areas 157
Totally NSSA 159
Totally Stubby Not Really Full Areas 160
When to Use Stub Areas 160
Aggregating Routes in OSPF 160
Filtering Routes in OSPF 162
Deploying OSPF on Specific Topologies 164
Redistribution into OSPF 164
External Route Metrics 164
External Route Selection at ABRs 167
Route Selection Between Processes 167
Full Mesh Topologies 167
Hub-and-Spoke Topologies 171
Treating the NBMA Interface as a Broadcast Interface 172
Treating the NBMA Interface as a Set of Point-to-Point Interfaces 174
Treating an NBMA Interface as a Broadcast Point-to-Multipoint Interface 175
Treating an NBMA Interface as a Nonbroadcast Point-to-Multipoint Interface 176
Summary of Interface and OSPF Link-Type Options 176
Reducing Flooding to the Spokes 176
Links Parallel to Area Boundaries 178
Dial Links 179
Point-to-point Broadcast Links 181
Case Study: OSPF Externals and the Next Hop 182
Case Study: Troubleshooting OSPF Neighbor Adjacencies 184
Review Questions 187

Chapter 5 IS-IS Network Design 189

Deploying IS-IS on a Three-Layer Hierarchy 190
The Entire Network as a Single Routing Domain 190

The Core as the L2 Domain 193
Merging the Core and Distribution Layers into Level 2 194
Mixing and Overlapping the Level 1/Level 2 Border 195
Deploying IS-IS on a Two-Layer Hierarchy 197
Working with IS-IS Routing Areas 198
Leaking Routes into an L1 Routing Domain 203
Aggregating Routes in IS-IS 204
Deploying IS-IS on Specific Topologies 204
Redistribution 204
Full Mesh Topologies 205
Hub-and-Spoke Topologies 209
Point-to-Point Links 209
Broadcast Interfaces 210
Point-to-Point Broadcast Links 211
Links Parallel to Area Boundaries 212
Other Considerations in IS-IS Scaling 212
Metrics 213
Excessive Link-State Flooding 213
LSP Corruption 214
Maximum Number of Pseudonodes 215
Prefix-Driven Routing Table Installation 216
Hello Padding Suppression 217
Case Study: Troubleshooting IS-IS Neighbor Relationships 217
Review Questions 220

Part III Advanced Network Design 223

Chapter 6 BGP Cores and Network Scalability 225
Case Study: Troubleshooting BGP Neighbor Relationships 227
No IP Connectivity 228
eBGP Multihop 228
Other BGP Neighbor Problems 230
Logging Neighbor Changes 231
BGP in the Core 232
Case Study: Sample Migration 233
Scaling Beyond the Core 236
Dividing the Network into Pieces 237
Regional IGPs 238
BGP Network Growing Pains 239
BGP Update Generation Issues 239

Reducing the Number of Updates Generated 239
Case Study: Route Reflectors as Route Servers 245
External Connections 247
Case Study: Dual-Homed Connections to the Internet 247
Load Sharing on the Outbound Side 247
Load Sharing on the Inbound Side 250
Being a Transit AS 252
Case Study: Conditional Advertisement 253
Case Study: Route Dampening 255
Review Questions 258

Chapter 7 High Availability and Fast Convergence 261
Considerations in Fast Convergence 261
Network Meltdowns 262
Solving the Meltdown 263
Designing Routing Protocols Not to Melt 263
Do Not Report Everything You See 264
Non-Stop Forwarding 265
Graceful Restart 266
EIGRP Graceful Restart 267
OSPF Graceful Restart 270
IS-IS Graceful Restart 274
BGP Graceful Restart 277
Fast Down Detection 280
Detecting a Link or Adjacency Failure Using Polling 280
Bidirectional Forwarding Detection 283
Detecting a Link or Adjacency Failure Using Event-Driven Link Failure Notification 283
SONET 284
Frame Relay 285
Ethernet 288
Slowing Down When the Network Speeds Up 290
Link-State Exponential Backoff 291
Configuring OSPF Exponential Backoff for LSA Generation 292
Configuring OSPF Exponential Backoff for Running SPF 292
Configuring IS-IS Exponential Backoff 293
IP Event Dampening 293
Configuring IP Event Dampening 295
Calculating the Route Faster 296
EIGRP Feasible Successors 296
Link-State Partial SPF 299

Link-State Incremental SPF 300
Deploying GR and Fast Convergence Technologies 302
Graceful Restart Versus Fast Down Detection 302
How Fast Can GR Work? 302
Balancing Between GR and Fast Down Detection 303
Deploying Graceful Restart with BGP and an Interior Gateway Protocol (IGP) 304
Deploying Exponential Backoff for Fast Convergence 305
Setting SPF Exponential Backoff Timers 306
Review Questions 307

Chapter 8 Routing Protocol Security 309
Fundamentals of Routing and Security 309
Understanding What a Routing System Is 309
Thoughts on Authorization and Authentication 310
Defining Authentication and Authorization 311
Transiting Authentication and Authorization 311
Transiting Authorization in a Routing System 314
Trust and Security 316
Determining the Reasons for an Attack on the Routing System 317
Types of Attacks Against Routing Systems 318
Disrupting Peering 318
Transport-Level Attacks Against OSPF and IS-IS 318
Transport-Level Attacks Against EIGRP 320
Transport-Level Attacks Against Border Gateway Protocol (BGP) 321
Protocol-Layer Attacks 322
Falsifying Routing Information 323
Disrupting Routing Domain Stability 324
Protecting Routing Domain Legitimacy 326
Protecting Routers from Being Compromised 326
Use Passwords 326
Filter Access to Routers 328
Protecting Against Illegitimate Devices Joining the Routing Domain 330
MD5 Authentication 331
Issues with MD5 Peer Authentication 332
IPSec 333
Protecting Routers from Denial-of-Service Attacks 334
Edge Filters 335
The Generalized TTL Security Mechanism 335
Protecting Routing Information 337
Extranet Connections 337
Use an Exterior Gateway Protocol for All Extranet Connections 339
Filter Routes Aggressively at the Extranet Edge 339

Dampen Prefixes Aggressively at the Extranet Edge 340
Limiting Route Count at the Extranet Edge 341
Connections to the Internet 341
Route Filtering 341
Protecting Against Transit 342
Route Dampening 343
Future Directions in Routing Protocol Security 343
Protecting Against Illegitimate Devices Joining the Routing Domain 343
Secure Origin BGP (soBGP) 344
Begin at the Beginning: Who Are You? 345
The First Goal: Are You Authorized? 346
The Second Goal: Do You Really Have a Path? 347
Review Questions 349
References 349

Chapter 9 Virtual Private Networks 353
MPLS 353
MPLS Basics 354
Overlay Routing over MPLS VPNs 356
Peer-to-Peer (Redistributed) Routing over MPLS VPNs 357
BGP/MPLS VPNs 358
EIGRP 361
OSPF 369
IPSec 370
GRE 372
NHRP 372
Case Study: NHRP in an ATM Network 373
Dynamic Multipoint IPSec VPNs 376
Review Questions 379
References 379

Part IV Appendixes 381

Appendix A EIGRP for IP Basics of Operation 383

Appendix B OSPF Basics of Operation 399

Appendix C Integrated IS-IS Basics of Operation 411

Appendix D Border Gateway Protocol 4 Basics of Operation 421

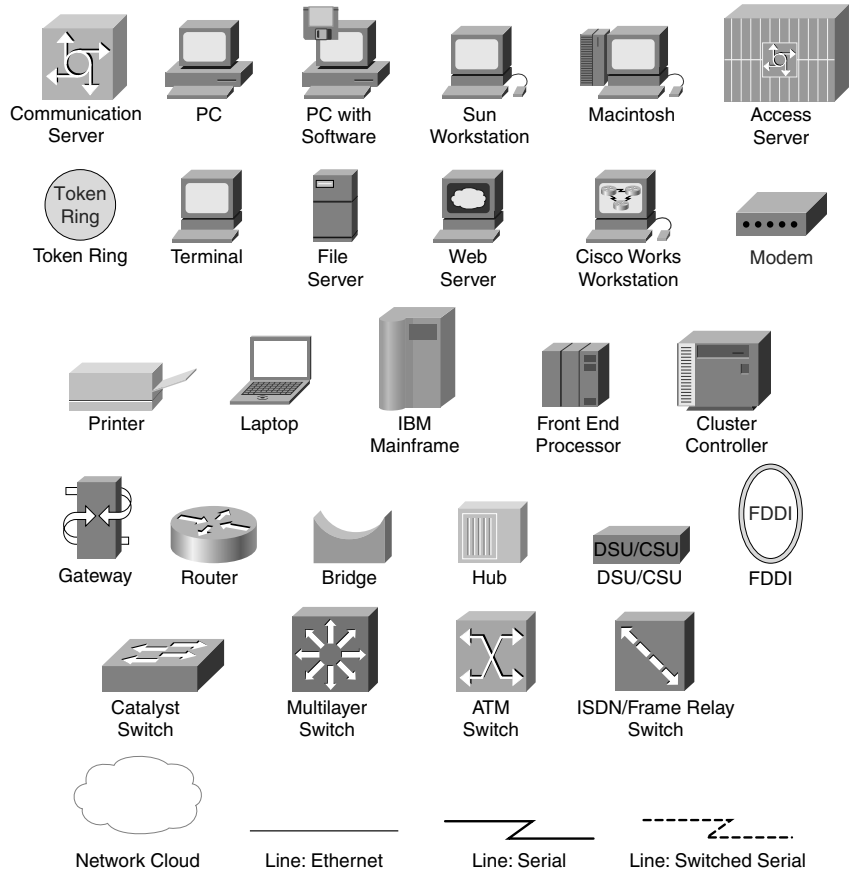
Appendix E IP Network Design Checklist 435

Appendix F Answers to Review Questions 441

Appendix G Which Routing Protocol? 457

Index 471

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 input manually 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

I first logged into the predecessor of the Internet—the Arpanet—in 1980. My task as a teaching assistant was to download two compilers for our new Computer Science Department VAX from a colleague at MIT. In the process I also learned about email and two games—Adventure and Zork. In line with today's environment, this led to a significant amount of time spent online.

The mechanics of how my session on a VAX in Halifax could move to another computer at MIT was hidden from me as a user, but fascinating to think about. As I began my working career as a systems programmer, I specialized in computer communications and have looked back.

The emergence of TCP/IP and routing protocols later in the 1980's permitted the growth of what we now know as the Internet. Today it has evolved from its simple origins in those early years to a collection of interconnected networks involving myriads of service providers, government agencies and private companies. The architecture and design of networks have become a science unto itself.

At Cisco Systems, Russ White, Alvaro Retana, and Don Slice have played an integral part in the support and design of customer networks. Their efforts have been recognized by numerous internal awards, IETF RFCs, drafts and publications. Indeed, they have progressed from using routing protocols for network design to completing the feedback loop and working with the routing community within Cisco and the IETF to improve the routing protocols themselves. One needs only to perform a search on the Google search engine with their names and IETF to get a sense of their involvement in the industry.

The complexity associated with overlaying voice and video onto an IP network involves thinking through latency, jitter, availability and recovery issues. *Optimal Routing Design* offers keen insights into the fundamentals of network architecture for these converged environments. As such, I recommend this book to any professional or student working in network architecture or design.

John Cavanaugh, CCIE No. 1066

Distinguished Services Engineer - Advanced Services, Cisco Systems, Inc.

Introduction

In 1998, when we first started writing *Advanced IP Network Design*, we had no idea that the future would bring us more and more deeply into the realms of routed network design or that we would work together in the same place in closely related teams for all of these years. We originally wrote *Advanced IP Network Design* to help answer some of the questions we heard on a regular basis as engineers working in the Cisco Technical Support Center Routing Protocol and Escalation teams.

In many ways, we wrote this book for the same reason: to help customers we meet on a daily basis with the answers to the questions we always hear. What is the best way to build addressing for my network? How do I redistribute between two protocols without blowing up my network? When and why should I use BGP?

In other ways, however, this book is completely different. Of course, the most obvious difference is that the authors have worked on thousands more networks and interacted with thousands of different customers since that book was written. Each time a network engineer approaches us with a new problem to be solved or we see a good solution for a problem, we learn more about network design.

Less obvious, though, are the lessons that failed networks and bad designs have given us. Each time we propose something that does not work, we learn new things about routing design that we did not know before, and we learn to watch for new problems that we might not have expected before. Our goal in this book was to amalgamate these experiences, both good and bad, into a readable, understandable whole so that network engineers at all skill levels can draw on them. We are in a position to see new networks, new problems, and new solutions every day; this book is an attempt to share that experience with other network engineers.

Who Should Read This Book?

Network engineers who want to understand the concepts and theory of designing and deploying a large-scale network, network engineers who are currently managing large-scale networks, and engineers who are studying for their CCIE or Cisco network design certifications will find this book useful. Readers should be familiar with basic routing protocols concepts, including the mechanics of how each protocol works, basic Cisco router configuration, and physical layer interconnectivity. Some review of routing protocol operation is provided in the appendixes, but these are by no means comprehensive reviews.

How This Book Is Organized

This book is broken into four distinct parts. Part I begins with a consideration of network design issues on a broad scale:

- Chapter 1, "Network Design Goals and Techniques," discusses the goals that a network designer needs to keep in mind, including tradeoffs among goals. You will find a good bit of discussion on the tradeoffs among network scaling, convergence speed, and resiliency.
- Chapter 2, "Applying the Fundamentals," discusses the basic techniques that are applicable to any network design, regardless of the routing protocol. Here we talk about hierarchy, addressing, summarization, and information hiding, all critical aspects of a good network design.

Part II of *Optimal Routing Design* covers each interior gateway protocol in depth, generally starting with a discussion on deploying the protocol on a three-layer hierarchy and then on a two-layer hierarchy. Each chapter then discusses deploying the protocol over specific topologies, such as full mesh and hub-and-spoke topologies. Each chapter ends with case studies that are specific to the protocol.

- Chapter 3, "EIGRP Network Design," covers the deployment and operation of EIGRP on large-scale networks. The operation of EIGRP on a number of specific topologies and specific techniques for deploying EIGRP are included.
- Chapter 4, "OSPF Network Design," covers the deployment and operation of OSPF on large-scale networks. The operation of OSPF on a number of specific topologies and specific techniques for deploying OSPF are included.
- Chapter 5, "IS-IS Network Design," covers the deployment and operation of IS-IS on large-scale networks. The operation of IS-IS on several specific topologies and specific techniques for deploying IS-IS are included.

Part III of the book leaves the IGP-specific realm and looks toward more advanced topics in network design.

- Chapter 6, "BGP Cores and Network Scalability," discusses when and how to use a BGP core in a large scale network and then moves into connections to outside networks, such as an Internet service provider or extranet.
- Chapter 7, "High Availability and Fast Convergence," goes into detail on the techniques and tradeoffs for reaching the magical five-9s of network uptime.
- Chapter 8, "Routing Protocol Security," covers some of the concepts surrounding securing a routing system, some baseline best practices, and some future work that is underway in this area.
- Chapter 9, "Virtual Private Networks," covers the concepts of virtual private networks and the various mechanisms used for creating them. This chapter includes various techniques for carrying routing information through a VPN.

Part IV of the book provides short appendixes dealing with the fundamentals of how each routing protocol that is discussed in the book works. These are not intended to be complete references, but rather just an introduction and a place to go when the corresponding chapter discusses some aspect of the protocol operation that you are not familiar with.

- Appendix A, "EIGRP for IP Basics of Operation," discusses the basic operation of EIGRP, including how neighbors are formed, the metrics used, the DUAL algorithm, and the processing of changed or withdrawn routing information.
- Appendix B, "OSPF Basics of Operation," covers the basic operation of OSPF, including how neighbors are formed, how information is flooded throughout the network, and how you can use the SPF algorithm to find loop-free paths through the network.
- Appendix C, "Integrated IS-IS Basics of Operation," discusses the basic operation of IS-IS, including how neighbors are formed, how information is flooded throughout the network, and how the SPF algorithm helps you find loop-free paths through the network.

- Appendix D, "Border Gateway Protocol 4 Basics of Operation," covers how BGP works, including how neighbors are built and how BGP ensures loop-free routing in an internetwork.
- Appendix E, "IP Network Design Checklist," provides a checklist that network designers can use to determine where they need to look in a network for problems and possible hidden issues and where to gain an understanding of the overall network design. This is useful mostly for engineers who are approaching a network for the first time.
- Appendix F, "Answers to Review Questions," provides the answers to the review question exercises found at the end of Chapters 1 through 9.
- Appendix G, "Which Routing Protocol?" provides an overview of the routing protocols, comparing their strengths and weaknesses. This is designed primarily for engineers who have a knowledge of one protocol and are trying to gain an understanding of the other protocols, or engineers who are considering which routing protocol to run on a specific new network design or if they should switch from one protocol to another.

Final Words

Overall, we have developed *Optimal Routing Design* to be read, not just used as a reference. We strongly believe that understanding network design with all the available protocols makes you a better network engineer. Learning how to deploy multiple protocols, even if you will never use them, helps you to understand and apply the underlying principles and find techniques to work around problems that you might encounter.

We hope that you find the time spent reading our little missive to be well spent—and we expect to welcome you on the list of excellent network designers! So, kick back, put your feet on your desk, and read through from the front to the back. You can tell your boss you are learning how to design your network to scale.

CHAPTER 3

EIGRP Network Design

The previous two chapters described many of the important network design techniques used to meet the design goals of high resiliency, manageability, and scalability. Now it is time to put these techniques into practice using the Cisco advanced distance vector routing protocol, Enhanced Interior Gateway Routing Protocol (EIGRP). These techniques will be applied to the networks shown in Figure 3-1.

For more information on how EIGRP functions, refer to Appendix A, "EIGRP for IP Basics of Operation." EIGRP has numerous advantages over its link-state routing protocol counterparts, but it also has limitations and behaviors that a network designer must understand to successfully implement a scalable EIGRP network. This chapter describes some of these behaviors and provides techniques that network designers can use to improve the performance and scalability of EIGRP networks.

This chapter helps you to do the following for both two-layer and three-layer hierarchical networks:

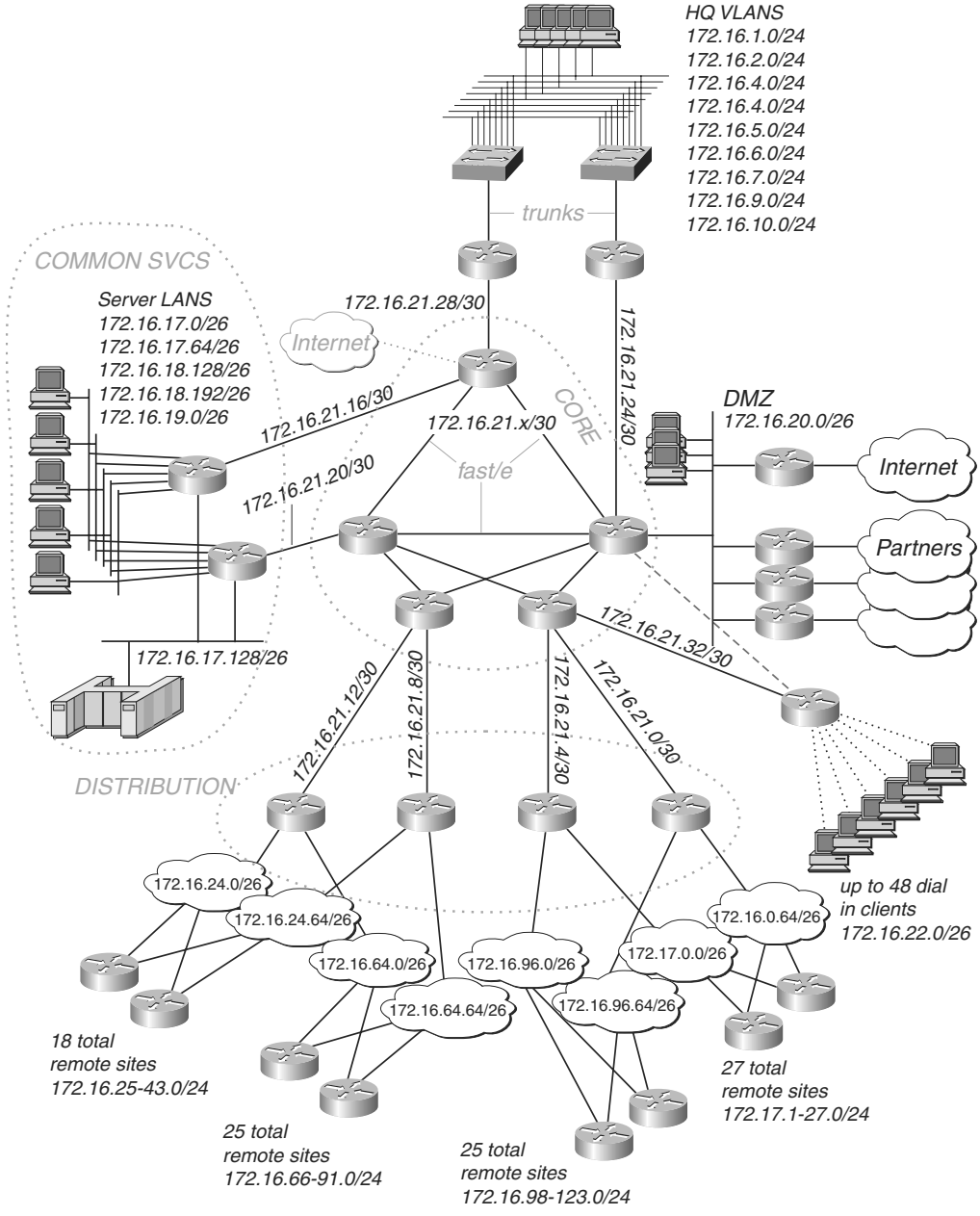
- Analyze summarization at each layer of the EIGRP network.
- Analyze the use of the stub feature for access routers.
- Analyze the best way to deal with external connections, common services, and dial-in clients.
- Explore case studies on summarization methods, query propagation, excessive redundancy, troubleshooting common problems, and redistribution issues.

Deploying EIGRP on a Large-Scale Three-Layer Hierarchical Network

Many networks have been built around the core, distribution, and access layer model, because it provides a well-defined separation of functions into the various portions of the network. It also provides an excellent topology to apply scalability improvement techniques such as summarization.

Using the network described in Figure 3-1, this section describes how you can implement the information hiding technique of summarization at each of the three layers: core, distribution, and access.

Figure 3-1 *Large-Scale Three-Layer Hierarchical Network*



Analyzing the Network Core for Summarization

The network core in EIGRP has the same requirements as those presented in Chapter 2, "Applying the Fundamentals." Adequate redundancy and bandwidth must be provided in the core to ensure rapid, reliable delivery of packets presented to it from the distribution layer and destined to common resources or other distribution layer routers. The core should present as little impediment to the delivery of packets as geographic distances and budgets allow. Network designs are much more scalable if it does not matter where a packet enters the core from the distribution layer. The core should appear to be a high-bandwidth service that the distribution layer uses to reach common resources and other distribution layer routers.

If the network has been designed well, including addressing, the edge of the network core will be an ideal place to summarize. The sections that follow discuss the best ways to summarize at the network core to provide maximum stability and resiliency. These methods include the following:

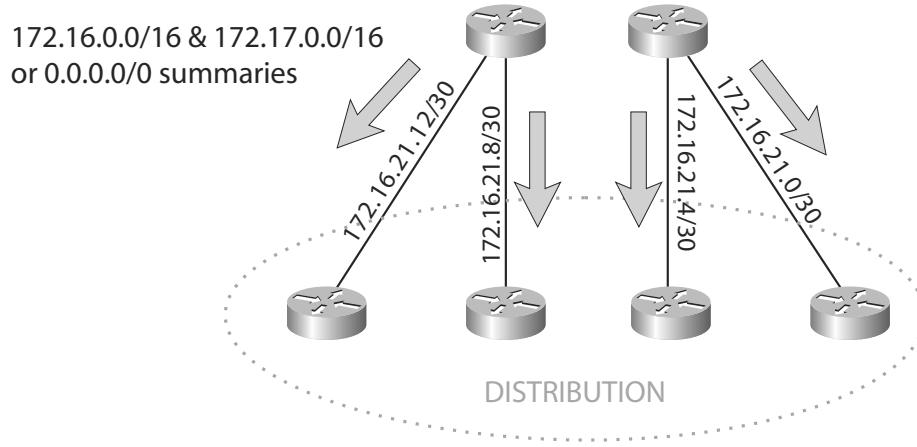
- Summarizing from the network core to the distribution layer
- Summarizing into the core at its edge

Summarizing from the Core to the Distribution Layer

The "Addressing and Summarization" section in Chapter 2 explained how stability and scalability are best when a network is implemented with good summarization. If your network core topology is robust enough to present a minimum of delay to transit packets and your IP addressing is well designed, you are free to summarize to the fullest from the core to the distribution layer.

In the example network shown in Figure 3-1, you can perform maximum summarization because the network core has adequate bandwidth and redundancy. You can put summarization statements on the serial links that connect the core to the distribution layer, either presenting only the two major network routes (172.16.0.0/16 and 172.17.0.0/16) or just the default route (0.0.0.0/0) to the distribution layer, as shown in Figure 3-2. Refer to the "Summarization Methods" case study later in this chapter for an examination of the various summarization techniques available in an EIGRP network.

Figure 3-2 *Summarizing Outbound from the Core*



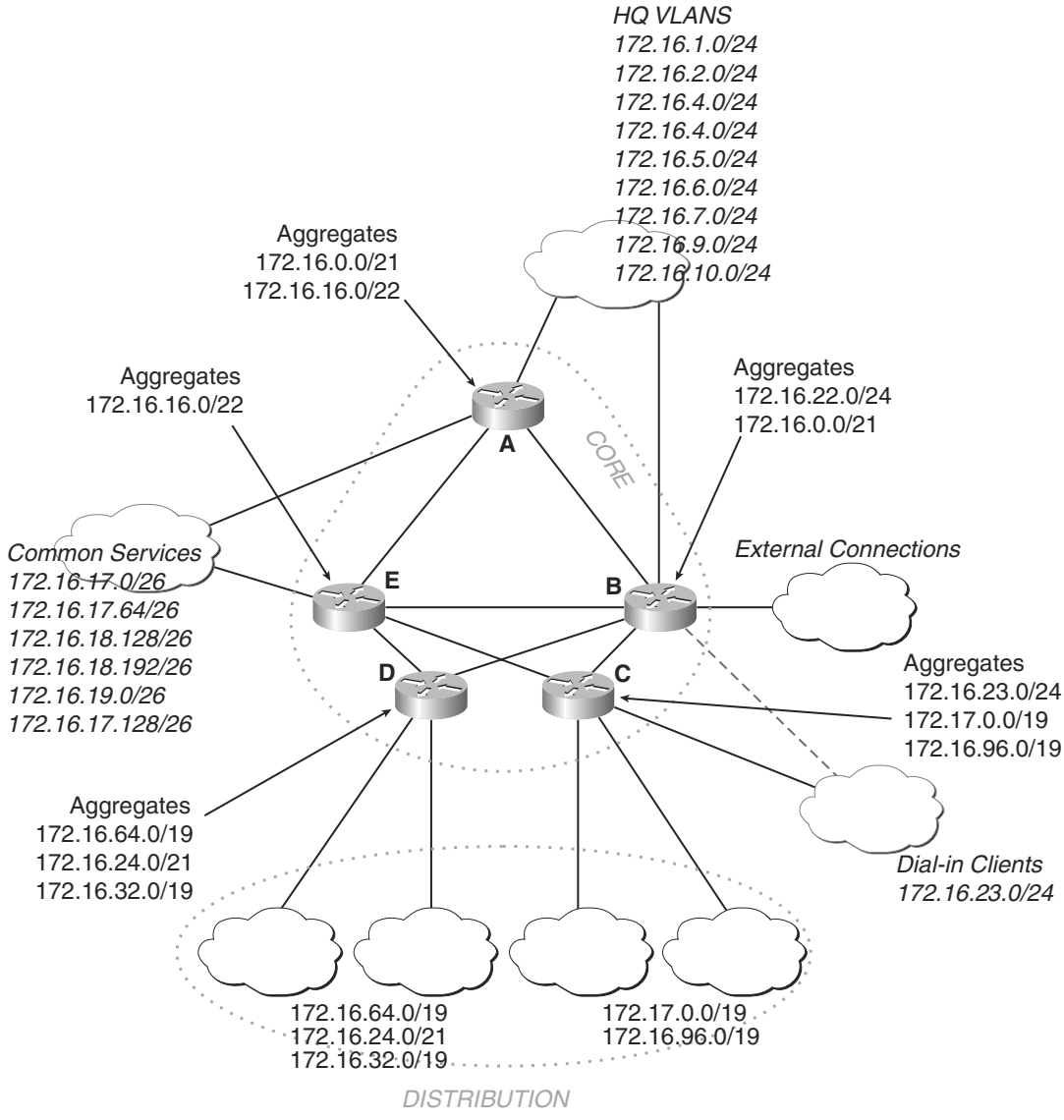
Minimizing the updates sent to the distribution layer routers from the core greatly reduces the query range and simplifies the process of bringing up neighbors across these critical links in the network. Refer to the later case study, "Controlling Query Propagation," for details on how important it is to limit the reach of queries in an EIGRP network.

If the destination subnet is closer in the topology to one core router than another, the shortest path from the distribution layer router to the target network might not be the one taken. (The traffic might take a suboptimal route.) If the network core presents minimal delay to traffic, the addition of an extra hop will not be significant when compared to increased stability.

Summarizing into the Core at Its Edge

Summarizing into the core at its edge is only useful if the distribution layer routers along the edge of the core are not also summarizing towards the core. As Figure 3-3 illustrates, the core routers could summarize toward the other core routers so that each core router has full component knowledge of the subnets inside of the regions to which it is connected but only summary knowledge of the other regions.

Figure 3-3 *Summarization into the Core from Its Edge*



The following list describes the routing advertisements resulting from the topology and configurations in Figure 3-3:

- Router A advertises 172.16.0.0/21 for the HQ VLANs and 172.16.16.0/22 for the common services out toward the other core routers.
- Router B advertises 172.16.22.0/24 for the external connections and 172.16.0.0/21 for the HQ VLANs toward the other core routers.
- Router C advertises 172.16.23.0/24 for the dial-in users, 172.17.0.0/19 for remote sites, and 172.16.96.0/19 for remote sites.
- Router D advertises 172.16.64.0/19, 172.16.24.0/21, and 172.16.32.0/19 for remote sites.
- Router E advertises 172.16.16.0/22 for the common services.

The advantage of this approach is that the core routers have full knowledge about all remote locations in their region and can choose the optimum route from the core router to the remote site. The disadvantage of this approach is that the core routers for each region are directly involved in the query path for any link failure inside of their region.

Should you summarize within the core of the network? Because this makes the configuration of the core more complicated and moves work from the distribution layer into the network core, you probably should not adopt this solution. In any case, you need to hold off on making a final decision until you have dealt with summarization in the distribution layer.

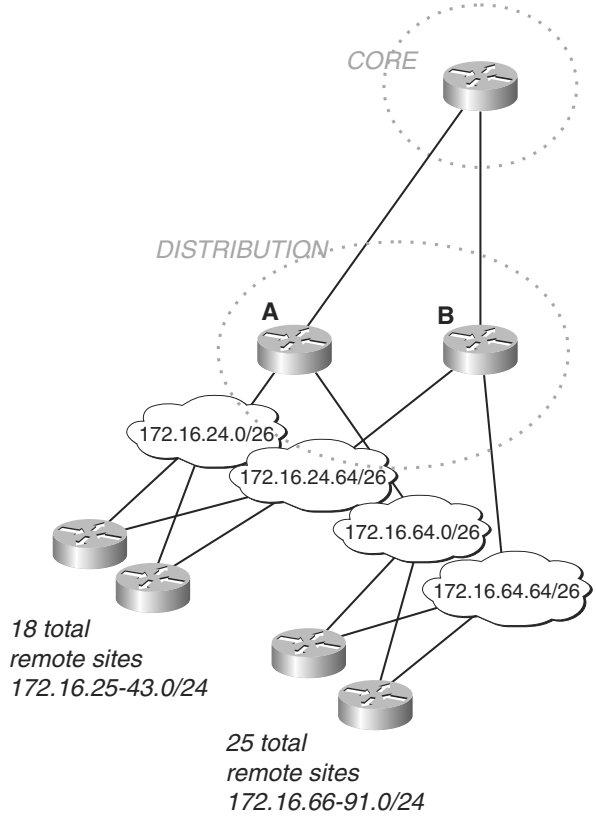
Analyzing the Network Distribution Layer for Summarization

The distribution layer goals in hierarchical networking are to summarize and aggregate traffic. The following sections on summarizing toward the network core and summarizing toward the remote sites give you a better idea of what you can do with summarization in the distribution layer.

Summarizing Toward the Network Core

You can apply summarization to the inbound links toward the core to limit their advertisements to one or more summary routes representing all the subnets that are reachable through a given distribution router. For example, in Figure 3-4, summarization is configured outbound on Router A and Router B on the serial links toward the core router.

Figure 3-4 *Summarization Between the Distribution Layer and Core*



In this network, Routers A and B can advertise the following routes to the core:

- 172.16.64.0/19
- 172.16.24.0/21
- 172.16.32.0/19

However, one problem can occur with this summarization method unless proper steps are taken. If both Router A and Router B advertise summaries representing the same sets of remote networks into the core, you can create a *routing black hole* if one of the distribution routers loses access to one of the remotes. For example, even if Router A loses its connection to the remote site advertising 172.16.64.0/24, it will continue advertising the 172.16.64.0/19 summary route. In this case, all packets destined to hosts within 172.16.64.0/24 forwarded to Router A will be dropped.

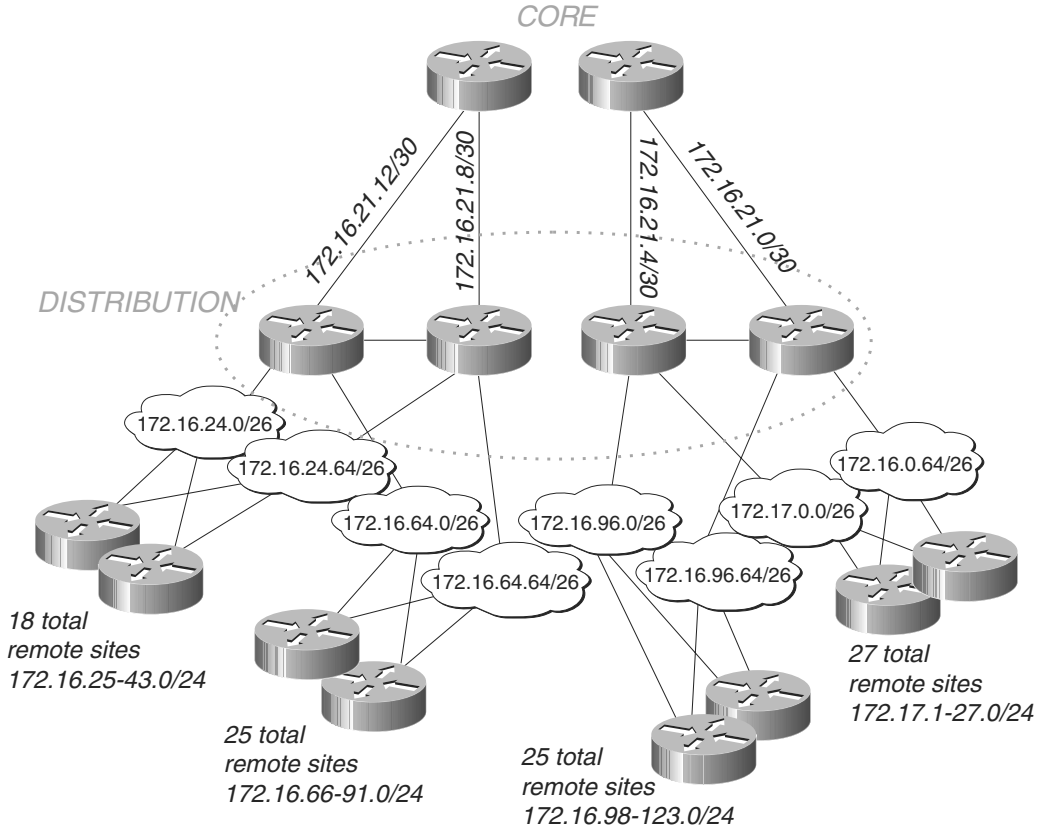
This problem has two solutions. The first solution is to summarize at the edge of the core into the core rather than between the distribution and core routers, as covered in the previous section, "Summarizing into the Core at Its Edge." This solution defeats the goals of the distribution layer, however, and causes queries for networks in the branches to be propagated into the core.

A second solution is to have another reliable link connecting the distribution layer routers within a region. Routes that are advertised over this link will not be summarized, but both distribution layer routers will contain all of the components from each other. The link between the distribution layer routers should be robust enough to support both any remote-to-remote traffic and traffic passed between the two routers advertising summaries in the case of multiple remote site link failures. On most corporate networks, remote site to remote site traffic is negligible, but there are some situations where the traffic levels can be a major consideration, for instance, when voice over is running between the remote sites.

Another, similar solution is to configure a tunnel between the distribution layer routers and use this link as an alternative path in the event of a distribution-access link failure. This technique is often used if the cost or availability of robust links from distribution router to distribution router precludes the use of a physical link.

Obviously, the preferred solution to the summarization toward the network core problem is to have a relatively high-speed and reliable link connecting the distribution layer routers within a region, given that little remote-to-remote traffic will exist. Figure 3-5 illustrates the new design.

Figure 3-5 *Links Between Distribution Layer Routers*



The first thing to note in Figure 3-5 is that no link exists between the two center distribution layer routers. A link here would cause too much route leakage between the distribution sublayers.

Summarizing Toward the Remote Sites

You should perform summarization on the interfaces outbound to the remote sites and toward the core. The purpose of this summarization is to limit the routing updates to the remote routers so that they contain only a default route or major net routes. Without the summarization, all the components in the region are sent to the remote sites. As explained later in this chapter in the case study "Troubleshooting Stuck-in-Active Routes," unnecessarily sending intraregion component routes to remotes causes the remote sites to be included in the query process, which is not good. The easiest way to create convergence problems in a large-scale EIGRP network is to do nothing about restricting the range of queries initiated when a route is marked active by a router. Each hop a query must take to resolve the reachability status of a specific destination increases the chances of a major convergence failure in your network.

In addition, if the routes are not summarized from the distribution routers to the remote routers, significantly more work and traffic are required to start up the distribution-to-remote neighbor relationship. Because smaller bandwidth links tend to be used between remote sites and the distribution layer, decreasing the EIGRP bandwidth requirements at startup is wise. You can use either a **summary-address** or a **distribute-list** statement to summarize routing information toward remote sites.

NOTE For more information on how to implement the **summary-address** and **distribute-list** statements, refer to the "Summarization Methods" case study later in this chapter.

In the section "Summarizing into the Core at Its Edge," you discovered that summarization within the network core has some advantages, but it also adds undesirable complexity. Summarizing from the distribution layer into the core decreases the EIGRP query range while reducing complexity within the network. Therefore, it is better to summarize into the core instead of within the core.

After you have decided to summarize from the distribution layer into the core, summarization within the core is unnecessary. Because each distribution layer router is sending only summary information to the core, you should not have much to summarize at the core edge into the core.

Analyzing Routing in the Network Access Layer

Normally, you can classify access layer routers as single-homed or dual-homed. The sections that follow present each type along with alternative methods of supporting them.

Single-Homed Sites

Single-homed sites are those that have only a single path into the rest of the network; single-homed remote sites typically have few routes to advertise upstream. True single-homed sites do not have dial backup or any other additional path into the distribution layer. As such, true single-homed remote sites tend to be less common.

Generally, you can handle singled-homed remote sites in two obvious ways:

- Running EIGRP out to them (allowing them to advertise their locally connected networks)
- Not running EIGRP out to them

If EIGRP is running out to the remote router of the single-homed remote site, the remote router can advertise any reachable destinations using EIGRP. In this case, the question becomes this: What should the distribution layer router to which the single-homed remote is connected advertise to the remote site?

By definition, a single-homed remote site really does not have routing decisions to make. That is, if the address is not local, it must be reachable through the link to the distribution layer. For this reason, limiting the routes that are sent from the distribution layer to the remote to the minimum number possible is particularly appropriate. Believe it or not, the minimum can be one or even none.

You can either send a single default route from the distribution layer router to the single-homed remote site, or you can filter out all updates from the distribution layer router to the remote site and define a static default route in the remote site pointing back to the distribution layer router. The latter is more efficient. In this way, the routes from the remote site are learned dynamically for delivery of traffic to the remote site, but a static route is used for the traffic that is inbound from the remote site.

If you do not want to run EIGRP between single-homed remote routers and the distribution layer router, you can use static routes at both routers. Because EIGRP is not running between the remote and the distribution layer routers, the distribution layer router cannot learn dynamically about destinations that are reachable at the remote site.

To provide the rest of the network with information about destinations that are available at each single-homed remote site, you can configure static routes at the distribution layer router pointing to the appropriate access router for each remote network. This is ideal when links to the remote sites are not robust. Because EIGRP is not running over the link, it is not affected a great deal if the link often fails. Therefore, it cannot create problems for the remainder of the network because of Stuck-in-Actives (SIAs).

The disadvantages of this approach are the administrative overhead of defining a multitude of static routes and then maintaining them when the network topology changes. Typically, you should only use this approach if you are trying to eliminate problem links from the query and update path for EIGRP.

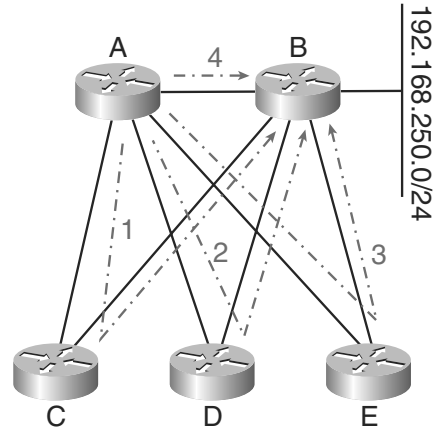
Dual-Homed Remotes

The second category of access layer routers, dual-homed remotes, is much more common than single-homed remotes. Some are permanent dual-homed remotes, like the remotes illustrated in Figure 3-1, with two or more low-speed connections to two different distribution routers from each remote site. Although the purpose of the two connections from the remote could be for load balancing, they are usually for redundancy. These important remote sites are connected in such a way that a Frame Relay permanent virtual circuit (PVC) failure or distribution layer router failure does not cause them to lose access to the core of the network.

Sites with a single permanent link combined with an on-demand backup link, such as a dial-up, ISDN dial-up, or on-demand switched virtual circuit, also need to be treated as if they are dual homed remotes. Even though such sites don't have two permanent connections into the network distribution or core layers, when the permanent link and the backup link are both in operation (which normally happens after a primarily link failure has been corrected, and the backup link has not yet been disconnected) the remote site will present all the same challenges as a dual homed remote.

Distribution layer routers that are attached to these dual-homed remotes see each of the remotes as an alternative path to reach elsewhere in the network. They appear to be transit paths or alternate paths through the network. For an example, look at Figure 3-6.

Figure 3-6 *Dual-Homed Remote as a Transit Path*



With a default EIGRP configuration that is running on all the routers shown in Figure 3-6, Router A sees four paths to the 192.168.250.0/24 network:

- Router C to Router B
- Router D to Router B
- Router E to Router B
- Through Router B

Router A would normally choose the route directly through Router B to reach 192.168.25.0/24, but if that route fails, Router A chooses between the remaining three routes or, possibly, load shares between them. This might be fine from a traffic standpoint; you can size the links to handle the load, and so forth.

From a network scaling perspective, however, this is more problematic. Router A sees each of these paths as a path through which it must query if the 198.162.250.0/24 network fails, and it holds each of these paths in its topology table, consequently wasting memory.

Summarizing outbound from the distribution layer, as discussed in the section "Summarizing Toward the Remote Sites," effectively limits the number of paths Router A sees to reach the 192.168.250.0/24 network. Because the remote routers will not have routes to this specific network through Router B, they cannot advertise it back to Router A.

This fact is important in the EIGRP network and most common EIGRP network designs because so many remotes are dual-homed. Summarizing to the greatest possible extent from the distribution layer into these remote site routers is important. Configure the distribution layer routers with distribution lists or summary address statements so that the access layer routers receive only a default route whenever possible.

Dual-Homed Remotes and Best Next Hop

Some remote sites might have links into geographically diverse locations with distinct sets of services available at each hub site. For instance, a single remote site might have links to New York City, where a mainframe with all the financial applications resides, and to San Jose, where all the human resources applications reside. In this situation, it may be better to direct traffic towards the hub location closest to the server (and application) the source host is trying to reach, rather than just routing to one of the two hubs based on a load sharing algorithm, or routing to the closest hub.

If a dual-homed remote site needs to select the best next hop to reach certain destinations (typically Data Centers or common services areas), specific routes to those destinations must be propagated to the remote routers so that path selection can take place. Of course, allowing these additional routes increases the work required to bring up the adjacency between the distribution router and the remote router and possibly allow the feedback of routes from distribution router to remote router to distribution router as described previously. How do you deal with this situation?

If a limited number of routes is being allowed from the distribution layer router to the remote router, the additional overhead of bringing up the link should not be severe. Limit the number of routes advertised to the remotes to a bare minimum.

What about those additional paths that the remote routers will be advertising back into the distribution layer? You need to eliminate the possibility of the distribution layer routers seeing the remote routers as transit paths back to other distribution layer routers.

You can prevent those routes from being readvertised from the remote routers back into the distribution layer by configuring distribution lists (filtering the routes advertised by the remote routers toward the distribution layer routers), allowing only the routes at that remote site in routing updates. In other words, the filters permit routes that originate at the remote site, and not routes that are learned via the links to the distribution layer.

Configuring route filters at the remote site's routers can prevent information learned through one hub from being forwarded to the other hub, and can also act as an insurance policy against remote site router misconfiguration disasters. A missing summary address statement or distribution list on a distribution router causes the remote site to learn more routes than it should, possibly causing havoc.

In some situations, a route that inadvertently leaks from the distribution layer toward a remote router might be the best route at the other distribution layer router, causing all the traffic to be routed through the remote site. This could be a disaster because it is not likely that the links to the remotes are provisioned to support the traffic that is transmitted through the site if this occurs. It could cause failed neighbors and network instability.

In the sample network shown in Figure 3-6, the distribution lists in the remotes are not necessary because every distribution router has the same level of summarization. To be safe, however, you should configure distribution lists.

Analyzing Use of the Stub Feature in Access Routers

Configuring a remote router as a stub, when used in conjunction with the summarization techniques described in the previous sections, can dramatically improve scaling in dual-homed remote routers. Because many networks are composed of large numbers of small access routers, which are either single- or dual-homed to the distribution layer, the stub feature is extremely valuable in many EIGRP networks. What does configuring a router as a stub actually do? Stubs limit the query scope and simplify the network topology, improving EIGRP network convergence.

NOTE Throughout this section, you will see discussion of controlling query propagation as an important part of configuring a remote router as a stub. Discussion of the importance of controlling query propagation occurs in the "Controlling Query Propagation" case study later in this chapter. In summary, queries are always propagated one hop past a summarization point. If you configure summarization on the distribution routers toward the remote routers (as recommended in the previous sections), queries are propagated one hop beyond the distribution layer routers, to the remote site routers in the access layer, even though the answer to the query is never found there. This is not much more work on the remote site routers, but it causes a great deal more work on the distribution layer routers, because they need to generate and track one query per remote router. Therefore, summarization succeeds at limiting updates to the remotes, but it still allows queries to reach the remote routers.

The active process is designed to find unknown loop free paths through the network. Why not take a short cut in the active process, and simply not search in places where you know an alternate path could not exist? You could cut down on the query range, and improve network convergence time, dramatically. In fact, there are routers a network designer knows, just by

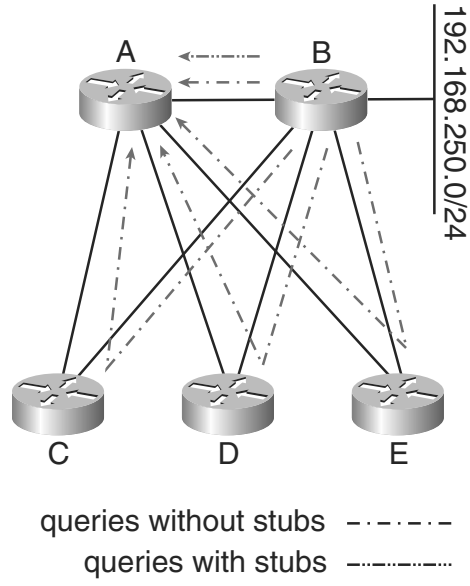
examining the network design itself, will never be used as an alternate path, or a transit path, no matter how many links in the network fail. In EIGRP terms, these are stub routers.

A stub router is a router on the edge of the network, or, in other words, a router with no routers farther from the core attached to it. If you view the network topology as a tree (the same way a link-state protocol would build a tree of the topology within a flooding domain), edge routers are always nodes at the farthest point possible from the center of the tree, and through which no traffic should (or would) ever pass. EIGRP allows the network designer to explicitly mark stub routers as stub routers. EIGRP will never search for an alternate path through a router marked as a stub.

How does configuring a router as a stub stop the router from receiving queries? When a router is configured as a stub, it flags itself as a stub by setting bits in its hello packet. Each neighbor of a stub router notes these flags and sets corresponding flags in the neighbor's data structure.

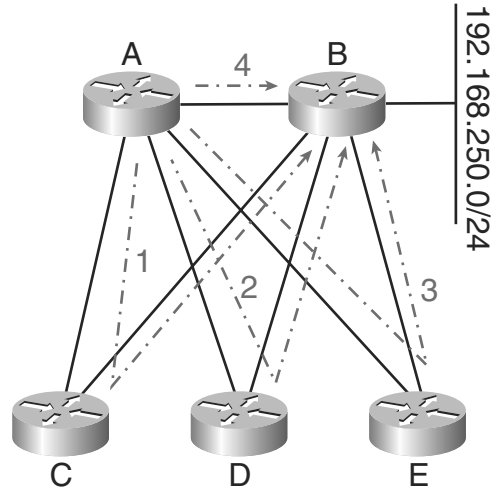
When the EIGRP process on a router loses all the successors and feasible successors for a route, it begins a diffusing update by marking the route active and sending queries to each of its neighbors (except, possibly, those attached to the same interface as the old successor). Before sending these queries, however, it looks at the peer information to determine if a peer is a stub router. If a peer is a stub router, it is removed from the list of neighbors to send queries to. Figure 3-7 illustrates the impact of declaring the remote routers as stubs. As illustrated in Figure 3-7, declaring your remote routers as stubs dramatically reduces the number of queries on the network. In that diagram, only one query is sent instead of many.

Figure 3-7 *Queries and Stub Routers*



Another important aspect of the stub feature is how it decreases the apparent complexity of the topology by limiting the types of routes that a router advertises. In Figure 3-8, Router A finds four alternate paths to 192.168.250.0/24, one through each remote router, and one directly to Router B.

Figure 3-8 *Paths Through Remotes and Stub Routers*



The stub feature significantly alters this behavior, simplifying the convergence process. When a router is defined as a stub, it must be configured with the types of routes that the stub router will advertise. By definition, an EIGRP stub router does not advertise dynamically derived routes (routes learned from other EIGRP neighbors). In Figure 3-8, this means that Routers C, D, and E will not advertise any route they learned from Router B. If Routers C, D, and E are configured as stub routers, Router A is left with one path to 192.168.250.0/24, through Router B.

The command syntax for configuring the stub feature is as follows:

```
rtrA(config)#router  eigrp 1
rtrA(config-router)#eigrp  stub ?
  connected      Do advertise connected routes
  receive-only   Set IP-EIGRP as receive only neighbor
  redistributed  Do advertise redistributed routes
  static         Do advertise static routes
  summary        Do advertise summary routes
```

Most of the options are relatively obvious:

- **connected** tells EIGRP to advertise connected routes only.
- **receive-only** tells EIGRP not to advertise routes, just to accept routes it receives from neighbors.
- **redistributed** was added a couple of years after the stub feature was created because of requirements given by customers in the Customer Proof of Concept labs at Cisco. This option allows a stub router to re-advertise routes learned through redistribution.

- **static** permits the router to advertise locally redistributed static routes. Although this option is no longer necessary (because the **redistributed** option was added), it has not been removed. That way, it will not surprise customers who have it defined in their configurations.
- **eigrp stub static** does not cause the redistribution of static routes, but allows only the advertisement of redistributed static routes. You still need to configure static route redistribution, using **redistribute static**, to redistribute static routes on the stub router.
- **summary** tells EIGRP to advertise locally created summary routes. Because these are a special category of local routes, they need to have their own operand.

You can define more than one operand on the **eigrp stub** command. For example, you can define the following command:

```
eigrp stub connected summary redistributed
```

This command causes EIGRP to advertise all connected, summary, and redistributed routes. If you do not define operands (you configure just **eigrp stub**), both connected and summary routes are advertised.

Analyzing Routes to External Connections

Another area to be concerned with is injecting information learned from other routing protocols into EIGRP. Typically, you would inject this information along the edge of the network, or from networks not originally planned to be a part of the EIGRP routing domain, such as the network of an aquired or partnering company. You can classify these external sites in two ways:

- Those that have a limited scope of addresses, such as connections from the routing domain into another company's network or other divisions of the company that fall under other administrative control.
- Those that do not have a limited scope of addresses, such as an Internet connection.

This section describes several methods to propagate information about these external destinations. First, if the external routing domain has a limited number of IP networks, you can redistribute the routes into EIGRP from the other routing domain.

NOTE Carefully consider the security of the routing system when redistributing routes from an external routing domain. See Chapter 8, "Routing Protocols Security," for more information on this topic.

Redistributing routes into EIGRP can be a reasonable choice if done correctly. If done poorly, however, redistribution can create a disaster. Refer to the "Redistribution" case study later in this chapter for techniques on preventing problems when redistributing routes from EIGRP into and from other routing protocols. The "Case Study: Redistribution" section focuses more exclusively on redistribution between Interior Gateway Routing Protocol (IGRP) and EIGRP for combining networks and for transitioning from IGRP to EIGRP.

NOTE If you have not already transitioned from IGRP to EIGRP in your network, you should. IGRP is being removed from Cisco IOS Software Release 12.3, so now is the perfect time to make the switch.

If the external connection is to the Internet, redistributing the routes into EIGRP is probably not a good idea, unless you enjoy cleaning up after complete network failures. The Internet has entirely too many routes; you would overpopulate the routing tables. Generally, from within a routing domain, you should use a default route to reach the nearest border with the Internet, and then use the more specific routing information on the border router to route correctly toward the Internet.

You can propagate information about the default route into EIGRP in two ways. First, you could define a static route to 0.0.0.0/0 and redistribute this route into EIGRP from a border router. One problem with this approach is routers configured with **ip summary-address eigrp** *AS* **0.0.0.0 0.0.0.0** will not forward traffic to a default route (0.0.0.0/0) learned from a neighboring router. Why not?

A local summary route has a default administrative distance of 5, whereas the external default route has an administrative distance of 170. Therefore, a redistributed static route will never be installed if a competing locally generated summary default route exists. Either the local router must have a static route with a better administrative distance than the summary, or the summary must be configured with an administrative distance higher than 170.

The second way to propagate information about the default route into EIGRP is to mark a route as a candidate default using the command **ip default-network**. However, this is not the preferred method of providing a default route into an EIGRP network.

NOTE The capability to distribute a default route through the command **default-information originate** is planned for a future release of Cisco IOS Software.

NOTE Cisco is planning to remove support for the command **ip default-network** in a future Cisco IOS release.

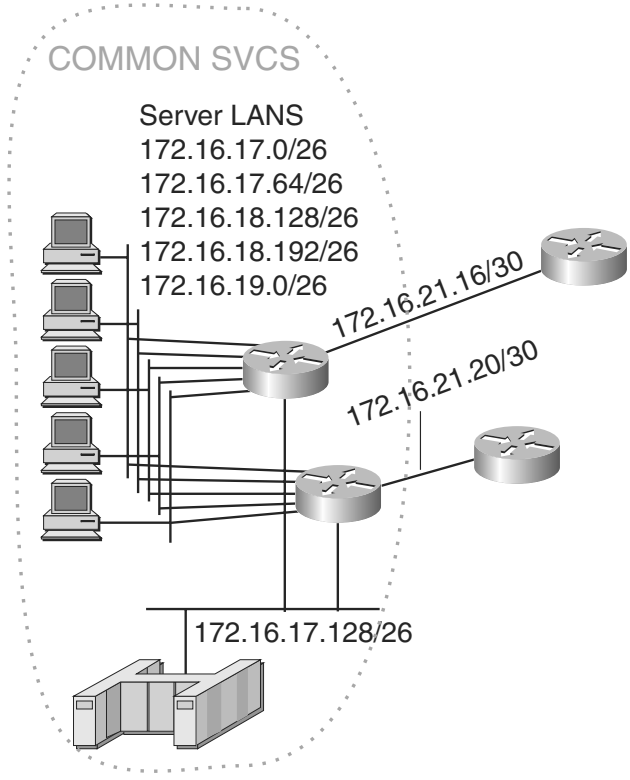
Analyzing Routes to the Common Services Area

In the network illustrated in Figure 3-9, common services are connected to the core through two distribution routers and via multiple, parallel Fast Ethernet links. Whether these are truly separate physical links or VLANs that are connected through switches, to EIGRP they present the appearance of multiple parallel paths interconnecting the two distribution routers. One of

the more typical errors that network designers make is to include all of these parallel paths as alternative paths for routes to reach much of the rest of the network.

Ideally, the servers on these segments point their default gateway to a Hot Standby Router Protocol (HSRP) address shared by the two distribution routers. This design allows the servers on these segments to adapt to a router or link failure almost immediately.

Figure 3-9 *Common Service Connections*



The networks that connect the servers to the routers are not designed for transit traffic; traffic is not expected to enter the common services distribution router from the core, go through one of the Fast Ethernet links used by the common services, and then exit through the other distribution router back to the core. EIGRP, however, does not know this, because every link between the two distribution routers appears as a possible path to every destination in the network. EIGRP treats each of these links as an alternate path, stores information about them in the topology table, and propagates queries through them. These alternate paths complicate the EIGRP convergence.

To eliminate the possibility of these networks being used for transit traffic, the network manager should run EIGRP on as few of these links as possible. Configuring **passive-interface** *interface* for an interface or subinterface removes EIGRP from these interfaces. Although EIGRP will continue to advertise the IP addresses for the interfaces that are declared passive, EIGRP Hellos will not be sent and neighbors will not be formed on them. This eliminates their use as transit paths for traffic.

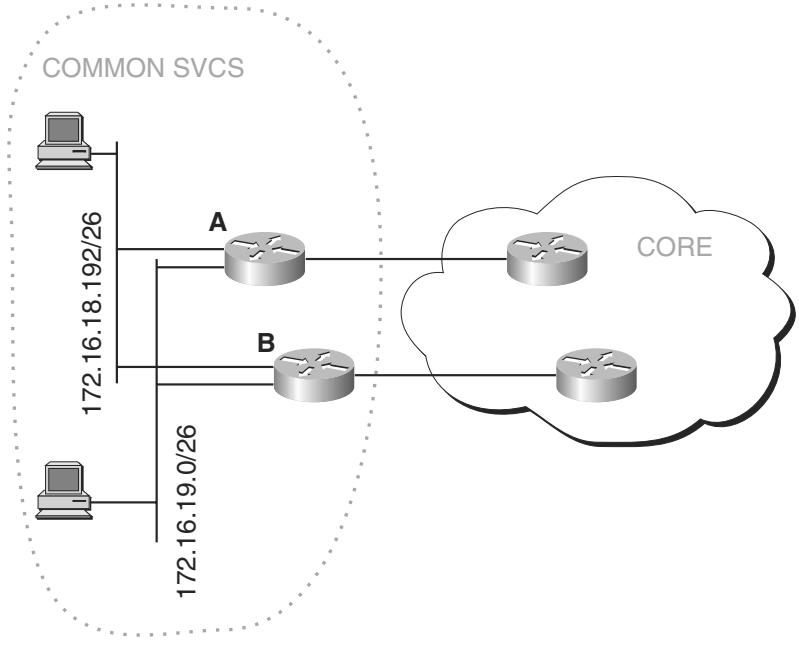
To prevent the rest of the routers in the network from going active on individual segments that support these servers, you should use the same strategy that is used everywhere else in the network. Summarize the subnets that reside on the common service Ethernet connections in both distribution layer routers so that they send only a single summary route to the core. If a single Ethernet connection goes down in the common services area, the remainder of the network does not start the query process to find an alternative path. The query stops at the first router that does not have knowledge of the specific subnet that has failed, which is a core router.

This strategy has one problem, though: It can create routing black holes in the same way that dual-homed remotes can. To understand why, examine Figure 3-10, which has all but two of the common services networks removed.

Router A and Router B both advertise a summary of 172.16.16.0/22, which covers the entire address range but does not overlap with other addresses in the network. If the Router A interface on the 172.16.18.192/26 network fails, Router A continues advertising the 172.16.16.0/22 summary toward the core. If, however, one of the core routers forwards a packet that is destined to the 172.16.18.192/26 network toward Router A, Router A drops it because it has no route for this destination. Even worse, it might send the packet back toward the core along its default route.

To resolve this situation, Router A must know that 172.16.18.192/26 is reachable through Router B. This is why you should run EIGRP over at least one of these parallel Ethernet links. To do this, do *not* put a **passive-interface** statement into the configuration for at least one Ethernet link. A better solution is to have one or two links between these routers for dedicated redundancy (with no servers or other devices on them) to account for just this situation.

Figure 3-10 *Simplified Common Services*



Analyzing Routes to Dial-In Clients

Dial-in access creates several issues and complications. This section discusses host routes created by the dial process and EIGRP bandwidth concerns.

Host Routes

Typically, dial in is handled through PPP. When a PPP session is initiated, a host route (/32) is created on the access server for the remote site, and the host route is removed when the call is dropped. If the number of dial-in clients is large, this can create a significant amount of network activity because the network reacts to these host routes appearing and disappearing.

You can eliminate this influx of network activity in EIGRP in two ways. First, you can define the command **no ip peer host-route** on the interface(s) of the access server, which stops the host route from being created in the first place.

Second, you can summarize the host routes learned via the dial interfaces, allowing only this summary route to be advertised toward the core. You can do this summarization either by configuring **ip summary-address** *autonomous system* **eigrp** on the links toward the core, or by configuring a **distribute-list out** on the links toward the core, as discussed in the "Summarization Methods" case study later in this chapter.

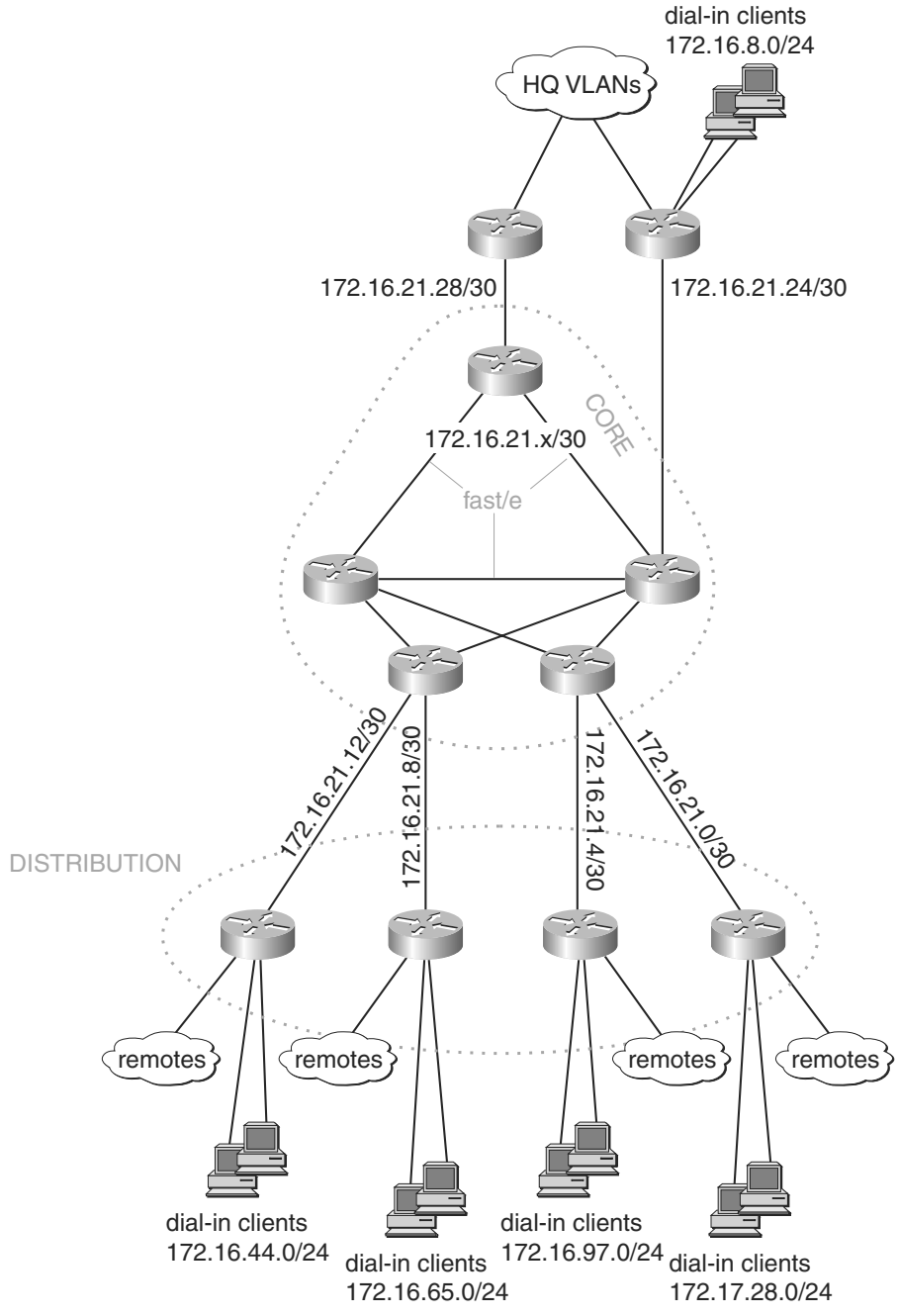
If the routes advertised by a router dialing into the network are normally summarized someplace other than the router accepting the dial-in connection, you can wind up with some major problems in your network. There are several problems with routers advertising routes towards the core of the network beyond the point where those routes are normally summarized.

As long as the dial-up link is up, the path through the dial-up link will be preferred towards the remote site. Once the primary link is fixed, it's common for the dial-up link to remain up for some time. In this situation, it's not desirable for the traffic to the remote site to continue to be routed over the dial-up link.

To make matters worse, once the primary link is repaired, the dialing router may actually leak more specific routes into the core of the network through the dial-up link, drawing all the traffic for every possible destination behind the summary onto the dial-up link. If you configure your dial-up links so a remote router will dial in to a destination between the summarization point for the routes advertised by that remote router and the core of the network, you need to make certain you take these possible problems into consideration in the network design.

Figure 3-11 illustrates the technique of making certain the dial-up links are terminated behind the summarization point, in relation to the network core.

Figure 3-11 *Addressing Dial-In Clients*



Bandwidth Issues

Bandwidth can be an issue when routers, rather than individual hosts, are dialing into an access server. EIGRP uses the bandwidth that is configured on the interface (using the **bandwidth** command) to determine the rate to pace EIGRP packets. EIGRP paces its packets so that it will not overwhelm the link by using 50 percent of the defined bandwidth by default. Because EIGRP relies on the bandwidth that is configured on the interface for packet pacing, it is important for the interface to be configured correctly. The interface should reflect the real bandwidth that is available on the link.

If EIGRP believes that the interface has more bandwidth than what is actually available, it can dominate the link, not allowing other traffic to flow. If EIGRP believes the interface has much less bandwidth than it actually does, it might not be able to successfully send all the updates, queries, or replies across the link because of the extended pacing interval.

To make things more complicated, the bandwidth that is used to determine the pacing interval is divided by the total number of remote peers on Integrated Services Digital Network (ISDN)

PRI and dialer interfaces in an attempt to fairly distribute the available bandwidth between the neighbors that are reachable through that interface.

With Frame Relay multipoint interfaces, this works fine. With ISDN or dialer interfaces, however, you never know how many neighbors will be dialed in. If only one Basic Rate Interface (BRI) is dialed in, the bandwidth should be defined as 64 kbps. If 23 BRIs are dialed in, the bandwidth should be 1.544 Mbps. Because the defined bandwidth does not change with the number of neighbors dialed in, you should set the bandwidth to make it work for both extremes by doing the following:

- Define the dial-in interfaces as dialer profiles instead of dialer groups or dialer interfaces. This allows you to set the bandwidth per dialed-in peer. However, it is an intense administrative approach.
- Summarize the EIGRP updates out of the dial link to make the amount of traffic so insignificant that it can fit across the link regardless of how much actual bandwidth is available. Refer to the earlier section titled “Summarizing Toward the Remote Sites” for more detail on this approach.

Deploying EIGRP on a Two-Layer Hierarchical Network

Now that you have had an opportunity to consider many of the techniques that are available to improve EIGRP stability and scalability in a three-layer hierarchical network design, you can explore another common design choice. Many companies that have smaller networks either geographically or topologically, or networks that have stricter latency requirements, use a two-layer network design instead of the traditional three-layer design. This section discusses how to use some of the techniques described in the previous sections on the three-layer hierarchy in the simpler, two-layer environment.

NOTE Some networks that have three layers from a switching or bridging perspective are actually two-layer networks from a routing perspective. A switched access layer combined with a routed distribution layer actually appears as one logical routing domain from the perspective of the routing protocol. They combine to form an aggregation layer.

As described in Chapter 2, a two-layer hierarchy consists of the core and aggregation layers. The sections that follow describe the scalability and design techniques that are appropriate for each of these two layers. The design principles that are outlined in the discussion of a three-layer hierarchy also apply in a two-layer hierarchy.

Summarization in the Core

The core in a two-layer hierarchy performs the same functions as the core in the three-layer hierarchy, moving traffic as quickly as possible. The biggest, fastest routers in the network reside at the network core. They are configured with minimal performance-degrading features to minimize latency and maximize performance. Typically, route policy, filtering, and summarization are avoided in the heart of the core.

Even though summarization in the center of the core is normally not encouraged, summarization from the core to the aggregation layer is often an excellent design choice. If the core is robust enough (and it should be), summarizing from the core to the aggregation layer can minimize the information known in the aggregation zones and minimize the number of queries sent into the aggregation zones. The design principles, problems, and solutions that are common in core-to-distribution and distribution-to-access layer summarization are also applicable in core-to-aggregation layer summarization.

Summarization in the Aggregation Layer

The aggregation layer within a two-layer hierarchy takes on the same attributes as the access and distribution layers, compressed into a smaller topological space within the network. Summarization toward the core of the network is the primary concern, with the same problems and solutions discussed in relation to summarization from the distribution layer in a three-layer hierarchy into the core.

Summary of EIGRP Network Design

The previous sections explored how you can apply the best summarization techniques to an EIGRP network to improve its scalability. Several techniques were discussed and numerous recommendations were made to summarize routes at various points in the network. These points include the following:

- Summarizing from the network core to the distribution layer
- Summarizing from the distribution layer to the network core
- Summarizing from the distribution layer to the remote sites
- Placing distribution lists on the remote routers to limit their advertisements to contain only those routes that originate at the remote site
- Summarizing from the common services area to the network core
- Implementing passive interfaces on all but one or two common services Ethernet/Fast Ethernet links
- Summarizing from the dial access servers into the network core

By taking these steps, the network will be robust and scalable. Adding more sites requires only that the same techniques are applied to the new routers. You can add new regions by using the same summarization/distribution list techniques to minimize the scope of queries and updates in the EIGRP network and providing the most robust, stable networking environment that is possible.

New Features in EIGRP

The sections that follow present several new features that you can use to solve tricky situations you might encounter in EIGRP networks. By using these new features, you can create the most effective design for a particular network.

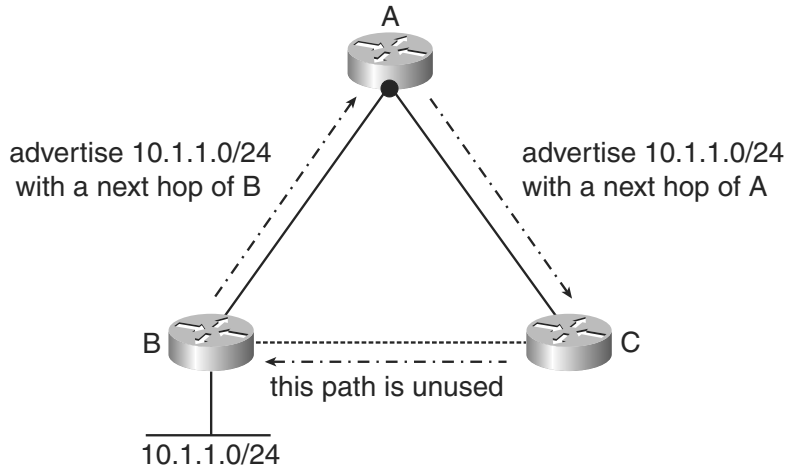
Third-Party Next Hop

Numerous problems are addressed by the EIGRP Third Party Next Hop feature. Generally, the Third Party Next Hop feature addresses the situation in which the best next hop to reach a destination is not known via an EIGRP neighbor. Prior to the creation of this feature, packets often took an extra hop to reach their destination. Two types of networks that are particularly susceptible to the extra hop problem are hub-and-spoke networks using nonbroadcast multiaccess (NBMA) networks such as Frame Relay or ATM multipoint technologies.

NBMA Hub-and-Spoke Network

The network shown in Figure 3-12 illustrates an NBMA hub-and-spoke topology to connect the distribution layer router (hub) to the access layer routers (spokes). Because most of the traffic typically flows from the access routers to the distribution routers and on to the core, the network designer often chooses not to define PVCs between the access routers. This works fine, but it can lead to the extra hop problem for any traffic that goes from one access router to another.

Figure 3-12 *Hub and Spoke with Extra Hop Problem*



In Figure 3-12, Router B sends an update containing 10.1.1.0/24 to Router A, which then propagates the route to Router C. This causes the network traffic originating on a host behind Router C and destined to a host behind Router B to go from Router C to Router A, where it is routed back out to Router B.

Some providers allow the network designer to provision switched virtual circuits (SVCs) to connect the access routers or to define the spoke-to-spoke PVCs without the broadcast option to minimize the routing complexity. In either case, a data plane connection exists between access routers, but an EIGRP neighbor relationship is nonexistent.

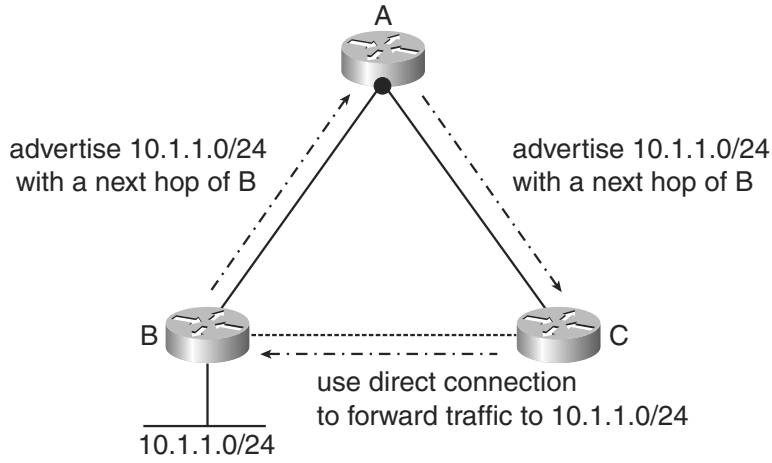
With the addition of the Third Party Next Hop feature, you can avoid this extra hop problem, without enabling the neighbor relationship between the access routers. By defining the command **no ip next-hop-self eigrp** *autonomous-system* on the NBMA interface of the hub router, the behavior changes significantly, as Figure 3-13 illustrates.

NOTE Strangely enough, prior to the coding of the next-hop feature, a **next-hop** field already existed in the update packet. Until the Third Party Next Hop feature, however, the field always contained **0.0.0.0**, which meant the receiver of the update was to use the sender's IP address as the next hop. This is what caused the extra-hop behavior illustrated in Figure 3-12.

With the new feature, things have changed. When the NBMA interface on the hub in Figure 3-13 is configured with **no ip next-hop-self eigrp** *autonomous-system*, EIGRP fills in the next-hop field in the updates and sends out the NBMA interface if the source and destination of the update are also reachable through the same NBMA interface. By definition, this means that the

router advertising the route and the router receiving the route must be on the same NBMA network.

Figure 3-13 *Hub and Spoke with Third Party Next Hop*



For example, in the network shown in Figure 3-13, the hub router (Router A) is connected via an NBMA multipoint network with Routers A, B, and C. The hub receives an update from Router B for network 10.1.1.0/24 with a source IP address of 10.1.2.2 (the NBMA interface on Router B). When Router A advertises 10.1.1.0/24 to Router C, it leaves the source, 10.1.2.2, in the next-hop field of the update.

When Router C receives the update, it has information about the correct next hop to use when reaching destinations on 10.1.1.0/24, so it can use the direct link between Router C and Router B to send the traffic, rather than the path through Router A.

In Example 3-1, the output of **show ip eigrp topology 10.1.1.0/24** on Router C shows a path to reach 10.1.1.0/24 with a next hop of 10.1.2.2, even though the associated **show ip eigrp neighbor** shows no neighbor relationship between Router A and 10.1.2.2. This could easily confuse support personnel if they do not understand this new feature.

Example 3-1 *EIGRP Topology Table Using* **no ip next-hop-self eigrp**

```
router-c#show ip eigrp topology 10.1.1.0 255.255.255.0
IP-EIGRP topology entry for 10.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 281600
  Routing Descriptor Blocks:
  10.1.2.2 (Serial0/1), from 10.1.2.1, Send flag is 0x0
      Composite metric is (3840000/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 1000 Kbit
        Total delay is 5000 microseconds
        Reliability is 255/255
        Load is 1/255
```

Example 3-1 *EIGRP Topology Table Using* **no ip next-hop-self eigrp** *(Continued)*

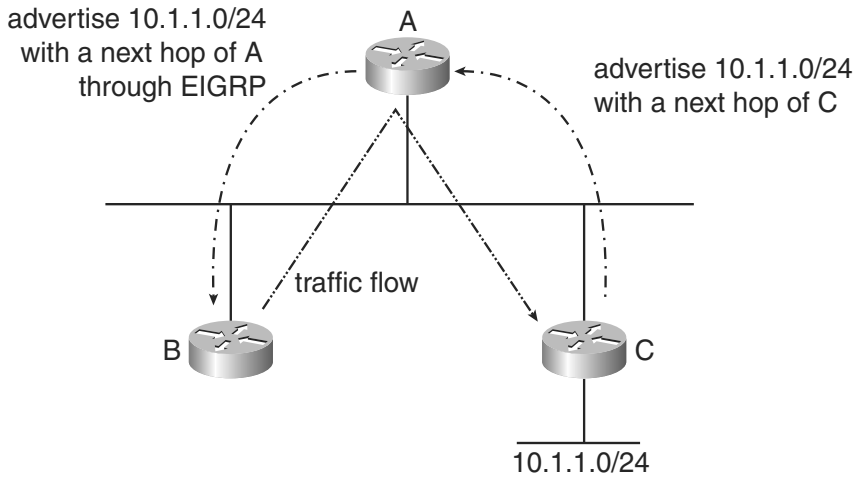
```
        Minimum MTU is 1500
        Hop count is 1
router-c#show ip eigrp neighbor
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q   Seq Type
                                            (sec)         (ms)       Cnt Num
0   10.1.2.1                Se3/0           14   1w2d       4   200  0   4
```

One extremely important requirement must be met for the next-hop feature to be implemented successfully. The link between Router B and Router C must be resilient at Layer 2, or traffic can be lost. In other words, EIGRP on Router A is trusting that it is appropriate to advertise a next hop of Router B to Router C even though it is unable to directly determine that such a path exists and is usable. Before you plan to use the next-hop feature, verify that the Layer 2 delivery mechanisms will heal Layer 2 connectivity if a link failure between the remote sites fails.

Redistributed Next Hop

Sometimes as a network designer, you encounter a situation in which the redistributing router experiences significant overhead when it shares a network with both EIGRP routers and routers that are running the redistributed protocol. To resolve this, you can configure only one of the two routers to redistribute from the external protocol into EIGRP, as Figure 3-14 illustrates.

Figure 3-14 *Redistributed Next Hop*



In this network, Router A redistributes between EIGRP and RIP and shares that same Ethernet segment with other EIGRP speakers, including Router B. Router A receives RIP routes from Router C, redistributes them into EIGRP, and then sends them back out the same interface to Router B through an EIGRP update. Example 3-2 shows the topology table for the redistributed route for this network.

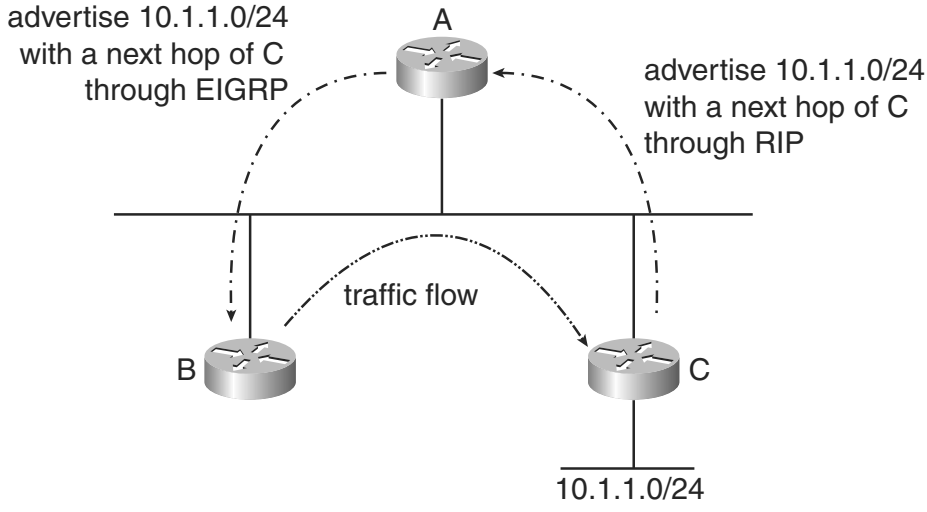
Example 3-2 *Topology Table for Redistributed Route*

```
router-b#show ip eigrp topology 10.1.1.0
IP-EIGRP (AS 100): Topology entry for 10.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  10.1.3.1 (Ethernet3/0), from 10.1.3.1, Send flag is 0x0
      Composite metric is (2172416/258560), Route is External
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      External data:
        Originating router is 10.1.3.1
        AS number of route is 0
        External protocol is RIP, external metric is 2
        Administrator tag is 0 (0x00000000)
```

Examining the EIGRP topology table on Router B, you can see the route to 10.1.1.0/24 with a next hop of Router A. Although Routers B and C connect to the same network segment, traffic between them must go through Router A, consuming more bandwidth on the network segment.

You can use the same solution that was used for the NBMA hub-and-spoke network in the previous section to solve the redistributed next-hop problem. As shown in Figure 3-15, the network administrator can configure a **no ip next-hop-self eigrp** *autonomous-system* command on the Ethernet interface that Router A shares with Routers B and C.

Figure 3-15 *Redistribution Next Hop with Third-Party Next Hop*



When Router A sends an EIGRP update to Router B, Router A looks for locally redistributed routes with next hops that are reachable via the same interface where the update is destined. If Router A finds a route matching this criterion, then Router A can determine the next-hop IP address from the routing table and insert this address in the next-hop field of the update.

When Router B receives this update, it installs the IP address from the next-hop field into the routing table as the next hop for this destination. Note that the output of **show ip route 10.1.1.0** in Example 3-3 looks similar to Example 3-1 for the NBMA hub-and-spoke topology.

Example 3-3 *Topology Table with* **no ip next-hop-self** *and Redistributed Routes*

```
router-b#show ip eigrp topology 10.1.1.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 10.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2172416
  Routing Descriptor Blocks:
  10.1.3.3 (Ethernet3/0), from 10.1.3.1, Send flag is 0x0
      Composite metric is (2172416/258560), Route is External
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      External data:
        Originating router is 10.1.3.1
        AS number of route is 0
        External protocol is RIP, external metric is 2
        Administrator tag is 0 (0x00000000)
```

Note in Example 3-3 that the next hop is set to the interface address of Router C. As a result, traffic that is destined to addresses on 10.1.1.0/24 flows directly from Router B to Router C instead of through Router A.

Enhanced Route Map Support

Although EIGRP has supported route maps in a limited fashion from the beginning, recent enhancements to EIGRP now allow much more robust and flexible use of route maps. First, you will learn what has always worked. Then you will move on to the new facilities that are available through the enhanced route map feature.

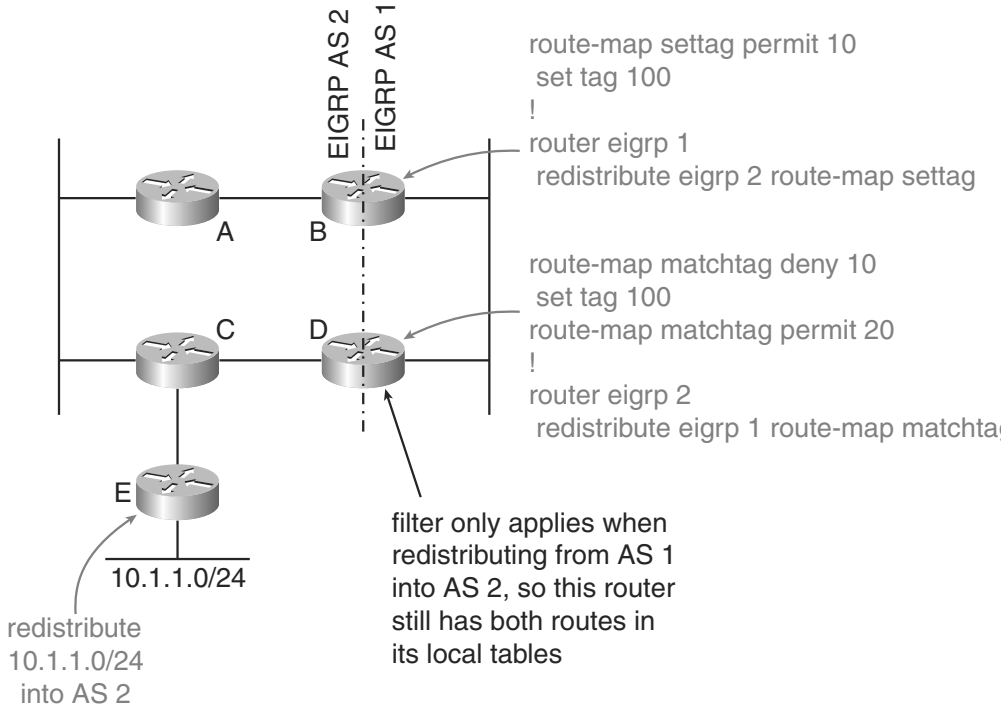
Before Enhanced Route Map Support

Before the route map enhancements were created, EIGRP supported the **route-map** command, but EIGRP had little route map capability. Some of the supported route map commands included **set tag**, **match tag**, and **set metric**. Unfortunately, the only time you could apply these **set** and **match** clauses was on a redistributing router. That is because EIGRP redistributed the

routes out of the routing table and into EIGRP. Therefore, not only could you set or match fewer things, but you could only do it on a minimal number of routers in your network.

Many network designers use the **route-map** command to filter routes based on setting and matching tags, but route maps only support filtering based on tags during redistribution, as Figure 3-16 illustrates.

Figure 3-16 *Using Enhanced Route Map Support*



NOTE The topology in Figure 3-16 is not recommended. This topology simply demonstrates the limitations of the route map support prior to the recent enhancements.

One of the dangers of this type of topology is the likelihood of creating routing loops or suboptimal routing because of information lost in the redistribution process. Normally, you should not connect multiple EIGRP autonomous systems at multiple points and redistribute at all of them. Occasionally, this sort of topology might be required. When it is required, you as the network designer should do everything you can to protect the network from routing loops.

In the network in Figure 3-16, routes that are leaving AS 1 are tagged with a value of 100 when they are redistributed into AS 2, and routes that are redistributed into AS 1 from AS 2 are tagged with a value of 200. When Router D redistributes routes from AS 2 to AS 1, it first tests the tag to make sure it is not a route that originated in AS 2. It seems like this approach will work fine.

Prior to the route map enhancements, however, you could apply the **route-map** command only on the **redistribution** statement. To see the limitation that this causes, follow an external AS 2 prefix that is being redistributed into AS 1. Prefix 10.1.1.0/24 exists in AS 2 as an external route (redistributed static or connected, possibly) and is propagated throughout AS 2. When prefix 10.1.1.0/24 reaches Router B, it is redistributed into AS 1 and a tag of 100 is applied.

10.1.1.0/24 is then propagated throughout AS 1 and eventually arrives at Router D. When Router D attempts to redistribute the route back into AS 2, redistribution is blocked by the route map's **match tag** filter. Unfortunately, the tag can be tested only as the route is being redistributed out of the routing table after being installed there in AS 1. This means that the routing loop will be blocked (the AS 2 route cannot be relearned in AS 2), but Router D can populate its local routing table with incorrect information. The route is filtered only after it is accepted in AS 1. What you really want to do is block routes that have a tag of 2 from being learned on Router D via AS 1.

Route Map Enhancements

One of the most significant enhancements to the route map support is the ability to filter routes as they are being received or sent, not just as they are being redistributed. This is accomplished via the following command:

```
router eigrp 1
  distribute-list route-map {route-map name} ?
     in
     out
     <cr>

  distribute-list route-map {route-map name} in ?
     serial
     ethernet
     ...
     <cr>
```

This permits the network designer to do the inbound filtering, which was impossible prior to the enhancements. By configuring **distribute-list route-map foo in Serial0**, for example, routes can have their tags tested prior to installing them in the topology table. This allows you to filter them prior to putting them in the routing table, which is quite an improvement over the old support.

In addition, more **match** and **set** clauses are supported by EIGRP than before the enhancements. As mentioned earlier, the only **match** clause that was supported was **match tag**, and the only set clauses were **set tag** and **set metric**. Again, you could apply these **set** and **match** clauses to

external routes only as they were redistributed into EIGRP. After the enhancements, more set and match clauses are available. You can apply them anywhere that you need:

- **match ip address**—Matches routes from the prefix list or access list.

  Routes can be filtered (denied) or have their attributes modified (via **set** clauses) based on whether a prefix matches the lists. Note that the direction of the distribution list defines what happens with this match clause. If it is applied on a **distribute-list in**, routes are accepted or rejected based on matching the supplied prefix or access list. If this **match** clause is applied via **distribute-list out**, routes are included or excluded from routing updates before sending the updates to neighbors.

- **match ip route-source**—Matches routes based on the source or neighbor list.

  The **route-source** that is supplied in this **match** clause is compared to the source of a received route, which allows you to make filtering choices or change route attributes based on the source of the route.

- **match ip route-source redistribution-source**—Matches external routes based on the originating router ID.

  Although this **match** clause seems similar to the **match ip route-source** clause, it is actually quite different. External routes include information about the router that performed the redistribution from the other protocol into EIGRP. This **match** clause allows you to take actions based on the router redistributing an external prefix into EIGRP. This **match** clause has no effect on internal routes, because no originating router is propagated in internal routes.

- **match interface**—Matches routes based on the interface that is used for the next hop.

  When this clause is used on **distribute-list in**, it limits the filter to routes that are received across the defined interface. On **distribute-list out**, the clause filters only routes that have a next-hop interface that matches the interface on the **match** clause.

- **match tag**—Matches internal or external routes based on the tag.

  This tag must have been set at some other point in the network via a **set tag** clause. As stated in the previous section, EIGRP has been able to set and match tags for years. A new capability included with the route map enhancements is the capability to set and match tags on internal routes. In the past, only external routes could be tagged, and only at the redistribution point. Now internal routes can also be tagged and filtered based on tags.

  One limitation with tags on internal routes, however, is that the number space is significantly smaller than with tags on external routes. External routes in EIGRP have always contained a 32-bit field to hold the tag value, which means that they can have values from 1 to 2^{32}. When you add the tag capability to internal routes,

however, the luxury of using a 32-bit field did not exist. Because one of EIGRP's requirements was that it always remain backward compatible, it was necessary to limit the tag on internal routes to a reserved field that was already available. This reserved field is only 8 bits wide, so the value of an internal tag can only be from 1 to 255. Although this is a significant limitation, it should certainly serve most, if not all, tagging requirements for internal routes.

- **match ip next-hop**—Matches routes based on the next hop.

  If you set the next hop for a route using the third-party next-hop feature, the next-hop and route-source fields of a route might be different. If the two fields are different, this **match** clause matches the next-hop field and filter or changes route attributes based on that next hop.

- **match metric [+–]**—Matches routes based on metric with deviation (+–).

  Filter or change route attributes based on the composite metric of a route. The deviation (+/–) allows you to match a metric within a certain range of values. It does not have to be an exact match.

- **match metric external {+–}**—Matches routes based on the external protocol metric.

  This is similar to the **match metric** [+–] command, except that the metric value it is testing is the metric of the external route at the point it is redistributed into EIGRP. If you display the topology table entry for an external route using **show ip eigrp topology** *network mask*, you see that external routes contain information on the metric from the original routing protocol. This **match** clause looks at that metric value, rather than the one inside EIGRP.

  The capability of looking at the metric from the other routing protocol before redistributing into EIGRP allows you to make filtering decisions or change route attributes based on the metric value of the external protocol. You can then favor a route taking you to the exit point of the EIGRP network that is closest to the destination in the external routing protocol.

- **match route-type external**—Matches external route based on external protocol and AS.

  This clause allows you to filter based on the external protocol (including AS number) that an external route is redistributed from. You can then deny or change route attributes if a route originally came into EIGRP from RIP, for example.

- **set metric**—Sets metric components (cannot decrease metric).
- **set tag**—Sets tag on internal or external routes.

Examples 3-4 through 3-8 provide some practical samples of how to use these **match** and **set** clauses to solve real problems.

With the configuration in Example 3-4, EIGRP filters (denies) routes matching **access-list 1** that are received from any neighbors on Serial 0/0. All other prefixes are permitted from neighbors on that interface or any other interface.

Example 3-4 *Using a Route Map to Select Which Routes to Advertise*

```
router eigrp 1
 distribute-list route-map stoproutes in Serial0/0

route-map stoproutes deny 10
 match ip address 1
route-map stoproutes permit 20
 match ip address 2

access-list 1 10.1.0.0 0.0.255.255
access-list 2 0.0.0.0 255.255.255.255
```

What if you want to deny different routes from different neighbors on multiple interfaces?

Using the commands in Example 3-5, EIGRP is instructed to filter (deny) routes matching **access-list 1** if they are received via interface Serial 0/0, and filter routes matching **access-list 2** if they are received on interface Serial 1/0. This gives much more flexibility in deciding which routes to accept.

Example 3-5 *Selective Filtering Based on Interface*

```
router eigrp 1
 distribute-list route-map stoproutes in

route-map stoproutes deny 10
 match interface Serial 0/0
 match ip address 1
route-map stoproutes deny 20
 match interface Serial 1/0
 match ip address 2

access-list 1 10.1.0.0 0.0.255.255
access-list 2 20.10.0.0 0.0.255.255
```

What if you want to accept specific routes from one neighbor on an interface, but not from another neighbor on the same interface?

The configuration in Example 3-6 tells EIGRP to filter (deny) routes matching **access-list 1**, but only if they are received from 10.1.1.1. This provides more granularity in filtering.

Example 3-6 *Selective Filtering Based on Neighbor*

```
router eigrp 1
 distribute-list route-map stoproutes in

route-map stoproutes deny 10
```

Example 3-6 *Selective Filtering Based on Neighbor (Continued)*

```
 match ip route-source 10.1.1.1
 match ip address 1
route-map stoproutes permit 20
 match ip address 2

access-list 1 10.1.0.0 0.0.255.255
access-list 2 0.0.0.0 255.255.255.255
```

What if you do not actually know which prefixes you want to filter, but you do know what part of the network they come from?

On the router that is injecting the prefixes into the network, enter the configuration in Example 3-7.

Example 3-7 *Setting Tags on Redistributed Routes*

```
router eigrp 1
 distribute-list route-map settag out Serial 0/0

route-map settag permit 10
 set tag 20
```

On the router where you want to do the filtering, enter the configuration in Example 3-8.

Example 3-8 *Filtering Routes Based on Tag*

```
router eigrp 1
 distribute-list route-map matchtag in
route-map matchtag deny 10
 match tag 20
```

Using the commands in Example 3-7 and 3-8, EIGRP tags every route being advertised into the network through Serial 0/0 from the first router and then filters the routes if they match the tag on the second router. This removes the need for knowing all the specific prefixes that are being injected at the first spot so that the filter can reflect those prefixes in the second router. This is much easier to manage than dealing with specific lists of prefixes.

By using the enhanced route map capabilities, you can define a much more specific filtering policy.

Enhanced EIGRP Active Process

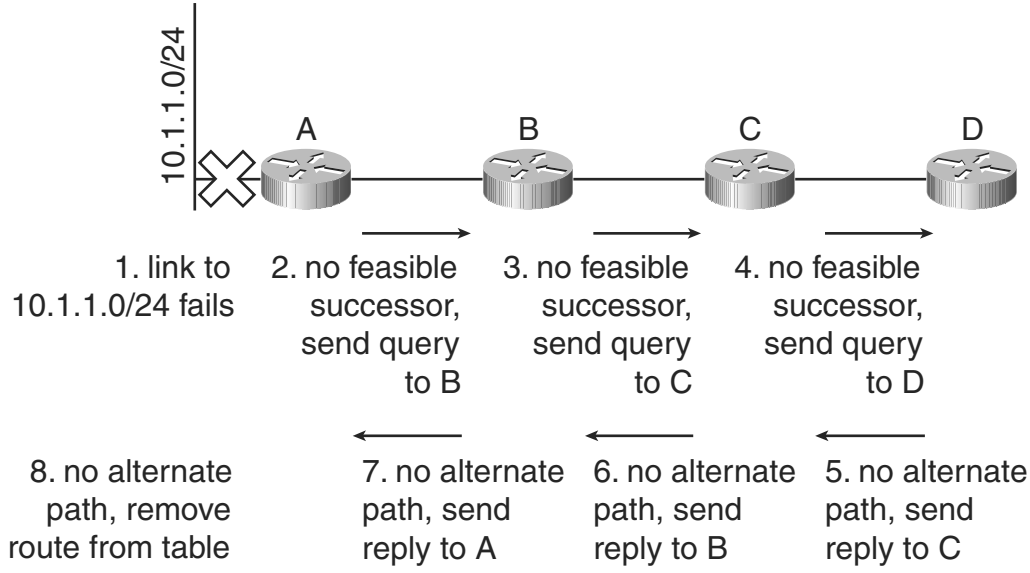
If you ask the network administrators of several large-scale EIGRP networks what their least favorite message to see in a log is, you would probably get a single, common answer—EIGRP SIAs. The active process in EIGRP is used to discover whether alternate paths to a specific destination exist, or whether existing alternate paths are loop free. If the query process fails, which generally happens only because a router does not reply to a query within a fixed time,

the EIGRP process on the originating router is outside the Diffusing Update Algorithm (DUAL) state machine. The only way out is to reset the relationship with the neighbor that has not replied to a query.

NOTE The EIGRP active process is covered in detail in Appendix A.

Begin by reviewing the active process before it was enhanced, using Figure 3-17.

Figure 3-17 *EIGRP Active Process*



The list that follows describes the labeled sequence of transactions as depicted in Figure 3-17.

1. The Router A link to 10.1.1.0/24 fails.
2. Router A examines its local topology table and finds it has no feasible successors for 10.1.1.0/24. (It has no alternate paths that are known to be loop free.) Router A marks the route as active, builds a query about 10.1.1.0/24, and sends it to Router B.
3. Router B receives this query, examines its local topology table, and determines that it has no feasible successor for 10.1.1.0/24. Router B marks the route as active, builds a query, and sends it to Router C.

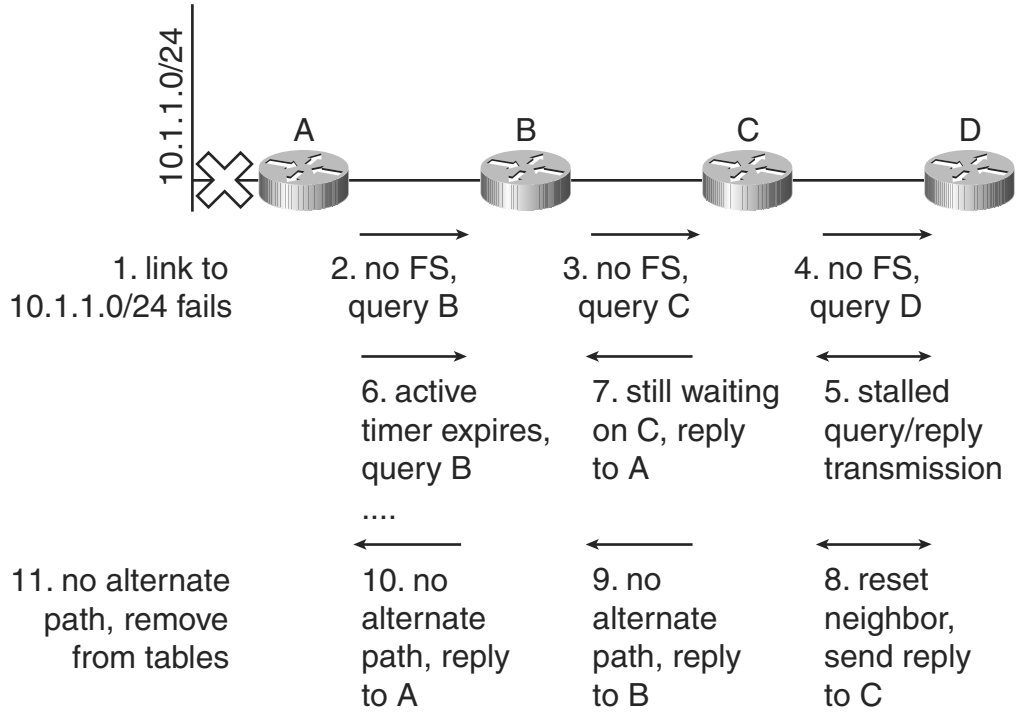
4. Router C receives this query, examines its local topology table, and determines that it has no feasible successor for 10.1.1.0/24. Router C marks the route as active, builds a query, and sends it to Router D.
5. Router D receives the Router C query, examines its local topology table, and determines that it has no feasible successor for 10.1.1.0/24. Router D has no other neighbors, so it marks 10.1.1.0/24 as unreachable and sends a reply to Router C.
6. Router C receives this reply and finds that it has no other paths through which it could reach 10.1.1.0/24. (Router C received the original query from Router B, and it just received a reply from Router D.) Router C marks 10.1.1.0/24 as unreachable and sends a reply to Router B.
7. Router B receives this reply and finds that it has no other paths through which it can reach 10.1.1.0/24. (Router B received the original query from Router A, and it just received a reply from Router C.) Router B marks 10.1.1.0/24 as unreachable and sends a reply to Router A.
8. Router A receives this reply and finds that it has no other possible paths to 10.1.1.0/24. Therefore, it marks the destination as unreachable, eventually removing 10.1.1.0/24 from its local routing and topology tables.

When Router A originally marks the route to 10.1.1.0/24 as active, it sets a 3-minute timer, called the *active timer*. If this timer expires before the Router B response is received, Router A resets its neighbor relationship with B. When this timer expires, the Router A EIGRP process has gone outside the DUAL finite state machine. Furthermore, no other alternative exists besides resetting the neighbor relationship to correct the problem.

What could go wrong with the active process? Suppose that Routers C and D have a problem communicating. Router D could be low on memory, or the link between them could be dropping a large percentage of the packets that are transmitted. While Router C is waiting on a reply from D, the active timer in Router A is still running. In fact, if it takes more than 3 minutes for the query to reach C in the first place and for D to respond to C, the active timer in Router A is guaranteed to expire before Router C receives the reply from Router D. This causes Router A to reset its neighbor relationship with Router B. There is obviously a problem here, because a glitch between Router C and Router D causes a neighbor relationship to be reset between Router A and Router B.

The EIGRP enhanced Active process (also known as the *SIA rewrite*) fixes this problem by adding a state so that the neighbor relationship reset happens where the actual network problem is. Figure 3-18 illustrates.

Figure 3-18 *The Enhanced EIGRP Active Process*



The list that follows describes the labeled sequence of transactions as depicted in Figure 3-18.

1. The Router A link to 10.1.1.0/24 fails.
2. Router A examines its local topology table and finds that it has no feasible successor to 10.1.1.0/24. It marks the route active, sets a 1-minute active timer, and sends a query about 10.1.1.0/24 to B.
3. Router B examines its local topology table and finds that it has no feasible successors for 10.1.1.0/24. It marks the route active, sets a 1-minute active timer, and sends a query about 10.1.1.0/24 to C.
4. Router C examines its local topology table and finds that it has no feasible successors for 10.1.1.0/24. It marks the route active, sets a 1-minute active timer, and sends a query about 10.1.1.0/24 to D.
5. The query/reply mechanism fails between Routers C and D. Both routers continue retransmitting.
6. The active timer in Router A expires. Router A builds an SIA query and transmits it to B.

7. Router B receives this SIA query and examines the state of its local topology table. Router B finds that it is still waiting on a reply from Router C, so it sends this information to Router A. This preserves the neighbor relationship between Routers A and B.
8. Routers C and D fail in their retransmission attempts and reset their neighbor relationship.
9. Router C now has no alternate path to 10.1.1.0/24, so it sends a reply to Router B.
10. Router B has no alternate path to 10.1.1.0/24, so it sends a reply to Router A.
11. Router A has no path to 10.1.1.0/24. It marks the route as unreachable and eventually removes 10.1.1.0/24 from its local topology and routing tables.

Case Study: Summarization Methods

You can use two basic tools to summarize routes in EIGRP:

- IP summary addresses
- Distribute lists

These two methods, which are uniquely useful, provide significantly different approaches to limiting the routing updates to a summary of the information. The best solution to a summarization problem is often a mixture of both approaches. One or both of these basic tools is applied in all three layers—core, distribution, and access—to provide the maximum in summarization, stability, and scalability. The next sections look at each tool so that you can understand the pros and cons of each.

IP Summary Addresses

The first summarization tool is an IP summary address, configured using the command **ip summary-address eigrp AS network mask** *distance*, applied to an interface. An IP summary address provides two related functions:

- An IP summary address creates a summary route in the routing table (identified as a summary route with a next-hop address of null0). It then propagates to any neighbors out of the interface with the summary address statement defined. This is called the *discard route*, which is created to prevent forwarding loops.
- An IP summary address filters out the components of the summary that would normally have been sent out of the interface with the **summary address** statement. In this way, an IP summary address sends *only* the summary information.

Although IP summary addresses are extremely flexible and powerful, they can be administratively wearisome and possibly error-prone. As mentioned previously, you need to apply the **summary-address** statement to each interface that you want to advertise the summary. On routers that contain dozens or even hundreds of interfaces and subinterfaces, you can have numerous **summary-address** statements to correctly define.

A summary route is created and sent only if EIGRP has an internal component of the summary. This means that if all components that make up the summary disappear, or only external (redistributed) components exist, the summary route is not installed and advertised.

One unfortunate side effect of the discard route is created when a IP summary address is configured. If the router that is generating the summary receives a route matching the summary (with the same network and mask) from another source, the router does not accept it. This is because the discard route that is generated by the **summary-address** command has an administrative distance of five by default, which is always better than the administrative distance of a dynamically learned route.

To illustrate, suppose that you have a router that is learning its default route through an external source (see Example 3-9) .

Example 3-9 *A Router Learning Its Default Route via an External Source*

```
router#show ip route
....
Gateway of last resort is 172.19.1.1 to network 0.0.0.0
....
D*EX 0.0.0.0/0 [170/2195456] via 172.19.1.1, 00:00:09, Serial0
```

You want to configure a **summary-address** statement that advertises the least number of routes possible out of interface serial 1 as follows:

```
router(config)#int serial 1
router(config-if)#ip summary-address eigrp 100 0.0.0.0 0.0.0.0
```

Example 3-10 shows the resulting routing table after this configuration.

Example 3-10 *Routing Table with Summary Default Route*

```
router#show ip route
....
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
....
D*   0.0.0.0/0 is a summary, 00:00:49, Null0
```

This is a problem. Any packets that should follow the default route directed toward 172.19.1.1 are sent to **null0**, the bit bucket.

To resolve this, you can use a new addition on the **ip summary-address** command:

```
router(config-if)#ip summary-address eigrp 100 0.0.0.0 0.0.0.0 200
```

The final **200** sets the administrative distance of this summary route to 200, effectively preventing the use of the discard route. Although the downstream router still receives only the 0.0.0.0/0 route, the summary is not installed in the routing table of this router because the administrative distance is higher than the external EIGRP route that you currently have. This feature is not available in all releases of Cisco IOS Software prior to Release 12.0(5)T, when the feature was integrated.

NOTE The discard route is created to prevent routing loops when a summary is configured. Using the administrative distance to prevent a discard route from being installed in the local routing table removes this protection. Use this feature carefully, generally only when summarizing toward a nontransit section of the network, such as a dual-homed remote site.

Distribute Lists

The second method that filters and summarizes routes in EIGRP involves defining a distribute list under the EIGRP configuration. This method uses a different approach than the **summary-address** statements, but it provides similar functionality. With the distribute list approach, you explicitly tell EIGRP which routes are allowed to be advertised out any or all interfaces. You enter the following command for this approach in EIGRP configuration mode:

```
distribute-list {access-list-number | prefix prefix-list-name} out [interface-name
  | routing-process | as-number]
```

The access list that is associated with the distribute list describes the route, or routes, that you can send out the interface defined under the **distribute-list** command. You can supply a wildcard mask in the access list so that more than one route is permitted under the same access list.

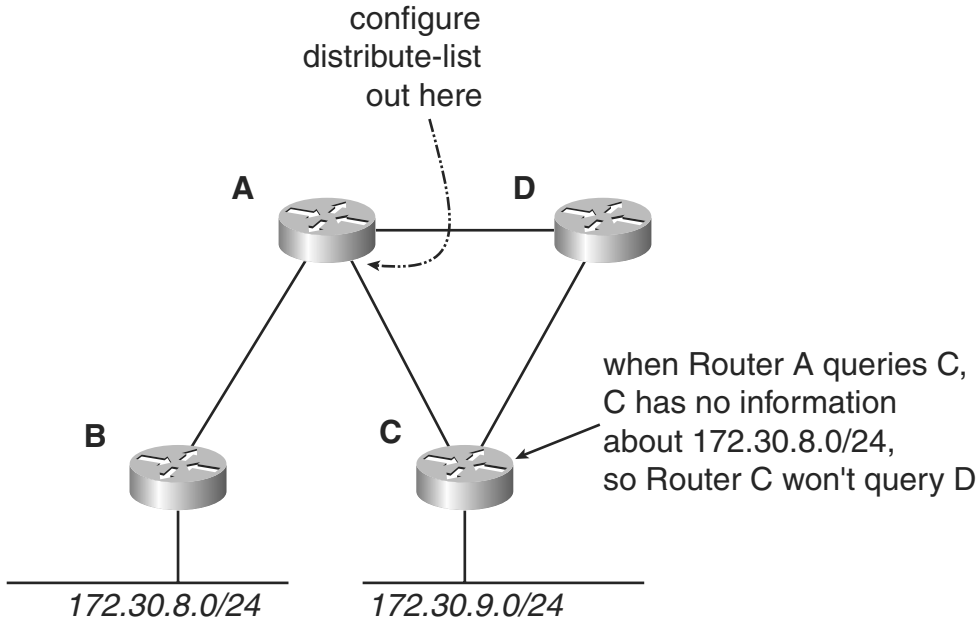
Alternatively, you can supply a prefix list instead of an access list. A prefix list is similar to an access list, but it is referenced by name instead of number and has a few additional options.

A key difference between distribute lists and summary addresses is that distribute lists do not automatically create the summary route you need to advertise. If the route that is permitted by the access list does not exist, the route is not sent. Typically, the network manager defines a static route to match the access list so that the route is always there to advertise. This static route can be floating (that is, with a high administrative distance) so that if the same route is learned from elsewhere, it is accepted and used. The static route is used only if the dynamically derived route disappears.

Case Study: Controlling Query Propagation

Not only do summarization statements and distribute lists limit the size and content of the updates that are sent to neighbors from a router, but they also control the scope of EIGRP query propagation. (See Appendix A for further details on the query process.) Consider a query propagating through the network as illustrated in Figure 3-19.

Figure 3-19 *Controlling Query Propagation*



If Router B loses its route to 172.30.8.0/24, which is directly attached, it queries each of its neighbors in search of a different path to reach this destination. Because Router B has only one neighbor, Router A is the only router that Router B queries. Router A then queries each of its neighbors, Router C and Router D, looking for an alternative path to 172.30.8.0/24. Router C queries Router D. Therefore, Router D receives two queries:

- One from Router A
- One from Router C

You know from looking at the network topology that Router D does not have a route to 172.30.8.0/24 unless Router A does. Why should you bother Router D with two queries about this network? Well, you can configure Router A so that Router D does not receive two queries.

A query stops propagating when it reaches a router that has no knowledge of the active route. Therefore, if you remove the knowledge that Router C has of 172.30.8.0/24, Router C does not propagate a query that it receives from Router A to Router D. This is where summarization and distribution lists come into play; they keep Router C from learning about 172.30.8.0/24.

On Router A, you can advertise a summary of all the routes available in the remainder of the network, 172.30.0.0/16, to Router C. When Router C receives a query for 172.30.8.0/24, it examines its local topology table and finds that it does not have a topology table entry for this particular destination network. When Router C discovers that it does not have alternate paths to 172.30.8.0/24, it replies to Router A noting that the active route is not reachable.

Case Study: A Plethora of Topology Table Entries

One of the common problems in an EIGRP network is the sheer number of alternate paths through which a given destination can be reached. Each alternate path in the topology table represents a query that must be generated if the path currently being used fails. These alternate paths, however, are not always obvious when you look at the topology table, as demonstrated in Example 3-11.

Example 3-11 *Alternate Paths to a Destination Are Not Always Displayed in a Topology Table*

```
router#show ip eigrp topology
IP-EIGRP Topology Table for process 100

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.19.2.128/25, 1 successors, FD is 2297856
         via 172.28.1.2 (2297856/128256), Serial0.1
P 172.19.10.0/24, 1 successors, FD is 2297856
         via 172.28.1.2 (2297856/128256), Serial0.1
```

The topology table in Example 3-11 shows what appear to be two destinations, each with a single path to reach it. However, the paths shown here are only a subset of what is known by EIGRP. This output does not show all the available paths. It shows only the ones that DUAL has calculated to be loop free.

To get a more accurate picture of which paths are available, you can execute **show ip eigrp topology all** or **show ip eigrp topology** for a particular destination, as demonstrated in Example 3-12.

Example 3-12 *Displaying All Paths to a Destination*

```
router#show ip eigrp topology all
IP-EIGRP Topology Table for process 100
 Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply, r - Reply status

P 172.19.2.128/25, 1 successors, FD is 2297856
         via 172.28.1.2 (2297856/128256), Serial0.1
         via 172.28.2.2 (3879455/2389454), Serial0.2
         via 172.28.3.2 (4893467/2389454), Serial0.3
         via 172.28.4.2 (4893467/2389454), Serial0.4
         via 172.28.5.2 (4893467/2389454), Serial0.5
         via 172.28.6.2 (4893467/2389454), Serial0.6
         via 172.28.7.2 (4893467/2389454), Serial0.7
         via 172.28.8.2 (4893467/2389454), Serial0.8
         via 172.28.9.2 (4893467/2389454), Serial0.9
         via 172.28.10.2 (4893467/2389454), Serial0.10
P 172.19.10.0/24, 1 successors, FD is 2297856
         via 172.28.1.2 (2297856/128256), Serial0.1
         via 172.28.2.2 (3879455/2389454), Serial0.2
         via 172.28.3.2 (4893467/2389454), Serial0.3
         via 172.28.4.2 (4893467/2389454), Serial0.4
```

Example 3-12 *Displaying All Paths to a Destination (Continued)*

```
         via 172.28.5.2 (4893467/2389454), Serial0.5
         via 172.28.6.2 (4893467/2389454), Serial0.6
         via 172.28.7.2 (4893467/2389454), Serial0.7
         via 172.28.8.2 (4893467/2389454), Serial0.8
         via 172.28.9.2 (4893467/2389454), Serial0.9
         via 172.28.10.2 (4893467/2389454), Serial0.10
router#show ip eigrp topology 172.19.10.0 255.255.255.0
IP-EIGRP topology entry for 172.19.10.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
  172.28.1.2 (Serial0.1), from 172.28.1.2, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
....
  172.28.2.2 (Serial0.2), from 172.28.2.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.3.2 (Serial0.3), from 172.28.3.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.4.2 (Serial0.4), from 172.28.4.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.5.2 (Serial0.5), from 172.28.5.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.6.2 (Serial0.6), from 172.28.6.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.7.2 (Serial0.7), from 172.28.7.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.8.2 (Serial0.8), from 172.28.8.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.9.2 (Serial0.9), from 172.28.9.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
....
  172.28.10.2 (Serial0.10), from 172.28.10.2, Send flag is 0x0
      Composite metric is (3879455/2389454), Route is Internal
```

Although this particular destination has only one successor, the number of different paths is numerous. This almost always indicates a topology that has too much redundancy; this router has at least ten neighbors, and each of them has a path to this destination. Unfortunately, no definite rules spell out how many paths are too many in the topology table. The number of alternative paths, however, indicates the total query paths in the network and, therefore, how much work the routers in the network need to do when converging on a topology change.

In general, avoid running EIGRP over multiple parallel links between two routers unless you intend transit traffic to be passed over all of them.

Case Study: Troubleshooting EIGRP Neighbor Relationships

EIGRP might experience problems establishing neighbor relationships for various reasons. To determine the source of the problem, the first thing to do is to add the command **eigrp log-neighbor-changes** under the router process in the configuration of every router. Doing so provides much more information about the cause of neighbor problems.

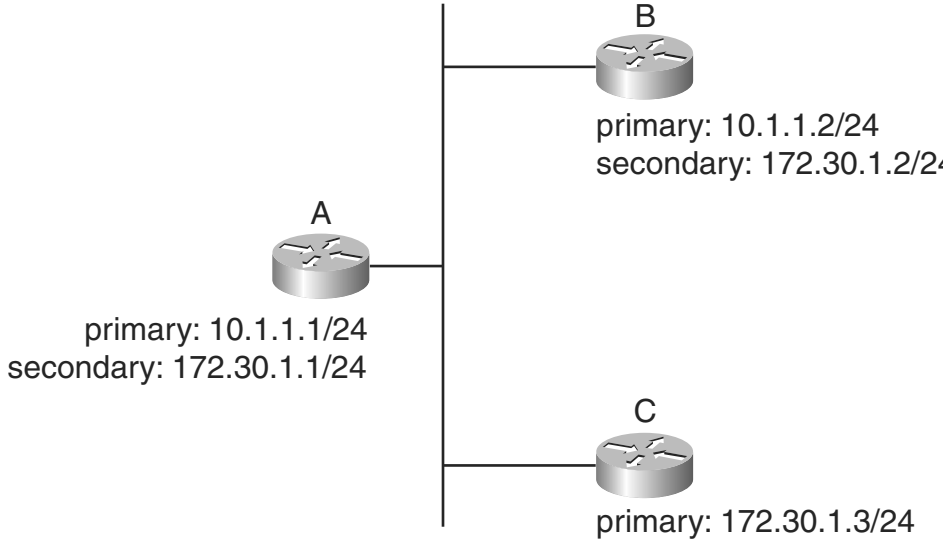
This case study describes two common problems that prevent EIGRP from establishing neighbors successfully:

- The first problem occurs when the primary addresses that are used by the routers trying to be neighbors do not belong to the same subnet.
- The second common problem occurs when the underlying media is failing to deliver either unicast or multicast traffic in one direction or both.

EIGRP Neighbor Relationships: Common Problem 1

Because Cisco routers permit the definition of both primary and secondary IP subnets on the same interface, many network implementers treat the primary and secondary addresses as equal. As Figure 3-20 reveals, this is not necessarily the case.

Figure 3-20 *EIGRP Neighbors with Different Primary Addresses*



In this network, Router C has its primary (and only) IP address in the same subnet as the secondary addresses of Routers A and B. You can see this easily by executing **show ip eigrp neighbors** on all three routers, as demonstrated in Example 3-13.

Example 3-13 **show ip eigrp neighbors** *with Primary/Secondary Address Mismatch*

```
router-a#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
1   172.30.1.3              Et0           13 00:00:15    0  5000  1  0
0   10.1.1.2                Et0           13 00:09:56   26   200  0  323
router-b#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
0   172.30.1.3              Et1           11 00:00:03    0  3000  1  0
1   10.1.1.1                Et1           11 00:11:09   23   200  0  3042
router-c#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
```

As the output in Example 3-13 indicates, Router A and Router B see Router C as a neighbor (a neighbor with a problem, however—note the Q count and lack of Smoothed Round Trip Time [SRTT]). Router C does not see Routers A or B as neighbors. This is because Routers A and B match the IP address of the source of the hello packet with any of its addresses on that interface. Because Router C falls in one of the subnets, Router A and Router B accept Router C as a neighbor.

NOTE The Q count, shown in **show ip eigrp neighbor**, indicates the number of items from the topology table that need to be sent to this neighbor. Some (or all) of these items might never be sent because of split-horizon, distribution lists, summaries, or other things. Therefore, the Q count does not indicate the number of packets that need to be sent or the number of routes that are being sent.

The SRTT, shown in **show ip eigrp neighbor**, indicates the average amount of time it takes for a neighbor to respond to packets that require an acknowledgement. It is a smoothed (or weighted) average over multiple transmit/acknowledgement cycles.

On the other hand, when Router C compares the source address of the received hellos, it does not match any of the addresses on that interface, so Router C rejects them. In some versions of IOS, the message **neighbor not on common subnet** printed on the console indicates this problem.

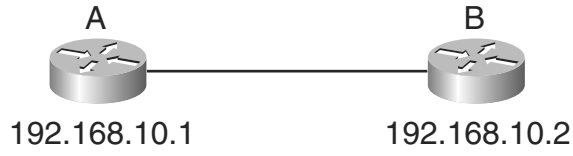
Because the source address of Router C is on a different subnet than Router A and Router B, a proper neighbor relationship is not established between Router C and the other two routers on

this subnet. To resolve this problem, Router C needs to be re-addressed so that its primary address is on the 10.1.1.0/24 subnet.

EIGRP Neighbor Relationships: Common Problem 2

Another problem often experienced with EIGRP neighbor establishment occurs when the underlying media fails to deliver unicast or multicast traffic in one direction or both. The remainder of this case study describes how it looks when you are missing multicast traffic in one direction using the network diagramed in Figure 3-21.

Figure 3-21 *EIGRP Neighbors with Multicast Delivery Problems*



Example 3-14 shows the **show ip eigrp neighbors** output for Router A.

Example 3-14 *Displaying the Router A Neighbors*

```
router-a#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
0   192.168.10.2            Se1           13 00:00:10    0  5000  1  0
```

Notice that Router B is seen in the neighbor table of Router A, but the Q count is not zero and the SRTT is not set to a value. If you have **eigrp log-neighbor-changes** configured (as you should), you also get messages on the console, or syslog, reporting that this neighbor is being restarted because the retransmit limit is exceeded. These symptoms indicate that you cannot get updates delivered and acknowledged to this neighbor, but you can see the neighbor hellos.

Now look at the **show ip eigrp neighbors** output for Router B in Example 3-15.

Example 3-15 *Displaying the Router B Neighbors*

```
router-b#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
```

Here, notice that Router B does not have Router A in its neighbor table. This indicates that the multicast packets that are sent by EIGRP as hellos are not being delivered to this neighbor. Common reasons for this include a missing broadcast keyword on a dialer map or **frame-relay map** statement, misconfiguration of Switched Multimegabit Data Service (SMDS) multicast groups, or other problem with the delivery mechanism.

Example 3-16 demonstrates a correct configuration for a multipoint Frame Relay interface.

Example 3-16 *Multipoint Frame Relay Configuration*

```
!
interface Serial 0
 encapsulation frame-relay
 ip address 172.30.14.1 255.255.255.0
 frame-relay map ip 172.30.14.2 100 broadcast
 frame-relay map ip 172.30.14.3 104 broadcast
 frame-relay map ip 172.30.14.4 210 broadcast
```

Note the **broadcast** keyword inserted at the end of each **frame-relay map** configuration command.

This symptom could also indicate that traffic from Router A is not being delivered to Router B. You can determine whether this is the case by pinging Router B from Router A. If the unicast ping works, but EIGRP is unable to see Router A from Router B, you should ping 224.0.0.10 (the multicast address of EIGRP) from Router A and see if Router B responds.

The router should forward a multicast ping to 224.0.0.10 onto every interface, and every adjacent EIGRP neighbor should respond to it. Example 3-17 demonstrates a neighbor having a packet delivery problem, and the use of the **ping** command to determine the scope of the problem. As you can see, the neighbor with a problem, 192.168.10.2, successfully responds to unicast pings but does not answer pings sent to the multicast address 224.0.0.10.

Example 3-17 *Troubleshooting Neighbor Problems*

```
router#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
4   192.168.10.2            Se1           14 00:00:05    0  3000  8  0
3   10.31.1.2               Se0.1         12 00:00:11  132   792  0  1668
2   10.31.2.2               Se0.2         12 00:00:12  131   786  0  1670
1   10.31.3.2               Se0.3         11 00:00:12  166   996  0  1669
0   10.1.2.1                Et0           10 1w4d       13   200  0  60131
router#ping 182.168.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/20 ms
router#ping 224.0.0.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.0.0.10, timeout is 2 seconds:

Reply to request 0 from 10.1.2.1, 12 ms
Reply to request 0 from 10.31.3.2, 112 ms
Reply to request 0 from 10.31.2.2, 104 ms
Reply to request 0 from 10.31.1.2, 100 ms
```

Example 3-17 *Troubleshooting Neighbor Problems (Continued)*

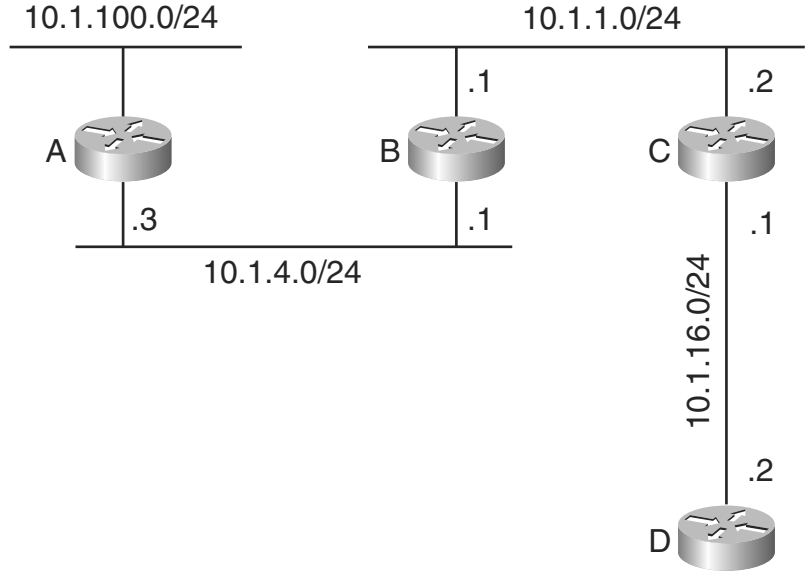
```
Reply to request 0 from 10.250.1.1, 12 ms
Reply to request 0 from 10.200.1.1, 12 ms
Reply to request 0 from 10.1.3.2, 12 ms
```

Case Study: Troubleshooting SIA Routes

SIA routes can be some of the most challenging problems to resolve in an EIGRP network. For more detail on the EIGRP active process, refer to Appendix A. In summary, a route becomes active when it goes down or its metric worsens, and no feasible successors exist. When a route goes active on a router, that router sends queries to all of its neighbors (except through the interface where the route was lost) and awaits the replies. A 3-minute timer starts when the router marks the route as active. If the timer expires without getting all the replies, the route that was active is considered stuck in active processing (thus the label "stuck in active" routes) and requires drastic actions.

Three minutes is an incredibly long time to a router. You need to understand why the replies could take longer than 3 minutes. Figure 3-22 shows a simple network that is reacting to a lost route so that you can understand how to troubleshoot it.

Figure 3-22 *Troubleshooting EIGRP SIA Routes*



Router A loses network 10.1.100.0/24 when its interface on that interface is shut down. Router A then goes active on the route and sends a query to Router B, which looks in its topology table for another successor, or feasible successor, for 10.1.100.0/24. In this case, Router B does not

have other successors or feasible successors. Therefore, it goes active on the route and sends a query to Router C. Router C goes through the same decision process, and the query continues on to Router D (and farther if possible).

During this entire process, the 3-minute timer of Router A has been running because a reply is not returned from Router B until it receives an answer from Router C, which is waiting on Router D. If something happens somewhere downstream (as it does in this case study), the timer on Router A might expire, and Router A considers the path through Router B unreliable. When that happens, Router A resets its neighbor relationship with Router B and tosses all routes previously learned through Router B. (Relearning these routes requires rebuilding the neighbor relationship.) This can be brutal if the link between Router A and Router B is a core link in your network.

You can see how to troubleshoot SIA routes on the example network in Figure 3-22. How do you know you are getting SIA routes? You know because you see messages in your log similar to this:

```
Jan  19 14:26:00: %DUAL-3-SIA: Route 10.1.100.0 255.255.255.0 stuck-in-active
  state in IP-EIGRP 1. Cleaning up
Jan  19 14:26:00: %DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.4.1 (Ethernet1) is
  up: new adjacency
```

The DUAL-3-SIA message identifies which route is getting stuck—10.1.100.0/24 in this case—but it does not reveal which neighbor did not answer. You need to have **log-neighbor-changes** configured (as recommended earlier) to get the message immediately after the DUAL-3-SIA message, stating new adjacency for the neighbor (or neighbors) that was reset because of the SIA. You can also tell which neighbors have been recently reset by looking for a short uptime in the **show ip eigrp neighbors** output. However, you cannot be sure that their reset condition was because of the SIA. Again, ensure that **log-neighbor-changes** is configured on every router. Also, send the log entries to the buffer via **logging buffered** or to a syslog server.

Because the log captured SIA messages, you need to try to determine where the source of the problem is. Ask the following two questions about SIA routes:

- Why are the routes going active?
- Why are they getting stuck?

You should work on both aspects of the problem, but the second is the most important by far and probably the most difficult to resolve. If you determine why a route is going active and resolve this part of the problem without determining why it became stuck, the next time a route goes active, it could become stuck again. Therefore, finding the cause of the stuck route is more important than finding the cause of the route going active.

Even though it is more important to find the cause of routes becoming stuck than why they went active, do not ignore why routes are going active. Using the **DUAL-3-SIA** messages printed to the router console, you can determine whether the routes that are going active are consistent. That is, are all of them /32 routes from dial-in clients coming and going, or are all of them the result of poor-quality lines at the fringes of the network? If all of them are host routes caused

by dial-in users, you should try to minimize these active routes through summarization or other methods. If the active routes are because of unstable links, you need to get these Layer 2 problems resolved.

How do you troubleshoot the stuck part of the SIA? If the SIA routes are happening regularly, and you are monitoring the routers during the time of the problem, this is a fairly straightforward job. If the problem happens infrequently, and you were not monitoring the routers when the problem happened, it is almost impossible to find the cause. For this case study, assume that the problem is happening regularly enough for you to catch the routes that are having problems.

Referring back to Figure 3-22, on Router A (where you are receiving the DUAL-3-SIA messages for 10.1.100.0/24), you look for active routes using the **show ip eigrp topology active** command, as demonstrated in Example 3-18. As you can see, the output reveals information about the state of the active route.

Example 3-18 **show ip eigrp topology active** *Output*

```
routerA#show ip eigrp topology active
IP-EIGRP Topology Table for process 1
 Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.1.100.0/24, 1 successors, FD is Inaccessible
    1 replies, active 00:01:23, query-origin: Local origin
         via Connected (Infinity/Infinity), Loopback0
    Remaining replies:
```

The **A** on the left side of the address shows that this is an active route. **active 00:01:23** reveals the duration of the wait on a reply to this query. It is normal in a large network to see routes go active, but if the amount of time that the routes stay active is more than a minute, something is certainly wrong, and SIAs might occur soon.

Notice the field Remaining replies; any neighbors that are listed under this field have not yet replied to this query. Depending on the timing of when the command is issued, you often see neighbors who have not replied with a lowercase r beside the address, but not under Remaining replies. For example (but not directly related to this case study), consider the output in Example 3-19.

Example 3-19 *Nonresponsive Neighbor Not Under Remaining replies*

```
router#show ip eigrp topology active
IP-EIGRP Topology Table for process 1 Codes:
P - Passive, A - Active, U - Update, Q - Query, R – Reply,
r - Reply status A 10.1.8.0 255.255.255.0, 1 successors, FD is 2733056
    1 replies, active 0:00:11, query-origin: Multiple Origins
         via 10.1.1.2 (Infinity/Infinity), r, Ethernet0
         via 10.1.5.2 (Infinity/Infinity), Serial1, serno 159
         via 10.1.2.2 (Infinity/Infinity), Serial0, serno 151
Remaining replies:
         via 10.1.1.1, r, Ethernet0
```

The first entry in the output identifies a neighbor that you are waiting on but that is not under the Remaining replies section. Keep your eye out for both forms.

Now the discussion gets back to troubleshooting. Because the **show ip eigrp topology active** on Router A revealed that you were waiting on neighbor 10.1.4.1 for 1 minute and 23 seconds, you know which neighbor to look at next: Router B. Log into Router B and execute the **show ip eigrp topology active** command again to see why Router A has not received an answer from Router B. Example 3-20 shows the resulting output.

Example 3-20 show ip eigrp topology active *Output for Router B*

```
router-b#show ip eigrp topology active
IP-EIGRP Topology Table for process 1 Codes:
P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status
A 10.1.100.0/24, 1 successors, FD is Inaccessible
    1 replies, active 00:01:36, query-origin: Successor Origin
         via 10.1.4.3 ((Infinity/Infinity), Ethernet
    Remaining replies:
         via 10.1.1.1, r, Ethernet0
```

Router B is still waiting on a reply from 10.1.1.1, which is Router C. Therefore, the next logical step is to log into Router C and see why it is not answering Router B. After you are on Router C, you issue the command **show ip eigrp topology active** again and get the results in Example 3-21.

Example 3-21 show ip eigrp topology active *Output from Router C*

```
router-c#show ip eigrp topology active
IP-EIGRP Topology Table for process 1 Codes:
P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status A 10.1.100.0/24, 1 successors, FD is Inaccessible, Q
    1 replies, active 00:01:49, query-origin: Successor Origin
         via 10.1.1.2 (Infinity/Infinity), Ethernet1
    Remaining replies:
         via 10.1.16.1, r, Serial0
```

Router C is in the same condition as Router A and Router B. Router C has not answered Router B because it is still waiting on an answer. Now log into 10.1.16.1, which is Router D, to see if this router is having the same problem. As Example 3-22 indicates, the output of **show ip eigrp topology active** on Router D provides different results.

Example 3-22 show ip eigrp topology active *Output from Router D*

```
router-d#show ip eigrp topology active
IP-EIGRP Topology Table for process 1
```

Router D is not waiting on anyone. Router C is waiting on Router D, but Router D is not waiting on replies from any other router. This indicates that the link between Router C and Router D is unreliable, and you need to start exploring why the communications between Routers C and D

are not working correctly. The first thing you need to establish is whether the neighbor relationship is up by issuing the **show ip eigrp neighbor** command, as demonstrated in Example 3-23.

Example 3-23 show ip eigrp neighbor *Output from Router D*

```
router-d#show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO   Q  Seq
                                        (sec)         (ms)       Cnt Num
0   10.1.16.2               Se0           14 00:10:27 1197  5000  1  741
router-d#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.16.2 (Serial0) is down:
  retry limit exceeded
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.16.2 (Serial0) is up: new adjacency
```

The Q count of 1 is not a promising sign. Then you get the error message **retry limit exceeded** on the console because you configured **eigrp log-neighbor-changes** on this router. The **retry limit exceeded** message indicates that acknowledgements are not being received for reliable packets. You need to determine why this is. By going back to Router C and checking the state of the neighbor relationship with Router D, you find the information in Example 3-24.

Example 3-24 show ip eigrp neighbor Output *from Router C*

```
router-c#show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
0   10.1.16.1               Se0           14 00:10:33  479  5000  1  1388
1   10.1.1.2                Et1           11 00:11:46   28   300  0  5318
RouterC#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.16.1 (Serial0) is down:
  retry limit exceeded
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.16.1 (Serial0) is up: new adjacency
```

Router C also complains about the inability to exchange reliable traffic with Router D. You need to use your normal troubleshooting skills to resolve this packet delivery problem. You need to issue pings, look at interfaces, and take the other normal steps to find the true cause of the problem.

Other common problems that can cause a router not to answer queries include the following:

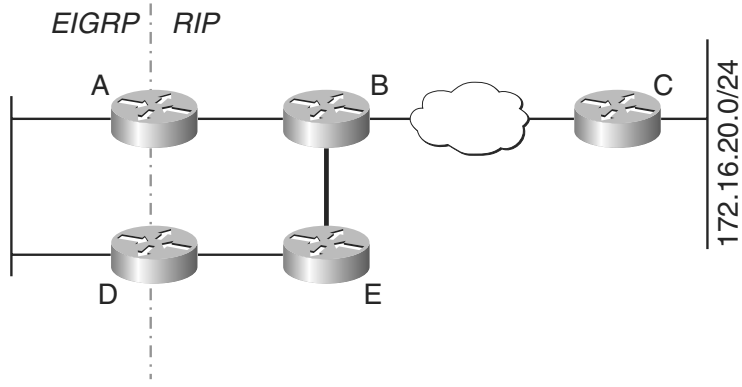
- Low memory.
- Congestion on the link, possibly caused by too many routes for the pipe to handle or by queue drops that are too small.
- MTU problems, possibly caused when small packets are delivered over the link, but not large packets.

Without taking the steps following the chain of waiting routers with the **show ip eigrp topology active** command, you never would have been able to find the failing link and start troubleshooting it.

Case Study: Redistribution

You often want to redistribute routes from EIGRP into other protocols and routes from other protocols into EIGRP. The main problem with redistribution between protocols is that it is easy to create redistribution routing loops. Look at Figure 3-23 to see why.

Figure 3-23 *Redistribution Routing Loop*



The list that follows describes the sequence of transactions depicted in Figure 3-23.

1. Router C advertises the 172.16.20.0/24 network to Router B. Assume that it has a metric of 3 hops when it reaches Router B.
2. Router B advertises this route with a metric of 4 hops to Router A.
3. Router A redistributes the route into EIGRP with some metric and advertises it to Router D.
4. Router D redistributes it back into Routing Information Protocol (RIP) with a default metric of 1 hop, for example, and advertises it to Router E.
5. Router E advertises this route to Router B with a metric of 2 hops, which is better than the route through Router C (which is, in fact, the correct route).

With the EIGRP use of an administrative distance of 170 for external sites, the preceding problem should not happen, should it? The example is simplified to make it clear. In reality, when Router D gets the route from Router A, Router D should prefer the route it had already received from RIP because it has an administrative distance of 120. What is the problem?

The problem occurs if Router E temporarily loses the route to 172.16.20.0/24 and withdraws it from Router D. If this happens, Router D advertises to Router E the route to 172.16.20.0/24

because of the redistribution from EIGRP. This means that the alternative path is working fine. Unfortunately, because the hop count on the redistribution is set to 1 because of the default metric, when Router E receives the real route back from Router B, it does not use it because the one it received from Router D is better. This is not what you want to happen.

This is a classic redistribution routing loop. How do you solve it? The easiest thing to do is to filter the destinations that are redistributed from RIP into EIGRP and from EIGRP into RIP.

Using Distribute Lists to Prevent Redistribution Routing Loops

The first, and simplest, way to handle this problem is to set up a distribute list specifically blocking the routes that you do not want to redistribute. For example, on Router D, you could build the distribute list in Example 3-25.

Example 3-25 *Using a Distribution List to Block Redistribution Routing Loops*

```
access-list 10 deny 172.16.20.0 0.0.0.255
access-list 10 permit any
!
router rip
 redistribute eigrp 100
 distribute-list 10 out serial 0
```

Assuming that Serial 0 is the link between Router D and Router E, this resolves the problem. RIP does not advertise the 172.16.20.0/24 route from Router D to Router E. If you have more than one connection back into the RIP side of the network, it can be difficult to manage the distribution lists that must be maintained.

Using Route Maps to Prevent Redistribution Routing Loops

An alternative to using a distribute list is to configure a route map on Router D, as demonstrated in Example 3-26.

Example 3-26 *Using a Route Map to Stop a Redistribution Routing Loop*

```
access-list 10 deny 172.16.20.0 0.0.0.255
access-list 10 permit any
!
route-map kill-loops permit 10
 match ip address 10
!
router rip
 redistribute eigrp 100 route-map kill-loops
```

This configuration allows only those networks that are permitted by access list 10 to be redistributed into RIP. This has the same effect as the distribute list used in the preceding solution, but it applies the filter in the redistribution rather than in the advertisement to Router D.

Another alternative is to match all external EIGRP routes in the route map, as demonstrated in Example 3-27.

Example 3-27 *Using a Route Map to Filter External Routes*

```
route-map kill-loops deny 10
 match route-type external
route-map kill-loops permit 20
```

However, this approach also destroys any external EIGRP routes that are learned from a protocol other than RIP. In other words, it prevents external destinations elsewhere in the EIGRP network from being reached by the hosts that are attached on the RIP side of the network.

Using Prefix Lists to Prevent Redistribution Routing Loops

In addition to using distribute lists and route maps to troubleshoot redistribution routing loops, you can use prefix lists. For example, you can configure Router D with the prefix lists in Example 3-28.

Example 3-28 *Using Prefix Lists to Prevent Redistribution Routing Loops*

```
ip prefix-list loop-list 10 deny 172.16.20.0/24
ip prefix-list loop-list 20 permit 0.0.0.0/0 le 32
!
route-map kill-loops permit 10
 match prefix-list loop-list
!
router rip
 redistribute eigrp 100 route-map kill-loops
```

Prefix lists allow you to match based on prefix length (the subnet mask) and the actual prefix (destination network). Many possibilities for filtering exist when this application is considered, but they are not covered here.

Setting the Administrative Distance to Troubleshoot Redistribution Routing Loops

Whereas all the previous mechanisms rely on the configuration (and maintenance) of an access list to prevent a redistribution routing loop, setting the administrative distance of all external routes learned by Router D from Router A does not rely on access lists. You can configure this technique using the **distance** command. On Router D, you would configure the following:

```
router eigrp 100
 distance 255 172.16.21.1 0.0.0.0
```

If the Router A address is 172.16.21.1, Router D assigns an administrative distance of 255 to any routes that it receives from Router A. A route that has an administrative distance of 255 is

never inserted into the routing table; therefore, it is not redistributed into RIP from EIGRP. (Redistribution always occurs from the routing table rather than any private databases that the various routing protocols use.)

The only problem with this approach is that Router D refuses all routes learned from Router A, including legitimate ones. You can remedy this by adding the access list back into the equation, as demonstrated in Example 3-29.

Example 3-29 *Using the* **distance** *Command with an Access List to Block Redistribution Loops*

```
access-list 10 permit 172.16.20.0 0.0.0.255
!
router eigrp 100
 distance 255 172.16.21.1 0.0.0.0 10
```

By providing an access list that identifies a particular range of addresses and blocks all others from this neighbor, you can accomplish slightly more selective filtering.

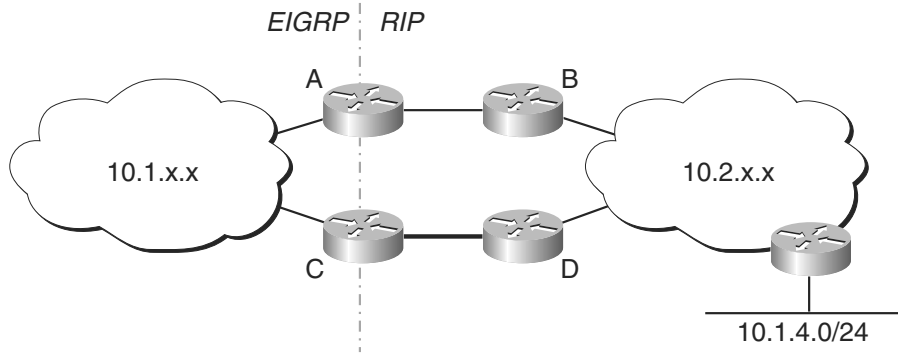
One additional limitation of this approach is that the **distance** command is applied to both internal and external routes. Therefore, if you are trying to limit the filtering to stop the receipt of external routes, you cannot use the **distance** command to accomplish it.

Using External Flags to Prevent Redistribution Routing Loops

All of the previously mentioned troubleshooting methods work, but they require either configuring a list of networks or removing the alternative route through the other protocol as a possible backdoor route in the case of failure. Tagging EIGRP externals to block routing loops resolves these two problems and is fairly straightforward to configure.

Connecting Router A to Router B and Router C to Router D has recently merged the two networks in Figure 3-24. At some point in the future, the network administrators intend to replace RIP with EIGRP; for now, they are redistributing between RIP and EIGRP on Routers A and C.

Figure 3-24 *Complex Redistribution Routing Loop*



This setup produces a classic redistribution routing loop:

1. Router B learns about some destination, such as 10.1.4.0/24, through RIP. Then it advertises this route to Router A.
2. Router A redistributes this route into EIGRP and advertises it to Router C.
3. Router C redistributes this route back into RIP and advertises it to Router D.
4. Router D advertises the route back to Router B (possibly with a better metric than Router B learned in the original advertisement).

Almost all of the EIGRP network in this figure uses addresses from the 10.1.0.0/16 address space, and almost all of the RIP network uses addresses from the 10.2.0.0/16 address space. However, some exceptions exist, such as the 10.1.4.0/24 network.

If it were not for the exceptions, this redistribution routing loop would be easy to resolve. You would simply prevent Router A and Router C from advertising routes in the 10.2.0.0/16 address range to Router B and Router D and prevent Router B and Router D from advertising routes in the 10.1.0.0/16 address range to Router A and Router C. Distribution lists combined with summarization would make this configuration easy.

Because of the exceptions, though, preventing this redistribution routing loop is more difficult. You could build distribution lists around the subnets present on each side and apply them on Router A, Router B, Router C, and Router D, but this adds some serious administrative overhead if many exceptions exist. Specific distribution lists would also require modification for each new exception added.

It is easier to use an automatic method to flag the routes learned through RIP on Router A and Router C. Then you can prevent any route that is flagged from being redistributed back into RIP. For example, Router A still learns about the 10.1.100.0/24 network through EIGRP and advertises this destination to Router B through RIP.

Router B still advertises 10.1.4.0/24 to Router A, which redistributes it into EIGRP and advertises it to Router C. However, Router A flags this route as coming from the RIP domain so that Router C does not advertise it back into RIP. Using some sort of tag like this means that adding a new network in the RIP AS should not require reconfiguration on the routers that are doing the redistribution. This type of routing loop is a good use for EIGRP administrator tags.

Administrator tags are applied and matched using route maps. On Router A and Router C, you create the route maps and then apply them to the redistribution between EIGRP and RIP by issuing the commands in Example 3-30.

Example 3-30 *Setting Administrative Tags on Redistribution*

```
route-map setflag permit 10
  set tag 1
route-map denyflag deny 10
  match tag 1
route-map denyflag permit 20
```

The **setflag** route map sets the administrator tag on any route to 1, whereas the **denyflag** route map denies routes with a flag of 1 and permits all others. On Router A and Router C, you apply these route maps to the redistribution between EIGRP and RIP by issuing the commands in Example 3-31.

Example 3-31 *Applying Tag Filtering on Redistribution*

```
router eigrp 4000
  redistribute rip route-map setflag
router rip
  redistribute eigrp 4000 route-map denyflag
```

As routes are redistributed from RIP to EIGRP, the **setflag** route map is applied, setting the EIGRP administrative tag to 1. As the routes are redistributed from EIGRP to RIP, the administrative tag is checked; if it is 1, the route is denied so that it is not redistributed.

Case Study: Retransmissions and SIA

Two timers that can interact in EIGRP to cause an SIA route in EIGRP are the SIA timer and the hold timer between two peers. How do these two relate? This section examines the two timers independently and then looks at how they interact.

The Hold Timer

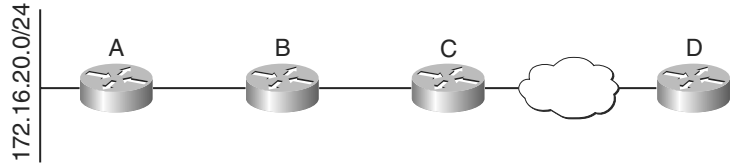
The obvious use for the hold timer is to determine how long to hold up a neighbor relationship without hearing EIGRP hellos. Each time a router receives a hello packet from a neighbor, it resets the hold timer to the hold time contained in the hello packet and decrements it once for each second that passes.

After the hold timer reaches zero, the neighbor is assumed dead. All paths through that neighbor are marked unusable (DUAL is run over these destinations to determine if the route needs to go active), and the neighbor is marked down.

However, the hold timer is also used by the EIGRP reliable transport mechanism as an outer bound on how long to wait for a neighbor to acknowledge the receipt of a packet. As mentioned in Appendix A, EIGRP attempts to retransmit 16 times or until retransmission has been occurring for as long as the hold timer, whichever is longer.

In the network depicted in Figure 3-25, assume that the Router D hold timer is 240 seconds. (Ignore the Hello timer because these are separate timers.)

Figure 3-25 *Interactions Between Hold Timers and SIA Timers*



If Router C sends a packet to Router D, and Router D does not acknowledge the packet, Router C continues retransmitting until it has retransmitted 16 times. Then Router C checks to see if it has been retransmitting for 240 seconds. If it has not, Router C continues sending the packet until it has been retransmitting for 240 seconds. After Router C has attempted retransmission for 240 seconds, it assumes that Router D is never going to answer and clear its neighbor relationship.

SIA Timer

The other timer that you need to concern yourself with is the SIA timer because it determines how long a query can be outstanding before the route is declared SIA and the neighbor relationship with the router that has not answered is torn down and restarted.

Prior to the SIA enhancements explained in the section "Enhanced EIGRP Active Process," the active timer is, by default, 3 minutes (although there has been talk of changing it). This means that a router waits 3 minutes after it has declared a route active until it decides that any neighbor that has not replied for this active route has a problem and restarts the neighbor.

Going back to Figure 3-25, this means that if Router A loses its connection to 172.16.20.0/24, it sends a query to Router B. If it does not receive a reply to that query within 3 minutes, it restarts its neighbor relationship with Router B. Note that two completely different things are being discussed here:

- How long to wait before getting an acknowledgement for a packet
- How long to wait for a reply to a query

Interaction Between the Hold Timer and the SIA Timer

You can work through an example of how these two timers interact. Assume that Router A in Figure 3-25 loses its connection to 172.16.20.0/24. Because it has no other paths to this destination, it marks the route as active and sends Router B a query.

Router B acknowledges the query and sends a query to Router C; Router C, in turn, acknowledges the query and sends a query to Router D. Router D, for some reason, never acknowledges the query. Router C begins retransmitting the query to Router D. It attempts to do so until it has retransmitted for the length of the hold timer.

For the entire time that Router C is trying to get an acknowledgement from Router D, the Router A SIA timer is running. Because the SIA timer is 3 minutes, and the Router D hold timer is 4 minutes, it is safe to assume that the Router A SIA timer will go off before Router C gives up retransmitting the query to Router D and clears the neighbor relationship.

Therefore, Router A registers an SIA and clears its neighbor relationship with Router B. It is important to remember when designing your network that the hold timer for any given link should never be more than or equal to the SIA timer for the entire network.

In this case, two solutions are possible:

- Reduce the Router D hold time to something less than the SIA timer (90 seconds, for example) by using the interface level command **ip eigrp hold-time**.
- Increase the SIA timer to something greater than the hold timer (five minutes, for example) by using the command **timers active** under the router EIGRP configuration.

Knowing which option to choose without more information is difficult. If the link between Router C and Router D is congested often enough that an acknowledgement takes 4 minutes to get through, it is probably going to be necessary to increase the SIA timer.

On the other hand, if it seems unreasonable to wait 4 minutes for a simple acknowledgement across a single link, it is better to decrease the hold timer on Router D. Remember to decrease the Hello timer, too, or you will have problems maintaining neighbor relationships. If a router is still sending hellos every 60 seconds, but the hold time is reduced to 90 seconds, the neighbor can be torn down if only two hellos are lost instead of three. The best practice is to always set the hold timer as a multiple of three hellos. If you reduce the hold time, you need to reduce the hello interval accordingly.

The two tradeoffs are as follows:

- The hold timer should be a reasonable amount of time, given the nature of the link and the likelihood of an EIGRP packet being delayed for a given period of time.
- The SIA timer bounds the time that the network is allowed to remain unconverged.

You need to balance these two tradeoffs for your network. There are no magic numbers (although there are defaults) .

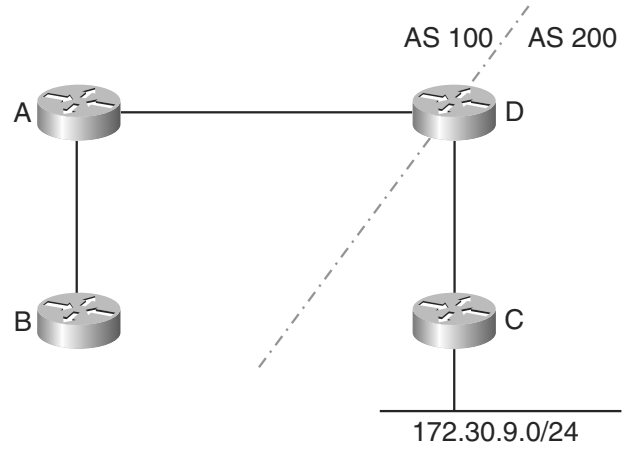
Case Study: Multiple EIGRP Autonomous Systems

One design that is used commonly in EIGRP to limit query range and improve stability is multiple autonomous systems, but is this really effective? Look at Figure 3-26 for some answers.

Begin by assuming that Router D is redistributing all the routes from AS 100 into AS 200 and all the routes from AS 200 into AS 100. If Router C loses its direct connection to 172.30.9.0/24, it notes that it has no feasible successor, places the destination in active state, and queries each of its neighbors.

When Router D receives this query, it looks through its topology table and, seeing no other routes to this destination within this AS, immediately sends a reply to Router C that this route is no longer reachable. Router C acknowledges the reply and removes the route from its topology table (so far, so good).

Figure 3-26 *Multiple EIGRP Autonomous Systems*

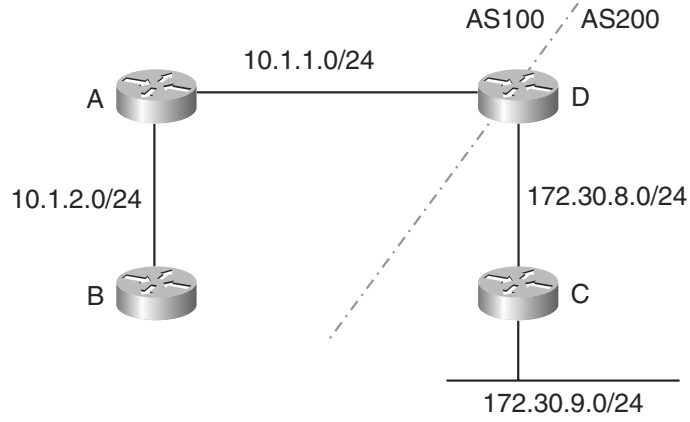


Return to Router D once more. Router D was redistributing this route into AS 100. When Router D loses the route, it goes active on the AS 100 topology table entry and queries its neighbors (in this case, Router A). Router A, in turn, queries Router B; the entire query process runs in AS 100 for this route.

In short, AS boundaries do not really stop queries. The query itself might stop, but a new query is generated at the AS border and propagated through the neighboring AS.

Therefore, AS boundaries do not help with query range issues, but can they really harm anything? Look at Figure 3-27 for a moment.

Figure 3-27 *Autosummarization Across an AS Boundary*



In the network that is illustrated, not only does Router D redistribute between AS 100 and AS 200, but an autosummary for the 10.0.0.0/8 network on Router D is also being advertised toward Router C, and an autosummary for 172.30.0.0/16 is being advertised toward Router A. Because of these autosummaries, the query range is bound at Router A for 172.30.9.0/24. In other words, Router B never receives a query about this network because Router A should not have information about it in its topology database.

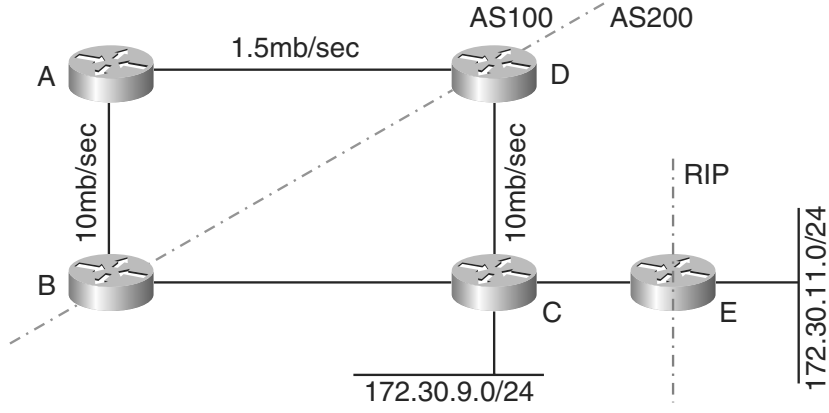
The problem is that EIGRP does not autosummarize externals unless an internal component exists in the topology table. Router D does not build summaries for the 10.0.0.0/8 and 172.30.0.0/16 networks automatically; it advertises all the components.

The really confusing part comes in if you decide to add something in the 10.0.0.0 network on Router B. Suppose that you add an Ethernet link to Router B and address it as 10.1.5.0/24. Router B summarizes this to be 10.0.0.0/8 and advertises it toward Router A (remember that this is an internal component), and Router A advertises it to Router D.

When Router D sees an internal component in the 10.0.0.0 network within AS 100, it begins summarizing the external sites toward Router A, advertising only the 10.0.0.0/8 route. This means that Router A has two routes to 10.0.0.0/8—a confusing situation at best.

What if you do not try to put a major net boundary on an AS boundary and rely on manual summarization? Multiple autonomous systems have no other problems, do they? As a matter of fact, they do. Look at Figure 3-28 for a third problem.

Figure 3-28 *Discontiguous Autonomous Systems*



Router B and Router D are redistributing between AS 100 and AS 200. Router E is redistributing from RIP into EIGRP AS 200. Router B receives two routes for 172.30.9.0/24:

- An internal route through Router C
- An external route through Router A

Which route does Router B choose? The route through Router A probably has a better metric, but Router B chooses the path through Router C because the administrative distance of internal routes is better than the administrative distance of external routes.

If all of these routers were in a single AS, Router B would choose the shortest path to 172.30.9.0/ 24; using multiple autonomous systems causes the routers to choose suboptimal routes.

Consider the route to 172.30.11.0/24 next. Which route does Router B choose for this destination? Router B should choose the route through Router A because both routes are externals. The administrative distances are the same for both routes.

However, the behavior in this instance is undefined. In other words, Router B could choose either route, regardless of which one has the better metric.

All in all, it is best to stick to one AS unless you have carefully considered all the issues involved in multiple AS designs. With good design, you can limit the query scope within the network through summarization and distribution lists.

If an EIGRP network grows large enough to need splitting, it is better to use a protocol other than EIGRP to do so (preferably BGP) .

Review Questions

1. What are the two basic tools you can use to summarize routes (or hide destination details) in EIGRP?
2. How can you tell that a route is a summary when you look at the routing table?
3. What is the default administrative distance for a summary route? What is the problem with this?
4. What bounds a query?
5. How far beyond one of the possible query bounds does a query travel?
6. What is the primary advantage to summarizing between core routers rather than between the distribution layer and core?
7. How is it possible to “black hole” packets when summarizing destinations behind dual-homed remotes into the core?
8. Why should summarization be configured outbound from the distribution layer routers toward access layer routers at remote sites?
9. What is the most common problem with dual-homed remotes? What options are available to resolve it?
10. What methods can you use to break a redistribution routing loop?

11 Under what conditions is the administrative distance ignored between EIGRP and IGRP?

12 What options do you have for generating a default route in EIGRP?

13 How can you prevent multiple parallel links within a network from being used as transit paths?

14 What does EIGRP use to pace its packets on a link?

This page intentionally left blank

Numerics

802.1x, 343–344

A

A bits, Frame Relay, 286
ABRs (area border routers), BGP, 234
abstraction through layering, 35–36
access layer, routing, 83
    best next hop, 86–87
    dual-homed remotes, 85–86
    single-homed sites, 84
access lists, 204
access-list command, 204
active process, EIGRP, 110–112
Active state, BGP neighbors, 227
add/drop multiplexer (ADM) error, 284
adjacencies
    building, 403–404
    detecing failures, 280–283
    IS-IS, 217, 318
        building, 415–416
    link flaps, 262
    on multiaccess networks, 405
    OSPF, 318
administrative distance
    BGP, 235
    preventing redistribution routing loops, 131–132
advertisements, BGP, 226
    conditional advertisements, 255
aggregation
    IS-IS, 190
    OSPF, 160–162
    versus summarization, 192
aggregation layer, summarization, 98
aging timer (link state), 214
AH (Authentication Header), 333
algorithms, BGP, 226
analyzing redundancy
    MTBF, 23–24
    MTTR, 24–25
application programming interfaces, 48
areas, 198, 201–202, 406
    boundaries, IS-IS, 212
AS_PATH length, BGP load sharing, 249
as-path access-list command, 343
ASPolicyCerts, BGP, 347
AS (autonomous systems)
    autosummarization, 137
    BGP, 226, 233
    discontiguous, 138
    EIGRP, 136–139
assigning IP addresses, 50–53
ATM (Asynchronous Transfer Mode), NHRP case study, 373–375
attached bits (IS-IS), 198
attacks
    BGP, 321
    disrupting peering, 318
        flooding, 318
        protocol-level attacks, 318
        transport-level attacks, 318
    disrupting routing domain stability, 324–325
    DoS, 334–335
        preventing via edge filters, 335
        preventing via GTSM, 335, 337
    EIGRP, 320
    falsifying routing information, 323–324
    IS-IS, 318, 320
    OSPF, 318, 320
    protocol-layer attacks, 322
    TCP resets, 321
AuthCerts, soBGP, 346
authentication, 311
    MD5, 331–333
    RADIUS servers, 328
    soBGP, 345–346
    transiting trust, 311–313
    versus authorization, 313
Authentication Header (AH), 333
authorization, 311
    versus authentication, 313
    RADIUS servers, 328
    soBGP, 346–347
    TACACs servers, 328
    transiting trust, 311–316
autosummarization, 137, 393

B

bandwidth command, 97
best next hop, 86–87
BFD (Bidirectional Forwarding Detection), 283
BGP (Border Gateway Protocol), 225–226, 233–236, 422
  attacks, 321
  conditional advertisements, 255
  confederations, 240
  core layer, 232–233
  deploying with GR, 304–305
  dividing, 237–238
  dual-homed connections, 247–253
  eBGP, 426
  extranets, 339
  filtering
    with distribution lists, 431
    with prefix lists, 431
    with route maps, 430
  GR, 277–279
  iBGP, 427
  IPSec, 334
  MD5 authentication, 331
  metrics, 423–425
  neighbors, troubleshooting, 227–228, 231
  next hop attribute, 429
  normal restarts, 277
  peers, 227
  route dampening, 255–257
  route reflectors, 242–245
  route servers, 245–247
  scaling, 236–237
  summarization, 432
  synchronization, 431–432
  transport-level attacks, 321
  updates, 239
bgp dampen command, 257
bgp dampening command, 340
bgp graceful-restart command, 279
bgp graceful-restart stalepath-time seconds command, 279
bgp graceful-restart-time seconds command, 279
BGP/MPLS VPNs, 358, 361
  implementing in EIGRP, 361–369
  implementing in OSPF, 369–370
Bidirectional Forwarding Detection (BFD), 283
binding EIGRP queries, 393–395
bit error rate (BER) errors, 284
black holes, 58, 81, 93
Border Gateway Protocol. *See* BGP
border routers, BGP, 234
broadcast interfaces, IS-IS, 210
broadcast links, IS-IS, 211
buffer flooding, 318
building IS-IS adjacencies, 415–416

C

calculating MTBF, 24
capability lls command, 273
case studiesm, OSPF external routes, 182
CDP (Cisco Discovery Protocol), 219
CE routers, 356
checklist for IP network design, 438
  network operations section, 435–437
  redistribution section, 438
  security section, 438
  topological layout section, 437
choke points, creating, 48
Cisco Discovery Protocol (CDP), 219
CLNS (connectionless network services)
  addressing, 412–413
  IS-IS, 189, 205
clusters, BGP, 242
commands
  bgp dampen, 257
  set metric-type internal, 233
  show ip bgp neighbor, 227–228
common services, EIGRP, 91–93
community strings, 425
comparing EIGRP and OSPF, 457
  convergence time, 459–462
  ease of troubleshooting, 458–459
  suitability of network designs, 462–468
complexity of network management, 25
  layering, 27
    functionality, separating, 32
    hiding information, 28–31
confederations, 240
Connect state (BGP neighbors), 227
connectionless network services. *See* CLNS
connection-oriented network services (CONS), 189
connections, BGP dual-homed connections, 247–253
CONS (connection-oriented network services), 189
control planes, 265
  GR, 266
convergence. *See also* fast convergence

decreasing speed, 296
EIGRP feasible successors, 296–299
link-state incremental SPF, 300–302
link-state partial SPF, 299–300
false injection attacks, 325
SONET, 284–285
core layer
BGP, 232–233
migration, 233–236
summarization, 77, 98
summarizing to core, 80
summarizing to distribution layer, 77–78
corruption, packet corruption, 214
creating choke points, 48
creating layers, 47
cryptography, key distribution, 345

D

dampening,
BGP route dampening algorithm, 226
event dampening, 293–295
dampening command, 295
DBDs (database descriptors), 404
debouncing, 263
default routes, BGP load sharing, 248
default-information originate command, 91
delay, 9
denial-of-service attacks. *See* DoS attacks
dense wavelength division multiplexing networks, SONET, 285
deploying OSPF
on three-layer hierarchy, 146–150
on two-layer hierarchy, 152
dial links, OSPF, 180
dial-in clients, EIGRP, 94
bandwidth issues, 97
host routes, 94
DIS (designated intermediate system), 210–211
discard routes, 116
disrupting
peering attacks, 318
routing domain stability attacks, 324–325
distance command, 131
distribute lists, 116
BGP filtering, 431
preventing redistribution routing loops, 130
distribution layer, summarizing, 80
toward core, 80–83
toward remote sites, 83
dividing BGP, 237–238
domains, routing, 326
illegitimate devices, thwarting, 330
IPSec, 333–334
MD5 authentication, 331–333
router compromise, avoiding, 326
filtering access, 328–330
using passwords, 326–328
DoS (denial-of-service) attacks, 334–335
preventing via edge filters, 335
preventing via GTSM, 335–337
down detection. *See* failure detection
DUAL, 383–384
dual-homed remotes, 85–86
best next hop, 86–87
DWDM (dense wavelength division multiplexing) networks, SONET, 285
dynamic multipoint IPSec VPNs, 376–378

E

eBGP, 426
peers, 226
eBGP-multihop, BGP neighbors, 229
edge filters, 335
edges, SPF, 299
EGPs (External Gateway Protocols), 225, 339
BGP. *See* BGP
EIGRP (Enhanced IGRP), 264
active process, 111–112
admin distance, 131–132
attacks, 320
bandwidth, 97
BGP/MPLS VPNs, implementing, 361–365
common services, 91–93
comparing with OSPF, 457–468
deploying on three-layer network, 75
access layer, 83
common services area, 91
core layer, 77
dial-in clients, 94
distribution layer, 80
stub routers, 87
deploying on two-layer network, 97
aggregation layer, 98
core layer, 98

dial-in clients, 94
    bandwidth issues, 97
    host routes, 94
discard routes, 116
distribute lists, 116, 130
DUAL, 383–384
dual-homed remotes, 85
external clients, 91
external flags, 132–134
feasible successors, 296–299
GR, 267–269
IP summary addresses, 114–115
load balancing, 396
loop detection, 388–390
MD5 authentication, 331
metrics, 387–388
multiple autonomous systems, 136–139
neighbor relationships, 120, 385–386
    mismatching primary addresses, 120–122
    multicast delivery problems, 122–123
neighbors, querying, 390–391
network design summary, 98–99
new features
    active process enhancements, 110–114
    enhanced route map support, 104–106
    route map enhancements, 106–110
    third-party next hop, 99–104
normal restart, 267
polling, 280
prefix lists, 131
queries
    bounding, 393–395
    propagation, controlling, 116–117
redistribution, 129–130
    setting admin distance, 131–132
    using distribute lists, 130
    using external flags, 132–134
    using prefix lists, 131
    using route maps, 130–131
route maps, 130–131
routing, access layer, 83–87
SIA routes, 124–128, 391–393
single-homed sites, 84
SRRT, 121
stub routing, 87–90, 394–395
summarization
    controlling query propagation, 116–117
    core layer, 77–80
    distribute lists, 116
    distribution layer, 80–83
    IP summary addresses, 114–115
    multiple topology table entries, 118–119
    stub routers, 87–90
timers, 134
    hold timers, 134–135
    hold/SIA timer interaction, 135–136
    SIA timer, 135
topology tables, 118–119
    clearing, 390–391
transport-level attacks, 320
troubleshooting
    neighbor relationships, 120–123
    SIA routes, 124–128
eigrp log-neighbor-changes command, 120
emergency network management, 18–20
enable password, 327
enable secret password, 327
Encapsulating Security Payload (ESP), 333
Enhanced Interior Gateway Routing Protocol. *See* EIGRP
EntityCerts, soBGP, 345
error checking, 215
errors
    ADM, 284
    BER, 284
    path errors, 284
    SONET, 284
ESP (Encapsulating Security Payload), 333
Established state (BGP neighbors), 228
Ethernet, failure detection, 288–289
event dampening, IP, 293–295
    default values, 295
    interface specific, 295
event reporting, limiting, 264–265
    GR, 266
        BGP, 277–279
        EIGRP, 267–269
        IS-IS, 274–276
        OSPF, 270–274
    NSF, 265
event-driven notification, detecting link/adjaceny failures, 283
exponential backoff, 291
    deploying, 305
        setting SPF timers, 306–307
    IS-IS, 293
    link-state generation timer, 291
    OSPF
        LSAs, 292
        SPF, 292
    SPF timer, 291
    versus IP event dampening, 293–295

extended access lists, IS-IS, 204
Exterior Gateway Protocols (EGPs), 225, 339
external connections, EIGRP, 91
external flags, preventing redistribution routing loops, 132–134
external route
external routes
  in OSPF, case study, 182
  injecting, 407
extranets, 337–338
  BGP, 339
  dampening prefixes, 340
  filtering routes, 339–340
  limiting route count, 341
  using EGPs, 339

F

failure detection, 280
  Ethernet, 288–289
  Frame Relay, 285–288
  measured responses, 290
    exponential backoff, 291
    IP event dampening, 293–295
    IS-IS exponential backoff, 293
    link-state exponential backoff, 291
    OSPF exponential backoff, LSAs, 292
    OSPF exponential backoff, SPF, 292
  SONET, 284–285
  using BFD, 283
  using Ethernet, 288–289
  using event-driven notification, 283
  using Frame Relay, 285–288
  using polling, 280–283
  using SONET, 284–285
falsifying routing information attacks, 323–324
fast convergence, 261–262
  deploying
    exponential backoff, 305–307
    GR versus fast failure detection, 302–304
    GR with BGP and an IGP, 304–305
  detecting failures, 280–289
  limiting reporting, 264, 271–273, 279
  network meltdowns, 263
    avoiding with routing protocol design, 263–264
    troubleshooting, 263
  slowing down, 290
fast hellos, 283
feasible successors, 460
  EIGRP, 296–299
feedback loops, 290
filtering
  routes in OSPF, 164
  with distribution lists, 431
  with prefix lists, 431
  with route maps, 430
flags, external flags, 132–134
flapping, 262
flaps, BGP, 340
flooding
  attacks, 318
  IS-IS
    domains, 197
    full mesh networks, 206
  link-state packets, 213–214
  LSAs, 403
forwarding planes, 265
Frame Relay
  A bits, 286
  detecting failures, 285–288
  multipoint configuration, 123
  point-to-multipoint configuration, 287
  point-to-point configuration, 287
  polling, 280
full mesh networks
  IS-IS, 205, 208–209
    flooding, 206
    mitigating single router failure, 208
  OSPF, 167, 170–171
  selecting suitable routing protocols, 465–466
functionality, separating, 32

G

Generalized TTL Security Mechanism (GTSM), 335–337
generation timer, link-state, 291
goals for network design, 5
  manageability, 13–14
    day-to-day maintenance, 14–16
    emergency management, 18–20
  reliability, 6
    and resiliency, 10
    network failures, defining, 12–13
    network recovery time, 13
    of packet delivery, 6– 9
  scalability, 20

GR (graceful restart), 266
BGP, 277–279
deploying with BGP and an IGP, 304–305
EIGRP, 267–269
high availability, 303
IS-IS, 274–275
configuring GR, 276
signaled GR, 275–276
lab performance, 302
mixing with non-GR routers, 304
OSPF, 270–274
using link local signaling, 271–272
using opaque LSAs, 272–273
graceful restart. *See* GR
Grace-LSAs, 273
GRE tunnels, 372
multipoint GRE tunnels, 376–378
GTSM (Generalized TTL Security Mechanism), 335–337

H

half-life, BGP route dampening , 256
HDLC (High-Level Data Link Control), polling, 280
hello interval, 303, 385
hello messages
fast hellos, 282
IS-IS, 217
padding, 217
hello packets, polling, 280
hiding
information, 28–31
layers within layers, 46
hierarchical network design
abstraction through layering, 35–36
choke points, creating, 48
hiding layers within layers, 46
horizontal layers, 36
layer functions, 38
aggregation of routing information, 39
controlling traffic, 42
defining routing policies, 41
forwarding traffic, 38
user attachment, 42
layers, creating, 47
selecting best design, 45
three-layer hierarchies, 44–45
two-layer hierarchies, 43–44
high availability, 303
hold timers, 134–135, 303
horizontal network layers, 36
host names, IS-IS LSPs, 200
HTML server passwords, 327
hub-and-spoke topologies, 171–177
IS-IS, 209
broadcast interfaces, 210
point-to-point links, 209
selecting suitable routing protocols, 463–464

I

iBGP, 427
peers, 226
synchronization, 431–432
Idle state, BGP neighbors, 227
ignore-lsp-errors command, 215
IGPs (Interior Gateway Protocols), 225
deploying with GR, 304–305
regional, 238
IGP-to-IGP redistribution, 62–64
incremental SPF, IS-IS, 302
incremental time, exponential backoff, 291
incremental updates, BGP, 239
incremental-spf command, 302
initial time, exponential backoff, 291
interfaces
debouncing, 263
null0, 234
intermediate systems
DIS election process, 211
parallel links, 212
selector bits, 216
serving as DIS, 215
Intermediate System-to-Intermediate System protocol. *See* IS-IS
Interior Gateway Protocols. *See* IGPs
Internet, 226
Internet connections, 341
protecting against transit, 342–343
route dampening, 343
route filtering, 341
ip address command, 208
IP addresses
assigning, 50–53
IS-IS, 209
summarizing, 54, 57
metrics, 61–62
suboptimal routing, 59–60

ip default-network command, 91
ip eigrp hold-time command, 136
IP event dampening, 293–295
    default values, 295
    interface specific, 295
ip hello-interval eigrp command, 282
ip hold-time eigrp command, 282
ip ospf dead-interval minimal hello-multiplier command, 281
ip ospf resynch-timeout command, 273
ip router isis command, 208
IP routes, IS-IS, 205
IP summary addresses, 114–115
IPSec, 333–334, 370
    AH, 333
    dynamic multipoint IPSec VPNs, 376–378
    ESP, 333
    transport mode, 333
    tunnel mode, 333
IS-IS (Intermediate System-to-Intermediate System), 189–190, 412
    adjacencies, building, 318, 415–416
    aggregation, 190
        versus summarization, 192–193
    aging timer, 214
    attacks, 318, 320
    blocked interfaces, 207
    CDP, 219
    Cisco router default adjacencies, 220
    CLNS, 189, 205
    configuring summarization, 204
    CONS, 189
    data transport, 318
    deploying on three layers, 190
        core as L2 domain, 193–194
        merging core and distribution in L2, 194–195
        mixing/overlapping L1/L2 border, 195–197
        single routing domain, 190, 193
    deploying on two layers, 197–198
    DIS election process, 210–211
    error checking, 215
    exponential backoff, 293
    flooding, 206–208
        domains, 197
    Frame Relay, 209
    full mesh networks, 205, 208–209
    GR, 274–275
        configuring GR, 276
        signaled GR, 275–276
    hub-and-spoke networks, 209
        broadcast interfaces, 210
        point-to-point links, 209
    incremental SPF, 302
    IP address space, 209
    IP integration, 417
    IP routes, 205
    links parallel to area boundaries, 212
    link-state flooding, 213–214
    LSP corruption, 214–215
    LSP flooding, 416
    mesh groups, 206
    metrics, 213, 415
    MPLS traffic engineering, 213
    multiple net statements, configuring, 418
    neighbor adjacencies
        correcting, 220
        different subnets, 218
        misconfigured NSAPs, 217
    neighbor loss, 417
    normal restart, 274
    NSAPs, 215
    path costs, 213
    point-to-point broadcast links, 211
    PRC, 293
    prefix-driven routing installation, 216
    pseudonode LSPs, 215–216
    redistribution, 204–205
    refresh interval, 214
    route leaking, 203
    route maps, 205
    route tags, 205
    router isis configuration mode, 201
    routing, 413–414
    routing areas, 198–202
        aggregating routes, 204
        leaking routes into L1 routing domain, 203–204
    routing domains, 190
        L1 versus L2, 198
        splitting single into multiple, 190
    routing loops, 205
    routing tables, 216–217
    selector bits, 216
    SPF
        calculation time, 306
        trees, 213

standards track RFCs, 411
static routes, 205
subinterfaces, 209
summarization, 190
versus aggregation, 192–193
suppressing hello padding, 217
tagging routes, 205
timers, 264
transport-level attacks, 318–320
wide metrics, 213
isis hello-interval minimal command, 282
isis hello-multiplier command, 282
isis link-type level-1-2 command, 212
isis mesh-group blocked command, 208
isis mesh-group command, 208
isis network point-to-point command, 211
isis priority command, 210
ispf command, 302
is-type level-1 command, 201

J-K-L

jitter, 9

key distribution, 345

layer functions, 27, 38
aggregation of routing information, 39
controlling traffic, 42
defining routing policies, 41
forwarding traffic, 38
hiding information, 28–31
separating, 32
user attachment, 42
layered network designs, selecting suitable routing protocols, 466–468
leaking routes (IS-IS), 204
leaves (SPF), 299
link flaps, 262
link local signaling, 271–272
links, detecting failures, 280–283
link-state advertisements. *See* LSAs
link-state exponential backoff, 291–292
link-state flooding, 213–214
link-state generation timer, 291
link-state incremental SPF, 300–302
link-state packets. *See* LSPs
link-state partial SPF, 299–300
link-state protocols
IS-IS, 412
adjacencies, building, 415–416
IP integration, 417
LSP flooding, 416
metrics, 415
neighbor loss, 417
routing, 413–414
OSPF, comparing with EIGRP, 457–468
load balancing, 396
load sharing, BGP, 249–252
loops, 310
detecting, 388–390
discard route, 116
redistribution routing loops, 129
TTL, 336
LSAs (link-state advertisements), 399–402
adjacencies
building, 403–404
on multiaccess networks, 405
age parameter, 402
generation time, 461
Grace-LSAs, 273
opaque LSAs, 272–273
OSPF exponential backoff, 292
reliable flooding, 403
throttling, 292
lsp-gen-interval command, 293
lsp-refresh-interval command, 214
LSPs (link-state packets)
attached bits, 198
error checking, 215
flooding, 213, 416
host names, 200
reflood storms, 215

M

manageability of networks, 13–14
day-to-day maintenance, 14–16
emergency management, 18–20
MARP (Multiaccess Reachability Protocol), 289
maximum time, exponential backoff, 291
max-lsp-lifetime command, 214
MD5 (Message Digest 5) authentication, 331–333
MED (Multi-Exit Discriminator), 251, 424
meltdowns, network, 263
avoiding via routing protocol design, 263–264
troubleshooting, 263
mesh groups, 206

metrics, 387–388
BGP, 423–425
IS-IS, 213, 415
OSPF external route metrics, 164–167
metric-style transition command, 213
metric-style wide command, 213
MPLS (Multiprotocol Label Switching), 213, 353–355
BGP/MPLS VPNs, 358, 361
implementing in EIGRP, 361–369
implementing in OSPF, 369–370
overlaying routing, 356–357
peer-to-peer routing over, 357
MTBF (mean time between failures), 23–24
MTTR (mean time to repair), 24–25
multiaccess networks, OSPF adjacencies, 405
Multiaccess Reachability Protocol. *See* MARP
multicast addresses, EIGRP, 122–123
multiple points of redistribution, 66
filters, 67–69
tags, 69–71
multipoint GRE tunnels, 376–378

N

narrow metrics, 415
neighbor adjacencies
IS-IS, misconfigured NSAPs, 217
OSPF, troubleshooting, 184–187
neighbor relationships
BGP, 226, 240
troubleshooting, 227–231
eBGP, 426–427
EIGRP, 385–386
hello interval, 385
net command, 210
network design goals, 5
manageability, 13–14
day-to-day maintenance, 14–16
emergency management, 18–20
reliability, 6
and resiliency, 10
network failures, defining, 12–13
network recovery time, 13
of packet delivery, 6–9
scalability, 20
network failures
defining, 12–13
MTBF, 23–24
MTTR, 24–25
network layer functions, 38
aggregation of routing information, 39
controlling traffic, 42
defining routing policies, 41
forwarding traffic, 38
user attachment, 42
network management, complexity of, 25–32
network recovery time, 13
network service access points. *See* NSAPs
next hop attribute (BGP), 429
NHRP (Next Hop Routing Protocol), 372–373
ATM network implementation, case study, 373–375
no ip next-hop-self command, 104
no ip next-hop-self eigrp command, 101
no ip peer host-route command, 94
no isis hello-padding command, 217
nodes, SPF, 299
Non-Stop Forwarding (NSF), 265
notification, event-driven, 283
NSAPs (network service access points), 412–413
misconfigured, 217
versus IP addresses, 413
NSAPs (network service access points, 215
NSF (Non-Stop Forwarding), 265
NSSAs (Not-So-Stubby Areas), 157–158
nsf command, 273
null0 interface, 234

O

opaque LSAs, 272–273
OpenConfirm state, BGP neighbors, 228
OpenSent state, BGP neighbors, 227
OSPF (Open Shortest Path First)
adjacency formation, 318
areas, 406
attacks, 318–320
comparing with EIGRP, 457–468
data transport, 318
deploying
on three-layer hierarchy, 146–150
on two-layer hierarchy, 152
dial links, 180
exponential backoff, 292
external routes

case study, 182
injecting, 407
metrics, 164–167
selecting at ABRs, 167
full mesh topologies, 167, 170–171
GR, 270–274
using link local signaling, 271–272
using opaque LSAs, 272–273
hello packets, 270
hub-and-spoke topologies, 171–177
implementing BGP/MPLS VPNs, 369–370
incremental SPF, 302
LSAs, 400–402
adjacencies, building, 403–405
age parameter, 402
reliable flooding, 403
throttling, 292
MD5 authentication, 331
neighor adjacencies, troubleshooting, 184–187
normal restart, 270
point-to-point broadcast links, 181–182
polling, 280
PRC, 293
restart signaling, 271
route aggregation, 160–162
route filtering, 164
route selection between processes, 167
router IDs, 399
SPF
calculation time, 306
throttling, 292
stub areas, flooding reduction, 153–155, 160
summarization, 144–145
timers, 264
transport-level attacks, 318–320
virtual links, 408
out-of-band resynchronization, OSPF, 271
output, BGP neighbors, 227
overlaying routing onto MPLS VPNs, 356–357

P

packet corruption, 214
packet filtering, 329–330
packet flooding, 318
partial route calculation (PRC), 293
passwords
configuration mode access, 327
console, 327
enable, 327
enable secret, 327
HTML server, 327
router access, 326
SSH, 326
Telnet, 326
virtual terminal, 326
path costs, IS-IS, 213
path errors, 284
path vector protocols, BGP. *See* BGP
PE (provider edge) routers, 356
peer groups, BGP, 239
peers, BGP, 226–227
peer-to-peer routing over MPLS VPNs, 357
point-to-point broadcast links, OSPF, 181–182
policies, BGP, 225
polling, detecing link/adjaceny failures, 280–283
port flooding, 318
pos delay triggers command, 284
pos threshold command, 284
PRC (partial route calculation), 293
prc-interval command, 293
prefix lists
BGP filtering, 431
preventing redistribution routing loops, 131
prefix-driven routing installation (IS-IS), 216
PrefixPolicyCerts, soBGP, 347
protocol-level attacks, 318, 322
pseudonodes, IS-IS, 215–216

Q

queries
controlling propagation, 116–117
EIGRP neighbors, 390–391
binding, 393–395
stub routers, 88

R

reachability information, 310
reaction times to failures, 290
redistribute static ip command, 205
redistributed next hop, EIGRP, 102–104
redistribution, 129–130
and connected routes, 65

BGP, 233–235
distribute lists, 130
external flags, 132–134
IGP-to-IGP, 62–64
into OSPF, 164–167
IS-IS, 204–205
multiple points of, 66
filters, 67–69
tags, 69–71
prefix lists, 131
route maps, 130–131
setting admin distance, 131–132
setting admin tags, 133
tag filtering, applying, 134
redistribution command, 204
redundancy
BGP, route reflectors, 244
network manageability, effect on , 25
resiliency, effect on, 21–22
scalability, effect on, 26–27
MTBF, 23–24
MTTR, 24–25
versus resiliency, 6
reflood storms, 215
regional IGPs, 238
reliability, 6
and resiliency, 10
network failures
defining, 12–13
recovery time, 13
of packet delivery, 6–7
delay and jitter budgets, 9
reporting, limiting, 264–265
GR, 266
BGP, 277–279
EIGRP, 267–269
IS-IS, 274–276
OSPF, 270–274
NSF, 265
resiliency, 11
and redundancy, 6, 21–22
restart acknowledgment (RA) bit, 275
restart request (RS) bit, 275
Restart TLV, 275
restarts
BGP
GR, 278
normal, 277
EIGRP
GR, 268
normal, 267
IS-IS
GR, 275
normal, 274
OSPF GR
using link local signaling, 271
using opaque LSAs, 272
OSPF normal, 270
reuse limit, BGP route dampening, 257
route calculation, decreasing convergence speed, 296
EIGRP feasible successors, 296–299
link-state incremental SPF, 300–302
link-state partial SPF, 299–300
route dampening algorithm, BGP, 226
route flaps, BGP, 340
route leaking, 203
route maps
BGP filtering, 430
EIGRP, 104–108
selecting routes to advertise, 109
selective filtering, 109
setting tags on redistributed routes, 110
IS-IS, 205
preventing redistribution routing loops, 130–131
route reflectors, BGP, 242–245
route servers, 245–247
route summarization, BGP, 432
route tags
IS-IS, 205
prefix-driven route table installation, 217
router IDs, 399
router isis configuration mode, 201, 204, 210, 213
routers, 310
control versus forwarding planes, 265
intermediate systems (IS-IS), 190
routing
access layer, 83
best next hop, 86–87
dual-homed remotes, 85–86
single-homed sites, 84
calculating routes, 296
IS-IS, 413–414
OSPF filtering, 164
SIA routes, 391–393
routing areas, IS-IS, 198–202
aggregating routes, 204
leaking routes into L1 routing domain, 203–204
routing attacks, 317
disrupting peering, 318
flooding, 318

protocol-level attacks, 318
transport-level attacks, 318
disrupting routing domain stability, 324–325
DoS attacks, 334–335
preventing via edge filters, 335
preventing via GTSM, 335–337
falsifying routing information, 323–324
transiting authorization, 314–316
routing domains, 190, 326
illegitimate devices,thwarting, 330
IPSec, 333–334
MD5 authentication, 331–333
IS-IS, 190
L1 versus L2, 198
L2 in the core, 194
overlapping L1/L2, 195
splitting single into multiple, 190
router compromise, avoiding, 326
filtering access, 328–330
using passwords, 326–328
routing loops, 129, 310
IS-IS, 205
preventing, 130–134
routing policies, 310
routing protocols, 309
comparing OSPF and EIGRP, 457
convergence time, 459–462
ease of troubleshooting, 458–459
suitability of network designs, 462–468
GR operation, 266
security, 343
802.1x, 343–344
soBGP, 344–348
routing tables
BGP load sharing, 249
IS-IS, 216–217

S

scalability, 20
and redundancy, 26–27
BGP, 236–237
secure origin BGP. *See* soBGP

security
attacks
BGP, 321
IS-IS, 318–320
OSPF, 318–320
protocol-layer attacks, 322
authentication, 311
transiting, 311
transiting trust, 311–313
authorization, 311
transiting trust, 311–313
brittleness, 316
extranets, 337–338
BGP, 339
dampening prefixes, 340
filtering routes, 339–340
limiting route count, 341
using EGPs, 339
Internet connections, 341
protecting against transit, 342–343
route dampening, 343
router filtering, 341
IPSec, 333
protecting information, 337
protocol-layer attacks, 322
RADIUS servers, 328

routing attacks
disrupting peering, 318
disrupting routing domain stability, 324–325
DoS attacks, 334–335
falsifying routing information, 323–324
routing protocols, 343
802.1x, 343–344
soBGP, 344–348
routing systems, 316
social engineering, 316
TACACs servers, 328
TCP, 322
trust, 311–313
selecting appropriate hierarchical networks, 45
selector bits, 216
separating network functionality, 32
set metric-type internal command, 233
shortest path first (SPF) algorithm. *See* SPF
show cdp neighbor detail command, 219
show clns neighbor command, 218
show ip bgp neighbor command, 227–228
show ip bgp neighbors command, 279
show ip eigrp neighbor command, 128
show ip eigrp neighbors command, 121–122
show ip eigrp topo command, 296–298

show ip eigrp topology active command, 126–127
show ip eigrp topology all command, 118
show ip eigrp topology command, 118
show ip interface brief command, 219
show ip ospf command, 292–293
show ip ospf neighbor detail command, 273
show ip ospf stat command, 306
show ip ospf timers rate-limit command, 292
show ip protocols command, 269
show ip route command, 115, 201
show isis data detail command, 215
show isis database command, 200
show isis database detail command, 202
show is-is nsf command, 276
show isis spf-log command, 306
SIA (stuck-in-active), 84, 391, 393
    routes, troubleshooting, 124–128
    timers, 135
signaling, link local, 271–272
single point of redistribution, 64
single-homed sites, 84
Smoothed Round Trip Time (SRRT), EIGRP, 121
soBGP (secure origin Border Gateway Protocol, 344–345
    authentication, 345–346
    authorization, 346–347
    internetwork topology mapping, 347–348
social engineering, 316
sockets, 48
SONET, 284–285
SoO attribute (EIGRP), 365–367
speakers, BGP, 238
SPF (shortest path first), 264
    calculation time, 291, 306
    exponential backoff, setting timers, 306–307
    flooding, 213
    incremental, IS-IS, 302
    IS-IS, 213
    link-state incremental SPF, 300–302
    link-state partial SPF, 299–300
    throttling, 292
spf-interval command, 293
SRRT (Smoothed Round Trip Time), EIGRP, 121
SSH (secure shell), passwords, 326
standards track RFCs for IS-IS, 411
stub areas, reducing flooding, 153–155, 160
stub routing, 87–90, 394–395
Stuck-in-Active. *See* SIA
subnetworks, BGP, 238
suboptimal routing, 59–60
summarization, 54, 57, 114, 144–145, 393
    aggregation layer, 98
    BGP, 432
    configuring in IS-IS, 204
    controlling query propagation, 116–117
    core layer, 77, 98
        summarizing into core, 80
        summarizing to distribution layer, 77–78
    discard routes, 116
    distribute lists, 116
    distribution layer, 80
        summarizing toward core, 80–83
        summarizing toward remote sites, 83
    IP summary addresses, 114–115
    IS-IS, 190
    metrics, 61–62
    multiple topology table entries, 118–119
    stub routers, 87–90
    suboptimal routing, 59–60
    versus aggregation, 192
summary command, 204
suppress adjacency (SA) bit, 275
suppress limit, BGP route dampening, 256–257
synchronization, iBGP, 431–432

T

tables, BGP, 234
    load sharing, 249
TCP attacks, 321–322
Telnet, passwords, 326
third-party next hop, EIGRP, 99
    NBMA hub-and-spoke networks, 99–102
    redistributed next hop, 102–104
three-layer hierarchies, 44–45
throttling, 292
Time To Live (TTL) mechanism, 336
timers, 324
    EIGRP, 134
        hold timers, 134–135
        hold/SIA timer interaction, 135–136
        SIA timer, 135
    IS-IS, 264
    link-state generation timer, 291
    link-state update generation, 305
    OSPF, 264
    SPF, 291
        exponential backoff, 306–307
timers active command, 136

timers lsa arrival command, 292
timers nsf route-hold command, 269
timers throttle lsa all command, 292
topologies
    BGP, 226
    full mesh, 167, 170–171
    hub-and-spoke, 171–177
topology maps, soBGP, 347–348
topology tables, 118–119
    EIGRP, clearing, 390–391
totally NSSAs, 159–160
totally stubby areas, 156
traffic engineering, IS-IS, 213
tranport mode, IPSec, 333
transit networks, BGP, 253
transport-level attacks, 318
    against BGP, 321
    against EIGRP, 320
    against OSPF/IS-IS, 318–320
troubleshooting
    EIGRP neighbor relationships, 120
        mismatching primary addresses, 120–122
        multicast delivery problems, 122–123
    OSPF neighbor adjacencies, 184–187
trust
    security aspects, 316
    transitive, 311–313
TSNRFA (totally stubby not really full area), 160
TTL (Time To Live) mechanism, 336
tunnel mode, IPSec, 333
two-layer hierarchies, 43–44

U-V

updates, BGP, 239

virtual links, 408
virtual terminal passwords, 326
VPNs
    MPLS, 353–355
        BGP/MPLS VPNs, 358, 361–370
        overlaying routing onto, 356–357
        peer-to-peer routing over, 357
    multipoint GRE tunnels, 376–378

W-X-Y-Z

wait timers, 324
wide metrics, 415
    IS-IS, 213

X.509vs certificate, soBGP, 345