CCNP Exam Preparation

# CCNP Routing and Switching Quick Reference

A quick review of CCNP Routing and Switching exam topics

Denise Donohue, CCIE® No. 9566
Brent Stewart

ciscopress.com

# CCNP

# Routing and Switching Quick Reference

Denise Donohue, CCIE No. 9566
Brent Stewart

**Cisco Press**
800 East 96th Street
Indianapolis, Indiana 46240 USA

# CCNP Routing and Switching Quick Reference

Denise Donohue, Brent Stewart

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

Printed in the United States of America

First Printing: January 2010

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

ISBN-13: 978-1-58720-284-1

ISBN-10: 1-58720-284-0

**Publisher**
Paul Boger

**Associate Publisher**
Dave Dusthimer

**Executive Editor**
Brett Bartow

**Project Editor**
Jennifer Gallant

**Managing Editor**
Patrick Kanouse

**Editorial Assistant**
Vanessa Evans

**Designer**
Sandra Schroeder

## Warning and Disclaimer

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

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

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

## Trademark Acknowledgments

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

## Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

**U.S. Corporate and Government Sales**    1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

**International Sales**    international@pearsoned.com

## Feedback Information

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

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

We greatly appreciate your assistance.

## About the Authors

**Denise Donohue, CCIE No. 9566,** is a senior solutions architect for ePlus Technology, a Cisco Gold partner. She works as a consulting engineer, designing networks for ePlus' customers. Prior to this role, she was a systems engineer for the data consulting arm of SBC/AT&T. She has co-authored several Cisco Press books in the areas of route/switch and voice. Denise has worked as a Cisco instructor and course director for Global Knowledge and was a network consultant for many years. Her areas of specialization include route/switch, voice, and data center.

**Brent Stewart, CCNP®, CCDP®, CCSI, MCSE,** is the manager of Connectivity Services at CommScope. He is responsible for designing and managing a large-scale worldwide voice, video, and data network. Previously he was a course director for Global Knowledge and participated in the development of BSCI with Cisco and has written and taught extensively on CCNA® and CCNP. Brent lives in Hickory, NC, with his beautiful wife, Karen, and their mischievous children Benjamin, Kaitlyn, Madelyn, and William.

## About the Technical Editors

**'Rhette (Margaret) Marsh** has been working in the networking and security industry for over ten years, and has extensive experience with internetwork design, IPv6, forensics, and greyhat work. She currently is a design consultant for Cisco in San Jose, CA, and works primarily with the Department of Defense and contractors. Before this, she worked extensively both in the financial industry as a routing and switching and design/security consultant and also in an attack attribution and forensics context. She currently holds a CCIE in Routing and Switching (CCIE No. 17476), CCNP, CCDP, CCNA, CCDA, CISSP and is working towards her Security and Design CCIEs. In her copious free time, she enjoys number theory, arcane literature, cycling, hiking in the redwoods, sea kayaking, and her mellow cat, lexx.

# Contents at a Glance

## ROUTE

Chapter 1 Planning for Complex Networks

Chapter 2 EIGRP

Chapter 3 OSPF

Chapter 4 Optimizing Routing

Chapter 5 Path Control

Chapter 6 BGP and Internet Connectivity

Chapter 7 Branch Office Connectivity

Chapter 8 Mobile Worker Connectivity

Chapter 9 IPv6 Introduction

Appendix A

Appendix B

## SWITCH

Chapter 1 Campus Network Design

Chapter 2 VLAN Implementation

Chapter 3 Spanning Tree

Chapter 4 InterVLAN Routing

Chapter 5 Implementing High Availability

Chapter 6 First Hop Redundancy

Chapter 7 Campus Network Security

Chapter 8 Voice and Video in a Campus Network

Chapter 9 Wireless LANs in a Campus Network

## TSHOOT

Chapter 1 Maintainance

Chapter 2 Troubleshooting Methodology

Chapter 3 Troubleshooting Tools

Chapter 4 Troubleshooting Switches

Chapter 5 Troubleshooting Routers

Chapter 6 Troubleshooting Security Features

Index

# Contents

## ROUTE

Chapter 1 Planning for Complex Networks 1

Network Design Models 1

Hierarchical Design Model 1

Enterprise Composite Model 2

Cisco Enterprise Architecture 7

SONA and IIN 7

Understanding Routing Protocols 9

Administrative Distance 9

Routing Protocol Characteristics 10

Building the Routing Table 11

Choosing a Route 11

Planning a Routing Implementation 12

Creating an Implementation Plan 12

Creating Implementation Documentation 13

Chapter 2 EIGRP 14

EIGRP Overview 14

EIGRP Messages 15

Packet Types 15

Neighbor Discovery and Route Exchange 15

EIGRP Route Selection 16

EIGRP Metric 16

Diffusing Update Algorithm (DUAL) 17

Route Selection Example 18

Planning an EIGRP Implementation 19

Basic EIGRP Configuration 20

Creating an EIGRP Default Route 20

Verify and Troubleshoot EIGRP 21

EIGRP Across a WAN 21

EIGRP over EoMPLS 21

EIGRP over MPLS 22

EIGRP over Frame Relay 23

WAN Bandwidth 24

**Customizing the EIGRP Configuration 25**

Passive Interface 25

Unicast Neighbors 25

Summarization 26

**Load Balancing 26**

**EIGRP Authentication 27**

**EIGRP Scalability 29**

EIGRP Stub 29

Active Process Enhancement 30

Graceful Shutdown 30

**Chapter 3 OSPF 31**

**OSPF Overview 31**

OSPF Network Structure 31

OSPF Metric 33

**Link State Advertisements (LSA) 34**

LSA Operation 34

LSDB Overload Protection 34

**LSA Types 35**

**OSPF Operation 35**

OSPF Packets 36

OSPF Neighbor Relationships 36

Establishing Neighbors and Exchanging Routes 37

**Planning for OSPF 37**

**Basic OSPF Configuration 38**

**Router ID 39**

**Verify and Troubleshoot OSPF 39**

**OSPF Network Types 40**

Designated Routers 41

Nonbroadcast Multiaccess (NBMA) Networks 42

OSPF over Layer 2 and Layer 3 MPLS 42

Advanced OSPF Configuration 42
OSPF Summarization 43
Passive Interface 43
OSPF Default Routes 44
Stub and Not-So-Stubby Areas 44
Virtual Links 45
OSPF Authentication 46

Chapter 4 Optimizing Routing 48
Controlling Routing Updates 48
Route Maps 48
Prefix Lists 52
Distribute Lists 54
Passive Interfaces 54
Using Multiple Routing Protocols 54
Configuring Route Redistribution 55
Seed Metric 55
Administrative Distance 56
Planning Route Redistribution 58
Redistribution Techniques 59
Redistribution Notes 59

Chapter 5 Path Control 61
Using Offset-lists 61
Using IOS IP SLA 62
Policy-Based Routing 64
OER and VRF 66

Chapter 6 BGP and Internet Connectivity 67
Planning an Internet Connection 67
To Route or Not to Route? 67
BGP Route Options 68
Types of ISP Connections 68
BGP Overview 69
BGP Databases 70
BGP Message Types 71

Internal and External BGP 71
BGP Next-Hop Selection 72
BGP Next Hop on a Multiaccess Network 73
BGP Synchronization Rule 73
**Configuring BGP 74**
BGP Network Command 74
BGP Peering 75
BGP Peering States 75
**BGP Path Selection 76**
BGP Attributes 76
BGP Path Selection Criteria 77
Influencing BGP Path Selection 78
**Filtering BGP Routes 78**
**BGP Authentication 79**
**Verifying BGP 80**
**Chapter 7 Branch Office Connectivity 83**
**Branch Office Design Considerations 83**
Small Branch Office Design 83
Medium Branch Office Design 84
Large Branch Office Design 84
**Implementing Branch Offices 84**
Verifying Existing Services 85
Configuring a Backup DSL Connection 85
Configuring an IPsec VPN 89
Configuring a Floating Static Route 90
Configuring Dynamic Routing over a GRE Tunnel 90
Load Sharing with EIGRP 92
**Chapter 8 Mobile Worker Connectivity 93**
Components of a Mobile Worker Solution 94
Implementing a Mobile Worker Solution 95

Chapter 9 IPv6 Introduction 99

IPv6 Addressing 99

Simplifying an IPv6 Address 100

Special Addresses 101

IPv6 Host Addressing 101

IPv6 Routing 104

Static Routing 105

RIPng for IPv6 105

EIGRP for IPv6 106

OSPFv3 107

MP-BGP for IPv6 109

RIPng Redistribution 109

Integrating IPv4 and IPv6 110

Tunneling IPv6 over IPv4 111

Using Address Translation 113

Static NAT-PT 113

Dynamic NAT-PT 113

IPv6 Link Types 114

Point-to-Point Links 114

Point-to-Multipoint Links 115

Multiaccess Links 115

Appendix A Understanding IPsec 117

IPsec Headers 117

Authentication Header 117

Encapsulating Security Payload 118

IPsec Modes 118

Authentication Methods 119

Encryption Methods 119

Symmetric Key Algorithms 119

Asymmetric Key Algorithm 120

Diffie-Hellman Key Exchange 120

**Key Management 120**

**Establishing an IPsec VPN 121**

**Configuring a Site-to-Site VPN Using IOS 121**

Configuring an ISAKMP Policy 122

Configuring an IPsec Transform Set 123

Configuring a Crypto ACL 123

Configuring a Crypto Map 124

Applying the Crypto Map to an Interface 124

Configuring an Optional Interface Access List 124

**Monitoring and Troubleshooting IPsec VPNs 125**

**Using GRE with IPsec 125**

**Configuring a GRE Tunnel Using IOS 126**

**Appendix B IPv6 Header Format 127**

## SWITCH

**Chapter 1 Campus Network Design 129**

**The Hierarchical Design Model 129**

**Core Layer 130**

Small Campus Design 131

Medium Campus Design 131

Data Center Design 131

Network Traffic Flow 133

**Service-Oriented Network Architecture 134**

**Planning a Network Implementation 135**

**Chapter 2 VLAN Implementation 137**

**VLAN Overview 137**

VLAN Planning 139

Creating a VLAN and Assigning Ports 139

Verifying VLAN Configuration 139

**VLAN Trunking 141**

Configuring a Trunk Link 142

VLANs Allowed on the Trunk 143

Best Practices for Trunking 143

**VLAN Trunking Protocol 143**

VTP Switch Roles 144

Configuring VTP 145

Verifying and Monitoring VTP 145

Adding a New Switch to a VTP Domain 146

**EtherChannels 146**

Configuring an EtherChannel 147

**Verifying an EtherChannel 148**

**Troubleshooting VLAN Issues 148**

Troubleshooting User Connectivity 148

Troubleshooting Trunking 149

Troubleshooting VTP 149

**Chapter 3 Spanning Tree 150**

**Understanding the Spanning Tree Protocol 150**

Spanning Tree Election Criteria 151

STP Election 152

Bridge Protocol Data Units 154

Spanning Tree Port States 155

Per-VLAN Spanning-Tree 155

Configuring Spanning Tree 156

Portfast 156

**Rapid Spanning Tree 157**

RSTP Port Roles 157

BPDU Differences in RSTP 157

RSTP Fast Convergence 158

**Multiple Spanning Tree 159**

**Spanning Tree Stability Mechanisms 159**

UplinkFast 159

BackboneFast 160

BPDU Guard 160

BPDU Filtering 161

Root Guard 161

Unidirectional Link Detection 161

Loop Guard 162

**Troubleshooting STP 163**

**Identifying a Bridging Loop 163**

**Spanning-Tree Best Practices 164**

**Chapter 4 InterVLAN Routing 166**

**InterVLAN Routing Using an External Router 166**

**InterVLAN Routing Using Multilayer Switches 167**

The Layer 2 and Layer 3 Forwarding Process 167

Understanding the Switching Table 168

MLS Interfaces 168

Understanding Switch Forwarding Architectures 170

CEF Switching 170

Configuring and Troubleshooting CEF 171

**Chapter 5 Implementing High Availability 173**

**Components of High Availability 173**

Redundancy 173

Technology 173

People 174

Processes 175

Tools 175

**Resiliency and High Availability 176**

Network Level Resiliency 176

Fast Failover 176

**Optimizing Redundancy 177**

NSF with SSO 177

Designing for Redundancy 178

Layer 2 Versus Layer 3 Access Design 178

Using Nonchassis Based Access Switches 181

**Network Management for High Availability 181**

Syslog 182

SNMP 182

IP SLA 183

Chapter 6 First Hop Redundancy 185

Hot Standby Router Protocol 185

HSRP States 186

Configuring HSRP 186

Virtual Router Redundancy Protocol 188

GLBP 189

Planning Router Redundancy Implementation 190

Chapter 7 Campus Network Security 191

MAC Address-Based Attacks 191

MAC Address Flooding 192

Port Security 192

Port-Based Authentication 193

VLAN-Based Attacks 194

Switch Spoofing 194

802.1Q Double-Tagging 195

VACLs 195

Private VLANs 196

Protected Ports 197

Spoof Attacks 197

DHCP Spoofing 198

IP Source Guard 198

ARP Spoofing 198

Securing Your Switch 199

Chapter 8 Voice and Video in a Campus Network 200

VoIP in a Campus Network 201

Preparing the Network for VoIP 202

Voice VLANs 203

QoS for VoIP 204

Configuring VoIP Support on a Switch 206

Video over IP 208

QoS Requirements for Video 208

**Chapter 9 Wireless LANs in a Campus Network 210**

**Cisco Unified Wireless Network 210**

**Characteristics of Wireless LANs 211**

Service Set Identifiers (SSID) 211

WLAN Topologies 211

Client Connectivity 212

**Cisco Wireless Network Components 213**

Autonomous (Stand-alone) APs 213

Lightweight Access Points 214

Wireless LAN Controllers 215

Hybrid Remote Edge Access Point (H-REAP) 216

**Integrating Wireless into the LAN 216**

Switch Configuration 217

Planning for a Wireless Implementation 217

## TSHOOT

**Chapter 1 Maintenance 219**

**Methodology 219**

**Common Tasks 220**

**Tools 223**

Configurations 223

Other Tools 228

**Chapter 2 Troubleshooting Methodology 230**

**Principles 230**

**Structured Troubleshooting 231**

**The Troubleshooting Method 232**

**Integrating Troubleshooting into Maintenance 233**

**Chapter 3 Troubleshooting Tools 235**

**IOS Filtering Tools 235**

**Output Redirection 242**

**IOS Troubleshooting Tools 242**

**Hardware Diagnostics 246**

**Working with External Tools 249**

Packet Sniffing 249

Netflow 250

SNMP and EEM 250

**Chapter 4 Troubleshooting Switches 252**

**Spanning Tree 254**

**VLANs 257**

**Switched Virtual Interfaces and InterVLAN routing 258**

**First-Hop Redundancy 258**

**Chapter 5 Troubleshooting Routing 261**

**Network Layer Connectivity 261**

**Routing Protocols 261**

**EIGRP 262**

Is the Correct Route Advertised? 262

Is the Correct Route Communicated? 263

Is There a More Desirable Path? 264

**OSPF 264**

Is the Correct Route Advertised? 264

Is the Correct Route Communicated? 264

Is There a More Desirable Path? 265

**BGP 265**

Is the Correct Route Advertised? 265

Is the Correct Route Communicated? 266

Is There a More Desirable Path? 266

Route Redistribution 267

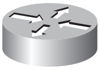
**Router Performance 268**

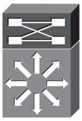
**Chapter 6 Troubleshooting Security Features 270**

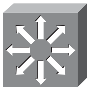
**Troubleshooting Security Features 271**

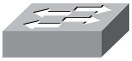
**Index 273**

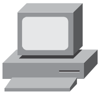
# Icons Used in This Book


Router


Route/Switch Processor


Multilayer Switch


Workgroup Switch


PC

# Command Syntax Conventions

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

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

# Introduction

The Cisco Certified Network Professional (CCNP®) Routing and Switching certification validates knowledge and skills required to install, configure and troubleshoot converged local and wide area networks with 100 to 500 or more nodes. With a CCNP Routing and Switching certification, a network professional demonstrates the knowledge and skills required to manage the routers and switches that form the network core, as well as edge applications that integrate voice, wireless, and security into the network.

The CCNP Quick Reference was written to help you prepare for the three exams in the CCNP Routing and Switching certification. Some readers tell us that they use this book before beginning their exam preparation, to find which areas they are weak in. This helps target their studying. Others use it after studying or taking the course as a concise learning resource during their final preparation for the exam.

This book will also help once your exams are over, when you need a quick answer about a technology, or a reminder about configuration steps.

# Who Should Read This Book?

Current and aspiring network engineers will find this book useful in two ways. First, those preparing for the CCNP Routing and Switching certification will appreciate the targeted review of exam topics. It will help them understand the technologies, not just memorize questions. This will lead to success on the exam and improved on-the-job performance. Secondly, the book serves as a reference for those not pursuing the certification. Its short descriptions and many examples come in handy when you need a fast answer to a question, or to configure something quickly. It deserves a place on every network engineer's bookshelf.

# How This Book Is Organized

The book is organized by exam, and in the order that most people take the exams. ROUTE is first, followed by SWITCH, and then TSHOOT. The topics within each exam section coorespond with those on the exam blueprint. This edition as a whole include more emphasis on planning and verifying network changes than the previous edition. There is more information on troubleshooting commands throughout the entire book, in addition to the TSHOOT section. Following is a description for each section.

## ROUTE:

### Chapter 1: Planning for Complex Networks

This chapter describes some different design models in use, provides an overview of routing protocols, and introduces methods of implementation planning.

### Chapter 2: EIGRP

This chapter provides an in-depth description of EIGRP operation and configuration, including neighbor establishment and route exchange. It covers using EIGRP with Frame Relay, Ethernet over MPLS (EoMPLS) and Layer 3 MPLS VPNs. It also includes planning an EIGRP implementation, ways to optimize EIGRP, and securing EIGRP through authentication.

### Chapter 3: OSPF

Chapter 3 describes OSPF's structure and operation. It covers OSPF design requirements, neighbor establishment, and LSA information. The configuration portion provides implementation planning, as well as OSPF configuration for LANs and WANs (including MPLS WANs.) The chapter additionally covers optimizing and securing OSPF.

### Chapter 4: Optimizing Routing

This chapter examines various methods of controlling routing updates, such as route maps, prefix lists, and distribute lists. It describes how to configure route maps, and how to use them for policy-based routing, controlling route redistribution, and tagging routes. Route redistribution planning and isses are described, redistribution configuration is demonstrated.

### Chapter 5: Path Control

Chapter 5 covers some additional ways to control the path that traffic takes. These include offset-lists, IP SLA, policy-based routing, Optimized Edge Routing, and VRFs. IP SLA tracking is shown along with route maps to provide backup routes when using policy-based routing.

### Chapter 6: BGP and Internet Connectivity

Whether or not to use BGP is the first topic discussed in Chapter 6. Different types of ISP connections are covered. The chapter gives an overview of BGP operation and basic configuration. BGP path selection is covered, along with ways to influence the path selection and ways to filter routes. Additionally, ways to verify BGP operation are shown.

### Chapter 7: Branch Office Connectivity

Chapter 7 considers branch office routing design and implementation. This includes a description of DSL with PPPoA, and IPsec VPNs. Dynamic routing over various types of tunnels is covered, and a GRE tunnel configuration is shown. The chapter ends with a look at EIGRP load sharing.

**Chapter 8: Mobile Worker Connectivity**
Chapter 8 covers the network changes necessary to allow mobile workers to connect to the corporate network. The services needed to enabled this are discussed, including NAT, DHCP, firewalling, VPNs, and routing.

**Chapter 9: IPv6 Introduction**
This rather in-depth introduction to IPv6 covers the IPv6 address format, ways for hosts to acquire their addresses, and IPv6 routing for RIP, EIGRP, OSPF, and BGP. It also includes strategies for integrating IPv4 and IPv6 such as various types of tunnels. IPv6 operation over various types of links is additionally covered.

**SWITCH:**

**Chapter 1: Campus Network Design**
Chapter 1 covers design considerations for small, medium, and large campuses, and Data Centers. It describes the Service-Oriented Network Architecture (SONA) and how it applies to campus design. The PPDIOO model is shown as a way to plan network implementations.

**Chapter 2: VLAN Implementation**
This chapter gives an overview of VLANs, then describes VLAN design and implementation planning. It covers trunking,VTP,and EtherChannel including best practices, configuration, and troubleshooting.

**Chapter 3: Spanning Tree**
Chapter 3 goes into detail on Spanning-Tree, Rapid Spanning-Tree, and Multiple Spanning Tree. It covers spanning-tree tuning mechanisms such as UDLD, loop guard, backbonefast, and BPDUguard. It also includes troubleshooting Spanning-Tree and Spanning-Tree best practices.

**Chapter 4: InterVLAN Routing**
Routing between VLANs using a router and using a multilayer switched are both covered in Chapter 4. This chapter additionally describes switch forwarding architectures and goes into detail on CEF operation and configuration.

**Chapter 5: Implementing High Availability**
This chapter describes the components of high availability - redundancy, technology, people, processes, and tools. It goes into depth on achieving network resiliency through correct network design. It also describes the role that network management tools Syslog, SNMP, and IP SLA play in a highly available network.

**Chapter 6: First Hop Redundancy**
Chapter 6 looks at HSRP, VRRP, and GLBP. It describes their operation, the differences between them, how to configure them, and how to tune them, It covers best practices planning and implementation.

### Chapter 7: Campus Network Security

This chapter is concerned with ways that the LAN might be attacked and its security compromised. It covers four types of attacks: MAC address attacks, VLAN-based attacks, spoofing attacks, and attacks against the switch itself. Prevention techniques are shown for each type of attack.

### Chapter 8: Voice and Video in a Campus Network

Chapter 8 describes how to prepare a network for voice over IP and video over IP. It covers the components needed such as a PoE switch, and the QoS requirements of voice and video. The chapter also shows how to configure a switch to support VoIP, including AutoQoS configuration.

### Chapter 9: Wireless LANs in a Campus Network

This chapter describes how to integrate wireless into the LAN. It provides an overview of wireless operation and components, and describes how introducing wireless impacts the LAN traffic. The chapter also discusses how to plan a wireless implementation and shows how to configure your switches when connecting to access points or controllers.

## TSHOOT

### Chapter 1: Maintainance

This chapter provides descriptions of tasks commonly used to maintain performance and prepare for problems, such as documentation and scheduled preventative maintenance. This chapter also describes some IOS tools - such as archiving, logging, and configuration rollback - that are valuable in this process.

### Chapter 2: Troubleshooting Methodology

This chapter focuses on minimizing time-to-repair. This chapter focuses on the techniques that can be applied to minimize downtime. The scientific method is suggested as a model for troubleshooting.

### Chapter 3: Troubleshooting Tools

Cisco IOS has a number of ways to extract data about the state of the machine. This chapter discusses IOS output filtering and redirection, network probing commands, and hardware diagnostics.

### Chapter 4: Troubleshooting Switches

Ethernet is ubiquitous in campus networks and data centers. More and more services are traveling on Ethernet, such as storage, virtualization, and telephony. This chapter describes troubleshooting the critical pieces: Spanning Tree, VLANs, InterVLaN routing, and gateway redundancy.

### Chapter 5: Troubleshooting Routers

This chapter covers troubleshooting link layer connectivity, OSPF, EIGRP, and BGP routing protocols, and router performance.

### Chapter 6: Troubleshooting Security Features

Security is a pervasive element of networks. Routers are both potential targets for attacks and platforms that can offer security services. This chapter describes troubleshooting security features.

*This page intentionally left blank*

# CHAPTER 1

# Planning for Complex Networks

## Network Design Models

Today's networks typically include voice, video, network management, mission-critical, and routing traffic in addition to bulk user traffic. Each type of traffic has different performance (bandwidth, delay, and jitter) and security requirements. Network design models provide a framework for integrating the many different types of traffic into the network.

Over the years, several models have been used to help describe how a complex network functions. These models are useful for designing a network and for understanding traffic flow within a more complex network. This section covers three models: the traditional Hierarchical Model, the Enterprise Composite Model, and the Cisco Enterprise Model.

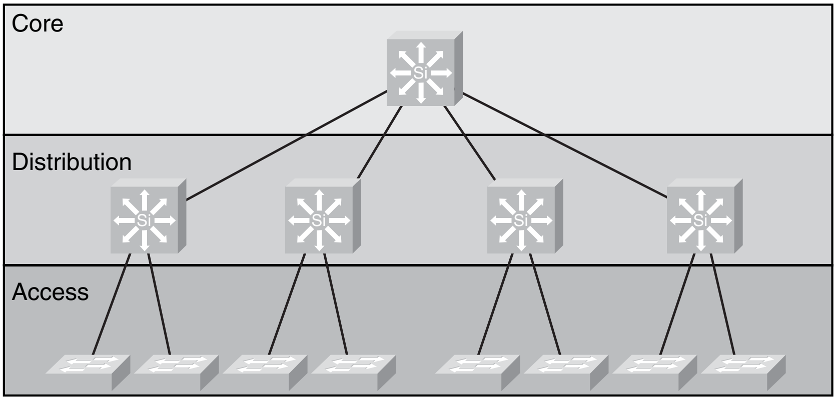
### Hierarchical Design Model

Network designers used the three-level *Hierarchical Design Model* for years. This older model provided a high-level idea of how a reliable network might be conceived, but it was largely conceptual because it didn't provide specific guidance. Figure 1-1 shows the Hierarchical Design Model.

This is a simple drawing of how the three-layer model might be built out for a campus network. A distribution Layer-3 switch is used for each building on campus, tying together the access switches on the floors. The core switches link the various buildings together.

This same three-layer hierarchy can be used in the WAN with a central headquarters, division headquarters, and units.

**Figure 1-1 Hierarchical Design Model**



The layers break a network in the following way:

- **Access layer:** Provides network access to workgroup end stations.
- **Distribution layer:** Intermediate devices provide connectivity based on policies.
- **Core layer:** Provides a high-speed switched path between distribution elements.

Redundant distribution and core devices, with connections, make the model more fault-tolerant. This early model was a good starting point, but it failed to address key issues, such as

- Where do wireless devices fit in?
- How should Internet access and security be provisioned?
- How do you account for remote access, such as dial-up or VPN?
- Where should workgroup and enterprise services be located?

## Enterprise Composite Model

A newer Cisco model—the Enterprise Composite Model—is significantly more complex and attempts to address the shortcomings of the Hierarchical

Design Model by expanding the older version and making specific recommendations about how and where certain network functions should be implemented. This model is a component of the Cisco Security Architecture for Enterprise (SAFE) Reference Architecture.

The Enterprise Model is broken into three large sections:

- **Enterprise Campus:** Switches that make up a LAN
- **Enterprise Edge:** The portion of the enterprise network connected to the larger world
- **Service Provider Edge:** The different public networks that are attached

The Enterprise Campus, as shown in Figure 1-2, looks like the old Hierarchical Design Model with added details. It features six sections:

- **Campus Backbone:** The core of the LAN
- **Building Distribution:** Connects subnets/VLANs and applies policy
- **Building Access:** Connects users to network
- **Management:** An out-of-band network to access and manage the devices
- **Edge Distribution:** A distribution layer out to the WAN
- **Server Farm:** For Enterprise services

The Enterprise Edge, as shown in Figure 1-3, details the connections from the campus to the WAN and includes

- E-commerce
- Internet connectivity
- Remote access
- WAN

**Figure 1-2 Enterprise Campus**

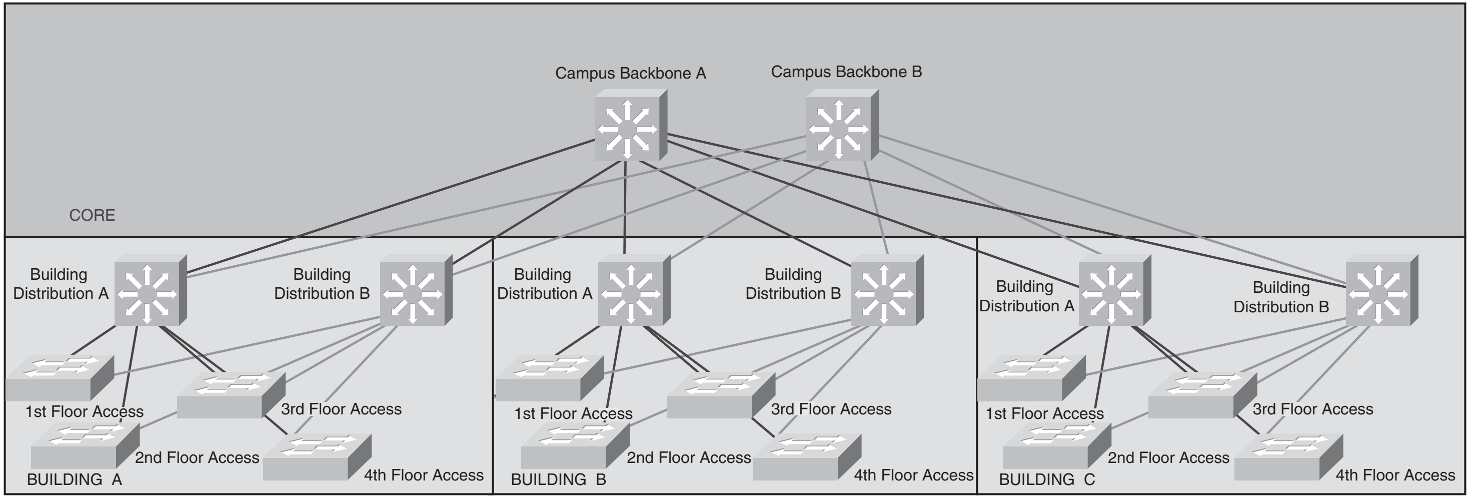
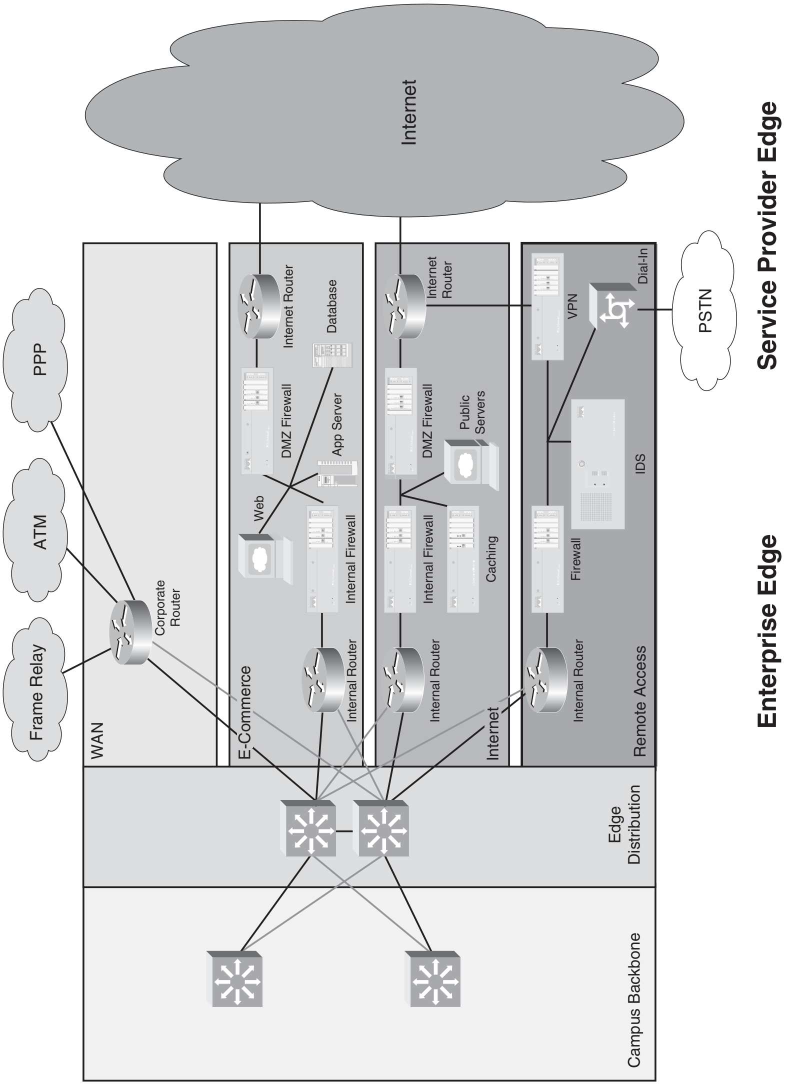


**Figure 1-3 Enterprise Edge**

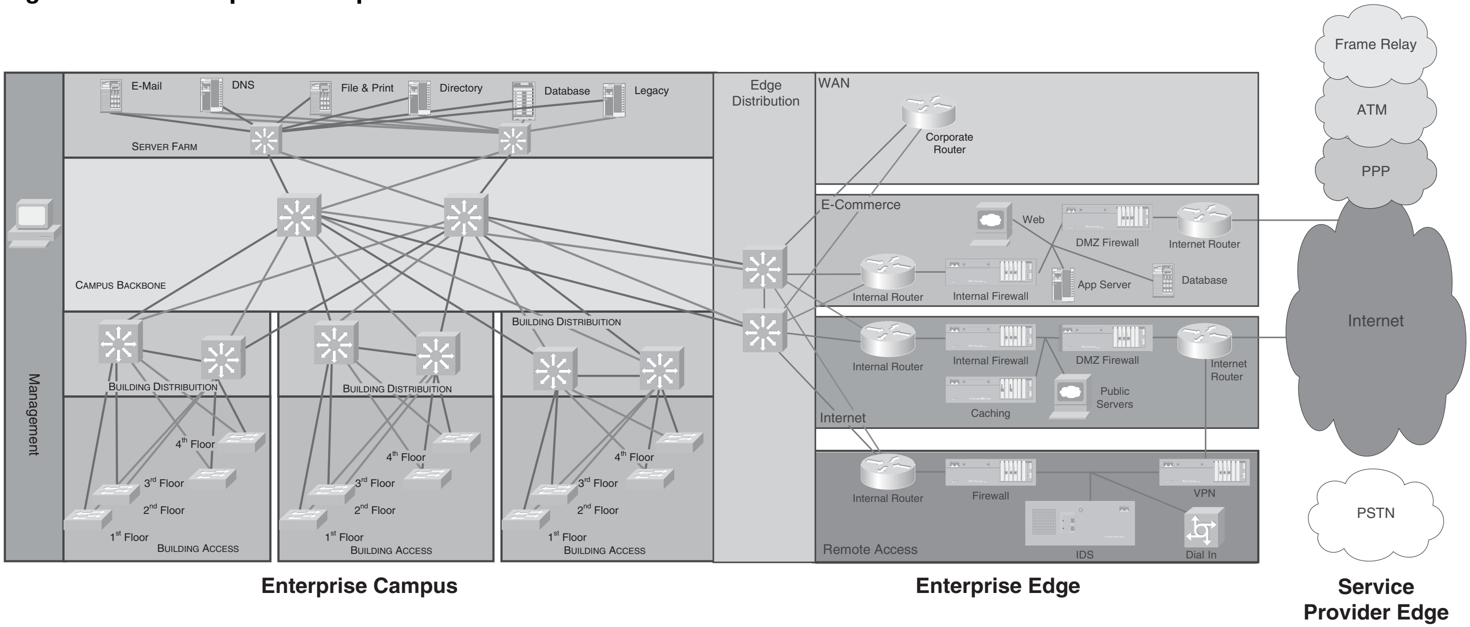


The Service Provider Edge is just a list of the public networks that facilitate wide-area connectivity and include

- Internet service provider (ISP)
- Public switched telephone network (PSTN)
- Frame Relay, ATM, and PPP

Figure 1-4 puts together the various pieces: Campus, Enterprise Edge, and Service Provider Edge. Security implemented on this model is described in the Cisco SAFE blueprint.

**Figure 1-4 Enterprise Composite Model**



## Cisco Enterprise Architecture

The Cisco Enterprise Architecture attempts to describe how all the network components integrate and work together. It includes Campus, Data Center, Branch, WAN, and Teleworker components.

The Campus Architecture component is basically the same as in the Composite model. It includes routing and switching integrated with technologies such as IP telephony and is designed for high availability with redundant links and devices. It integrates security features and provides QoS to ensure application performance. It is flexible enough to add advanced technologies such as VPNs, tunnels, and authentication management.

The Data Center component provides a centralized, scalable architecture that enables virtualization, server and application access, load balancing, and user services. Redundant data centers might be used to provide backup and business continuity.

The Branch Architecture extends enterprise services to remote offices. Network monitoring and management is centralized. Branch networks include access to enterprise-level services such as converged voice and video, security, and application WAN optimization. Resiliency is obtained through backup local call processing, VPNs, redundant WAN links, and application content caching.

The WAN component provides data, voice, and video content to enterprise users any time and any place. QoS, SLAs, and encryption ensure a high-quality secure delivery of resources. It uses IPsec or MPLS VPNs over Layer 2 or Layer 3 WANs, with either a hub-and-spoke or mesh topology.

Teleworker Architecture describes how voice and data are delivered securely to remote small or home office users. It leverages a standard broadband connection, combined with VPN and identity-based access. An IP phone can also be used.

# SONA and IIN

Modern converged networks include different traffic types, each with unique requirements for security, QoS, transmission capacity, and delay. These include

- Voice signaling and bearer
- Core application traffic, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM)

- Database transactions
- Multicast multimedia
- Network management
- Other traffic, such as web pages, email, and file transfer

Cisco routers can implement filtering, compression, prioritization, and policing. Except for filtering, these capabilities are referred to collectively as QoS.

Although QoS is a powerful tool, it is not the only way to address bandwidth shortage. Cisco espouses an idea called the Intelligent Information Network (IIN).

IIN describes an evolutionary vision of a network that integrates network and application functionality cooperatively and enables the network to be smart about how it handles traffic to minimize the footprint of applications. IIN is built on top of the Enterprise Composite Model and describes structures overlaid on to the Composite design as needed in three phases.

Phase 1, "Integrated Transport," describes a converged network, which is built along the lines of the Composite model and based on open standards. This is the phase that the industry has been transitioning to recently. The Cisco Integrated Services Routers (ISR) are an example of this trend.

Phase 2, "Integrated Services," attempts to virtualize resources, such as servers, storage, and network access. It is a move to an "on-demand" model.

By "virtualize," Cisco means that the services are not associated with a particular device or location. Instead, many services can reside in one device to ease management, or many devices can provide one service. An ISR brings together routing, switching, voice, security, and wireless. It is an example of many services existing on one device. A load balancer, which makes many servers look like one, is an example of one service residing on many devices.

VRFs are an example of taking one resource and making it look like many. Some versions of IOS are capable of having a router present itself as many virtual router (VRF) instances, allowing your company to deliver different logical topologies on the same physical infrastructure. Server virtualization is another example. The classic example of taking one resource and making it appear to be many resources is the use of a virtual LAN (VLAN) and a virtual storage area network (VSAN).

Virtualization provides flexibility in configuration and management.

Phase 3, “Integrated Applications,” uses application-oriented networking (AON) to make the network application-aware and to enables the network to actively participate in service delivery.

An example of this Phase 3 IIN systems approach to service delivery is Network Admission Control (NAC). Before NAC, authentication, VLAN assignment, and antivirus updates were separately managed. With NAC in place, the network can check the policy stance of a client and admit, deny, or remediate based on policies.

IIN enables the network to deconstruct packets, parse fields, and take actions based on the values it finds. An ISR equipped with an AON blade might be set up to route traffic from a business partner. The AON blade handles many functions, including examining traffic, recognizing an application, and rebuilding XML files in memory. Corrupted XML fields might represent an attack (called *schema poisoning*), and the AON blade can react by blocking that source from further communication. In this example, routing, an awareness of the application data flow, and security are all combined to enable the network to contribute to the success of the application.

Services-Oriented Network Architecture (SONA) applies the IIN ideal to Enterprise networks. SONA breaks down the IIN functions into three layers:

- **Network Infrastructure:** Hierarchical converged network and attached end systems
- **Interactive Services:** Resources allocated to applications
- **Applications:** Includes business policy and logic

# Understanding Routing Protocols

Routing protocols pass information about the structure of the network between routers. Cisco routers support multiple routing protocols, but the ROUTE exam covers only EIGRP, OSPF, and BGP. This section compares routing protocols and calls out some key differences between them.

## Administrative Distance

Cisco routers are capable of supporting several IP routing protocols concurrently. When identical prefixes are learned from two or more separate sources, Administrative Distance (AD) is used to discriminate between the paths. AD is a poor choice of words; *risk-factor* is a more descriptive name. All other things being equal, routers choose paths advertised by the protocol with the lowest AD. AD can be manually adjusted.

Table 1-1 lists the default values for various routing protocols.

**Table 1-1 Routing Protocols and Their Default Administrative Distance**

| Information Source | AD |
|---|---|
| Connected | 0 |
| Static | 1 |
| External BGP (Border Gateway Protocol) | 20 |
| Internal EIGRP (Enhanced IGRP) | 90 |
| IGRP (Internet Gateway Routing Protocol) | 100 |
| OSPF (Open Shortest Path First) | 110 |
| IS-IS (Intermediate System to Intermediate System) | 115 |
| RIP (Routing Information Protocol) | 120 |
| ODR (On Demand Routing) | 160 |
| External EIGRP | 170 |
| Internal BGP | 200 |
| Unknown | 255 |

## Routing Protocol Characteristics

Two things should always be considered in choosing a routing protocol: fast convergence speed and support for VLSM. EIGRP, OSPF, and BGP all meet these criteria. There are important distinctions between them, as described here:

- EIGRP is proprietary, so it can be used only in an all-Cisco network; however, it is simple for network staff to configure and support.
- OSPF is an open standard, but it is a bit more difficult for network staff to implement and support.
- BGP is also an open standard but is typically used to exchange routes with routers external to your network. It can be very complex to implement, and fewer network engineers understand it well.

Table 1-2 compares routing protocols.

**Table 1-2 Comparison of Routing Protocols**

| Property | EIGRP | OSPF | BGP |
|---|---|---|---|
| Method | Advanced distance vector | Link state | Path vector |
| Summarization | Auto and manual | Manual | Auto and Manual |
| VLSM | Yes | Yes | Yes |
| Convergence Speed | Very fast | Fast | Slow |
| Timers: Update (hello/dead) | Triggered (LAN 5/15, WAN 60/180) | Triggered, but LSA refreshes every 30 minutes (NBMA 30/120, LAN 10/40) | Triggered (60/180) |
| Network Size | Large | Large | Very large |

# Building the Routing Table

The router builds a routing table by ruling out invalid routes and considering the remaining advertisements. The procedure is

1. For each route received, verify the next hop. If invalid, discard the route.
2. If multiple identical, valid routes are received by a routing protocol, choose the lowest metric.
3. Routes are identical if they advertise the same prefix and mask, so 192.168.0.0/16 and 192.168.0.0/24 are separate paths and are each placed into the routing table.
4. If more than one specific valid route is advertised by different routing protocols, choose the path with the lowest AD.

# Choosing a Route

Routers look at the routing table to decide how to forward a packet. They look for a match to the destination IP address. Rarely will a route match the destination IP address exactly, so the router looks for the longest match. For instance, suppose a packet is bound for the IP address 10.1.1.1. The routing table has a route for 10.1.0.0/16, one for 10.1.1.0/24, and a default route of 0.0.0.0. The default route matches 0 bits of the destination address, the 10.1.0.0 route matches 16 bits of the destination address, and the 10.1.1.0

route matches 24 bits of the destination address. The 10.1.1.0 route is the longest match, so it will be used to forward the packet.

# Planning a Routing Implementation

It is critical to take a structured approach to planning a routing implementation and to document thoroughly once you are done. Taking an ad-hoc approach could lead to network instability, suboptimal routing, or scalability problems.

Four commonly used models include

- **Cisco Lifestyle Services:** Uses the PPDIOO model (Prepare, Plan, Design, Implement, Operate, and Optimize.) Network engineers at the CCNP level are involved with the implementation planning during the Design phase, and the Implementation itself during the Implement phase.
- **IT Infrastructure Library (ITIL):** Emphasizes business requirements and processes as they relate to IT. Implementation and implementation planning are part of its best practices.
- **Fault, Configuration, Accounting, Performance, and Security (FCAPS):** Has five network management categories. Implementation and implementation planning are under the Configuration management category.
- **Telecommunications Management Network (TMN):** Based on the FCAPS model. Implementation and implementation planning are one of its building blocks.

Each approach includes identifying requirements, creating an implementation plan, implementing the changes, verifying your work, and then documenting it.

## Creating an Implementation Plan

To create an implementation plan you need to know what the network looks like now, and what it should look like when you are done. This involves gathering information about the current network parameters such as IP addressing, physical connectivity, routing configuration, and equipment. Compare the current state to what is required. Be sure to include any site-specific requirements and any dependencies on the existing network.

An implementation plan includes most of the following, some of which might be site-specific:

- A checklist of tasks to be done
- Tools and resources needed
- The schedule of work, coordinated with all needed resources
- Device configurations
- Verification processes and tests

## Creating Implementation Documentation

Documentation should be kept up-to-date, accurate, and accessible. It includes network information, tools and resources used, implementation tasks, verification methods, device configurations, performance measurements, and possibly screen shots or pictures.

# CHAPTER 2

# EIGRP

## EIGRP Overview

Enhanced Interior Gateway Routing Protocol (EIGRP) is a Cisco proprietary, advanced distance vector, classless routing protocol that uses a complex metric based on bandwidth and delay. The following are some features of EIGRP:

- Fast convergence.
- Support for VLSM.
- Partial updates conserve network bandwidth.
- Support for IP, AppleTalk, and IPX.
- Runs directly over IP, using protocol number 88.
- Support for all Layer 2 (data link layer) protocols and topologies.
- Sophisticated metric that supports load-balancing across unequal-cost paths.
- Use of multicast (and unicast where appropriate) instead of broadcasts.
- Support for authentication.
- Manual summarization at any interface.
- Uses multicast 224.0.0.10.

EIGRP's function is controlled by four key technologies:

- **Neighbor discovery and maintenance:** Periodic hello messages
- **The Reliable Transport Protocol (RTP):** Controls sending, tracking, and acknowledging EIGRP messages
- **Diffusing Update Algorithm (DUAL):** Determines the best loop-free route
- **Protocol-independent modules (PDM):** Modules are "plug-ins" for IP, IPX, and AppleTalk versions of EIGRP

EIGRP uses three tables:

- The neighbor table is built from EIGRP hellos and used for reliable delivery.
- The topology table contains EIGRP routing information for best paths and loop-free alternatives.
- EIGRP places best routes from its topology table into the common routing table.

# EIGRP Messages

EIGRP uses various message types to initiate and maintain neighbor relationships, and to maintain an accurate routing table. It is designed to conserve bandwidth and router resources by sending messages only when needed and only to those neighbors that need to receive them.

## Packet Types

EIGRP uses five packet types:

- **Hello:** Identifies neighbors and serves as a keepalive mechanism
- **Update:** Reliably sends route information
- **Query:** Reliably requests specific route information
- **Reply:** Reliably responds to a query
- **ACK:** Acknowledgment

EIGRP is reliable, but hellos and ACKs are not acknowledged. The acknowledgment to a query is a reply.

If a reliable packet is not acknowledged, EIGRP periodically retransmits the packet to the nonresponding neighbor as a unicast. EIGRP has a window size of one, so no other traffic is sent to this neighbor until it responds. After 16 unacknowledged retransmissions, the neighbor is removed from the neighbor table.

## Neighbor Discovery and Route Exchange

When EIGRP first starts, it uses hellos to build a neighbor table. Neighbors are directly attached routers that have a matching AS number and k values.

(The timers don't have to agree.) The process of neighbor discovery and route exchange between two EIGRP routers is as follows:

**Step 1.** Router A sends out a hello.

**Step 2.** Router B sends back a hello and an update. The update contains routing information.

**Step 3.** Router A acknowledges the update.

**Step 4.** Router A sends its update.

**Step 5.** Router B acknowledges.

When two routers are EIGRP neighbors, they use hellos between them as keepalives. Additional route information is sent only if a route is lost or a new route is discovered. A neighbor is considered lost if no hello is received within three hello periods (called the *hold time*). The default hello/hold timers are as follows:

- 5 seconds/15 seconds for multipoint circuits with bandwidth greater than T1 and for point-to-point media
- 60 seconds/180 seconds for multipoint circuits with bandwidth less than or equal to T1

The exchange process can be viewed using **debug ip eigrp packets**, and the update process can be seen using **debug ip eigrp**. The neighbor table can be seen with the command **show ip eigrp neighbors**.

# EIGRP Route Selection

An EIGRP router receives advertisements from each neighbor listing the advertised distance (AD) and feasible distance (FD) to a route. The AD is the metric from the neighbor to the network. FD is the metric from this router, through the neighbor, to the destination network.

## EIGRP Metric

The EIGRP metric is shown in Figure 2-1.

**Figure 2-1 EIGRP Metric**

$$metric = 256(k1 \times \frac{10^7}{BW_{\min}} + \frac{k2 \times BW_{\min}}{256 - load} + k3 \times \sum delays)(\frac{k5}{reliability + k4})$$

The k values are constants. Their default values are k1 = 1, k2 = 0, k3 = 1, k4 = 0, and k5 = 0. If k5 = 0, the final part of the equation (k5 / [rel + k4]) is ignored.

$BW^{min}$ is the minimum bandwidth along the path—the choke point bandwidth.

Delay values are associated with each interface. The sum of the delays (in tens of microseconds) is used in the equation.

Taking the default k values into account, the equation simplifies to the one shown in Figure 2-2.

**Figure 2-2 EIGRP Metric Simplified**

$$metric = 256(\frac{10^7}{BW_{\min}} + \sum delays)$$

If default k values are used, this works out to be 256 (BW + cumulative delay).

Bandwidth is the largest contributor to the metric. The delay value enables us to choose a more direct path when bandwidth is equivalent.

## Diffusing Update Algorithm (DUAL)

DUAL is the algorithm used by EIGRP to choose best paths by looking at AD and FD. The path with the lowest metric is called the *successor* path. EIGRP paths with a lower AD than the FD of the successor path are guaranteed loop-free and called *feasible successors*. If the successor path is lost, the router can use the feasible successor immediately without risk of loops.

After the router has chosen a path to a network, it is *passive* for that route. If a successor path is lost and no feasible successor is identified, the router sends out queries on all interfaces in an attempt to identify an alternate path. It is *active* for that route. No successor can be chosen until the router receives a reply to all queries. If a reply is missing for 3 minutes, the router becomes *stuck in active (SIA)*. In that case, it resets the neighbor relationship with the neighbor that did not reply.

Three common causes for SIA routes are

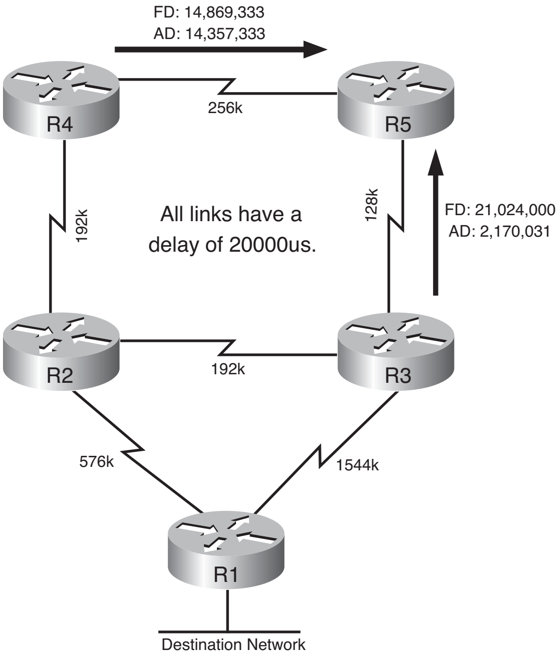
- CPU or memory usage is so high on the neighbor that it cannot process the query or reply.

- The link between the routers drops packets. Enough packets get through to maintain the neighbor relationship, but some queries or replies are dropped.
- Unidirectional link, so the router never receives packets from its neighbor.

## Route Selection Example

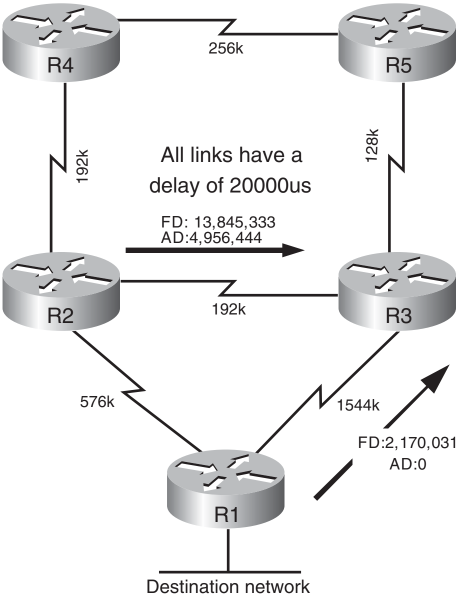
The following diagrams show EIGRP advertisements to R3 and R5 about a destination network connected to R1. In Figure 2-3, R5 chooses R4 as the successor path because it offers the lowest feasible distance. The AD from R3 indicates that passing traffic through R3 will not loop, so R3 is a feasible successor.

**Figure 2-3 EIGRP Path Selection, Part One**



How does R3 choose its path? Figure 2-4 shows the path selection process for R3.

**Figure 2-4 EIGRP Path Selection, Part Two**



R1 will be its successor because it has the lowest metric. However, no feasible successor exists because R2's AD is greater than the successor path metric. If the direct path to R1 is lost, R3 has to query its neighbors to discover an alternative path. It must wait to hear back from R2 and R5 and will ultimately decide that R2 is the new successor.

# Planning an EIGRP Implementation

When planning an EIGRP implementation, gather the following information:

- **Current network setup and future requirements:** Document the IP addressing used and the network topology, including links types, bandwidth, and utilization. A good IP addressing design allows summarization at various points in the network.
- **Network design:** Although EIGRP does not require a hierarchical network design, it can perform more efficiently within that type of network.
- **Plans for EIGRP scaling options:** These would include summarization, stub areas, and changes in interface metrics to improve bandwidth utilization.

Your final implementation plan needs to include detailed parameters such as the exact topology, IP networks to be advertised, EIGRP AS number, lists of routers to run EIGRP, and any nondefault metrics to be used. It needs to list implementation tasks for each router in the network. Finally it needs to provide verification tasks for each router such as verifying neighbors, IP routing tables, EIGRP topology tables, and network connectivity.

## Basic EIGRP Configuration

EIGRP is configured by entering router configuration mode and identifying the networks within which it should run. When setting up EIGRP, an autonomous system number must be used (7 is used in the example). Autonomous system numbers must agree for two routers to form a neighbor relationship and to exchange routes.

```
Router(config)# router eigrp 7
Router(config-router)# network 192.168.1.0
```

The wildcard mask option can be used with the network command to more precisely identify EIGRP interfaces. For instance, if a router has two interfaces—fa0/0 (192.168.1.1/27) and fa0/1 (192.168.1.33/27)—and needs to run EIGRP only on fa0/0, the following command can be used:

```
Router(config-router)# network 192.168.1.0 0.0.0.1
```

In this command, a wildcard mask of 0.0.0.1 matches only two IP addresses in network 192.168.1.0–192.168.1.0 and 192.168.1.1. Therefore, only interface fa0/0 is included in EIGRP routing.

To ensure that the correct metric is calculated, or to influence the metric, you might want to configure the bandwidth on the interface. Use the interface command:

```
R1(config)# interface s0/0/0
R1(config-if)# bandwidth kbps
```

## Creating an EIGRP Default Route

Figure 2-5 shows a simple two-router network. You can configure EIGRP on R1 to advertise a default route to R3 in these ways:

- R1 can specify a default network:

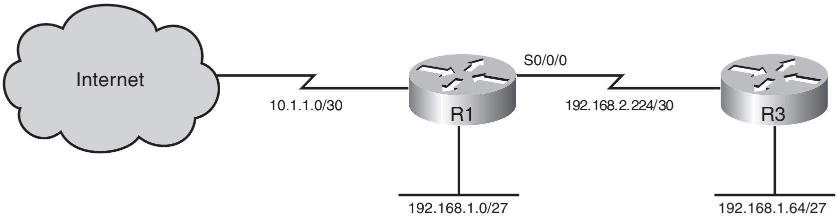
  ```
  R1(config)# ip default-network 10.0.0.0
  ```

  R3 now sees a default network with a next hop of R1.

- Create a static default route and then include network 0.0.0.0 in EIGRP:

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2
R1(config)# router eigrp 7
R1(config-router)# network 0.0.0.0
```

**Figure 2-5 EIGRP Default Route**



# Verify and Troubleshoot EIGRP

The most straightforward way to troubleshoot EIGRP is to inspect the routing table, **show ip route**. To filter the routing table and show only the routes learned from EIGRP, use the **show ip route eigrp** command. The **show ip protocols** command verifies autonomous system, timer values, identified networks, and EIGRP neighbors (routing information sources).

The command **show ip eigrp topology** shows the EIGRP topology table and identifies successors and feasible successors. Use **show ip eigrp neighbors** to verify that the correct routers are neighbors, and use **show ip eigrp traffic** to show the amount and types of EIGRP messages. The command **show ip eigrp interfaces** lists the interfaces participating in EIGRP and any neighbors found out those interfaces, along with some other statistics.

# EIGRP Across a WAN

EIGRP can be used across many types of WAN links. This section examines how it operates over some of them.

## EIGRP over EoMPLS

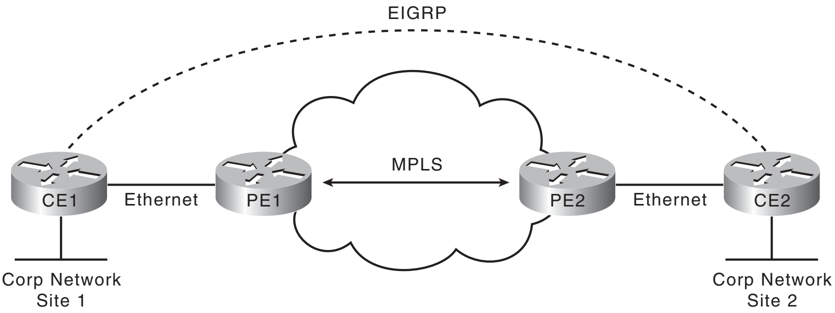
MPLS can provide either a Layer 2 or a Layer 3 connection. In MPLS terminology, your WAN edge routers are called CE (customer edge) routers, and the ISP's WAN edge routers are called PE (provider edge) routers. Within the ISP's network are P (provider) routers, but they should not be visible to the CE.

Ethernet over MPLS (EoMPLS) leverages Any Transport over MPLS (AToM) to provide a Layer 2 connection such as Metro Ethernet. With EoMPLS, the CE routers appear to have a point-to-point Ethernet connection across the WAN. In reality, each CE router has an Ethernet connection to its local PE router.

Figure 2-6 shows how this works. The PE1 router receives Ethernet frames from CE1, encapsulates them into an MPLS packet, and then forwards them across the WAN to PE2, which is the local router connected to CE2. PE2 decapsulates the packet, rebuilds the Ethernet frame, and sends it to the CE2.

It is important to understand that CE1 and CE2 build an EIGRP neighbor relationship with each other. The ISP routers are not involved in routing with the CE routers. Additionally, the PE routers do not learn any MAC addresses or participate in Spanning Tree.

**Figure 2-6 Using EIGRP with EoMPLS**



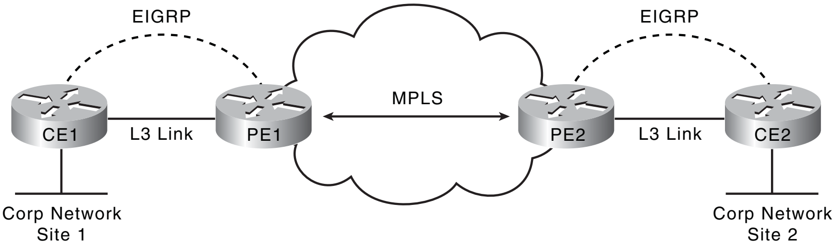
## EIGRP over MPLS

PE routers are involved in routing when you use EIGRP over Layer 3 MPLS VPNs, however. The connection between the CE and PE routers is a Layer 3 connection. Each connected PE and CE router are EIGRP neighbors. The PE router is just another neighbor to the CE router; it is not aware of the MPLS network or the ISP's P routers.

In Figure 2-7, CE1 creates an EIGRP neighbor relationship with PE1. CE1 sends routing updates about its networks to PE1, which installs the routes in the correct Virtual Routing and Forwarding (VRF) table and then transmits them across the WAN as MPLS packets to PE2. PE2 is an EIGRP neighbor to CE2, so it forwards the route advertisements as normal EIGRP updates.

When using EIGRP over MPLS, the customer and the provider need to use the same basic EIGRP configuration such as AS number and authentication.

**Figure 2-7 EIGRP with MPLS**



## EIGRP over Frame Relay

One issue with using EIGRP over Frame Relay is that one physical interface can support multiple logical connections, each identified by a Data Link Connection Identifier (DLCI). These are Layer 2 connections and must be mapped to a Layer 3 neighbor IP address. This mapping can be done either dynamically or statically. Multipoint interfaces are used in partial and full mesh topologies.

Dynamic mapping uses Inverse ARP. Routers form EIGRP neighbor adjacencies only with routers that they connect to via a Frame Relay virtual circuit (VC). Static mapping requires manual configuration under each interface but enables routers without VC connections to become neighbors. The static mapping command is given under interface configuration mode:

```
frame-relay map ip remote-ip-address local-dlci broadcast
```

The **broadcast** keyword is required because Frame Relay is, by default, a nonbroadcast medium. Static mapping can be used with both physical multipoint interfaces and subinterfaces. Note that a multipoint interface stays up if one DLCI is active, so a neighbor loss might not be detected until the hold timer expires.

Frame Relay can emulate physical point-to-point links by using point-to-point subinterfaces. This is used in a hub-and-spoke topology. Neighbor loss is detected much more quickly on point-to-point links for two reasons:

- The default timers are shorter, 5 second hold timer and 15 second dead timer.
- The subinterface goes down when its associated DLCI goes down.

## WAN Bandwidth

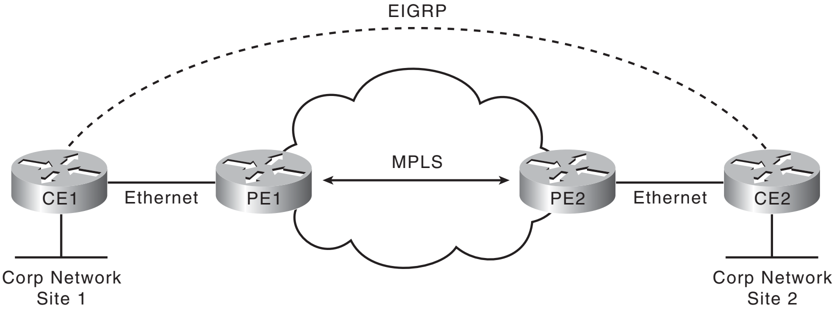
By default, EIGRP limits itself to bursting to half the link bandwidth. This limit is configurable per interface using the **ip bandwidth-percent** command. The following example assumes EIGRP AS 7 and limits EIGRP to one quarter of the link bandwidth:

```
Router(config)# int s0/0/0
Router(config-if)# ip bandwidth-percent eigrp 7 25
```

The real issue with WAN links is that the router assumes that each link has 1544 kbps bandwidth. If interface Serial0/0/0 is attached to a 128 k fractional T1, EIGRP assumes it can burst to 768 k and could overwhelm the line. This is rectified by correctly identifying link bandwidth:

```
Router (config)# int serial 0/0/0
Router (config-if)# bandwidth 128
```

**Figure 2-8 EIGRP with Frame Relay**



In this example, R1 has a 256 kbps connection to the Frame Relay network and two permanent virtual circuits (PVCs) with committed information rates (CIR) of 128 Kpbs and 64 Kbps. EIGRP divides the interface bandwidth evenly between the number of neighbors on that interface. What value should be used for the interface bandwidth in this case? The usual suggestion is to use the CIR, but the two PVCs have different CIRs. You can use the bandwidth-percent command to allow SNMP reporting of the true bandwidth value, while adjusting the interface burst rate to 25 percent, or 64 kbps.

```
R1(config)# int serial 0/0/0
R1 (config-if)# bandwidth 256
R1 (config-if)# ip bandwidth-percent eigrp 7 25
```

A better solution is to use point-to-point subinterfaces and identify bandwidth separately. In the following example, s0/0/0.1 bursts to 64 k, and s0/0/0.2 bursts to 32 k, using EIGRP's default value of half the bandwidth.

```
R1(config)# int serial 0/0/0.1 point-to-point
R1(config-if)# bandwidth 128
R1(config-if)# frame-relay interface-dlci 100
!
R1(config)# int serial 0/0/0.2 point-to-point
R1(config-if)# bandwidth 64
R1(config-if)# frame-relay interface-dlci 101
```

In cases where the hub interface bandwidth is oversubscribed, it might be necessary to set bandwidth for each subinterface arbitrarily low and then specify an EIGRP bandwidth percent value over 100 to allow EIGRP to use half the PVC bandwidth.

# Customizing the EIGRP Configuration

EIGRP provides some ways to customize its operation, such as passive interface, unicast neighbors, route summarization, unequal-metric load balancing, and authentication. This section describes how to configure these.

## Passive Interface

The **passive-interface** command prevents either routing updates or hello messages from being sent out an interface. RIP does not send updates when it enabled; EIGRP and OSPF do not send hellos, and thus they don't discover neighbors or form an adjacency out that interface. To disable the protocol on one interface, use the routing protocol configuration command **passive-interface** *interface*. To turn off the protocol on all interfaces, use **passive-interface default**. You can then use **no passive-interface** *interface* for the ones that should run the protocol, as shown here:

```
Router(config)# router eigrp 7
Router(config-router)# passive-interface default
Router(config-router)# no passive-interface s0/0/0
```

## Unicast Neighbors

EIGRP usually uses a multicast to IP address 224.0.0.10 for its messages. You can configure it to use a unicast address instead with the routing protocol configuration command **neighbor** *ip-address.* The IP address must be in the same subnet as one of the router's own interfaces.

## Summarization

EIGRP defaults to automatically summarizing at classful network boundaries. Automatic summarization is usually disabled using the following command:

```
Router(config-router)# no auto-summary
```

Summaries can be produced manually on any interface. When a summary is produced, a matching route to null0 also becomes active as a loop prevention mechanism. Configure a summary route out a particular interface using the **ip summary-address eigrp** *autonomous_system* command. The following example advertises a default route out FastEthernet0/1 and the summary route 172.16.104.0/22 out Serial0/0/0 for EIGRP AS 7.

```
Router(config)# int fa0/1
Router(config-if)# ip summary-address eigrp 7 0.0.0.0 0.0.0.0
!
Router(config)# int s0/0/0
Router(config-if)# ip summary-address eigrp 7 172.16.104.0
  255.255.252.0
```

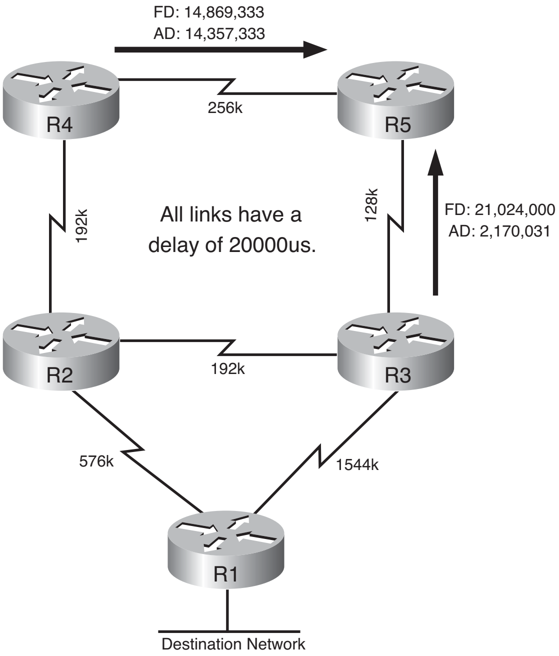
# Load Balancing

EIGRP, like most IP routing protocols, automatically load balances over equal metric paths. What makes EIGRP unique is that you can configure it to proportionally load balance over *unequal* metric paths. The **variance** command is used to configure load balancing over up to six loop-free paths with a metric lower than the product of the variance and the best metric. Figure 2-9 shows routers advertising a path to the network connected to R1.

By default, R5 uses the path through R4 because it offers the lowest metric (14,869,333). To set up unequal cost load balancing, assign a variance of 2 under the EIGRP process on R5, which multiplies the best metric of 14,869,333 by 2 to get 29,738,666. R5 then uses all loop-free paths with a metric less than 29,738,666, which includes the path through R3. By default, R5 load balances over these paths, sending traffic along each path in proportion to its metric.

```
R5(config)# router eigrp 7
R5(config-router)# variance 2
```

**Figure 2-9 EIGRP Unequal-cost Load Balancing**



# EIGRP Authentication

By default, no authentication is used for any routing protocol. Some protocols, such as RIPv2, IS-IS, and OSPF, can be configured to do simple password authentication between neighboring routers. In this type of authentication, a clear-text password is used. EIGRP does not support simple authentication. However, it can be configured to authenticate each packet exchanged using an MD5 hash created from a preconfigured, shared password. This is more secure than clear text because only the message digest is exchanged, not the password. The password is called the *key*.

EIGRP authenticates each of its packets and verifies the source of each routing update by including the hash in each one. If the hash value does not match, the packet is silently dropped.

To implement EIGRP authentication, first create a plan:

- Look at the current configuration to determine the AS number and interfaces where it will be configured.
- Decide the authentication type. (For EIGRP this must be MD5.)
- Decide the key strings, and how many keys will be used.
- Optionally decide the key lifetimes.

To configure the router for EIGRP authentication, follow these steps:

**Step 1.** Configure a key chain to group the keys.

**Step 2.** Configure one or more keys within that key chain. The router checks all inbound packets against the list of keys and uses the first valid one it finds.

**Step 3.** Configure the password or authentication string for that key. Repeat Steps 2 and 3 to add more keys if desired.

**Step 4.** Optionally configure a lifetime for the keys within that key chain. If you do this, be sure that the time is synchronized between the two routers.

**Step 5.** Enable authentication and assign a key chain to an interface.

**Step 6.** Designate MD5 as the type of authentication.

Example 2-1 shows a router configured with EIGRP authentication. It shows configuring a lifetime for packets sent using key 1 that starts at 10:15 and lasts for 300 seconds. It also shows configuring a lifetime for packets received using key 1 that starts at 10:00 and lasts until 10:05. Router clocks must be synchronized when using lifetimes, so use an NTP server.

**Example 2-1 Configuring EIGRP Authentication**

```
Router(config)# key chain RTR_Auth
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string mykey
Router(config-keychain-key)# send-lifetime 10:15:00 300
Router(config-keychain-key)# accept-lifetime 10:00:00 10:05:00
!
Router(config)# interface s0/0/0
Router(config-if)# ip authentication mode eigrp 10 md5
Router(config-if)# ip authentication key-chain eigrp 10 RTR_Auth
```

Verify your configuration with the **show key chain** command. **show ip eigrp neighbors** is also useful, as no neighbor relationship will be formed if authentication fails. Using the **debug eigrp packets** command should show packets containing authentication information sent and received, and it enables you to troubleshoot configuration issues. The debug output lists an authentication mismatch message if authentication does not succeed.

# EIGRP Scalability

Four factors influence EIGRP's scalability:

- The number of routes that must be exchanged
- The number of routers that must know of a topology change
- The number of alternate routes to a network
- The number of hops from one end of the network to the other (topology depth)

To improve scalability, summarize routes when possible, try to have a network depth of no more than seven hops, and limit the scope of EIGRP queries.

## EIGRP Stub

*Stub routing* is one way to limit queries. A stub router is one that is connected to no more than two neighbors and should never be a transit router. This feature is commonly used in a hub-and-spoke topology. When a router is configured as an EIGRP stub, it notifies its neighbors. The neighbors then do not query that router for a lost route. An EIGRP stub router still receives all routes from its neighbors by default.

Under router configuration mode, use the command **eigrp stub [receive-only|connected|static|summary|redistributed]**. Table 2-1 lists each of the command options and their affect.

**Table 2-1 eigrp stub Command Options**

| Command Option | Affect |
|---|---|
| `receive-only` | Prevents the router from advertising any networks, including its own. Cannot be combined with any other option. |
| `connected` | Enables the router to advertise connected routes. These must either be included in a network statement or redistributed into EIGRP. Enabled by default. |
| `static` | Enables the router to advertise static routes. They must be redistributed into EIGRP before they will be advertised. |
| `summary` | Enables the router to advertise summary routes, both those created manually and automatically. Enabled by default. |
| `redistributed` | Allows the router to advertise routes redistributed into EIGRP from another protocol or AS. |

## Active Process Enhancement

The Active Process Enhancement enables routers to use *SIA-Queries* and *SIA-Replies* to prevent the loss of a neighbor unnecessarily during SIA conditions. A router sends its neighbor a SIA-Query after no reply to a normal query. If the neighbor responds with a SIA-Reply, the router does not terminate the neighbor relationship after 3 minutes, because it knows the neighbor is available.

## Graceful Shutdown

*Graceful shutdown* is another feature that speeds network convergence. Whenever the EIGRP process is shut down, the router sends a “goodbye” message to its neighbors. Ironically, the goodbye message is sent in a “hello” packet. The neighbors can then immediately recalculate any paths that used the router as the next hop, rather than waiting for the hold timer to expire.

# CHAPTER 3

# OSPF

## OSPF Overview

OSPF is an open-standard, classless routing protocol that converges quickly and uses cost as a metric. (Cisco IOS automatically associates cost with bandwidth.)

OSPF is a link-state routing protocol and uses Dijkstra's Shortest Path First (SPF) algorithm to determine its best path to each network. The first responsibility of a link-state router is to create a database that reflects the structure of the network. Link state routing protocols learn more information on the structure of the network than other routing protocols and thus can make more informed routing decisions.

OSPF routers exchange Hellos with each neighbor, learning Router ID (RID) and cost. Neighbor information is kept in the adjacency database.

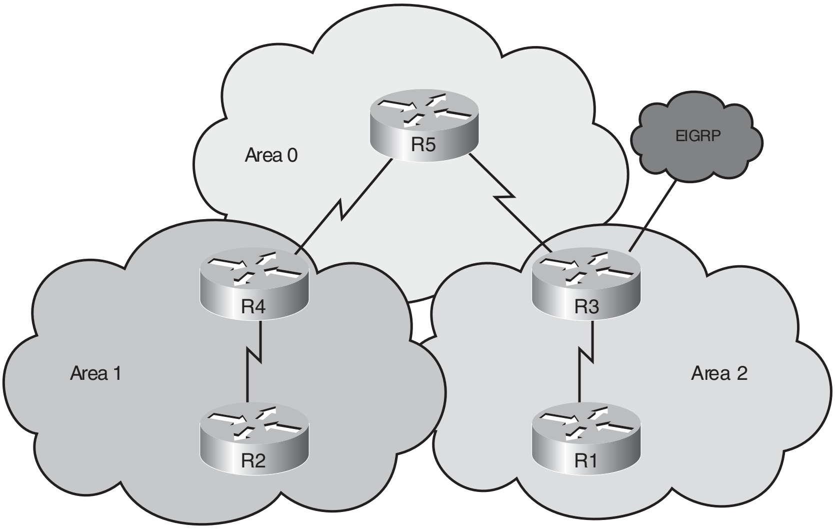
The router then constructs the appropriate Link State Advertisements (LSA), which include information such as the RIDs of, and cost to, each neighbor. Each router in the routing domain shares its LSAs with all other routers. Each router keeps the complete set of LSAs in a table—the Link State Database (LSDB).

Each router runs the SPF algorithm to compute best paths. It then submits these paths for inclusion in the routing table, or forwarding database.

## OSPF Network Structure

OSPF routing domains are broken up into areas. An OSPF network must contain an area 0 and might contain other areas. The SPF algorithm runs within an area, and interarea routes are passed between areas. A two-level hierarchy to OSPF areas exists; area 0 is designed as a transit area, and other areas should be attached directly to area 0 and only to area 0. The link-state database must be identical for each router in an area. OSPF areas typically contain a maximum of 50 routers to 100 routers, depending on network volatility. 3-1 shows a network of five routers that has been divided into three areas: area 0, area 1, and area 2.

**Figure 3-1 OSPF Areas**



Dividing an OSPF network into areas does the following:

- Minimizes the number of routing table entries
- Contains LSA flooding to a reasonable area
- Minimizes the impact of a topology change
- Enforces the concept of a hierarchical network design

Following are several types of areas:

- **Backbone area:** Area 0, which is attached to every other area.
- **Regular area:** Nonbackbone area; its database contains both internal and external routes.
- **Stub area:** It's database contains only internal routes and a default route.
- **Totally Stubby Area:** Cisco proprietary area designation. Its database contains routes only for its own area and a default route.
- **Not-so-stubby area (NSSA):** Its database contains internal routes, routes redistributed from a connected routing process, and optionally a default route.
- **Totally NSSA:** Cisco proprietary area designation. Its database contains only routes for its own area, routes redistributed from a connected routing process, and a default route.

OSPF defines router roles as well. One router can have multiple roles:

- An internal router has all interfaces in one area. In Figure 3-1, R1, R2, and R5 are all internal area routers. They maintain a link-state database for their own area only.
- Backbone routers have at least one interface assigned to area 0. R3, R4, and R5 are backbone routers.
- An Area Border Router (ABR) has interfaces in two or more areas. In Figure 3-1, R3 and R4 are ABRs. ABRs separates LSA flooding areas, can summarize area routes, and can source default routes. They maintain a link-state database for each area to which they are connected.
- An Autonomous System Boundary Router (ASBR) has interfaces inside and outside the OSPF routing domain. In Figure 3-1, R3 additionally functions as an ASBR because it has an interface in an EIGRP routing domain.

## OSPF Metric

By default, Cisco assigns a cost to each interface that is inversely proportional to 100 Mbps (100,000,000 bps). The cost for each link is then accrued as the route advertisement for that link traverses the network. 3-2 shows the default OSPF formula.

**Figure 3-2 OSPF Cost Formula**

$$\text{Cost}=\frac{100\ \text{Mbps}}{\text{Bandwidth}}$$

The default formula doesn't differentiate between interfaces with speeds faster than 100 Mbps. It assigns the same cost to a Fast Ethernet interface and a Gigabit Ethernet interface, for example. In such cases, the cost formula can be adjusted using the **auto-cost** command under the OSPF routing process. Values for bandwidth (in kbps) up to 4,294,967 are permitted (1 Gbps is shown in the following line):

```
Router(config-router)# auto-cost reference-bandwidth 1000
```

The cost can also be manually assigned under the interface configuration mode. The cost is a 16-bit number, so it can be any value from 1 to 65,535.

```
Router(config-if)# ip ospf cost 27
```

# Link State Advertisements (LSA)

Each router maintains a database, called the *link-state database (LSDB)*, containing the latest received LSAs. A separate LSDB is maintained for each area connected to the router.

## LSA Operation

Each LSA is numbered with a sequence number, and a timer is run to age out old LSAs. The default timer is 30 minutes.

When a LSA is received, it's compared to the LSDB. If it is new, it is added to the database, and the SPF algorithm is run. If it is from a Router ID that is already in the database, the sequence number is compared, and older LSAs are discarded. If it is a new LSA, it is incorporated in the database, and the SPF algorithm is run. If it is an older LSA, the newer LSA in memory is sent back to whoever sent the old one.

OSPF sequence numbers are 32 bits. The first legal sequence number is 0x80000001. Larger numbers are more recent. The sequence number changes only under two conditions:

- The LSA changes because a route is added or deleted.
- The LSA ages out. (LSA updates are flooded within the area every half hour, even if nothing changes.)

The command **show ip ospf database** shows the age (in seconds) and sequence number for each router.

## LSDB Overload Protection

Because each router sends an LSA for each link, routers in large networks might receive—and must process—numerous LSAs. This can tax the router's CPU and memory resources, and adversely affect its other functions. LDSB overload protection monitors the number of LSAs received and placed into the LSDB. If the specified threshold is exceeded for one minute, the router enters the "ignore" state by dropping all adjacencies and clearing the OSPF database. The router resumes OSPF operations after things have been normal for a specified period. Be careful because this feature disrupts routing when invoked.

Configure LSDB overload protection with the OSPF router process command **max-lsa** *maximum-number* [*threshold-percentage*] [**warningonly**][**ignore-time** *minutes*] [**ignore-count** *number*] [**reset-time** *minutes*].

# LSA Types

OSPF uses different types of LSAs to advertise different types of routes, such as internal area or external routing domain. Many of these are represented in the routing table with a distinctive prefix. Table 3-1 describes these LSA types.

**Table 3-1 OSPF LSA Types**

| Type | Description | Routing Table Symbol |
|---|---|---|
| 1 | Router LSA. Advertises intra-area routes. Generated by each OSPF router. Flooded only within the area. | O |
| 2 | Network LSA. Advertises routers on a multiaccess link. Generated by a DR. Flooded only within the area. | O |
| 3 | Summary LSA. Advertises interarea routes. Generated by an ABR. Flooded to adjacent areas. | O IA |
| 4 | Summary LSA. Advertises the route to an ASBR. Generated by an ABR. Flooded to adjacent areas. | O IA |
| 5 | External LSA. Advertises routes in another routing domain. Generated by an ASBR. Flooded to adjacent areas.<br>E1–metric increases at each router as it is passed through the network.<br>O E2–metric does not increase (this is the default). | O |
| 6 | Multicast LSA. Used in multicast OSPF operations. | |
| 7 | Not-so-stubby area (NSSA) LSA. Advertises routes in another routing domain. Generated by an ASBR within a not-so-stubby area.<br>N1–metric increases as it is passed through the network.<br>O N2–metric does not increase (default). | O |
| 8 | External attributes LSA. Used in OSPF and BGP interworking. | |
| 9,10,11 | Opaque LSAs. Used for specific applications, such as OSPF and MPLS interworking. | |

# OSPF Operation

OSPF uses several different message types to establish and maintain its neighbor relationships and to maintain correct routing information. When preparing for the exam, be sure you understand each OSPF packet type and the OSPF neighbor establishment procedure.

## OSPF Packets

OSPF uses five packet types. It does not use UDP or TCP for transmitting its packets. Instead, it runs directly over IP (IP protocol 89) using an OSPF header. One field in this header identifies the type of packet being carried. The five OSPF packet types follow:

- **Hello:** Identifies neighbors and serves as a keepalive.
- **Link State Request (LSR):** Request for a Link State Update (LSU). Contains the type of LSU requested and the ID of the router requesting it.
- **Database Description (DBD):** A summary of the LSDB, including the RID and sequence number of each LSA in the LSDB.
- **Link State Update (LSU):** Contains a full LSA entry. An LSA includes topology information; for example, the RID of this router and the RID and cost to each neighbor. One LSU can contain multiple LSAs.
- **Link State Acknowledgment (LSAck): Acknowledges all other OSPF packets (except Hellos).**

OSPF traffic is multicast to either of two addresses: 224.0.0.5 for all OSPF routers or 224.0.0.6 for all OSPF DRs.

## OSPF Neighbor Relationships

OSPF routers send out periodic multicast packets to introduce themselves to other routers on a link. They become neighbors when they see their own router ID included in the Neighbor field of the Hello from another router. Seeing this tells each router that they have bidirectional communication. In addition, two routers must be on a common subnet for a neighbor relationship to be formed. (Virtual links are sometimes an exception to this rule.)

Certain parameters within the OSPF Hellos must also match for two routers to become neighbors. They include

- Hello/dead timers
- Area ID
- Authentication type and password
- Stub area flag

OSPF routers can be neighbors without being adjacent. Only adjacent neighbors exchange routing updates and synchronize their databases. On a point-to-point link, an adjacency is established between the two routers when they can communicate. On a multiaccess link, each router establishes an adjacency only with the DR and the backup DR (BDR).

Hellos also serve as keepalives. A neighbor is considered lost if no Hello is received within four Hello periods (called the dead time). The default Hello/dead timers are as follows:

- 10 seconds/40 seconds for LAN and point-to-point interfaces
- 30 seconds/120 seconds for nonbroadcast multiaccess (NBMA) interfaces

## Establishing Neighbors and Exchanging Routes

The process of neighbor establishment and route exchange between two OSPF routers is as follows:

**Step 1.** **Down state:** OSPF process not yet started, so no Hellos sent.

**Step 2.** **Init state:** Router sends Hello packets out all OSPF interfaces.

**Step 3.** **Two-way state:** Router receives a Hello from another router that contains its own router ID in the neighbor list. All other required elements match, so routers can become neighbors.

**Step 4.** **Exstart state:** If routers become adjacent (exchange routes), they determine which one starts the exchange process.

**Step 5.** **Exchange state:** Routers exchange DBDs listing the LSAs in their LSD by RID and sequence number.

**Step 6.** **Loading state:** Each router compares the DBD received to the contents of its LS database. It then sends a LSR for missing or outdated LSAs. Each router responds to its neighbor's LSR with a Link State Update. Each LSU is acknowledged.

**Step 7.** **Full state:** The LSDB has been synchronized with the adjacent neighbor.

# Planning for OSPF

Planning an OSPF implementation is more stringent than planning for EIGRP, because OSPF has specific network design requirements. Gather the following information:

- **Current network setup and future requirements:** Document the IP addressing used and the network topology, including links types, bandwidth, and utilization. Document the current router utilization. Create an IP addressing design that allows summarization at the ABRs.

- **Network design:** OSPF requires a hierarchical network design. You must create a backbone area (area 0) and normal areas. Area 0 must be contiguous. Normal areas must be connected to area 0 either directly or via a virtual link. Decide where the area boundaries should fall. Ensure that the normal areas have sufficient connectivity to area 0, and that all ABRs have the resources to handle the OSPF traffic in addition to the user traffic.
- **Plans for OSPF scaling options:** These would include summarization and stub areas.

Your final implementation plan needs to include detailed parameters such as the exact topology, IP networks to be advertised, OSPF process number, lists of routers to run OSPF, and any changes needed to the default interface metric. It needs to list implementation tasks for each router in the network. Finally it needs to provide verification tasks for each router such as verifying neighbors, IP routing tables, OSPF topology tables, and network connectivity. Document the new network configurations.

## Basic OSPF Configuration

OSPF is configured by entering router configuration mode and identifying the range of interface addresses on which it should run and the areas they are in. When setting up OSPF, a process ID must be used (8 is used in the example), but the process ID does not need to agree on different OSPF devices for them to exchange information. The network statement uses a wildcard mask and can specify any range from a single address to all addresses. Unlike EIGRP, the wildcard mask is not optional. The following example shows a router configured as an ABR. Interfaces falling with the 192.168.1.0 network are placed in area 0, and interfaces falling within the 172.16.1.0 network are placed in area 1.

```
Router(config)# router ospf 8
Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
Router(config-router)# network 172.16.1.0 0.0.0.255 area 1
```

Alternatively, you can enable OSPF directly on an interface, rather than using a network statement. This is especially helpful on unnumbered interfaces and enables more granular control over which interfaces run OSPF.

```
Router(config)# int s 0/0/0
Router(config-if)# ip ospf 8 area 0
```

The **ip ospf area** interface command takes precedence over a **network** command.

## Router ID

The SPF algorithm maps the shortest path between a series of nodes. This causes an issue with IP because an IP router is not identified by a single IP address; its interfaces are. For this reason, a single IP address is designated as the “name” of the router: the Router ID (RID).

By default, the RID is the highest loopback IP address. If no loopback addresses are configured, the RID is the highest IP address on an active interface when the OSPF process is started. The RID is selected when OSPF starts and—for reasons of stability—is not changed until OSPF restarts. The OSPF process can be restarted by rebooting or by using the command **clear ip ospf process**. Either choice affects routing in your network for a period of time and should be used only with caution.

A loopback interface is a virtual interface, so it is more stable than a physical interface for RID use. A loopback address is configured by creating an interface and assigning an IP address.

```
Router(config)# interface loopback0
Router(config-if)# ip address 10.0.0.1 255.255.255.255
```

The loopback address does not need to be included in the OSPF routing process, but if you advertise it, you can ping or trace to it. This can help in troubleshooting.

A way to override the default RID selection is to statically assign it using the OSPF **router-id** command. Router ID is typically statically assigned for predictability should a process be forced to unexpectedly restart.

```
Router(config)# router ospf 8
Router(config-router)# router-id 10.0.0.1
```

## Verify and Troubleshoot OSPF

The neighbor initialization process can be viewed using the **debug ip ospf adjacencies** command. The neighbor table can be seen with **show ip ospf neighbors**, which also identifies adjacency status and reveals the designated router and backup designated router. Use the **debug ip ospf packet** command to view all OSPF packets in real time.

Often, the first place OSPF issues are noticed is when inspecting the routing table: **show ip route**. To filter the routing table and show only the routes learned from OSPF, use **show ip route ospf**.

The command **show ip protocols** offers a wealth of information for any routing protocol issue. Use this command to verify parameters, timer values, identified networks, and OSPF neighbors (routing information sources).

Use **show ip ospf** to verify the RID, timers, and counters. Because wildcard masks sometimes incorrectly group interfaces to areas, another good place to check is **show ip ospf interface**. This shows the interfaces on which OSPF runs and their current correct assigned area.

## OSPF Network Types

The SPF algorithm builds a directed graph—paths made up of a series of points connected by direct links. One of the consequences of this directed-graph approach is that the algorithm has no way to handle a multiaccess network, such as an Ethernet VLAN. The solution used by OSPF is to elect one router, called the Designated Router (DR), to represent the entire segment. Point-to-point links fit the SPF model perfectly and don't need any special modeling method. On a point-to-point link, no DR is elected, and all traffic is multicast to 224.0.0.5.

OSPF supports five network types:

- **NBMA:** Default for multipoint serial interfaces. RFC-compliant mode that uses DRs and requires manual neighbor configuration.
- **Point-to–multipoint (P2MP):** Doesn't use DRs so adjacencies increase logarithmically with routers. Resilient RFC-compliant mode that automatically discovers neighbors.
- **Point-to-multipoint nonbroadcast (P2MNB):** Proprietary mode that is used on Layer 2 facilities where dynamic neighbor discovery is not supported. Requires manual neighbor configuration.
- **Broadcast:** Default mode for LANs. Uses DRs and automatic neighbor discovery. Proprietary when used on WAN interface.
- **Point-to–point (P2P):** Proprietary mode that discovers neighbors and doesn't require a DR.

If the default interface type is unsatisfactory, you can statically configure it with the command **ip ospf network** under interface configuration mode:

```
Router(config-if)# ip ospf network point-to-multipoint
```

When using the NBMA or P2MP nonbroadcast mode, neighbors must be manually defined under the routing process:

```
Router(config-router)# neighbor 172.16.0.1
```

The command **show ip ospf interface** displays the network type for each link.

## Designated Routers

On a multiaccess link, one of the routers is elected as a DR and another as a backup DR (BDR). All other routers on that link become adjacent only to the DR and BDR, not to each other. (They stop at the two-way state.) The DR is responsible for creating and flooding a network LSA (type 2) advertising the multiaccess link. NonDR (DROTHER) routers communicate with DRs using the IP address 224.0.0.6. The DRs use IP address 224.0.0.5 to pass information to other routers.

The DR and BDR are elected as follows:

**Step 1.** A router starting the OSPF process listens for OSPF Hellos. If none are heard within the dead time, it declares itself the DR.

**Step 2.** If Hellos from any other routers are heard, the router with the highest OSPF priority is elected DR, and the election process starts again for BDR. A priority of zero removes a router from the election.

**Step 3.** If two or more routers have the same OSPF priority, the router with the highest RID is elected DR, and the election process starts again for BDR.

After a DR is elected, elections do not take place again unless the DR or BDR are lost. Because of this, the DR is sometimes the first device that comes online with a nonzero priority.

The best way to control DR election is to set OSPF priority for the DR and BDR for other routers. The default priority is one. A priority of 0 means that a router cannot act as DR or BDR; it can be a DROTHER only. Priority can be set with the **ip ospf priority** command in interface configuration mode.

```
Router(config)# int fa 0/1
Router(config-if)# ip ospf priority 2
```

## Nonbroadcast Multiaccess (NBMA) Networks

Routing protocols assume that multiaccess links support broadcast and have full-mesh connectivity from any device to any device. In terms of OSPF, this means the following:

- All Frame Relay or ATM maps should include the broadcast attribute.
- The DR and BDR should have full virtual circuit connectivity to all other devices.
- Hub-and-spoke environments should either configure the DR as the hub or use point-to-point subinterfaces, which require no DR.
- Partial-mesh environments should be configured using point-to-point subinterfaces, especially when no single device has full connectivity to all other devices. If there is a subset of the topology with full connectivity, that subset can use a multipoint subinterface.
- Full-mesh environments can be configured using the physical interface, but often logical interfaces are used to take advantage of the other benefits of subinterfaces.
- It might be necessary to statically identify neighbor IP addresses.

## OSPF over Layer 2 and Layer 3 MPLS

Layer 2 and Layer 3 MPLS-based solutions were described in Chapter 2, “EIGRP.” A Layer 2 connection uses EoMPLS, and OSPF operates just as it would on any other Ethernet network. It forms a neighbor relationship with the CE router across the WAN, and they elect a DR and BDR. The OSPF network type is Multiaccess Broadcast.

A Layer 3 MPLS VPN requires that the CE routers form an OSPF neighbor relationship with their connected PE router. The PE router appears to the enterprise as just another router within their network. The OSPF network type is determined by the type of link between the CE and PE. Carefully consider your area design when using this type of WAN.

# Advanced OSPF Configuration

OSPF provides many different ways to customize its operation to fit your network needs. This section discusses route summarization, passive interfaces, default routes, stub areas, and virtual links.

## OSPF Summarization

Summarization helps all routing protocols scale to larger networks, but OSPF especially benefits because its processes tax the memory and CPU resources of the routers. The SPF algorithm consumes all CPU resources when it runs. Summarization prevents topology changes from being passed outside an area and thus saves routers in other areas from having to run the SPF algorithm. OSPF's multiple databases use more memory the larger they are. Summarization decreases the number of routes exchanged, and thus the size of the databases. It localizes the impact of a topology change. OSPF can produce summaries within a classful network (VLSM) or summaries of blocks of classful networks (CIDR). There are two types of summarizations:

- **Inter-area (LSA type 3) route summarizations** are created on the ABR under the OSPF routing process using the **area range** command. A summary route will be advertised as long as at least one subnet within the summary is active in the area. The summary route's metric is the lowest cost route within the summary range. The router automatically creates a static route for the summary, pointing to Null0.

  The following command advertises 172.16.0.0/12 from area 1:

  ```
  Router(config-router)# area 1 range 172.16.0.0 255.240.0.0
  ```

- **External (LSA type 5) route summarization** is done on an ASBR using the **summary-address** command under the OSPF routing process. It can also be done on the ABR of a NSSA to summarize type 7 routes before advertising them as type 5. The router automatically creates a static route for the summary, pointing to Null0. The following example summarizes a range of external routes to 192.168.0.0/16 and injects a single type 5 route into OSPF.

  ```
  Router(config-router)#summary-address 192.168.0.0
    255.255.0.0
  ```

## Passive Interface

The **passive-interface** command prevents OSPF from sending Hello messages out an interface. Thus an OSPF router does not discover neighbors or form an adjacency out that interface. To disable the protocol on one interface, use the routing protocol configuration command **passive-interface** *interface*. To turn off the protocol on all interfaces, use **passive-interface default**. You can then use **no passive-interface** *interface* for the ones that should run the protocol. See Chapter 2 for a configuration example.

## OSPF Default Routes

The default route is a special type of summarization; it summarizes all networks down to one route announcement. This provides the ultimate benefit of summarization by reducing routing information to a minimum:

- Routers have a smaller routing table.
- Less use of router resources to advertise multiple routes.
- Routers do not need to keep information on external routes.

A default route is injected into OSPF as a type 5 route. There are several ways to use the router IOS to place a default route into OSPF. The best-known way is to use the **default-information** command under the OSPF routing process. This command, without the keyword **always**, advertises a default route learned from another source (such as a static route) into OSPF. If the **always** keyword is present, OSPF advertises a default even if that route does not already exist in the routing table. The **metric** keyword sets the starting metric for this route.

```
Router(config-router)# default-information originate [always]
[metric metric]
```

Alternatively, a default summary route can also be produced using the **summary-address** command or the **area range** command. These commands can cause the router to advertise a default route pointing to itself.

Reducing routing information in nonbackbone areas is a common requirement because these routers are typically the most vulnerable in terms of processor and speed, and the links that connect them usually have the least bandwidth. A specific concern is that an area will be overwhelmed by external routing information.

## Stub and Not-So-Stubby Areas

Another way to reduce the route information advertised is to make an area a *stub* area. Configuring an area as a stub area forces its ABR to drop all external (type 5) routes and replaces them with a default route. To limit routing information even more, an area can be made *totally stubby* using the **no-summary** keyword on the ABR only. In that case, all interarea and external routes are dropped by the ABR and replaced by a default route. The default route starts with a cost of 1; to change it, use the **area default-cost** command. The example that follows shows area 2 configured as a totally stubby area, and the default route injected with a cost of 5:

```
Router(config-router)# area 2 stub no-summary
Router(config-router)# area 2 default-cost 5
```

Stub areas are attractive because of their low overhead. They do have some limitations, including the following:

- Stub areas can’t include a virtual link.
- Stub areas can’t include an ASBR.
- Stubbiness must be configured on all routers in the area.
- Area 0 cannot be a stub area.

Another kind of stub area is a *not-so-stubby area* (NSSA). NSSA is like a stub or totally stub area but enables an ASBR within the area. External routes are advertised as type 7 routes by the ASBR. The ABR converts them to type 5 external routes when it advertises them into adjacent areas. NSSA is configured with the **area nssa** command under the OSPF routing process. The **no-summary** keyword on the ABR configures the area as *totally NSSA*; this is a Cisco proprietary feature. By default, the ABR does not inject a default route back into an NSSA area. Use the **default-information-originate** keyword on the ABR or ASBR to create this route.

```
Router(config-router)# area 7 nssa [no-summary] [default-
 information-originate]
```

## Virtual Links

OSPF requires that all areas be connected to area 0 and that area 0 must be contiguous. When this is not possible, you can use a virtual link to bridge across an intermediate area. Virtual links

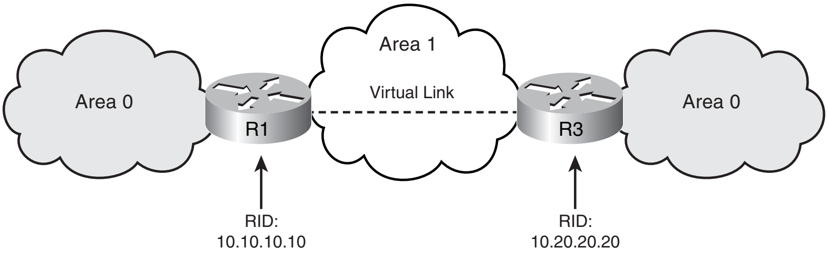
- Connect areas that do not have a physical link to area 0. (This should be a temporary solution.)
- Connect a discontiguous area 0 (when merging two company networks, for instance. This should also be a temporary solution!)

Figure 3-3 shows a virtual link connecting two portions of the backbone area 0.

Area 1 is the transit area for the virtual link. Configure each end of a virtual link on the ABRs of the transit area with the command **area** *area-number* **virtual-link** *router-id*. Each end of the link is identified by its RID. The area listed in the command is the transit area, not the area being joined by the link. The configuration for R1 is

```
R1(config)# router ospf 1
R1(config-router)# area 1 virtual-link 10.20.20.20
```

**Figure 3-3 OSPF Virtual Link**



The configuration for R3 is

```
R3(config)# router ospf 1
R3(config-router)# area 1 virtual-link 10.10.10.10
```

Verify that the virtual link is up with the **show ip ospf virtual-links** command. Additionally, virtual interfaces are treated as actual interfaces by the OSPF process, and thus, their status can be verified with the **show ip ospf interface** *interface-id* command.

## OSPF Authentication

For security purposes, you can configure OSPF to authenticate every OSPF packet and the source of every OSPF routing update. By default, the router does no authentication. OSPF supports two types of authentication:

- Simple (plain text) authentication
- MD5 authentication

The following example shows a router configured for simple password authentication in OSPF area 1, using a password (or *key*) of "simple." Note that authentication commands are necessary both under the OSPF process and the interface configuration. All OSPF neighbors reachable through an interface configured for authentication must use the same password. You can, however, use different passwords for different interfaces.

```
Router(config)# int gi0/0
Router(config-if)# ip ospf authentication-key simple
Router(config-if)# ip ospf authentication
Router(config-if)# !
Router(config-if)# router ospf 1
Router(config-router)# area 1 authentication
```

The next example shows the same router configured for OSPF MD5 authentication for area 0, using a password of "secure." Note that the commands are slightly different. The optional keyword **message-digest** is required in two of the commands, and a key number must be specified. Any neighbors reachable through the Gi0/1 interface must also be configured with the same key.

```
Router(config-router)# int gi0/1
Router(config-if)# ip ospf message-digest-key 2 md5 secure
Router(config-if)# ip ospf authentication message-digest
Router(config-if)# !
Router(config-if)# router ospf 1
Router(config-router)# area 0 authentication message-digest
```

Use the following commands to verify and troubleshoot OSPF authentication:

- **debug ip ospf adj:** The debug shows an error message if there is a key mismatch.
- **show ip ospf neighbor:** If a neighbor relationship has been established, you can assume the authentication worked properly.
- **show ip route:** Verify that route information is being exchanged between the two authenticating routers.

# CHAPTER 4

# Optimizing Routing

There are times when you need to go beyond just turning on a routing protocol in your network. You might need to control exactly which routes are advertised or redistributed, or which paths are chosen. You might also need to use multiple routing protocols. Network performance can suffer when routing is not optimized. Excessive routing updates lead to extra CPU usage because of the amount of routing information and the frequency of updates. Running multiple protocols requires extra router resources and might result in suboptimal paths. Incorrectly configured route filters can lead to routing issues.

## Controlling Routing Updates

Cisco IOS provides several ways to control routing updates:

- Route Maps
- Prefix Lists
- Distribute Lists
- Passive Interface

When a route update arrives at a router's interface, the router checks to see if a route filter is associated with that interface. If not, the update processes normally. If there is a filter, the router checks for an entry matching the update. If there is no matching entry, the update is dropped. If a matching entry exists, the router processes the update based on instructions in the filter.

### Route Maps

Route maps are a bit like programs that use an if/then/else decision-making capability. They *match* traffic against certain conditions and then set specified options for that traffic. Each statement has a sequence number, statements are read from the lowest number to highest, and the router stops reading when it gets a match. The sequence number can be used to insert or delete statements. Like an access list, there is an implicit "deny" at the end of each route map; any traffic not matched with a route map statement is denied. Some uses for route maps include

- **Filtering redistributed routes:** Use the **route-map** keyword in the redistribute command.
- **Policy-based routing:** To specify which traffic should be policy routed, based on very granular controls.
- **BGP policy:** To control routing updates and to manipulate path attributes.

## Route Map Syntax

Route maps are created with the global command:

```
Router(config)# route-map {tag} permit | deny [sequence_number]
```

Each statement in a route map begins this same way, with the same route map name but different sequence numbers, and with match and set conditions below it. *Permit* means that any traffic matching the match conditions is processed by the route map statement. *Deny* means that any traffic matching the match conditions is not processed by the route map statement.

## Route Map Match and Set Conditions

Each route map statement can have from none to multiple **match** and **set** conditions. If no **match** condition exists, the statement matches anything, similar to a "permit any" in an access list. If there is no **set** condition, the matching traffic is either permitted or denied, with no other conditions being set.

Multiple match conditions on the same line use a logical OR. For example, the router interprets **match a b c** as "**match a** or **b** or **c**." Multiple match conditions on different lines use a logical AND. For example, the router interprets the following route map statement as "**match a** and **b** and **c**":

```
route-map Logical-AND permit 10
 match a
 match b
 match c
```

In route redistribution, some common conditions to **match** include

- **ip address:** Refers the router to an access list that permits or denies networks.
- **ip address prefix-list:** Refers the router to a prefix-list that permits or denies IP prefixes.

- **ip next-hop:** Refers the router to an access list that permits or denies next-hop IP addresses.
- **ip route-source:** Refers the router to an access list that permits or denies advertising router IP addresses.
- **length:** Permits or denies packets based on their length in bytes.
- **metric:** Permits or denies routes with the specified metric from being redistributed.
- **route-type:** Permits or denies redistribution of the route type listed, such as internal or external.
- **tag:** Routes can be labeled (tagged) with a number, and route maps can look for that number.

In route redistribution, some common conditions to set are

- **metric:** Sets the metric for redistributed routes.
- **metric-type:** Sets the type, such as E1 for OSPF.
- **tag:** Tags a route with a number that can be matched on later by other route maps.

## Controlling Route Redistribution Using Route Maps

The following configuration example shows a route map named BGP-LP with three statements that control which routes will be redistributed from OSPF into BGP. The router has already been configured with two access lists, numbered 23 and 103 (not shown.) The first route map statement, with sequence number 10, is a *permit* statement. The **match** condition tells it to use access list 23. Any traffic permitted by access list 23 matches this statement and will be redistributed into BGP. Any traffic explicitly denied by access list 23 will not be redistributed into BGP. The **set** condition tells it to set a BGP local preference for all traffic that matches statement 10. Traffic not matching access list 23 will be checked against the second route map statement.

The second route map statement, sequence number 20, is a *deny* statement that matches access list 103. Any traffic permitted by access list 103 will be denied by this statement and thus will not be redistributed. Any traffic explicitly denied by access list 103 will be ignored by this statement and checked against the next route map statement. This route map statement has no **set** conditions. Traffic not matching route map statements 10 or 20 will be checked against statement 30.

The third route map statement, sequence number 30, is a *permit* statement with no **match** or **set** conditions. This statement matches everything and sets nothing, thus permitting all other traffic without changing it. Without this statement, all other traffic would be denied.

Lastly, the route map is applied to the redistribution command to filter routes redistributed from OSPF into BGP:

```
Router(config)# route-map BGP-LP permit 10
Router(config-route-map)# match ip address 23
Router(config-route-map)# set local-preference 200
Router(config-route-map)# !
Router(config-route-map)# route-map BGP-LP deny 20
Router(config-route-map)# match ip address 103
Router(config-route-map)# !
Router(config-route-map)# route-map BGP-LP permit 30
!
Router(config)# router bgp 65001
Router(config-router)# redistribute ospf 1 route-map BGP-LP
```

## Policy-Based Routing Using Route Maps

Policy-based routing overrides the normal routing process. Normal routing is done based on the destination IP address. Policy-based routing is based on source IP address or interface, or packet length. Create a route map statement that matches an access list, a specific IP address, or a packet length range. Then set either a next-hop IP address or an outbound interface for any traffic that matches the statement. Next, apply the route map either to an inbound interface or to the router itself to control locally generated traffic. The following configuration example shows a route map named LOCAL that matches the source addresses in access list 1. It assigns a next-hop IP address of 10.1.1.1 to this traffic. Because it is applied to the local router, it will be used only for traffic generated by the router itself.

```
Router(config)# route-map LOCAL
Router(config-route-map)# match ip address 1
Router(config-route-map)# set ip next-hop 10.1.1.1
!
Router(config)# ip local policy route-map LOCAL
```

Route map INT, shown in the following example, has no match condition and thus matches all traffic. It sets an outbound interface. Because it is applied to an interface, its policy routes all inbound traffic from that interface:

```
Router(config)# route-map INT
Router(config-route-map)# set interface fa0/1
!
```

```
Router(config)# int fa0/0
Router(config-if)# ip policy route-map INT
```

Verify policy routing with the **debug ip policy** command. See Chapter 5, "Path Control," for more information on policy-based routing.

### Tagging Routes Using a Route Map

Another use for a route map is to tag routes as they are redistributed from one protocol to another. Then you can deny tagged routes from being redistributed back into the original protocol. For instance, supposed you are mutually redistributing routes between OSPF and EIGRP. You can tag EIGRP routes as you redistribute them into OSPF. Then when you redistribute OSPF routes back into EIGRP, you can deny those tagged routes. The following example illustrates this.

```
Router(config)# route-map EIGRP2OSPF deny 5
Router(config-route-map)# match tag 1
Router(config-route-map)# route-map EIGRP2OSPF permit 10
Router(config-route-map)# set tag 2
!
Router(config)# route-map OSPF2EIGRP deny 5
Router(config-route-map)# match tag 2
Router(config-route-map)# route-map OSPF2EIGRP permit 10
Router(config-route-map)# set tag 1
!
Router(config)# router eigrp 1
Router(config-router)# redistribute ospf 2 route-map OSPF2EIGRP
  metric 1 1 1 1 1500
!
Router(config-router)# router ospf 2
Router(config-router)# redistribute eigrp 1 route-map EIGRP2OSPF
  subnets
```

## Prefix Lists

A prefix list matches both the subnet, or *prefix*, and the number of bits in the subnet mask. Similar to an access list, it consists of one or more statements permitting or denying prefixes. Routers evaluate the prefix statements in order, stopping if they find a match. There is an implicit "deny all" at the end of the prefix list. The command syntax follows:

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

The meaning of each command field is detailed in Table 4-1.

**Table 4-1 The ip prefix-list Command**

| Command Field | Meaning |
|---|---|
| `list-name` | Gives a name to the prefix list. Prefix lists are named, not numbered. |
| `seq number` | [Optional] Assigns a sequence number to the prefix list statement. Statements are numbered in increments of 5 by default, enabling a statement to be inserted between two others by using the **seq** option. |
| `deny \| permit` | Denies or permits the matching prefix. |
| *`network/length`* | Configures the prefix and number of bits that must be matched. If no ge or le option is given, the length also equals the length of the subnet mask. |
| `ge` *`ge-length`* | [Optional] Stands for “greater than or equal to.” Specifies the minimum number of bits a subnet mask must have to match the statement. |
| `le` *`le-length`* | [Optional] Stands for “lesser than or equal to.” Specifies the maximum number of bit a subnet mask can have to match the statement. |

Some sample prefix lists include

- **ip prefix-list CCNP permit 0.0.0.0/0:** Permits only a default route.
- **ip prefix-list CCNP permit 0.0.0.0/0 le 32:** Permits all routes (equivalent to a “permit any” in an access list.) The prefix 0.0.0.0/0 means that none of the prefix bits must be matched. “Le 32” means that the subnet mask must be less than or equal to 32. Thus any network will match this statement.
- **ip prefix-list CCNP permit 0.0.0.0/0 ge 32:** Permits only host routes. The prefix 0.0.0.0/0 means that none of the prefix bits must be matched. “Ge 32” means that the subnet mask must be exactly 32 bits, thus this statement matches only host routes.
- **ip prefix-list CCNP permit 10.0.0.0/8 ge 24 le 24:** Permits any route whose first 8 bits equal 10, with a subnet mask of exactly 24 bits.

Before taking the ROUTE exam, be sure you understand and can interpret prefix lists.

Prefix lists can be used in a route map to control redistribution of networks. They can also be applied to a BGP neighbor to filter routing updates to that neighbor.

## Distribute Lists

A distribute list enables you to filter both routing updates and routes being redistributed, through the use of an access list. Configure an access list that permits the routes to be advertised or redistributed, and then link that access list to the routing process with the **distribute-list** command, given under router configuration mode. This command has two options:

- **distribute-list** *access-list* **in–**Filters updates as they come in an interface. For OSPF, this controls routes placed in the routing table but not the database. For other protocols, this controls the routes the protocol knows about.
- **distribute-list** *access-list* **out**–Filters updates going out of an interface and also updates being redistributed out of another routing protocol into this one.

## Passive Interfaces

The **passive-interface** command is another way to control routing updates because it prevents any updates from sending out an interface that is marked as passive. OSPF and EIGRP do not send Hello messages out a passive interface, and thus do not discover any neighbors. RIP does not send updates out a passive interface but listens for inbound updates. The EIGRP and OSPF chapters have a more in-depth description of this command.

# Using Multiple Routing Protocols

There are several reasons you might need to run multiple routing protocols in your network. Some include

- Migrating from one routing protocol to another, where both protocols will run in the network temporarily
- Applications that run under certain routing protocols but not others
- Areas of the network under different administrative control (Layer 8 issues)
- A multivendor environment in which some parts of the network require a standards-based protocol

## Configuring Route Redistribution

Route redistribution is used when routing information must be exchanged among the different protocols or routing domains. Only routes that are in the routing table and learned via the specified protocol are redistributed. Each protocol has some unique characteristics when redistributing, as shown in Table 4-2.

**Table 4-2 Route Redistribution Characteristics**

| Protocol | Redistribution Characteristics |
|---|---|
| `RIP` | Default metric is Infinity. Metric must be set, except when redistributing static or connected routes, which have a metric of 1. |
| `OSPF` | Default metric is 20. Can specify the metric type; the default is E2. Must use **subnets** keyword or only classful networks are redistributed. |
| `EIGRP` | Default metric is Infinity. Metric must be set, except when redistributing static or connected routes, which get their metric from the interface. Metric value is "bandwidth, delay, reliability, load, MTU." Redistributed routes have a higher administrative distance than internal ones. |
| `Static/Connected` | To include local networks not running the routing protocol, you must redistribute connected interfaces. You can also redistribute static routes into a dynamic protocol. |
| `BGP` | Metric (MED) is set to IGP metric value. |

You can redistribute only between protocols that use the same protocol stack, such as IP protocols, which cannot advertise IPX routes. To configure redistribution, issue this command under the routing process that is to receive the new routes:

```
Router(config-router)# redistribute {route-source} [metric metric]
 [route-map tag]
```

# Seed Metric

Redistribution involves configuring a routing protocol to advertise routes learned by another routing process. Normally, protocols base their metric on an interface value, such as bandwidth, but a redistributed route is not associated with an interface. Protocols use incompatible metrics, so the redistributed routes must be assigned a new metric compatible with the new protocol.

A route's starting metric is called its *seed metric*. Set the seed metric for all redistributed routes with the **default-metric** [*metric*] command under the routing process. To set the metric for specific routes, either use the **metric** keyword when redistributing or use the **route-map** keyword to link a route map to the redistribution. After the seed metric is specified, it increments normally as the route is advertised through the network (except for certain OSPF routes).

## Administrative Distance

When a router receives routes to the same destination network from more than one routing process, it decides which to put in the routing table by looking at the administrative distance (AD) value assigned to the routing process. The route with the lowest AD is chosen. Table 4-3 shows administrative distance values.

**Table 4-3 Administrative Distance**

| Routing Information Source | Administrative Distance |
|---|---|
| Connected interface | 0 |
| Static route | 1 |
| EIGRP summarized route | 5 |
| BGP external route | 20 |
| EIGRP internal route | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| EIGRP external route | 170 |
| BGP internal route | 200 |
| Unknown | 255 |

AD can be changed for all routes of a process or only for specific routes within a process. The command for all IGPs except EIGRP is

```
Router(config-router)# distance administrative_distance {address
  wildcard-mask} [access-list-number | name]
```

Using the **address/mask** keywords in the command changes the AD of routes learned from the neighbor with that IP address. An entry of **0.0.0.0**

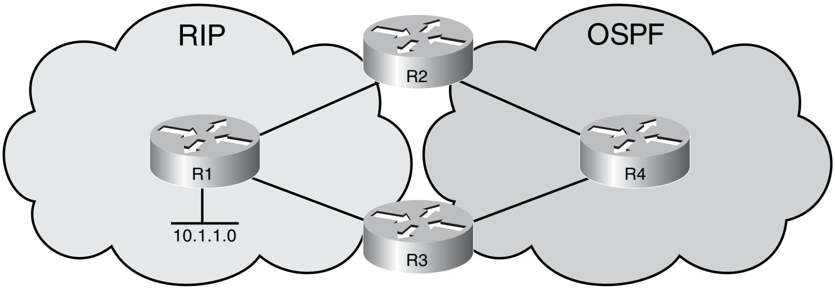
**255.255.255.255** changes the AD of all routes. Specifying an access list number or name changes the AD only on networks permitted in the ACL.

EIGRP and BGP have different AD values for internal and external routes, so you have to list those separately when using the command with those protocols. BGP also enables you to change the AD for locally generated routes. For these protocols, the commands are

```
Router(config-router)# distance eigrp internal-distance external-
  distance
Router(config-router)# distance bgp external-distance internal-
  distance local-distance
```

Route redistribution can cause suboptimal routing; one way to correct this is to adjust AD. Figure 4-1 shows a network with two routing domains: RIP and OSPF.

**Figure 4-1 Controlling Routing with AD**



R2 redistributes its RIP routes into OSPF. These routes inherit OSPF's AD when they are advertised to R4, which then advertises them to R3 as OSPF routes.

R3 now knows about the 10.1.1.0 network from two routing processes: RIP, with an AD of 120, and OSPF, with an AD of 110. The shortest path is the RIP route through R1. The OSPF path goes through R4 and R2, and then to R1—a much longer path. But, based on AD, R3 puts the OSPF path in its routing table.

To prevent this, increase the AD of the redistributed RIP routes when OSPF advertises them. Note that this doesn't change all OSPF routes, just the ones learned from RIP. The commands given on R2 (the router doing the initial redistribution) are shown here:

```
Router(config)# access-list 10 permit 10.1.1.0
!
Router(config)# router ospf 1
Router(config-router)# redistribute rip subnets
Router(config-router)# distance 125 0.0.0.0 255.255.255.255 10
```

The AD is increased to 125 for routes from all neighbors, if they match the network permitted in access list 10. Now R3 hears about the 10.1.1.0 network from RIP with an AD of 120, and from OSPF with an AD of 125. The RIP route is put into the routing table based on its lower AD.

Routing protocols that assign a higher AD to external routes, EIGRP and BGP, accomplish a similar result automatically. OSPF can be configured to do so with the **distance ospf external** command.

## Planning Route Redistribution

Plan carefully before redistributing routes between protocols. Different protocols have incompatible routing information and different convergence times. First, decide which is the core, or main, protocol and which is the edge protocol. Decide if you will do one-way or two-way, and single point or multipoint redistribution.

One-way redistribution involves redistributing routes from the edge routing protocol into the core protocol. Static or default routes must be used in the edge protocol. Two-way redistribution involves redistributing routes mutually between both core and edge protocols. No static routes are needed because both protocols know all routing information.

One-way and two-way redistribution at just one router within the network is considered safe because traffic between administrative domains has only one exit point, thus routing loops are not a problem. Redistribution at multiple routers within the network can cause routing loops and suboptimal routing.

With multipoint one-way redistribution:

- Use a routing protocol that uses different ADs for external and internal routes (EIGRP, OSPF, and BGP).
- Ensure that the AD of the redistributed external routes is higher than the AD of the protocol where they originated.

Multipoint two-way redistribution adds the following considerations:

- Ensure that only internal routes are redistributed from each protocol. You can do this by tagging the routes and then filtering based on tags when redistributing.
- Adjust the metric of the redistributed routes.

- Consider using a default route to avoid multipoint two-way redistribution.

## Redistribution Techniques

Try to design your route redistribution as safely as possible. The options include

- Redistribute all edge information into the core, but send only a default route into the edge.
- Redistribute all edge information into the core, but redistribute multiple static routes into the edge.
- Redistribute routes in both directions, but filter to prevent routes from being redistributed back into their original administrative domain.
- Redistribute all routes in both directions, but increase the AD for external routes.

## Redistribution Notes

The IPv6 commands to redistribute routes between protocols or between multiple instances of a protocol are just like the ones in IPv4. Under the routing protocol configuration mode, issue the command **redistribute** *route-source* and specify any options such as a route map if desired.

Some points to remember about redistributing routes follow:

- A router redistributes only routes learned by the source protocol. For instance, if you redistribute connected routes into the protocol, it will advertise them but not redistribute them.
- When redistributing routes into BGP, you can use the keyword **include-connected** to get the connected routes into BGP.
- When you redistribute routes between two OSPF processes, the routes are advertised into the new process as Type 2.
- You generally want to include the **subnets** keyword on routes distributed from another routing protocol into OSPF. Otherwise, only routes that use their default classful subnet mask are redistributed.
- Be sure to specify a seed metric when redistributing routes into RIP. Otherwise the routes start with a metric of 16, which RIP interprets as “unreachable.”

- If you redistribute in multiple places, check the path that traffic takes. You might run into suboptimal routing. A way to fix this is to tune the administrative distance for some of the routes.
- BGP does not redistribute routes learned via IBGP into an IGP by default. To change this behavior, use the router configuration command **bgp redistribute-internal**.

# CHAPTER 5

# Path Control

In general, link redundancy is a good thing, but it can also lead to some network problems. You might want to manually control the route taken by some or all of your traffic to provide a predictable and deterministic traffic flow. Path control can prevent suboptimal routing, ensure path availability, provide optimized performance for specific applications, and provide load sharing among various paths.

A good path control strategy understands that traffic is bidirectional and considers both inbound and outbound traffic. Asymmetric routing—where traffic exits via one link and enters via another—is not inherently bad. But you might need to minimize it when using stateful devices such as firewalls, or with sensitive applications such as voice. A good routing plan requires a good design. Design IP addresses that can be summarized and use redistribution and passive interfaces appropriately.

The previous chapter covered several ways to influence paths: route maps, prefix lists, distribute lists, administrative distance, and route tags. This chapter covers some others: Offset-lists, IP SLA, Policy-based Routing (PBR), Optimized Edge Routing, and Virtual Routing and Forwarding (VRF).

## Using Offset-lists

An offset-list is a way to increase the metric of routes. You might do this to cause a router to choose what it would normally consider a less desirable path, or to load balance over paths that would normally have unequal methods. The command uses an access list, so you can also use this command to send a subset of traffic over a different path. Historically, offset-lists were used with RIP because it only looked at hop count, and thus might choose a slow path that had a lower hop count than a fast path. Only RIP and EIGRP support offset-lists.

Configure an offset-list with the command **offset-list** {*access-list-number* | *access-list-name*} {**in** | **out**} *offset* [*interface-type interface-number*]. This command is given in router configuration mode. The offset amount is added to the hop count in RIP and is added to the delay value in EIGRP. Be careful when changing an EIGRP metric value and test thoroughly. The following example shows a delay value of 2000 added to the EIGRP metric of the two routes permitted in access list Offset, when they are advertised in interface FastEthernet 0/0.

```
R1(config)# ip access-list standard Offset
R1(config-std-nacl)# permit 172.31.1.0 0.0.0.255
R1(config-std-nacl)# permit 172.16.1.0 0.0.0.255
!
R1(config)# router eigrp 100
R1(config-router)# offset-list Offset in 2000 fa0/0
```

You can configure more than one offset-list. If you specify an interface, the offset-list is considered an extended offset-list and has precedence over a normal offset-list.

## Using IOS IP SLA

IP SLA is a feature that enables a Cisco router or switch to simulate specific types of traffic and send it either to an IP address or to a receiver, called a "responder." IP SLA probes can simulate various types of traffic, such as HTTP, FTP, DHCP, UDP jitter, UDP echo, HTTP, TCP connect, ICMP echo, ICMP path echo, ICMP path jitter, and DNS, and can report statistics such as path jitter. It has highly granular application configuration options such as TCP/UDP port numbers, TOS byte, and IP prefix bits. With IP SLA you can measure network performance and host reachability. This is useful for path control because it enables you to switch to a backup path if network performance on the primary path degrades, or if a link failure occurs someplace in the primary path.

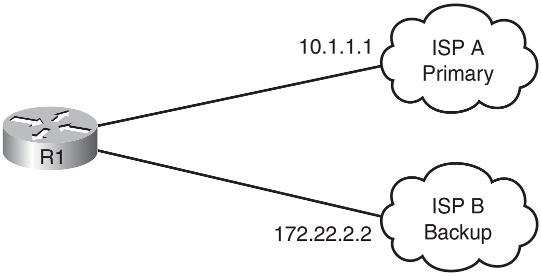
To use IP SLA for path control, you must

1. Create a monitor session on the probe source device.
2. Define the probe by specifying traffic type, destination IP address, and any other desired variables such as DSCP value.
3. Schedule the probe beginning and ending times.
4. Define a tracking object that is linked to the monitor session.
5. Link the tracking object to a static route.

The destination can be any trusted device that responds to the traffic you send. To use IP SLA for bringing up a backup link, choose a destination that will reflect problems in the ISP's network. One benefit of using a Cisco device as the responder is that it can add time stamps to help measure latency and jitter. These time stamps take into account the device processing time so that the measurement reflects only network latency. The configuration of a Cisco responder is simple. Use the global command **ip sla responder.**

Figure 5-1 shows a network with two connections to the Internet, one primary and one backup. The edge router has a static default route pointing to each provider.

**Figure 5-1 Using IP SLA for Path Control**



In the following example, router R1 is configured to conditionally announce a default route based on the IP SLA probe response. Two IP SLA monitor sessions are configured to send a ping every 10 seconds to a DNS server within each ISP's network. A separate tracking object tracks reachability to each server. The two default route statements tell the router to give the primary route an administrative distance of 2 if the tracked object is reachable. The backup route is assigned an administrative distance of 3 if its tracked object is reachable.

```
R1(config)# ip sla 1
R1(config-ip-sla)# icmp-echo 10.1.1.50
R1(config-ip-sla-echo)# frequency 10
!
R1(config)# ip sla 2
R1(config-ip-sla)# icmp-echo 171.22.2.52
R1(config-ip-sla-echo)# frequency 10
!
R1(config)# ip sla schedule 1 life forever start-time now
R1(config)# ip sla schedule 2 life forever start-time now
!
R1(config)# track 1 ip sla 1 reachability
R1(config)# track 2 ip sla 2 reachability
!
!Primary route
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1 2 track 1
```

```
!Backup route
R1(config)# ip route 0.0.0.0 0.0.0.0 172.22.2.2 3 track 2
```

Under normal circumstances, the default route with an AD of 2 would be installed in the IP routing table instead of the one with an AD of 3. But when the primary DNS server is not reachable, the primary route is withdrawn, and the backup route with an AD of 3 is installed in the routing table.

Additionally, you can combine IP SLA tracking with policy-based routing to provide redundancy for specific traffic types on a per-interface basis. For instance, after an access list permitting the interesting traffic is specified, a route-map such as the following can be configured. In the example route-map, x.x.x.x and y.y.y.y represent different next hops to send the traffic specified in ACL 101. The **track 1** and **track 2** keywords tie the configuration back to the SLA groups configured in the previous example. You then apply the policy map under the incoming interfaces just as with normal policy routing.

```
route-map REDUNDANT permit 10
 match ip address 101
 set ip next-hop verify-availability x.x.x.x 10 track 1
 set ip next hop verify-availability y.y.y.y 20 track 2
!

interface FastEthernet0/1
 description LAN Interface
 ip policy route-map REDUNDANT
```

## Policy-Based Routing

Normal IP routing chooses a path based on the destination IP address. Policy-based routing (PBR) lets you route traffic based on other variables. It uses a route-map to match traffic and then sets either a next-hop address or an exit interface. It can also mark the traffic that it policy routes. Any traffic not matched in the route-map is routed normally. Policy-based routing can be applied both to traffic entering the router and to traffic originated by the router.

Some benefits of policy routing include

- Ability to route based on traffic source, and other attributes
- Ability to set QoS markings
- Ability to force load sharing between unequal paths
- Ability to allocate traffic among multiple paths based on traffic attributes

Use the following steps to configure policy-based routing:

1. Configure a route-map that matches the desired traffic and uses the **set** command to define the actions for that traffic.
2. Optionally enable fast-switched PBR. CEF-switched PBR is enabled automatically whenever CEF is enabled.
3. Apply the route map either to an incoming interface or to traffic generated by the router.
4. Verify the configuration.

Some typical attributes to match in the route map include source or destination address with an access list and packet length. If no match criteria are specified, all packets are considered a match.

You can choose to set IP precedence, but the most typical setting determines how the traffic leaves the router. There are four ways to do this. If there are multiple **set** statements, the router evaluates them in this order:

1. **set ip next-hop** *ip-address*: When this command is given, the router checks to see if the next-hop address is reachable. If so, it forwards the traffic toward that address. If not, it uses the routing table.
2. **set interface** *interface-type interface-number:* Multiple interfaces can be listed. When this command is given, the router checks that it has an explicit route for the destination network in its routing table before forwarding the traffic out the specified interface. If it does not, this command is ignored. A default route is not considered an explicit route. Listing multiple interfaces under the set command allows for redundancy if the first interface fails or goes down. The router uses the first active interface listed.
3. **set ip default next-hop** *ip-address*: If the routing table contains an explicit route for the destination network, that route is used and this command is ignored. If no explicit route exists, this command is executed. A default route is not considered an explicit route.
4. **set default interface** *interface-type interface-number*: If the routing table contains an explicit route for the destination network, that route is used and this command is ignored. If no explicit route exists, traffic is forwarded out the specified interf*ace.* A default route is not considered an explicit route.

To apply the PBR route-map to an interface, use the interface command **ip policy route-map** *name*. To apply PBR to packets originated by the router itself, use the global command **ip local policy route-map** *name*. The following example shows a PBR route map that matches traffic in access list 101, and policy routes it to a next hop of 10.1.1.1. The policy is applied to interface fa0/0, so all traffic entering that interface is evaluated against the route-map. Packets not permitted by access list 101 are routed normally (destination routed.)

```
route-map Policy permit 10
 match ip address 101
 set ip next hop 10.1.1.1
!
interface FastEthernet0/0
 ip policy route-map Policy
```

PBR can also use IP SLA tracking, described previously in the IP SLA section. Verify your configuration with the commands **show ip policy** and **show route-map** {*name*}.

## OER and VRF

With OER, Border Routers monitor WAN performance and report the information to a Master Controller router. If WAN performance falls within configured ranges, no change is made to the default routing. If performance begins to degrade for a specific link or network, the Controller notifies its Border Routers to reroute traffic, perhaps by adding a static route or changing routing protocol parameters.

VRFs are a way to segment traffic. You might think of them as Layer 3 VLANs. Just as VLANs create virtual switches with segregated CAM tables, VRFs create virtual routers with segregated routing tables. This lets you separate guest traffic from employee traffic, for instance. Traffic for different VRFs can be routed over different paths.

# CHAPTER 6

# BGP and Internet Connectivity

## Planning an Internet Connection

Consider your company's needs when planning your Internet connection. If all you need is one-way connectivity to enable internal users to connect to sites on the Internet, a private IP address space with Network Address Translation (NAT) should suffice. If external users need to connect to resources, such as servers, inside your network, you need some public IP addresses. You might combine these with private addresses and NAT for your users.

If external users must connect to your internal resources, you should plan the following:

- How many public IP addresses will you need?
- Should you get your IP addresses from your ISP or acquire your own? If you elect to acquire your own addresses, you also need a public Autonomous System (AS) number.
- What link type and speed will you need to support all the external connections plus your internal users?
- Will you use static or dynamic routing?
- How much redundancy will you need? This includes link redundancy and ISP redundancy.

### To Route or Not to Route?

If your ISP connection is a Layer 2 circuit emulation, there is no need to run a routing protocol with the ISP.

If you use MPLS VPNs, you either use static routes or run a dynamic routing protocol with the ISP edge router. This might be either one of the IGPs (EIGRP, OSPF, or RIP) or Border Gateway Protocol (BGP).

If you need only a default route pointing to your ISP, static routes work. The provider needs to create static routes pointing to your network and redistribute them into its routing protocol.

BGP is a good choice if you connect to multiple ISPs, you need to control how traffic enters or exits your company, or you need to react to Internet topology changes.

## BGP Route Options

You have a choice of three ways to receive BGP routes from an ISP:

- **Default routes from each provider:** This is simple to configure and results in low use of bandwidth and router resources. The internal network's IGP metric determines the exit router for all traffic bound outside the autonomous system. No BGP path manipulation is possible, so this can lead to suboptimal routing if you use more than one ISP.
- **Default routes plus some more specific routes:** This option results in medium use of bandwidth and router resources. It enables you to manipulate the exit path for specific routes using BGP so that traffic takes a shorter path to networks in each ISP. Thus path selection is more predictable. The IGP metric chooses the exit path for default routes.
- **All routes from all providers:** This requires the highest use of bandwidth and router resources. It is typically done by large enterprises and ISPs. Path selection for all external routes can be controlled via BGP policy routing tools.

## Types of ISP Connections

A site with a single ISP connection is *single-homed.* This is fine for a site that does not depend heavily on Internet or WAN connectivity. Either use static routes, or advertise the site routes to the ISP and receive a default route from the ISP.

A *dual-homed* site has two connections to the same ISP, either from one router or two routers. One link might be primary and the other backup, or the site might load balance over both links. Either static or dynamic routing would work in this case.

*Multihoming* means connecting to more than one ISP at the same time. It is done for redundancy and backup if one ISP fails, and for better performance if one ISP provides a better path to frequently used networks. This also gives you an ISP-independent solution. BGP is typically used with multihomed connections.

You can take multihoming a step further and be *dual-multihomed*, with two connections to multiple ISPs. This gives the most redundancy. BGP is used with the ISPs and can be used internally also.

## BGP Overview

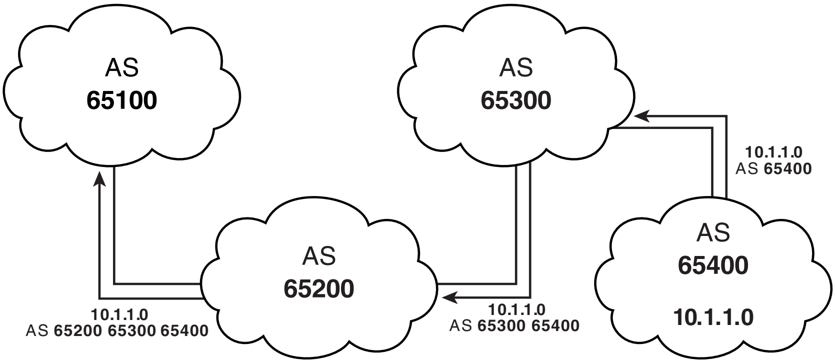
BGP is an external gateway protocol, meant to be used between different networks. It is the protocol used between Internet service providers (ISPs) and also can be used between an Enterprise and an ISP. BGP was built for reliability, scalability, and control, not speed. Because of this, it behaves differently from the protocols covered so far in this book:

- BGP stands for Border Gateway Protocol. Routers running BGP are termed *BGP speakers*.
- BGP uses the concept of autonomous systems (AS). An *autonomous system* is a group of networks under a common administration. The Internet Assigned Numbers Authority (IANA) assigns AS numbers: 1 to 64511 are public AS numbers and 64512 to 65535 are private AS numbers.
- Autonomous systems run Interior Gateway Protocols (IGP) within the system. They run an Exterior Gateway Protocol (EGP) between them. BGP version 4 is the only EGP currently in use.
- Routing between autonomous systems is called *interdomain routing*.
- The administrative distance for EBGP routes is 20. The administrative distance for IBGP routes is 200.
- BGP neighbors are called peers and must be statically configured.
- BGP uses TCP port 179. BGP peers exchange incremental, triggered route updates and periodic keepalives.
- Routers can run only one instance of BGP at a time.
- BGP is a path-vector protocol. Its route to a network consists of a list of autonomous systems on the path to that network.
- BGP's loop prevention mechanism is an autonomous system number.

> When an update about a network leaves an autonomous system, that autonomous system's number is prepended to the list of autonomous systems that have handled that update. When an autonomous system receives an update, it examines the autonomous system list. If it finds its own autonomous system number in that list, the update is discarded.

In Figure 6-1, BGP routers in AS 65100 see network 10.1.1.0 as having an autonomous system path of 65200 65300 65400.

**Figure 6-1 BGP AS-Path Advertisement**



Use BGP when the AS is multihomed, when route path manipulation is needed, or when the AS is a transit AS. (Traffic flows through it to another AS, such as with an ISP.)

Do not use BGP in a single-homed AS, with a router that does not have sufficient resources to handle it, or with a staff that does not have a good understanding of BGP path selection and manipulation.

## BGP Databases

BGP uses three databases. The first two listed are BGP-specific; the third is shared by all routing processes on the router:

- **Neighbor database:** A list of all configured BGP neighbors. To view it, use the **show ip bgp summary** command.
- **BGP database, or RIB (Routing Information Base):** A list of networks known by BGP, along with their paths and attributes. To view it, use the **show ip bgp** command.

- **Routing table:** A list of the paths to each network used by the router, and the next hop for each network. To view it, use the **show ip route** command.

## BGP Message Types

BGP has four types of messages:

- **Open:** After a neighbor is configured, BGP sends an open message to try to establish peering with that neighbor. Includes information such as autonomous system number, router ID, and hold time.
- **Update:** Message used to transfer routing information between peers. Includes new routes, withdrawn routes, and path attributes.
- **Keepalive:** BGP peers exchange keepalive messages every 60 seconds by default. These keep the peering session active.
- **Notification:** When a problem occurs that causes a router to end the BGP peering session, a notification message is sent to the BGP neighbor and the connection is closed.

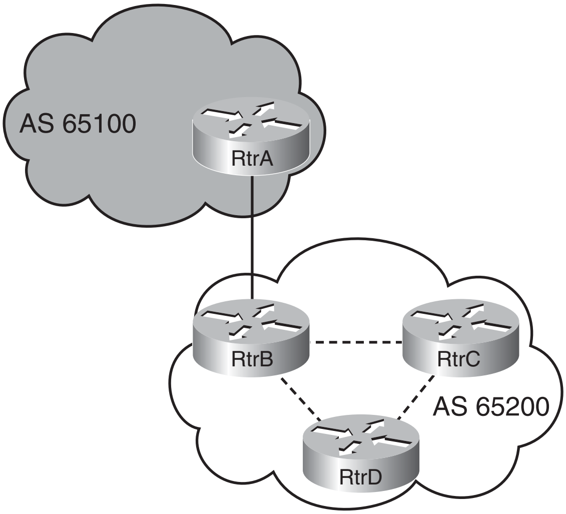
## Internal and External BGP

Internal BGP (IBGP) is a BGP peering relationship between routers in the same autonomous system. External BGP (EBGP) is a BGP peering relationship between routers in different autonomous systems. BGP treats updates from internal peers differently than updates from external peers.

Before any BGP speaker can peer with a neighbor router, that neighbor must be statically defined. A TCP session must be established, so the IP address used to peer with must be reachable.

In Figure 6-2, Routers A and B are EBGP peers. Routers B, C, and D are IBGP peers.

**Figure 6-2 Identifying EBGP and IBGP Peers**



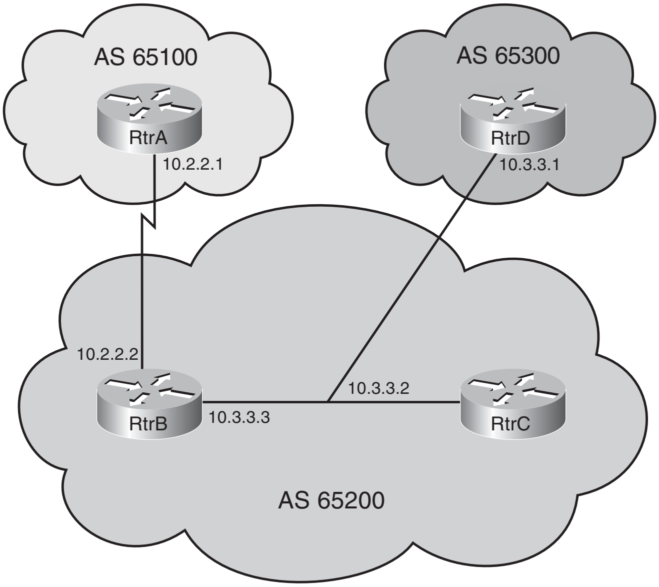
## BGP Next-Hop Selection

The next hop for a route received from an EBGP neighbor is the IP address of the neighbor that sent the update.

When a BGP router receives an update from an EBGP neighbor, it must pass that update to its IBGP neighbors without changing the next-hop attribute. The next-hop IP address is the IP address of an edge router belonging to the next-hop autonomous system. Therefore, IBGP routers must have a route to the network connecting their autonomous system to that edge router. For example, in Figure 6-3, RtrA sends an update to RtrB, listing a next hop of 10.2.2.1, its serial interface. When RtrB forwards that update to RtrC, the next-hop IP address will still be 10.2.2.1. RtrC needs to have a route to the 10.2.2.0 network to have a valid next hop.

To change this behavior, use the **neighbor** [*ip address*] **next-hop-self** command in BGP configuration mode. In Figure 6-3, this configuration goes on RtrB. After you give this command, RtrB advertises its IP address to RtrC as the next hop for networks from AS 65100, rather than the address of RtrA. Thus, RtrC does not have to know about the external network between RtrA and RtrB (network 10.2.2.0).

**Figure 6-3 BGP Next-Hop Behavior**



## BGP Next Hop on a Multiaccess Network

On a multiaccess network, BGP can adjust the next-hop attribute to avoid an extra hop. In Figure 6-3, RtrC and RtrD are EBGP peers, and RtrC is an IBGP peer with RtrB. When C sends an update to D about network 10.2.2.0, it normally gives its interface IP address as the next hop for D to use. But because B, C, and D are all on the same multiaccess network, it is inefficient for D to send traffic to C, and C to then send it on to B. This process unnecessarily adds an extra hop to the path. So, by default, RtrC advertises a next hop of 10.3.3.3 (RtrB's interface) for the 10.2.2.0 network. This behavior can also be adjusted with the **neighbor** [*ip address*] **next-hop-self** command.

## BGP Synchronization Rule

The BGP synchronization rule requires that when a BGP router receives information about a network from an IBGP neighbor, it does not use that information until a matching route is learned via an IGP or static route. It also does not advertise that route to an EBGP neighbor unless a matching route is in the routing table. In Figure 6-3, if RtrB advertises a route to RtrC, then RtrC does not submit it to the routing table or advertise it to RtrD unless it also learns the route from some other IGP source.

Recent IOS versions have synchronization disabled by default. It is usually safe to turn off synchronization when all routers in the autonomous system run BGP. To turn it off in earlier IOS versions, use the command **no synchronization** under BGP router configuration mode.

# Configuring BGP

Before beginning to configure BGP, gather the network requirements you need, which should include the following:

- Whether you need to run IBGP for internal connectivity
- External connectivity to the ISP
- Configuration parameters such as neighbor IP addresses and their AS number, and which networks you will advertise via BGP

Table 6-1 lists the basic BGP configuration commands and their functions.

**Table 6-1 Basic BGP Configuration Commands**

| Command | Description |
|---|---|
| **router bgp** *AS-number* | Starts the BGP routing process on the router. |
| **neighbor** *ip-address* **remote-as** *AS-number* | Sets up peering between BGP routers. IP address must match the source of routing updates. |
| **neighbor peer**-*group-name* **peer-group** | Creates a peer group to which you can then assign neighbors. |
| **neighbor** *ip-address* **peer-group** *peer-group-name* | Assigns a neighbor to a peer group. |
| **neighbor** *ip-address* **next-hop-self** | Configures a router to advertise its connected interface as the next hop for all routes to this neighbor. |
| **neighbor** *ip-address* **update-source** *interface-type number* | Configures a router to use the IP address of a specific interface as the source for its advertisements to this neighbor. |
| **no synchronization** | Turns off BGP synchronization. |
| **network** *prefix* [**mask** ***subnet-mask***] | Initiates the advertisement of a network in BGP. |

## BGP Network Command

In most IGPs, the network command starts the routing process on an interface. In BGP, the command tells the router to originate an advertisement for that network. The network does not have to be connected to the router; it just has to be in the routing table. In theory, it can even be a network in a different autonomous system (not usually recommended).

When advertising a network, BGP assumes you are using the default classful subnet mask. If you want to advertise a subnet, you must use the optional keyword **mask** and specify the subnet mask to use. Note that this is a subnet mask, not the inverse mask used by OSPF and EIGRP network statements. The routing table must contain an exact match (prefix and subnet mask) to the network listed in the network statement before BGP advertises the route.

## BGP Peering

BGP assumes that external neighbors are directly connected and that they are peering with the IP address of the directly connected interface of their neighbor. If not, you must tell BGP to look more than one hop away for its neighbor, with the **neighbor** *ip-address* **ebgp-multihop** *number-of-hops* command. You might use this command if you are peering with loopback interface IP addresses, for instance. BGP assumes that internal neighbors might not be directly connected, so this command is not needed with IBGP. If you do peer with loopback IP addresses, you must change the source of the BGP packets to match the loopback address with the **neighbor** *ip-address* **update-source** *interface* command.

To take down the peering session with a neighbor but keep the neighbor configuration, use the **neighbor** *ip-address* **shutdown** command.

## BGP Peering States

The command **show ip bgp neighbors** shows a list of peers and the status of their peering session. This status can include the following states:

- **Idle:** No peering; router is looking for neighbor. Idle (admin) means that the neighbor relationship has been administratively shut down.
- **Connect:** TCP handshake completed.
- **OpenSent, or Active:** An open message was sent to try to establish the peering.
- **OpenConfirm:** Router has received a reply to the open message.
- **Established:** Routers have a BGP peering session. This is the desired state.

You can troubleshoot session establishment with debug commands. Use **debug ip bgp events** or **debug ip bgp ipv4 unicast** (in IOS versions 12.4 and up) to see where the process fails. Some common failure causes include AS number misconfiguration, neighbor IP address misconfiguration, a neighbor with no neighbor statement for your router, and a neighbor with no route to the source address of your router's BGP messages.

# BGP Path Selection

IGPs, such as EIGRP or OSPF, choose routes based on lowest metric. They attempt to find the shortest, fastest way to get traffic to its destination. BGP, however, has a different way of route selection. It assigns various attributes to each path; these attributes can be administratively manipulated to control the path that is selected. It then examines the value of these attributes in an ordered fashion until it can narrow all the possible routes down to one path.

## BGP Attributes

BGP chooses a route to a network based on the attributes of its path. Four categories of attributes exist as follows:

- **Well-known mandatory:** Must be recognized by all BGP routers, present in all BGP updates, and passed on to other BGP routers. For example, AS path, origin, and next hop.
- **Well-known discretionary:** Must be recognized by all BGP routers and passed on to other BGP routers but need not be present in an update, for example, local preference.
- **Optional transitive:** Might or might not be recognized by a BGP router but is passed on to other BGP routers. If not recognized, it is marked as partial, for example, aggregator, community.
- **Optional nontransitive:** Might or might not be recognized by a BGP router and is not passed on to other routers, for example, Multi-Exit Discriminator (MED), originator ID.

Table 6-2 lists common BGP attributes, their meanings, and their category.

**Table 6-2 BGP Attributes**

| Attribute | Meaning |
|---|---|
| AS path | An ordered list of all the autonomous systems through which this update has passed. Well-known, mandatory. |
| Origin | How BGP learned of this network. i = by *network* command, e = from EGP, ? = redistributed from other source. Well-known, mandatory. |
| Local Preference | A value telling IBGP peers which path to select for traffic leaving the AS. Default value is 100. Well-known, discretionary. |
| Multi-Exit Discriminator (MED) | Suggests to a neighboring autonomous system which of multiple paths to select for traffic bound into your autonomous system. Lowest MED is preferred. Optional, non-transitive. |
| Weight | Cisco proprietary, to tell a router which of multiple local paths to select for traffic leaving the AS. Highest weight is preferred. Only has local significance. |

## BGP Path Selection Criteria

BGP tries to narrow its path selection down to one best path; it does not load balance by default. To do so, it examines the path attributes of any loop-free, synchronized (if synchronization is enabled) routes with a reachable next-hop in the following order:

1. Choose the route with the highest weight.
2. If weight is not set, choose the route with the highest local preference.
3. Choose routes that this router originated.
4. Choose the path with the shortest Autonomous System path.
5. Choose the path with the lowest origin code (i is lowest, e is next, ? is last).
6. Choose the route with the lowest MED, if the same Autonomous System advertises the possible routes.
7. Choose an EBGP route over an IBGP route.
8. Choose the route through the nearest IGP neighbor as determined by the lowest IGP metric.
9. Choose the oldest route
10. Choose a path through the neighbor with the lowest router ID.
11. Choose a path through the neighbor with the lowest IP address.

To enable BGP to load balance over more than one path, you must enter the command **maximum-paths** *number-of-paths*. BGP can load balance over a maximum of six paths.

## Influencing BGP Path Selection

BGP was not created to be a fast protocol; it was created to enable as much administrative control over route path selection as possible. Path selection is controlled by manipulating BGP attributes, usually using route maps. You can set a default local preference by using the command **bgp default local-preference** and a default MED for redistributed routes with the **default-metric** command under the BGP routing process. But by using route maps, you can change attributes for certain neighbors only or for certain routes only. The earlier section on route maps contains an example of using a route map to set a local preference of 200 for specific redistributed routes. This is higher than the default local preference of 120, so routers within the AS are more likely to prefer that path than others.

Route maps can also be applied to routes sent to or received from a neighbor. The following example shows a simple route map that sets a MED value and adds two more copies of its AS number to the AS path on all routes advertised out to an EBGP neighbor:

```
route-map MED permit 10
 set metric 50
 set as-path prepend 65001 65001
!
router bgp 65001
 neighbor 10.1.1.1 route-map MED out
```

When attributes are changed, you must tell BGP to apply the changes. Either clear the BGP session (**clear ip bgp** * ) or do a soft reset (**clear ip bgp * soft in | out**). Routers using recent IOS versions do a route refresh when the session in cleared inbound.

# Filtering BGP Routes

You can combine route maps with prefix lists to filter the routes advertised to or received from a BGP peer, to control routes redistributed into BGP, and to set BGP attributes for specific routes. Prefix lists alone can be applied to a neighbor to filter route updates.

To use a prefix list, plan the implementation by determining the requirements. Then create a prefix list to match the networks to be filtered. Permit the networks you want to allow to be advertised and deny all others. Next apply the prefix list to the BGP neighbor, inbound or outbound. The next example shows a prefix list that permits only summary routes in the 172.31.0.0 network. All other routes are denied by default. The prefix list is then applied to BGP neighbor 10.1.1.1 outbound, so only these routes will be advertised to that peer:

```
ip prefix-list Summary permit 172.31.0.0/16 le 20
!
router bgp 65001
neighbor 10.1.1.1 prefix-list Summary out
```

To verify the results of your configuration use the command **show ip prefix-list**. To clear the counters shown in that command, use the **clear ip prefix-list** command.

Combine a prefix list with a route map to set attributes on the routes allowed in the prefix list. In the following example, prefix list Summary is used again. A route map sets the Med for those routes to 100 when they are advertised. It sets a Med of 200 for all other routes advertised. The route map is then applied to BGP neighbor 10.1.1.1 outbound:

```
route-map CCNP permit 10
match ip address prefix-list Summary
set metric 100
route-map CCNP permit 20
set metric 200
!
router bgp 65001
neighbor 10.1.1.1 route-map CCNP out
```

# BGP Authentication

BGP supports MD5 authentication between neighbors, using a shared password. It is configured under BGP router configuration mode with the command **neighbor** {*ip-address* | *peer-group-name*} **password** *password*. When authentication is configured, BGP authenticates every TCP segment from its peer and checks the source of each routing update. Most ISPs require authentication for their EBGP peers.

Peering succeeds only if both routers are configured for authentication and have the same password. If a router has a password configured for a neighbor, but the neighbor router does not, a message such as the following displays on the console while the routers attempt to establish a BGP session between them:

```
%TCP-6-BADAUTH: No MD5 digest from [peer's IP address]:11003 to
 [local router's IP address]:179
```

Similarly, if the two routers have different passwords configured, a message such as the following will display on the screen:

```
%TCP-6-BADAUTH: Invalid MD5 digest from [peer's IP
 address]:11004 to [local router's IP address]:179
```

# Verifying BGP

One of the best commands to verify and troubleshoot your BGP configuration is **show ip bgp** to see the BGP topology database**.** This is such an important command that it’s worth looking at in depth. The command output lists a table of all the networks BGP knows about, the next hop for each network, some of the attributes for each route, and the AS path for each route. The sample output from this command was taken from an actual Internet BGP peer.

```
route-server>show ip bgp
BGP table version is 22285573, local router ID is 12.0.1.28
Status codes: s suppressed, d damped, h history, * valid, > best, i
 - internal,
       r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network      Next Hop          Metric LocPrf Weight Path
* 3.0.0.0      12.123.137.124                  0     7018 2914 9304
 80 i
*>             12.123.1.236                    0     7018 2914 9304
 80 i
* 3.51.92.0/23 12.123.137.124                  0     7018 ?
*              12.122.125.4      2366          0     7018 ?
*>             12.123.1.236                    0     7018 ?
* 8.6.6.0/24   12.123.137.124                  0     7018 701 14744
 14744 14276 i
*              12.123.145.124                  0     7018 701 14744
 14744 14276 i
*>             12.123.1.236                    0     7018 701 14744
 14744 14276 i
```

Networks are listed in numerical order, smallest to largest. The first three columns list each route’s status. An asterisk (*) in the first column means that the route has a valid next hop. Some other options for the first column include the following:

- **“s” for suppressed:** BGP knows about this network but is not advertising it, usually because it is part of a summarized route.
- **“d” for dampened:** BGP can stop advertising a network that flaps (goes up and down) too often until it is stable for a period of time.
- **“h” for history:** BGP knows about this network but does not currently have a valid route to it.
- **“r” for RIB failure:** The route was advertised to BGP but it was not installed in the IP routing table. This might be because of another protocol having the same route with a better administrative distance.
- **“S” for stale:** Used with nonstop forwarding to indicate that the route is stale and needs to be refreshed when the peer is reestablished.

The second column has a greater-than sign (>) beside the route that was selected as the best path to that network. In the example, the second route was selected for network 3.0.0.0.

The third column is blank in the example, which means that the router learned all the routes from an external neighbor. A route learned from an IBGP neighbor would have an “I” in the third column.

The fourth column lists the networks. Those without a subnet mask, such as network 3.0.0.0, use their classful mask. As seen in the example, when the router learns about the same network from multiple sources, it lists only the network once.

The fifth column lists the next-hop address for each route. As you learned in the previous sections on BGP next hops, this might or might not be a directly connected router. A next-hop of 0.0.0.0 means that the local router originated the route.

If a Med value was received with the route, it is listed in the Metric column. Notice that the advertisement for network 3.51.92.0/23 from the router at 12.122.125.4 has a large Med value of 2366. Because the default Local Preference is used for each of the routes shown, no local preference value is displayed. The default Weight value of 0 is listed, however.

The ninth column shows the AS path for each network. Reading this field from left to right, the first AS number shown is the adjacent AS this router learned the route from. After that, the AS paths that this route traversed are shown in order. The last AS number listed is the originating AS. In the example, our router received an advertisement about network 3.0.0.0 from its neighbor AS 7018, which heard about it from AS 2914, which heard about it from AS 9304. And AS 9304 learned the route from AS 80, which originated it. A blank AS path means that the route was originated in the local AS.

**Note**

In the AS Path column, note that network 8.6.6.0 shows AS 14744 twice in its AS path list. Most likely AS 14744 has prepended an extra copy of its AS number to make the path through it less attractive than the path through other autonomous systems. In this case it did not work because the only paths to 8.6.6.0 this router knows about all go through AS 14744.

The last column shows how BGP originally learned about the route. Networks 3.0.0.0 and 8.6.6.0 show an "i" for their origin codes. This means that the originating router had a network statement for that route. Network 3.51.92.0 shows a "?" as its origin. This means that the route was redistributed into BGP; BGP considers it an "incomplete" route. You will likely never see the third possibility, an "e," because that means BGP learned the route from the Exterior Gateway Protocol (EGP), which is no longer in use.

Some other useful commands for verifying and troubleshooting BGP include

- **show ip bgp rib-failure:** Displays routes that were not inserted into the IP routing table and the reason they were not used.
- **show ip bgp summary:** Displays the memory used by the various BGP databases, BGP activity statistics and a list of BGP neighbors.
- **show ip bgp neighbors:** Displays details about each neighbor. Can be modified by adding the neighbor IP address.
- **show ip bgp neighbors** *address* **[received | routes | advertised]:** Lets you monitor the routes received from and advertised to a particular neighbor.

You can search for "Internet route servers" to find listings of BGP routers that enable public telnet access for viewing their BGP tables. Trying some of these commands on a public route server can help you become familiar with them.

# CHAPTER 7

# Branch Office Connectivity

The needs of branch offices are changing. This is due to the adoption of unified networks that support voice, video, and data; the consolidation of IT resources; and the physical mobility of many users.

## Branch Office Design Considerations

Some design considerations for branch offices include

- **Connectivity technologies:** What WAN options are available?
- **Resiliency:** How much downtime can the site tolerate? Are there alternate WAN paths available?
- **Routing:** Will the WAN support routing protocols?
- **Services:** Are services such as NAT, WAN optimization, and QoS needed at the branch?
- **Security and compliance:** What security is needed, where will it be placed, and how will that affect routing?
- **Mobility:** Do teleworkers use this branch for VPN access?

Strive for a consistent design across your branch offices, with a structured method of handling change management and configuration. Branch offices have different needs from campus locations, but you can still have a common design foundation by creating standard designs for different size offices. Each category of branch office is its own "place in the network." Categorize offices not only by the number of users they have, but also by how critical the branch is. The following office profiles are meant as a baseline, not a recommendation for every network.

### Small Branch Office Design

A small branch office typically leverages an ISR router to provide multiple services such as WAN and PSTN connectivity, NAT, WAN optimization, firewall, and DHCP. Its WAN connectivity might be a T1 primary link with a cable or DSL backup link using an IPsec VPN. You might run a routing protocol or simply use floating static routes. The infrastructure typically consists of Layer 2 switching—either internal to the router or using an external switch, computers, phones, and printers.

This design is cost-effective and provides minimum devices to manage. However, network resiliency suffers because the router is a single point of failure.

### Medium Branch Office Design

A medium-sized branch office requires some additional resiliency and network equipment. There typically are redundant WAN routers with dual connections to a private WAN using either MPLS or Frame Relay. The routers will be higher capacity devices but might still provide services such as firewall, NAT, DHCP, and WAN optimization. The network might use a FHP such as HSRP. The infrastructure typically consists of either Layer 2 or Layer 3 external switches, computers, phones, and printers.

This design is more resilient than the small office design. However, the dual paths add path control complexity and dynamic routing is needed to accomplish load sharing across the links. Documenting traffic flows becomes more important.

### Large Branch Office Design

A large branch office is similar to a campus design in that it typically uses a layered design with redundancy at all but the access layer. Stand-alone devices for firewalls and WAN optimization might be used, along with multilayer switches. This branch can provide services to other branches and can thus benefit from an MPLS WAN with its any-to-any connectivity. The infrastructure is engineered for high availability. It typically consists of dual WAN access routers, dual distribution switches, and dual firewalls.

This design adds more complexity to the routing structure and might require route redistribution and filtering. Regulatory requirements can lead to deploying overlay VPNs such as Dynamic Multipoint VPN (DMVPN) or Group Encrypted Transport VPN (GET VPN).

## Implementing Branch Offices

A full branch office implementation plan aims to integrate the new design without disrupting the existing users. It would include the following items:

- Deployment strategy
- Network diagrams
- Installation and site tests
- Site survey results
- Installation guidelines
- Device-specific configuration templates
- Test and acceptance plan
- Documentation of the new network

## Verifying Existing Services

Because a CCNP-level engineer should handle the device configuration, we start at that step of the plan. The implementation is likely an upgrade to an existing branch office network, so the first step is to identify and document the current configuration of every device and the services it provides. These services might include NAT, HSRP, ACLs, firewall, or redirection services such as PBR or WCCP. The IP address schema must also be documented.

To see the NAT settings, you might use the commands **show ip nat translations** and **show ip nat statistics**. To document DHCP, use the commands **show ip dhcp pool** and **show ip dhcp server statistics**. The **show access-lists** and **show ip interface** commands give you information about ACLs and where they are applied. To see what IOS firewall rules are in place, use the commands **show ip inspect interfaces** and **show zone-pair security**. Verify HSRP operation with **show standby brief**. The command **show ip policy** displays any PBR policies.

## Configuring a Backup DSL Connection

Assume that this branch already has a WAN connection, and you are adding redundancy by provisioning a backup DSL connection. Voice does not use all the available bandwidth on a phone line; it uses frequencies up to only approximately 3 kHz. DSL was created to use the space between 3 kHz and 1 MHz to send data traffic over a telephone local loop. Thus, both voice and data can be sent simultaneously over the same connection. (Some variants of DSL use the entire spectrum, however, so no voice can be sent.) DSL is a physical layer medium that extends between the subscriber's DSL modem and the provider's DSL Access Multiplexer (DSLAM).

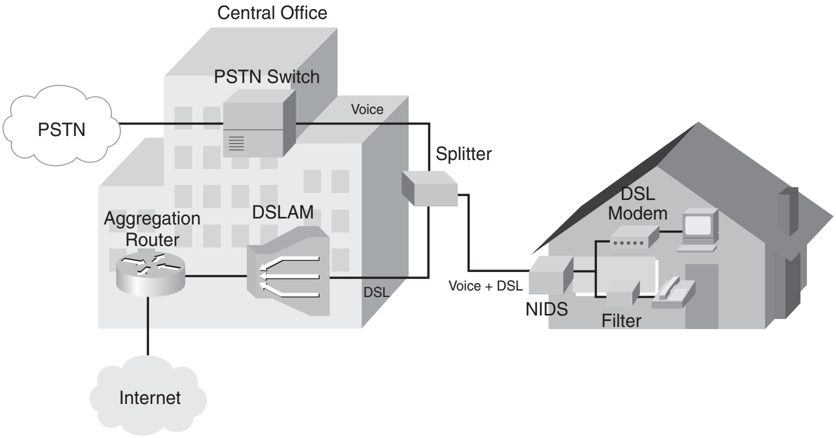
Asymmetrical DSL has higher downstream (from the provider's Central Office to the subscriber) bandwidth than upstream (from the subscriber to the CO.) Symmetrical DSL has the same bandwidth both downstream and upstream. You sometimes see these referred to as "asynchronous" and "synchronous" DSL.

The various types of DSL include

- **ADSL:** Asymmetric DSL supports both voice and data. Downstream bandwidth goes up to 8 Mbps; upstream goes up to 1 Mbps. Two other versions, ADSL2 and ADSL2+, provide 24 Mbps downstream and 1.5 Mbps upstream. The maximum distance from the CO is 18,000 feet, or 5.46 km.
- **VDSL:** Very-high-rate DSL can be either symmetric or asymmetric and can carry voice along with data. Maximum symmetric bandwidth is 26 Mbps; maximum asymmetric is 52 Mbps downstream and 13 Mbps upstream. The maximum distance from the CO is 4,500 feet, or 1.37 km.
- **SDSL:** Symmetric DSL carries only data, with a maximum for both downstream and upstream of 768 kbps. The distance limitation is 22,000 feet, or 6.7 km. It is a proprietary technology that uses only one twisted pair of wires.
- **HDSL:** High-data-rate DSL uses two twisted pairs of wires to achieve a maximum symmetrical bandwidth of 2.048 Mbps. Its maximum distance from the CO is 12,000 feet, or 3.7 km. HDSL carries only data, no voice.
- **G.SHDSL:** Symmetric High-speed DSL has a symmetrical data rate of 2.3 Mbps and the longest maximum distance: 28,000 feet, or 8.52 km. It also carries only data, no voice.

Figure 7-1 shows how ADSL components work together in a typical residential implementation. The telephone company's Central Office forwards both POTS and DSL data traffic over the same line to the subscriber. The line enters at the Network Interface Device (NIDS) and branches toward the telephone and the PC. A low-pass filter blocks everything but voice frequencies from reaching the phone. A DSL modem (or router with a DSL interface) forwards data to the PC. When the Central Office receives traffic from the subscriber, a splitter sends voice frequencies to the PSTN switch and DSL frequencies to the DSLAM. The DSLAM sends data traffic to a router for forwarding to the Internet.

**Figure 7-1 Components of an ADSL System**



Recall that DSL is a Layer 1—Physical Layer—technology. Following are the three methods of carrying data at Layer 2 over DSL:

- **Bridging:** Based on RFCs 1483 and 2684. Ethernet traffic is just bridged from the subscriber PCs, through the DSL modem and the DSLAM, to a provider router. Is not as secure or scalable as other methods.
- **Point-to-Point Protocol over Ethernet (PPPoE):** The most common Layer 2 method of carrying data over DSL. PPP traffic is encapsulated in Ethernet frames.
- **Point-to-Point Protocol over ATM (PPPoA):** PPP packets are routed over ATM between the subscriber equipment and the provider.

In the following example we use PPoA, which requires a CPE router because traffic is routed from the subscriber PCs to the aggregation router. The PPP session is established between the CPE router and the aggregation router. Either Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP) authentication can be used. Multiple users are supported if the CPE router is configured to do DHCP and NAT. Traffic between the CPE router and the aggregation router is encapsulated as ATM at Layer 2.

When configuring PPPoA you must set up the internal Ethernet interface, a dialer interface, NAT or PAT, DHCP, and a static default route. Because this is ATM, you must configure Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) information on the external interface to match that of the provider. The type of ATM encapsulation must be specified, PPPoA must be enabled, and the ATM interface must be linked to the virtual dialer interface. A dialer pool is associated with PVC. The final configuration is shown in the following examples.

Internal Ethernet interface:

```
interface FastEthernet0/1
 description Internal interface
 ip address 172.16.1.1 255.255.255.0
 ip nat inside
```

Dialer interface:

```
interface Dialer1
 ip address negotiated
 ip mtu 1492
 ip nat outside
 encapsulation ppp
 dialer pool 1
 ppp authentication chap
 ppp chap password 0 dslpass
```

Port address translation (PAT)

```
access-list 100 permit ip 172.16.1.0 0.0.0.255 any
ip nat inside source list 100 interface Dialer1 overload
```

DHCP:

```
ip dhcp pool Users
  import all
  network 172.16.1.0 255.255.255.0
  default-router 172.16.1.1
```

Static default route:

```
ip route 0.0.0.0 0.0.0.0 Dialer1
```

External ATM interface:

```
interface ATM1/0
 description DSL interface
 no ip address
 dsl operating-mode auto
 pvc 1/100
 encapsulation aal5mux ppp dialer
 dialer pool-member 1
```

To verify and troubleshoot the DSL configuration, use the commands **show dsl interface** atm *number*, **debug atm events**, **debug ppp authentication**, **show ip route**, **ping**, and **traceroute**.

## Configuring an IPsec VPN

IPsec is not covered in depth on the ROUTE exam, but you need to understand it well enough to verify the configuration and add routing across it. This sample branch uses an IPsec VPN to connect to the headquarters when the backup DSL link is active.

When IPsec establishes a VPN between two peer hosts, it sets up a security association (SA) between them. SAs are unidirectional, so each bidirectional data session requires two. The Internet Security Association and Key Management Protocol (ISAKMP) defines how SAs are created and deleted.

An IPsec transform set defines how VPN data will be protected by specifying the IPsec protocols that will be used. You can specify up to four transforms, and the algorithm to use with each. You can also configure either tunnel or transport mode. (Tunnel is default.)

You use a crypto ACL to identify traffic that should be protected by the IPsec VPN. Any traffic permitted in the ACL will be sent over the VPN. Traffic denied by the ACL will not be dropped; it will simply be sent normally.

A crypto map pulls together the transform sets and crypto ACLs and associates them with a remote peer. After the crypto map is configured, it must be applied to an interface for it to take effect. It is applied at the *outgoing* interface—the one that VPN traffic uses to reach the other end of the VPN. You might need to use a static route or otherwise adjust your routing to force traffic bound for the VPN destination networks to use the correct outgoing interface.

To verify the IPSEC VPN, use the following commands:

- **show crypto map:** Shows the crypto ACLs, any peers, and the interface where the crypto map is applied
- **show crypto isakmp sa:** Shows information about the ISAKMP security associate negotiation process
- **show crypto IPsec sa:** Shows the settings used by current SAs, including tunnel status and peers

## Configuring a Floating Static Route

The IPsec tunnel can be used solely as a backup link, or you can load balance between it and the primary link. To use it as a backup link, you can configure a floating static route. A floating static route is one with an administrative distance greater than the primary route. If the primary route is active, the static route will not be placed into the routing table due to its higher AD. But when the primary route is down, the static route will be used.

The command syntax for a floating static route is: **ip route** *destination-network next-hop-address administrative-distance*. You can find the AD of the primary route by using the command **show ip protocols**. EIGRP has an AD of 90, so you might use 100 as the AD of a floating static route when the primary route is learned via EIGRP.

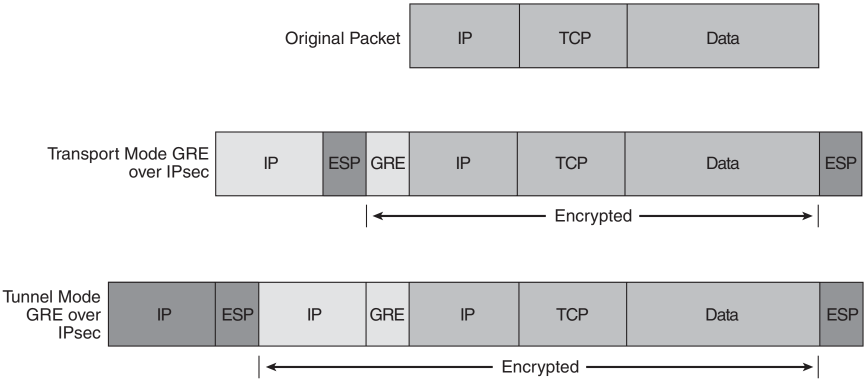
## Configuring Dynamic Routing over a GRE Tunnel

To use the IPsec tunnel as an "always on" connection, you need to send routing updates over it. However, IPsec VPNs do not carry broadcast or multicast traffic. You need to create a tunnel within the IPsec tunnel to carry the routing traffic. Four ways to do this include

- **DMVPN:** Creates multipoint tunnels on-demand. Good for scenarios when spoke-to-spoke connections are needed.
- **GET VPN:** Creates encrypted multipoint tunnels on-demand. Good for scenarios when secure spoke-to-spoke connections are needed.
- **Virtual Tunnel Interface (VTI):** Creates an always-on tunnel that carries unicast and multicast traffic. Enables you to configure the routing protocol on the tunnel interface, saving the 4 extra bytes required for a GRE header.
- **Generic Routing Encapsulation (GRE):** GRE is a tunneling protocol that can support multiple Layer 3 protocols. It enables the use of multicast routing protocols across the tunnel. It adds a 20-byte IP header and a 4-byte GRE header. GRE does not encrypt traffic or use any strong security measures to protect the traffic. GRE can be used along with IPsec to provide data source authentication, data confidentiality, and assurance of data integrity. GRE over IPsec tunnels are typically configured in a hub-and-spoke topology over an untrusted WAN to minimize the number of tunnels that each router must maintain.

In this section, we configure a GRE tunnel to carry EIGRP traffic over the IPSEC tunnel. This basically creates a tunnel within a tunnel, as shown in Figure 7-2.

**Figure 7-2 GRE over IPSEC Tunnel**



Configuring a GRE tunnel is fairly easy. Follow these steps on the routers at each end of the tunnel:

1. Create a loopback interface to use as the tunnel endpoint. Using a loopback rather than a physical interface adds stability to the configuration.
2. Create the GRE tunnel interfaces.
3. Add the tunnel subnet to the routing process so that it exchanges routing updates across that interface.
4. Add GRE traffic to the crypto access list, so that IPsec encrypts the GRE tunnel traffic.

The following example shows a tunnel interface configured for GRE. The **mode** command is shown only as a reference; because it is the default, it would not normally appear in the configuration:

```
interface Tunnel1
 ip address 172.16.5.2 255.255.255.0
 tunnel source Serial0/0
 tunnel destination 10.1.1.1
 tunnel mode gre ip
```

To verify the configuration, use the **show ip route** command and look for the remote routes in the routing table. They should have a next hop of the tunnel interface. A **traceroute** shows traffic going across the tunnel. The

**show crypto ipsec sa** command output shows increased traffic as the traceroute goes through the tunnel.

## Load Sharing with EIGRP

In the EIGRP chapter you saw how EIGRP can load balance across unequal-cost links. Because you have an always-on IPsec tunnel, you might want to use that in addition to your primary route. Use the **variance** *multiplier* command under EIGRP configuration mode. Look in the EIGRP topology table to find the metric for the best route and the secondary route. Determine what you need to multiply the best route's metric by for it to be more than the secondary route's metric. That is the variance multiplier number that you should use to configure EIGRP to load balance over both paths.

# CHAPTER 8

# Mobile Worker Connectivity

Mobile workers might be in a home office, with an always-on secure connection to the corporate network, managed centrally by the corporation (business-ready mobile worker.) Or they might be truly mobile, connecting via a laptop or public computer to the corporate network (traditional mobile worker.) In planning for either type of mobile worker, consider the network access technologies and infrastructure services needed, such as the following:

- **Bandwidth requirements:** Because mobile workers use the same applications as office workers—email, other applications, voice, video, real-time collaboration—they need sufficient bandwidth. Typical remote access technologies include residential cable, DSL, and wireless.
- **Connection security:** Use site-to-site VPNs for permanent home users and remote access VPNs for mobile users. These can be either IPsec or Secure Sockets Layer (SSL) VPNs.
- **Corporate security:** Because a remote environment is less controlled than an office environment, use firewalls, intrusion prevention services (IPS), and URL filtering to protect the corporate network from remote users.
- **User authentication:** Use network access control (NAC), AAA servers, or other authentication mechanisms to protect access to corporate resources.
- **QoS:** If voice and video are used, determine how you will prioritize that traffic, and how you will address the differences in upload and download speeds of common broadband connections.
- **Management:** Support for remote workers is more complex when they are not under corporate control. Provide methods to push security policies and updates to mobile workers.

Table 8-1 compares the two types of mobile workers and the ability to offer these services to each.

**Table 8-1 Comparison of Traditional and Business-Ready Mobile Workers**

| | Traditional Mobile Worker | Business-Ready Mobile Worker |
|---|---|---|
| **Access to applications and services** | Basic | Full |
| **Voice and video support** | Limited to none | Yes |
| **QoS** | No (best effort) | Yes |
| **Security** | Relies on end-user | Controlled by corporate IT |
| **Remote management** | No | Yes |

## Components of a Mobile Worker Solution

There are three major groups of components in a mobile worker solution: corporate devices, devices located at the remote site, and optional additional services.

The corporate components include headend routers, devices to terminate the VPNs such as ASA firewalls, authentication services, and central management devices.

The remote site components for a Business-ready solution include broadband access, a VPN router with QoS capabilities, and a computer. There can also be a wireless access point, an IP phone, and a video telephony camera. A traditional mobile worker might have only a laptop, perhaps with a softphone on it.

The corporation might offer additional services to the mobile worker, such as IP telephony, voice mail, or contact center.

The corporation also needs to decide which type of VPN services to offer: IPsec, SSL, or both. IPsec requires a client on the endpoint (computer or router), but it is a well-proven technology that provides full access to all network applications. It is a good choice for mobile workers using company-managed devices.

SSL is also a well-proven technology and can be used with or without a client. When used from a web browser, it provides access even from nonmanaged devices, and the portals can be customized to provide appropriate access for employees or business partners. Some applications might not work well through the web portal, however.

## Implementing a Mobile Worker Solution

Some of the things you need to implement in a mobile worker solution are

- The VPN solution, either IPsec or SSL
- A firewall solution, for stateful Layer 3–7 protection
- Intrusion prevention to defend against worms and viruses
- Wireless security, if wireless is used, with encryption and 802.1x authentication.
- QoS for voice and video
- Ports at the remote site for printers and other devices
- PoE ports for IP phones

Cisco Easy VPN can simplify the deployment of all these services and policies to remote users. Easy VPN enables a server to push down VPN configuration to a client. It is a way to create site-to-site VPNs without manually configuring each remote router. Thus it is good for remote sites without technical support. It can also be used with software clients for remote users.

Cisco Easy VPN dynamically handles the following items:

- Negotiating VPN tunnel parameters
- Establishing the VPN tunnel based on those parameters
- NAT, PAT, or ACL configuration
- User authentication
- Managing encryption and decryption keys
- Authenticating, encrypting, and decrypting traffic

Cisco Easy VPN has two components: a server and a remote client. The *Easy VPN Server* can be a Cisco router, ASA Firewall, or Cisco VPN concentrator. It contains security policies and pushes those to remote clients. The *Easy VPN Remote* can be a Cisco router, ASA Firewall, a hardware client, or a software client. It contacts the server and receives policies from it to establish the IPsec tunnel. The steps to configure the headend for Easy VPN are

1. Allow IPsec traffic through the edge firewall or access list.
2. Create the IP address pool for the VPN clients.
3. Verify the IPsec VPN configuration.

4. Ensure that corporate devices have routes to the VPN subnets.
5. Tune NAT to bypass VPN traffic.

## Allow IPsec Traffic

IPsec uses the following ports and protocols:

- Encapsulating Security Payload (ESP)—IP protocol 50
- Authentication Header—IP protocol 51
- ISAKMP—UDP port 500
- NAT Traversal (NAT-T)—UDP port 4500

These protocols and ports must be allowed in through the firewall from the outside for an IPsec tunnel to be established. If your network is protected by a firewall appliance or module, coordinate these changes with your security personnel. If your network is protected by an IOS firewall, determine whether you have a zone-based firewall or a classic IOS firewall. The command **show zone-pair security** produces output if you have a zone-based firewall. The command **show ip inspect interfaces** produces output if you have a classic IOS Firewall. The command **show access-lists** will show you any access lists associated with the firewall that will need to be modified to allow IPsec traffic.

Fortunately, configuring IPsec and firewalls is outside the scope of this exam, so you need to understand it only in concept.

## Create the Address Pool

VPN clients have a public Internet address until they reach your network but need a private inside address to access network resources. The addresses can be given by a router acting as a DHCP server, by other DHCP servers, or by AAA servers.

Plan your address pool so that you have enough client addresses for the maximum number of users and so that the subnets can be summarized when their routes are advertised to internal devices. Be sure that the client address range is not used anyplace else in the network.

To configure a DHCP pool on a router, use the command:

```
ip local pool pool-name first-address last-address [mask subnet-
  mask]
```

For the exam, you are not required to add this IP address pool to the VPN configuration. In theory you would request a change ticket for your VPN support team to make that change.

## Verify the IPsec VPN

You are likewise required to understand only the components of a VPN, not to know how to configure it. The two main things you need to know are the function of a crypto map and the commands to verify IPsec connectivity.

A crypto map is described in Chapter 7, "Branch Office Connectivity." It basically groups the VPN settings to apply them to an interface. It consists of the crypto ACL, which defines the traffic to be encrypted, security policies such as how to protect the data, and other security parameters such as peer IP address or tunnel lifetime. The crypto map is applied to the exit interface for traffic needing to be protected.

Commands to verify IPsec connectivity include

- **show crypto map:** Shows the crypto ACLs, any peers, and the interface where the crypto map is applied
- **show crypto isakmp sa:** Shows information about the ISAKMP security associate negotiation process
- **show crypto IPsec sa:** Shows the settings used by current SAs, including tunnel status and peers
- **show crypto engine connections active:** Shows the status of any IPsec tunnels

## Route to the VPN Subnets

Internal resources need to have a route to the VPN subnets. One way to do this is to make the VPN subnet a part of an existing subnet applied to a router interface. This subnet should already be advertised to the rest of the network, so no changes to routing are needed. The disadvantage is that you need to enable proxy-ARP on the router interface connected to the subnet.

A second option is to use a separate subnet for the VPN clients and then inject that subnet into the routing protocol. An IPsec feature called Reverse Route Injection (RRI) does that dynamically. It injects a host route into the routing table for each client while they are connected and then withdraws that route when they disconnect. You would then redistribute those routes into the routing protocol. Alternatively, you can create a static route for the VPN subnets and redistribute the static route into your routing protocol.

## Tune NAT

VPN traffic should bypass the NAT process as it leaves your network. If the same edge router is doing NAT and terminating the VPNs, the router processes the NAT service before the IPsec service. It is simple to configure NAT bypass because NAT uses an access list to identify which traffic to translate. Modify the access list to deny traffic destined to the VPN subnet. Be sure to permit all other traffic. You can then match that access list in a route-map and use the route-map to modify the NAT traffic. Your final configuration might look something like this:

```
access-list 101 deny ip any vpn-subnet vpn-mask
access-list 101 permit ip any any
!
route-map NAT permit 10
match ip address 101
!
ip nat inside source route-map NAT pool NAT-POOL overload
```

Be sure to verify your configuration by checking the IP address of a VPN client, pinging internal resources, checking the routing table, and the IPsec SAs.

# CHAPTER 9

# IPv6 Introduction

IPv6 is an extension of IP with several advanced features:

- Larger address space.
- No more need for NAT.
- Simpler header for increased router efficiency.
- No more broadcasts.
- Stateless autoconfiguration.
- Built-in support for Mobile IP.
- Built-in support for IPsec security.
- Rich transition features.
- Easy IP address renumbering.
- Capability to have multiple addresses per interface.
- Routers create link-local addresses for use by IGPs.
- As with IPv4, the addresses can be provided by the ISP or can be provider independent.

The primary adoption of IPv6 is driven by the need for more addresses. Given the growth in Internet use and the emergence of large groups of Internet users worldwide, this is a significant requirement. Another reason to use IPv6 is growth in the size of the current Internet routing table. IPv4 addresses are not summarized enough to keep the size down, increasing the load on Internet routers. Additionally, although the use of NAT has postponed the need for IPv6, it breaks TCP/IP's end-to-end networking model.

## IPv6 Addressing

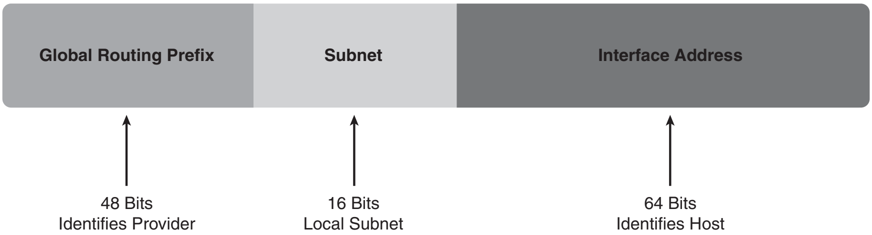
IPv4 addresses are 32-bits long and written in dotted decimal, whereas IPv6 addresses are 128 bits and written in hexidecimal. They are typically divided into a 64-bit network portion and a 64-bit host portion. The first 48 bits of the network portion are considered as Global Address Space. These bits consist of the following elements (see Figure 9-1):

- The first three bits (/3) of a unicast address are always 001.
- The next 13 bits (/16) identify the Top-Level Aggregator (TLA); the upstream ISP.
- The next 24 bits (/40) identify the next-level aggregator, or regional ISP.

Enterprises are assigned /48 addresses and have 16 bits of subnetting available.

The host portion of the address is last 64 bits. The subnet mask is specified using Classless Interdomain Routing (CIDR) notation. Figure 9-1 shows the address components.

**Figure 9-1 IPv6 Address Structure**



## Simplifying an IPv6 Address

There are two ways to shorten the representation of an IPv6 address. Take the example address 2001:0000:0001:0002:0000:0000:0000:ABCD.

- Leading zeros can be omitted. Doing this would shorten the preceding address to 2001:0:1:2:0:0:0:ABCD.
- Sequential zeros can be shown as double colons. *This is allowed only once per address*. Adding this would simplify the above address even further, to 2001:0:1:2::ABCD.

For the exam, be sure that you can distinguish between correct and incorrect IPv6 addresses. For instance, the address 2001::1:2::ABCD is incorrect because it uses double colons twice.

## Special Addresses

IPv6 does not support broadcasts but replaces broadcasts with multicasts. IPv6 also uses Anycast, which involves using the same address on two devices. Anycast can be used to implement redundancy and has been back-ported to IPv4.

Each IPv6 system must recognize the following addresses:

- Its unicast addresses
- Link local address (begins with FE80/10)
- Loopback (::1/128)
- All-nodes multicast (FF00::1)
- Site-local multicast (FF02::2)
- Solicited-nodes multicast (FF02::1:FF00/104)
- Default route (::/0)

Additionally, some systems also use the following addresses:

- IPv4 compatible address (::/96 | 32-bit, IPv4 address).
- Second unicast address shared with another system (anycast).
- Additional multicast groups.
- Routers must support subnet-router anycast (all zeros EUI-64).
- Routers must support local all-routers multicast (FF01::2), link-local (FF02::2), and site-local (FF05:2).
- Routers must support routing protocol multicast groups.

## IPv6 Host Addressing

An IPv6 host can obtain an IP address by manual assignment, by manually assigning the network address only, by using stateless autoconfiguration, or by using DHCPv6. IPv6 is not enabled by default on Cisco routers. To enable IPv6 routing, the command is **ipv6 unicast-routing** at the global configuration mode.

To ping any IPv6 address, including link-local addresses, use the command **ping ipv6** *destination-address* **source** *exit-interface*. Note that you must specify a source.

## Manual IP Address Assignment

To manually assign an IPv6 address to a router interface, use the command **ipv6 address** *ipv6-address/prefix-length*. The following example shows a router interface with two IPv6 addresses. In the first address, note leading zeros are omitted in two of the quartets. In the second address, note the use of the double colons:

```
RouterA# configure terminal
RouterA(config)# ipv6 unicast-routing
!
RouterA(config)# interface fastethernet0/0
RouterA(config-if)# ipv6 address
 2001:0:aabb:1:2222:3333:4444:5555/64
RouterA(config-if)# ipv6 address 2001:0:aabb:2::1 /64
```

## Manual Network Assignment

The router can create its own IPv6 address when it knows its network. If the end system has a 64-bit MAC address, it concatenates the network prefix and its MAC address to form an IPv6 address. If the end system has a 48-bit MAC address, it flips the global/local bit (the 7th bit) and inserts 0xFFEE into the middle of the MAC address. The resulting 64-bit number is called the EUI-64 address. The prefix and EUI-64 address are concatenated to form the host IPv6 address. The command is **ipv6 address** *ipv6-prefix::/prefix-length* **eui-64**.

The following example shows this command and the resulting link-local and global unicast address. Note the interface MAC address and how it relates to the IPv6 addresses.

```
RouterA(config)# interface fastethernet0/0
RouterA(config-if)# ipv6 address 2001:8:1234:aabb::/64
!
R1# show int fa 0/0
FastEthernet0/0 is up, line protocol is up
 Hardware is MV96340 Ethernet, address is 001d.a188.33c1 (bia
 001d.a188.33c1)
!
R1# show ipv6 int fa0/0
FastEthernet0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::21D:A1FF:FE88:33C1
 [TEN]
 No Virtual link-local address(es):
 Global unicast address(es):
  2001:8:1234:AABB:21D:A1FF:FE88:33C1, subnet is
 2001:8:1234:AABB::/64 [EUI/TEN]
 Joined group address(es):
  FF02::1
  FF02::2
```

## Stateless Autoconfiguration

One big benefit of IPv6 is stateless autoconfiguration, the capability of a host to automatically acquire an IP address without needing DHCP. It uses its link-local address and the Neighbor Discovery Protocol (NDP) to do this.

Each device creates a link-local address for itself based on the prefix FE80:: and the interface MAC address. This address is only valid on the local network. It then uses NDP to make sure that the address is unique.

**Note**

IGP routing protocols use the link-local address to form neighbor relationships. It is also the next-hop address that is installed in the routing table by IGPs.

NDP has several functions in IPv6, including the following:

- **Duplicate Address Discovery (DAD):** The host uses Neighbor Solicitation (NS) to send a message to its own address. No response means that the link-local address is unique.
- **Neighbor Discovery:** Similar to ARP, the host discovers the link-local address of neighbors using an NS message. This is ICMP type 135. Neighbors respond with an ICMP type 136 message.
- **Router Discovery:** Ipv6 routers periodically send Router Advertisements (RAs) listing the network prefix. When a host comes online it immediately sends a Router Solicitation (RS) message, asking for prefix information, rather than waiting for the RA. This is sent to the All-routers multicast address.

To configure stateless autoconfiguration, use the interface command **ipv6 address autoconfig**. Acquiring an address involves the following steps:

1. The host creates a link-local address.
2. It sends an NS message to its link-local address out the interface.
3. If there is no reply, DAD declares the address unique.
4. If the host doesn’t receive an RA, it sends an RS.
5. A router on the subnet sends an RA, listing its interface prefix.

6. The host uses that prefix and the interface MAC address to create its IPv6 address.

Use the command **show ipv6 interface** to verify your configuration. The following example shows this command and the resulting IPv6 address.

```
R4(config)# int fa 0/0
R4(config-if)# ipv6 address autoconfig
!
R4# show ipv6 int fa0/0
FastEthernet0/0 is up, line protocol is up
 IPv6 is enabled, link-local address is FE80::21D:A1FF:FE6C:D238
 No Virtual link-local address(es):
 Global unicast address(es):
  2001:8::21D:A1FF:FE6C:D238, subnet is 2001:8::/64 [EUI/CAL/PRE]
   valid lifetime 2591828 preferred lifetime 604628
```

## Renumbering

IPv6 supports easy network renumbering. Note in the previous example that lifetimes are listed for the subnet address. When it is time to change the subnet, you can configure the router to advertise the old prefix with a short lifetime and a new prefix with a longer lifetime. You can even configure the router to expire a prefix at a certain date and time. The router sends out an RA with both prefixes and their lifetimes. Hosts then update their addresses. Anyone who has had to renumber a large range of IPv4 addresses can testify to how useful this feature is!

# IPv6 Routing

Routing with IPv6 will seem very familiar to you.  The same IGPs—RIP, EIGRP, and OSPF—are used as in IPv4; they have been adapted to carry IPv6 routes.  BGP extensions allow it to do IPv6 routing. The same rules for metric and administrative distance apply. The commands are very similar too.  The main difference in commands is that you need to specify that the command pertains to IPv6, since IPv4 is the default.  One big configuration difference is that the **network** command is no longer used by IGPs to initiate routing.  It is enabled at each interface instead. BGP does still use the **network** command to designate which networks to advertise.

## Static Routing

Static routing with IPv6 works exactly like it does with version 4. Aside from understanding the address format, there are no differences. The syntax for the IPv6 static route command is

```
Router(config)# ipv6 route ipv6-prefix/prefix-length {ipv6-address |
 interface-type
interface-number [ipv6-address]} [administrative-distance]
[administrative-multicast-distance | unicast | multicast] [tag tag]
```

The following examples show the command in context as it might be applied. The first line shows a recursive static route that lists a next-hop address. The second line shows a directly connected static default route that lists an outbound interface. The third line shows a fully specified static route, which lists both the next-hop address and the outbound interface.

```
RouterA(config)# ipv6 route 2001:0:1:2::/64 2001:0:1:1::1
RouterA(config)# ipv6 route ::/0 serial1/0/0
RouterA(config)# ipv6 route 2001:0:1:2::/64 serial1/0/1
 2001:0:1:1::1
```

Verify your configuration with the command **show ipv6 route**.

## RIPng for IPv6

RIP next generation (RIPng) is the IPv6 version of RIP and is defined in RFC 2080. Like RIPv2 for IPv4, RIPng is a distance vector routing protocol that uses a hop count for its metric, has a maximum hop count of 15, and uses split horizon. It uses UDP and still has an administrative distance of 120. RIPng also uses periodic multicast updates—every 30 seconds—to advertise routes. The multicast address is FF02::9. (RIP v2 uses IPv4 address 224.0.0.9.) The source address of RIPng updates is the link-local address of the outbound interface.

There are two important differences between the old RIP and the next-generation RIP. First, RIPng supports multiple concurrent processes, each identified by a process number. (This is similar to OSPFv2.) Second, RIPng is initialized in global configuration mode and then enabled on specific interfaces. There is no **network** command in RIPng.

The following example shows the syntax used to apply RIPng to a configuration. Notice that the syntax is similar to traditional RIP. You must first enable IPv6 routing. The global command to enable RIPng is optional; the router creates it automatically when the first interface is enabled for RIPng. You might need the command for additional configuration, such as disabling split horizon for a multipoint interface, as shown in the example.

```
Router(config)# ipv6 router rip process
Router(config-rtr)# no split-horizon
!
Router(config)# interface type number
Router(config-if)# ipv6 rip process enable
```

Like RIP for IPv4, troubleshoot RIPng by looking at the routing table (**show ipv6 route [rip]**), by reviewing the routing protocols (**show ipv6 protocols**), and by watching routing updates propagated between routers (**debug ipv6 rip**).

## EIGRP for IPv6

EIGRP has been expanded to support IPv6, although you need to verify that your specific version of IOS is capable of doing this. EIGRP for IPv6 is based on the IPv4 version, and the two can run in tandem on the same router and on the same interfaces. EIGRP is still an advanced distance vector routing protocol that uses a complex metric. EIGRP still has a reliable update mechanism and uses DUAL to retain fall-back paths. Like EIGRP in IPv4, it sends multicast hellos every 5 seconds. (But the multicast address is now FF02::A.) Messages are exchanged using the interface link-local address as the source address. This leads to the possibility that two routers with interfaces on different subnets can now form an EIGRP adjacency.

Like RIPng, there is no more **network** command; EIGRP routing is enabled at each interface. You must assign a router ID in the format of a 32-bit IPv4 address. It does not need to be a routable address. One important thing to note is that the protocol starts off in the shutdown state. You must **no shut** it before routing will begin. Auto-summarization is disabled by default in IPv6 EIGRP.

The following example shows how to enable IPv6 EIGRP:

```
Router(config)# ipv6 unicast-routing
!
Router(config)# ipv6 router eigrp AS
Router(config-rtr)# router-id ipv4-address
Router(config-rtr)# no shut
!
Router(config)# interface type number
Router(config-if)# ipv6 eigrp AS
```

Like EIGRP for IPv4, troubleshoot by looking at the routing table (**show ipv6 route**), by reviewing the routing protocols (**show ipv6 protocols**), and by monitoring neighbors (**show ipv6 eigrp neighbors**).

IPv6 EIGRP can summarize routes at the interface, and the stub feature is also available, just as with the IPv4 version. The following example shows a sample configuration for IPv6 EIGRP, with both summarization and stub routing enabled. Notice that the routing protocol is enabled under each interface:

```
RouterA(config)# ipv6 router eigrp 1
RouterA(config-rtr)# router-id 10.255.255.1
RouterA(config-rtr)# stub connected summary
!
RouterA(config)# interface fastethernet0/0
RouterA(config-if)# description Local LAN
RouterA(config-if)# ipv6 address 2001:0:1:1::2/64
RouterA(config-if)# ipv6 eigrp 1
!
RouterA(config-if)# interface serial 1/0/1
RouterA(config-if)# description point-to-point line to Internet
RouterA(config-if)# ipv6 address 2001:0:1:5::2/64
RouterA(config-if)# ipv6 eigrp 1
RouterA(config-if)# ipv6 summary address eigrp 1 2001:0:1/24
```

## OSPFv3

OSPFv3 was one of the first routing protocols available for IPv6 and because of its open-standard heritage, it is widely supported in IPv6. OSPFv3, which supports IPv6, is documented in RFC 2740. Like OSPFv2, it is a link-state routing protocol that uses the Dijkstra algorithm to select paths. Routers are organized into areas, with all areas touching area 0.

OSPFv3 routers use the same packet types as OSPFv2, form neighbors in the same way, flood and age LSAs identically, and support the same NBMA topologies and techniques such as NSSA and on-demand circuits. It can run concurrently with OSPFv2 because each version maintains its own databases and runs a separate SPF calculation.

OSPFv3 differs from its predecessors principally in its new address format. OSPFv3 advertises using multicast addresses FF02::5 and FF02::6 but uses its link-local address as the source address of its advertisements. This means that OSPF can form adjacencies with neighbor routers that are not on the same subnet. Multiple instances of OSPFv3 can run on each link. Authentication is no longer built in but relies on the underlying capabilities of IPv6.

OSPFv3 configuration is similar to RIPng and EIGRP. The routing process is created and routing properties are assigned to it. As with EIGRP, you must create a router ID in 32-bit dotted decimal format. The router ID is not automatically created in OSPFv3.  Interfaces are associated with the OSPF process under interface configuration mode.

Assuming that **ipv6 unicast-routing** and interface IP addresses are already in place, the commands to implement basic OSPFv3 are shown in the following example.

```
Router(config)# ipv6 router ospf process-id
Router(config-rtr)# router-id 32bit-address
!
Router(config-rtr)# interface type number
Router(config-if)# ipv6 ospf process-id area area
```

As illustrated in the following example, route summarization is still configured under the OSPF routing process. Stub routing is also configured under the routing process, using the same commands as with OSPFv2. The default costs and interface priorities can be overridden at each interface. This example shows how these commands might look on an actual router.

```
RouterA(config)# ipv6 unicast-routing
!
RouterA(config)# ipv6 router ospf 1
RouterA(config-rtr)# router-id 10.255.255.1
RouterA(config-rtr)# area 1 range 2001:0:1::/80
RouterA(config-rtr)# area 1 stub no summary
!
RouterA(config-rtr)# interface fastethernet0/0
RouterA(config-if)# ipv6 address 2001:0:1:1::2/64
RouterA(config-if)# ipv6 ospf 1 area 1
RouterA(config-if)# ipv6 ospf cost 10
RouterA(config-if)# ipv6 ospf priority 20
!
RouterA(config-if)# interface serial 1/0/0
RouterA(config-if)# ipv6 address 2001:0:1:5::1/64
RouterA(config-if)# ipv6 ospf 1 area 0
```

Troubleshoot OSPFv3 just like OSPFv2. Start by looking at **show ipv6 route** to verify routes have been advertised. Assuming the route is in the routing table, test reachability using **ping ipv6**. You can also look at the OSPF setup using **show ipv6 ospf** *process* **interface**, **show ipv6 ospf**, or **show ipv6 ospf database**.

## MP-BGP for IPv6

Multiprotocol BGP (RFC 2858) involves two new extensions to BGP4 that enable BGP to carry reachability information for other protocols, such as IPv6, multicast IPv4, and MPLS. The extensions enable NEXT_HOP to carry IPv6 addresses and NLRI (network layer reachability information) to an IPv6 prefix. An **address-family** command is added to the BGP configuration to enable this.

Router ID must be manually configured in an all-IPv6 implementation and is a 32-bit dotted decimal number. Unlike the IGPs, configuration is done under the BGP router configuration mode, not at the interface. Neighbors are configured under the global BGP configuration mode but must be activated under the IPv6 address family mode. Any policies or networks relevant to this mBGP extension are also configured under the address family.

The following example shows the BGP commands as they might be applied.

```
RouterA(config)# ipv6 unicast-routing
!
RouterA(config)# router bgp 65000
RouterA(config-rtr)# router-id 10.255.255.1
RouterA(config-rtr)# neighbor 2001:0:1:1:5::4 remote-as 65001
RouterA(config-rtr)# address-family ipv6 unicast
RouterA(config-rtr-af)# neighbor 2001:0:1:5::4 activate
RouterA(config-rtr-af)# network 2001:0:1::/48
```

To verify your BGP configuration, use the commands **show bgp ipv6 unicast summary** and **show ipv6 route bgp**.

## RIPng Redistribution

You can run multiple instances of RIPng on the same router by giving them different process tags in the global RIP configuration. Be sure to use the correct tag when you enable RIP on each interface.

**Note**

The process tag is case-sensitive.

An interesting thing about RIPng is that the multiple instances exchange routing information with each other if they use the same multicast group and UDP port number. To keep the route information separate, you need to configure each instance to use a different port. Do this under the global

RIPng configuration mode for each process. You can keep the default multicast group:

```
R1(config)# ipv6 router rip Process1
R1(config-rtr)# port 1010 multicast-group ff02::9
!
R1(config)# ipv6 router rip Process2
R1(config-rtr)# port 1011 multicast-group ff02::9
```

Remember to do this on all routers in the RIP process. If you need to share routes between the two processes, you can control the redistribution by configuring it on the desired routers. You can further control it by using a route map to modify the redistribution. With a route map you can set the seed metric for specific routes, or filter routes that should not be redistributed, just as you can with IPv4 routing. The command to redistribute Process2 routes into Process1 would look like this:

```
R1(config)# ipv6 router rip Process1
R1(config-rtr)# redistribute rip Process2 route-map Filter
```

Redistribution between other IPv6 routing protocols use the same commands and follow most of the same rules as IPv4 routing protocols.

## Integrating IPv4 and IPv6

There are several strategies for migrating from IPv4 to IPv6. Each of these strategies should be considered when organizations decide to make the move to IPv6 because each has positive points to aiding a smooth migration. It should also be said that there does not have to be a global decision on strategy—your organization might choose to run dual-stack in the United States, go completely to IPv6 in Japan, and use tunneling in Europe. The transition mechanisms include

- **Dual stack:** Running IPv6 and IPv4 concurrently on the same interface.
- **Tunneling:** Routers that straddle the IPv4 and IPv6 worlds encapsulate IPv6 traffic inside IPv4 packets.
- **Translation:** Using an extension of NAT, NAT protocol translation (NAT-PT), to translate between IPv4 and IPv6 addresses.

## Tunneling IPv6 over IPv4

A tunnel serves as a virtual point-to-point link between IPv6 domains. It doesn't matter what the underlying IPv4 structure is if there is IP reachability between the tunnel endpoints. This exam covers five ways to tunnel IPv6 over IPv4:

- Manual Tunnels
- GRE Tunnels
- 6to4 Tunnels
- IPv4-Compatible IPv6 Tunnels
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

### Manual Tunnels

When you manually create the tunnel, the source and destination IP addresses are IPv4 addresses because IPv4 is the transport protocol. You might want to use loopback addresses for increased stability. IPv6 addresses go on the tunnel interfaces because IPv6 is the passenger protocol. Because IPv6 considers the tunnel a point-to-point link, the address of each end of the tunnel is in the same subnet. Include the command **tunnel mode IPv6IP** in tunnel configuration mode to enable IPv6 over IP encapsulation.

To verify your configuration you can use the commands **debug tunnel** or **show interface tunnel** *int-number*.

### GRE Tunnels

GRE is the default tunnel mode for Cisco routers. It provides more flexibility because it is protocol-agnostic. It can carry multiple protocols and can use multiple protocols for its transport, including IPv6 and routing protocols.

Configuring an IPv4 GRE tunnel to carry IPv6 traffic is the same as configuring a manual tunnel except you do not have to specify the tunnel mode because GRE is the default. You can allow a routing protocol on the tunnel interface, too. The process is the same as enabling it on a physical interface.

To configure a completely IPv6 GRE tunnel, use IPv6 interface addresses as the tunnel source and destination. Give the tunnel endpoints IPv6 addresses, too. You need a command to identify that the transport protocol is IPv6. That command, given in tunnel configuration mode, is **tunnel mode gre ipv6**.

## 6to4 Tunnels

This technique dynamically creates tunnels that IPv6 considers point-to-multipoint interfaces. You use the reserved prefix 2002::/16 in your IPv6 domain and then add the IPv4 address of the dual-stack router on the other side of the IPv4 domain as the next 32 bits of the network address. This means you need to translate that IP address into hexadecimal.

When IPv6 traffic arrives at an edge dual-stack router with a destination IPv6 prefix of 2002::/16, the router looks at the first 48 bits, derives the embedded IPv4 address from them, and uses it to determine the packet destination. The router then encapsulates the IPv6 packet in an IPv4 packet with the extracted IPv4 address as the packet destination.

Configure a tunnel as before, using IPv4 addresses as the source, but do not manually specify a destination. Give the tunnel an IPv6 address as previously described, with the tunnel destination embedded in its prefix. The tunnel mode command is **tunnel mode ipv6ip 6to4**.

Each router needs a route to its peer on the other side of the IPv4 network. The only current options for this are static routes and BGP.

## IPv4-Compatible IPv6 Tunnels

This type of tunnel has been deprecated. It encodes the IPv4 address of the tunnel source in the lowest 32 bits of the IPv6 tunnel address and then pads the rest of the bits with zeros. It uses the tunnel mode command **tunnel mode ipv6ip auto-tunnel**.

## ISATAP Tunnels

ISATAP tunnels are similar to the other two tunnels techniques in that an IPv4 address is encoded into the IPv6 address. It is meant to be used within a site, between hosts and routers, although it can be used between sites.

The tunnel source address is an IPv4 address. Do not specify a tunnel destination. The IPv6 address of the tunnel itself combines the network prefix, 0000:5EFE, and the 32-bit IPv4 tunnel source address. The IPv4 address is encoded into the least significant 32 bits of the address. You can use any network prefix. The tunnel interface link-local address still starts with FE80 and then uses 0000:5EFE plus the encoded IPv4 address.

For instance, the link-local address of a tunnel that uses 10.8.8.8 as its source is

FE80::5EFE:A08:808

The unicast IPv6 address of that same tunnel interface, assuming that prefix 2001:1:2:3/64 was assigned to the interface, is

2001:1:2:3:0:5EFE:A08:808

ISATAP tunnels do not support multicast. A route is needed to the tunnel destination if it is in a different subnet; this can be either a static route or a BGP route.

## Using Address Translation

Instead of replacing IPv4, there are several ways to coordinate the functioning of IPv4 and v6 concurrently. NAT-Protocol Translation is an example of this coexistence strategy. NAT-PT does bidirectional translation between IPv4 and IPv6 addresses. Use it when hosts using IPv4 need to establish a session with hosts using IPv6, and vice versa. If hosts communicate using DNS names, a DNS Application Layer Gateway (DNS ALG) can resolve names to both IPv4 and IPv6 addresses.

To enable NAT-PT on a router, use the command **ipv6 nat** on each interface in which traffic needs to be translated. You must also configure at least one NAT-PT prefix. This is used to decide which traffic to NAT; only traffic matching the prefix will be translated. This is configured either at the global configuration mode (to apply to the entire router) or at the interface configuration mode (to apply only to traffic on that interface.) The command is **ipv6 nat prefix** *prefix/prefix-length*.

## Static NAT-PT

You can use either static or dynamic mapping of addresses. To configure static translation of an IPv6 address to an IPv4 address, use the global command:

```
ipv6 nat v6v4 source ipv6-address ipv4-address
```

To configure static mapping of an IPv4 address to an IPv6 address, use the global command:

```
ipv6 nat v4v6 source ipv4-address ipv6-address
```

## Dynamic NAT-PT

Dynamic mapping draws from a pool of addresses to temporarily assign to hosts. You need to create a pool of addresses and then configure NAT-PT to use that pool. You can optionally control the traffic to be mapped by using an access list or route map. To create the pool and enable NAT-PT for IPv4 to IPv6 translation, use the global commands:

```
ipv6 nat v4v6 pool name start-ipv6 end-ipv6, prefix-length prefix-
  length
ipv6 nat v4v6 source list {access-list-number | name} pool name
```

To create the pool and enable NAT-PT for IPv6 to IPv4 translation, use the global commands:

```
ipv6 nat v6v4 pool name start-ipv4 end-ipv4 prefix-length prefix-
  length
ipv6 nat v6v4 source {list access-list-name | route-map map-name}
  pool name
```

Verify NAT-PT operation with the commands **show ip nat translations**, **show ip nat statistics**, **show ipv6 nat translations**, and **show ipv6 nat statistics**.

# IPv6 Link Types

IPv6 recognizes three types of links:

- Point-to-point
- Point-to-multipoint
- Multiaccess

## Point-to-Point Links

Recall that an IPv6 interface uses its MAC address to create its link-local address. A serial link has no MAC address associated with it, so it uses one from an Ethernet interface. You can manually configure the link-local address to make it more recognizable. Be sure to begin the IPv6 address with the link-local prefix FE80.

Point-to-point links do not necessarily need global unicast addresses. The routers can communicate with only link-local addresses, but you could not reach those interfaces from off the network because the link-local is not a routable address.

## Point-to-Multipoint Links

For point-to-multipoint links, such as Frame Relay, you must map the destination IPv6 address to the correct DLCI, just as with IPv4. The difference is that with IPv6 you must also map the link-local address to the DLCI because it is used as the next hop for routing. So for each DLCI, you must have at least two mappings: the remote router's IPv6 global unicast address and the remote router's link-local address. The map command is

```
frame relay map ipv6 destination-address out dlci dlci-number broad-
  cast
```

In a hub-and-spoke topology, the hub must be configured for IPv6 unicast routing for the spokes to communicate with each other.

## Multiaccess Links

Devices on multiaccess links, such as Ethernet, build a table mapping destination Layer 3 addresses to Layer 2 addresses, whether you use IPv4 or IPv6. IPv4 uses a separate protocol, ARP, to do this. In IPv6 the process is built into the IPv6 protocol with the Neighbor Discovery process. It uses ICMPv6.

An IPv6 device sends a Neighbor Solicitation (NS) multicast with a prefix of FE02. The neighbor responds with a Neighbor Advertisement (NA) message listing its MAC address. As with ARP, these mappings have a set lifetime (called the *reachable time*), so an NS can also be sent periodically to verify that a neighbor is still reachable.

To add a static entry to the Neighbor Discovery table, use the command **ipv6 neighbor** *ipv6-address interface-type interface-number hardware-address*. A static address does not age out of the table.

Display the mappings with the **show ipv6 neighbors** command.

*This page intentionally left blank*

## APPENDIX A

# Understanding IPsec

You are not required to have detailed knowledge of IPsec for the ROUTE exam. This Appendix is intended to help solidify your understanding of the technology. It can also serve as a command reference if you need to actually configure IPsec.

IPSecurity, or IPsec, is a set of rules for securing data communications across a public, untrusted network such as the Internet. It provides

- Data confidentiality by encrypting portions of a packet
- Data integrity by ensuring the packet has not been altered in transit
- Data source authentication to ensure the data originated with a trusted source
- Antireplay protection to ensure that packets are not copied and sent

IPsec standards do not specify exactly how packets should be encrypted or authenticated; it relies on other protocols to accomplish those functions. For encryption it can use Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES). For authentication it can use Hash-based Message Authentication Codes (HMAC). An HMAC combines a hash function such as Message Digest 5 (MD5), and Secure Hash Algorithm 1 (SHA-1) with a shared secret key. MD5 uses a 128-bit hash, whereas SHA-1 uses a 160-bit hash; however, IPsec uses only 96 bits of the SHA-1 hash.

## IPsec Headers

IPsec defines two types of headers: Authentication Header and Encapsulating Security Payload.

### Authentication Header

Authentication Header (AH) is IP protocol number 51. It authenticates the packet, including the IP header, but does not encrypt the packet payload. AH works by creating an MD5 or SHA-1 hash from the IP header (except any changeable fields such as Time to Live) and the packet payload. It sends this hash in an AH header after the Layer 3 IP header. The receiving host also creates a hash value from the IP header and packet payload, and compares the two hashes. If they match, the packet was unchanged during transit. A shared key creates the hashes, so a match also serves to authenticate the source of the packet, which is rarely used without ESP.

## Encapsulating Security Payload

Encapsulating Security (ESP), IP protocol number 50, encrypts packet payloads and can optionally authenticate and do integrity checks by using it with AH. It adds a header and a trailer to the packet. When used with AH, the packet is encrypted first and then put through the hash mechanism.

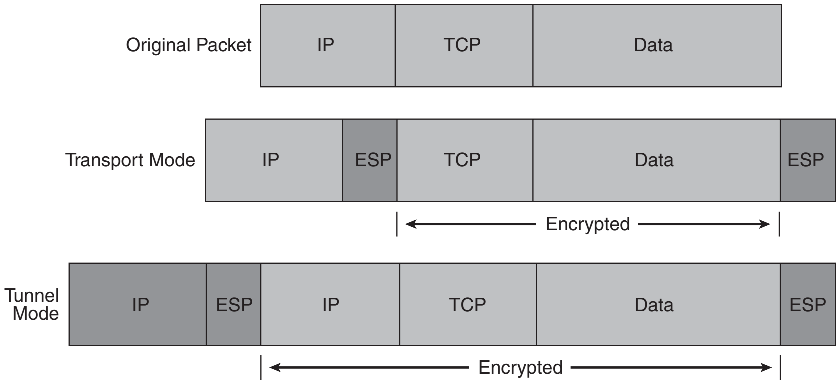
# IPsec Modes

IPsec can operate in either transport mode or tunnel mode. The headers differ based on the mode used:

- Transport Mode IPsec uses the original IP header. The data payload can be encrypted, and the packet can be authenticated from the ESP header back. Transport mode is often used with Generic Routing Encapsulation (GRE) tunnels because GRE hides the original IP address.
- Tunnel Mode IPsec replaces the original IP header with a tunnel header. The ESP header is placed after the new header, before the original one. The original IP header can be encrypted along with the data payload, and the packet can be authenticated from the ESP header back. Tunnel mode adds approximately 20 bytes to the packet.

Figure A-1 shows the packet headers in the two IPsec modes.

**Figure A-1 Transport Mode Versus Tunnel Mode IPsec**



Tunnel mode ESP can cause problems when used with Network Address Translation (NAT). The original TCP or UDP header is encrypted and hidden, so there are no Layer 4 port numbers for NAT to use. *NAT Traversal* detects the existence of a NAT device and adds a UDP header after the tunnel IP header. NAT can then use the port number in that UDP header.

## Authentication Methods

Several authentication methods are supported with IPsec VPNs:

- Username and password
- A one-time password
- Biometric features, such as fingerprint
- Preshared key values
- Digital certificates

## Encryption Methods

IPsec encryption uses key values to encrypt and decrypt data. Keys can be either symmetric or asymmetric. Symmetric keys use the same value to both encrypt and decrypt the data, which include DES, 3DES, and AES. Asymmetric keys use one value to encrypt the data and another one to decrypt it. Diffie-Hellman and RSA use asymmetric keys.

**NOTE**

RSA is not an acronym; it is the initials of the last names of the algorithm's inventors: Ron Rivest, Adi Shamir, and Len Adleman.

### Symmetric Key Algorithms

DES uses a 56-bit key and can be broken fairly easily. It is a block cipher that encrypts 64-bit blocks of data at a time.

3DES is also a block cipher, but it encrypts each block, decrypts it, and then encrypts it again. A 56-bit key is used each time, thus equaling a key length of 168 bits. It is more secure than DES but also requires more processing power.

AES is a stronger block cipher encryption method than DES or 3DES. It uses a 128-bit data block and a key length of 128 bits, 192 bits, or 256 bits. AES has been approved for use with government classified data.

### Asymmetric Key Algorithm

RSA uses asymmetric keys and can be used for signing messages and encrypting them. A public key encrypts or signs the data. It can be decrypted only with a private key held by the receiver. RSA is slower than symmetrical key algorithms but more secure if a large enough key is used. A key length of 2048 bits is recommended.

### Diffie-Hellman Key Exchange

The Diffie-Hellman protocol solves the problem of exchanging keys over an insecure network. Each device creates a public key and a private key. They exchange their public keys in the open, unencrypted. They each then use the other device's public key and their own private key to generate a shared secret key that each can use.

## Key Management

The public key infrastructure (PKI) manages encryption and identity information such as public keys and certificates. It consists of the following components:

- Peer devices that need to communicate securely.
- Digital certificates that validate the peer's identity and transmit their public key.
- Certificate authorities (CA), also known as trustpoints, that grant, manage, and revoke certificates. This can be a third-party CA or an internal one. Cisco has an IOS Certificate Server.
- Optional Registration authorities (RA) that handle certificate enrollment requests.
- A way to distribute certificate revocation lists (CRL), such as HTTP or Lightweight Directory Access Protocol (LDAP).

PKI credentials, such as RSA keys and digital certificates, can be stored in a router's NVRAM. The can also be stored in USB eTokens on routers that support them.

# Establishing an IPsec VPN

When IPsec establishes a VPN between two peer hosts, it sets up a security association (SA) between them. SAs are unidirectional, so each bidirectional data session requires two. The Internet Security Association and Key Management Protocol (ISAKMP) defines how SAs are created and deleted. Following are five the basic steps:

1. **Interesting traffic arrives at the router:** "Interesting" traffic is that which should be sent over the VPN. This is specified by a crypto access list. Any traffic not identified as "interesting" is sent in the clear, unprotected.

2. **Internet Key Exchange (IKE) Phase One:** Negotiates the algorithms and hashes to use, authenticates the peers, and sets up an ISAKMP SA. This has two modes: Main and Aggressive. Main mode uses three exchanges during Phase One. Aggressive mode sends all the information in one exchange. The proposed settings are contained in *Transform Sets* that list the proposed encryption algorithm, authentication algorithm, key length, and mode. Multiple transform sets can be specified, but both peers must have at least one matching transform set or the session is torn down.

3. **IKE Phase Two:** Uses the secure communication channel created in Phase One to set up the SAs for ESP and AH, negotiating the SA parameters and settings to be used to protect the data transmitted. This periodically renegotiates the SAs. SAs have lifetimes that can be measured in either the amount of data transferred or length of time. Might do an additional Diffie-Hellman key exchange during Phase Two.

4. **Data is transferred along the VPN between the two peers:** It is encrypted by one peer and decrypted by the other, according to the transform sets negotiated.

5. **Tunnel termination:** The IPsec session drops because of either direct termination or time out.

# Configuring a Site-to-Site VPN Using IOS

Following are six steps to configuring a site-to-site IPsec VPN using Cisco IOS commands:

1. Configure the ISAKMP policy.

2. Configure the IPsec transform set or sets.

3. Configure a crypto access control list (ACL).
4. Configure a crypto map.
5. Apply the crypto map to the outgoing interface.
6. Optionally configure and apply an ACL that permits only IPsec or IKE traffic.

## Configuring an ISAKMP Policy

To configure an ISAKMP policy, first create the policy and then give the parameters. These parameters might include such things as type of encryption, type of hash, type of authentication, SA lifetime, and Diffie-Hellman group. The following example shows an ISAKMP policy configuration, along with the options available with each parameter. Options can vary based on IOS version.

```
IPSEC_RTR(config)# crypto isakmp policy ?
  <1-10000>  Priority of protection suite
IPSEC_RTR(config)# crypto isakmp policy 1
!
IPSEC_RTR(config-isakmp)# encryption ?
  3des  Three key triple DES
  aes   AES - Advanced Encryption Standard.
  des   DES - Data Encryption Standard (56 bit keys).
IPSEC_RTR(config-isakmp)# encryption 3des
!
IPSEC_RTR(config-isakmp)# hash ?
  md5  Message Digest 5
  sha  Secure Hash Standard
IPSEC_RTR(config-isakmp)# hash sha
!
IPSEC_RTR(config-isakmp)# authentication ?
  pre-share  Pre-Shared Key
  rsa-encr   Rivest-Shamir-Adleman Encryption
  rsa-sig    Rivest-Shamir-Adleman Signature
IPSEC_RTR(config-isakmp)# authentication pre-share
!
IPSEC_RTR(config-isakmp)# group ?
  1  Diffie-Hellman group 1
  2  Diffie-Hellman group 2
  5  Diffie-Hellman group 5
IPSEC_RTR(config-isakmp)# group 2

IPSEC_RTR(config-isakmp)# lifetime ?
  <60-86400>  lifetime in seconds
IPSEC_RTR(config-isakmp)# lifetime 300
```

## Configuring an IPsec Transform Set

An IPsec transform set defines how VPN data will be protected. It specifies the IPsec protocols that will be used. You can specify up to four transforms and the algorithm to use with them. You can also configure either tunnel or transport mode. (Tunnel is default.) The transforms include

- AH with either MD5 or SHA-1
- ESP encryption using DES, 3DES, AES, or others
- ESP authentication using MD5 or SHA-1
- Compression using the Lempel-Ziv-Stac (LZS) algorithm

The following example shows a transform set with ESP encryption and authentication. Note that these commands are all given as part of the same command.

```
IPSEC_RTR# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
IPSEC_RTR(config)# crypto ipsec transform-set TRANSFORM1 esp-aes 192
 esp-md5-hmac
```

## Configuring a Crypto ACL

You use a crypto ACL to identify traffic that should be protected by the IPsec VPN, in particular the interesting traffic that brings up the tunnel. Any traffic permitted in the ACL is sent over the VPN. Traffic denied by the ACL is not dropped—it is simply sent normally.

The following example shows a crypto ACL that permits traffic from two internal networks—172.16.1.0 and 172.16.4.0—if it is bound to the server network of 10.6.3.0.

**NOTE**

When configuring the crypto ACL on the router at the other end of the tunnel, be sure to reverse the source and destination IP addresses.

```
IPSEC_RTR(config)access-list 172 permit ip 172.16.1.0 0.0.0.255
 10.6.3.0 0.0.0.255
IPSEC_RTR(config)access-list 172 permit ip 172.16.4.0 0.0.0.255
 10.6.3.0 0.0.0.255
```

## Configuring a Crypto Map

A crypto map pulls together the transform sets and crypto ACLs, and associates them with a remote peer. You can use a sequence number when configuring a crypto map. Multiple crypto maps with the same name but different sequence numbers form a crypto map set. Traffic is evaluated against each crypto map depending on its sequence number to see if it should be protected. This permits more complex and granular traffic filtering.

The following example shows a crypto map that links the transform set and ACL configured in previous examples.

```
IPSEC_RTR(config)# crypto map TO_SERVERS 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
IPSEC_RTR(config-crypto-map)# set peer 10.1.1.1
IPSEC_RTR(config-crypto-map)# match address 172
IPSEC_RTR(config-crypto-map)# set transform-set TRANSFORM1
```

## Applying the Crypto Map to an Interface

After the crypto map is configured, it must be applied to an interface for it to take effect. It is applied at the *outgoing* interface—the one that VPN traffic uses to reach the other end of the VPN. You might need to use a static route or otherwise adjust your routing to force traffic bound for the VPN destination networks to use the correct outgoing interface.

The following example shows the crypto map TO_SERVERS applied to interface serial 0/0/0. Note that the router replies with a message that ISAKMP is now enabled:

```
IPSEC_RTR(config)# int s0/0/0
IPSEC_RTR(config-if)# crypto map TO_SERVERS
IPSEC_RTR(config-if)#
01:19:16: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

## Configuring an Optional Interface Access List

You might want to have an interface ACL on the VPN interface. Typically you would permit only IPsec-related traffic, and perhaps routing protocol traffic, in and out that interface. Keep in mind the following port numbers when configuring the ACL:

- ESP is IP protocol 50.
- AH is IP protocol 51.

- IKE uses UDP port 500.
- NAT traversal uses UDP port 4500.

The source and destination addresses should be the IP addresses of the outgoing VPN interfaces. The following example shows an ACL that permits IPsec traffic between two hosts.

```
IPSEC_RTR(config)# access-list 101 permit ahp host 10.1.1.2 host
 10.1.1.1
IPSEC_RTR(config)# access-list 101 permit esp host 10.1.1.2 host
 10.1.1.1
IPSEC_RTR(config)# access-list 101 permit udp host 10.1.1.2 eq isakmp
 host 10.1.1.1
IPSEC_RTR(config)# access-list 101 permit udp host 10.1.1.2 host
 10.1.1.1 eq isakmp
!
IPSEC_RTR(config)# interface s 0/0/0
IPSEC_RTR(config-if)# ip address 10.1.1.2 255.255.255.252
IPSEC_RTR(config-if)# ip access-group 101 out
```

# Monitoring and Troubleshooting IPsec VPNs

Some useful IOS commands for monitoring your IPsec VPNs include

- **show crypto isakmp sa:** Shows all the IKE security associations currently active on the router. Look for a status of QM_IDLE to verify that the SA is active.
- **show crypto ipsec sa:** Shows the parameters used by each SA and traffic flow. Look for the count of packets being encrypted and decrypted to verify the VPNs operation.

To troubleshoot VPN problems, first verify IP connectivity. If that exists, review your configuration one more time. If the configuration looks correct on both peers, you can view detailed information about the IKE negotiations by using the command **debug crypto isakmp**.

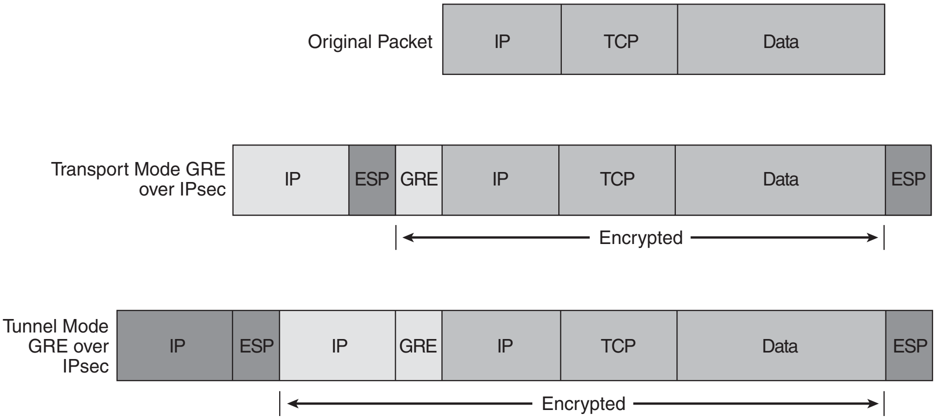
# Using GRE with IPsec

GRE is a tunneling protocol that can support multiple Layer 3 protocols, such as IP, IPX, and AppleTalk. It also enables the use of multicast routing protocols across the tunnel. It adds a 20-byte IP header and a 4-byte GRE header, hiding the existing packet headers. The GRE header contains a flag field and a protocol type field to identify the Layer 3 protocol being transported. It might optionally contain a tunnel checksum, tunnel key, and tunnel sequence number. GRE does not encrypt traffic or use any strong security measures to protect the traffic.

GRE can be used along with IPsec to provide data source authentication and data confidentiality and ensure data integrity. GRE over IPsec tunnels are typically configured in a hub-and-spoke topology over an untrusted WAN to minimize the number of tunnels that each router must maintain.

Figure A-2 shows how the GRE and IPsec headers work together.

**Figure A-2 GRE over IPsec Headers**



## Configuring a GRE Tunnel Using IOS

To configure GRE using IOS commands, you must first configure a logical tunnel interface. GRE commands are then given under that interface. You must specify a source and destination for the tunnel; the source is a local outgoing interface. You might also give the tunnel interface an IP address and specify the tunnel mode. GRE is the default mode.

The following example shows a tunnel interface configured for GRE. The **mode** command is shown only as a reference because it is the default; it would not normally appear in the configuration.

```
interface Tunnel1
 ip address 172.16.5.2 255.255.255.0
 tunnel source Serial0/0/0
 tunnel destination 10.1.1.1
 tunnel mode gre ip
```

# APPENDIX B

# IPv6 Header Format

Although this specific material is not tested on the ROUTE exam, it might help you gain a better understanding of the structure of IPv6.

The IPv6 header is similar to the IPv4 header. The largest changes have to do with the larger addresses, aligning fields to 64-bit boundaries, and moving fragmentation to an extension header.

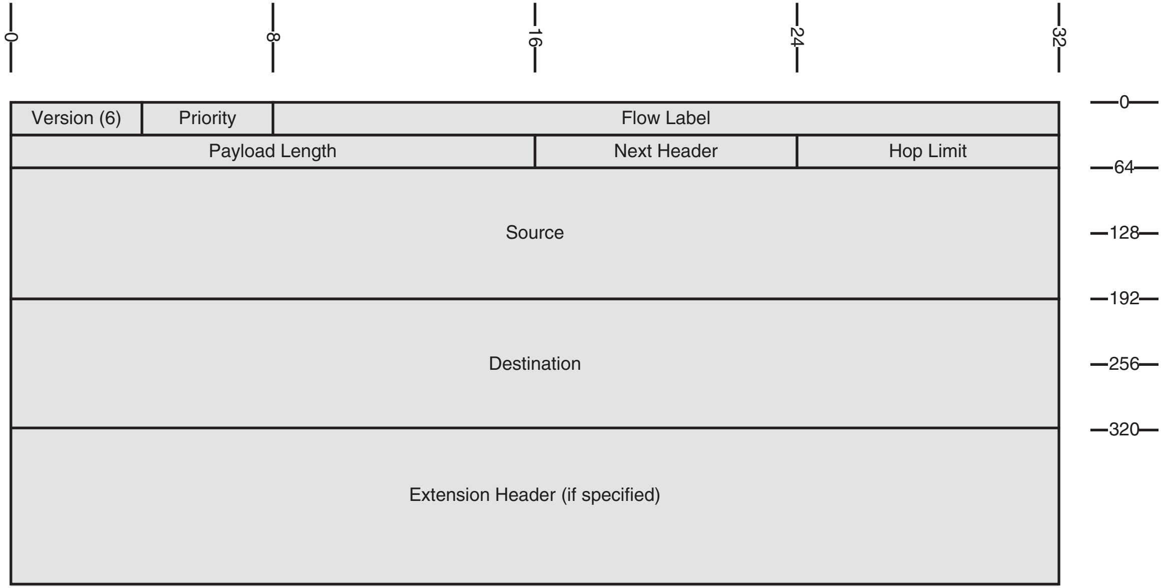
The fields are

- **Version:** 6.
- **Priority:** Similar to DSCP in version 4, this 8-bit field describes relative priority.
- **Flow:** 20-bit flow label enables tagging in a manner similar to MPLS.
- **Length:** The length of the data in the packet.
- **Next Header:** Indicates how the bits after the IP header should be interpreted. Can indicate TCP or UDP, or it can show an extension header.
- **Hop Limit:** Similar to TTL.
- **Source and Destination:** IPv6 addresses.

Zero or more extension headers can follow, including

- **Hop-by-hop options:** Options for intermediate devices.
- **Destination options:** Options for the end node.
- **Source routing:** Specifies “way stations” that the route must include.
- **Fragmentation:** Divides packets.
- **Authentication:** Attests to source. Replaces the AH header from IPsec.
- **Encryption:** Replaces the IPsec ESP header.

**Figure B-1 IPv6 Header**



# SWITCH

## CHAPTER 1

# Campus Network Design

An enterprise campus generally refers to a network in a specific geographic location. It can be within one building or span multiple buildings near each other. A campus network also includes the Ethernet LAN portions of a network outside the data center. Large enterprises have multiple campuses connected by a WAN. Using models to describe the network architecture divides the campus into several internetworking functional areas, thus simplifying design, implementation, and troubleshooting.

## The Hierarchical Design Model

Cisco has used the three-level Hierarchical Design Model for years. The hierarchical design model divides a network into three layers:

- **Access:** Provides end-user access to the network. In the LAN, local devices such as phones and computers access the local network. In the WAN, remote users or sites access the corporate network.
  - High availability via hardware such as redundant power supplies and redundant supervisor engines. Software redundancy via access to redundant default gateways using a first hop redundancy protocol (FHRP).
  - Converged network support by providing access to IP phones, computers, and wireless access points. Provides QoS and multicast support.
  - Security through switching tools such as Dynamic ARP Inspection, DHCP snooping, BPDU Guard, port-security, and IP source guard. Controls network access.
- **Distribution:** Aggregation point for access switches. Provides availability, QoS, fast path recovery, and load balancing.
  - High availability through redundant distribution layer switches providing dual paths to the access switches and to core switches.

    Use of FHRP protocols to ensure connectivity if one distribution switch is removed.

    - Routing policies applied, such as route selection, filtering, and summarization. Can be default gateway for access devices. QoS and security policies applied.
    - Segmentation and isolation of workgroups and workgroup problems from the core, typically using a combination of Layer 2 and Layer 3 switching.

- **Core:** The backbone that provides a high-speed, Layer 3 path between distribution layers and other network segments. Provides reliability and scalability.
    - Reliability through redundant devices, device components, and paths.
    - Scalability through scalable routing protocols. Having a core layer in general aids network scalability by providing gigabit (and faster) connectivity, data and voice integration, and convergence of the LAN, WAN, and MAN.
    - No policies such as ACLs or filters that would slow traffic down.

A set of distribution devices and their accompanying access layer switches are called a switch block.

## Core Layer

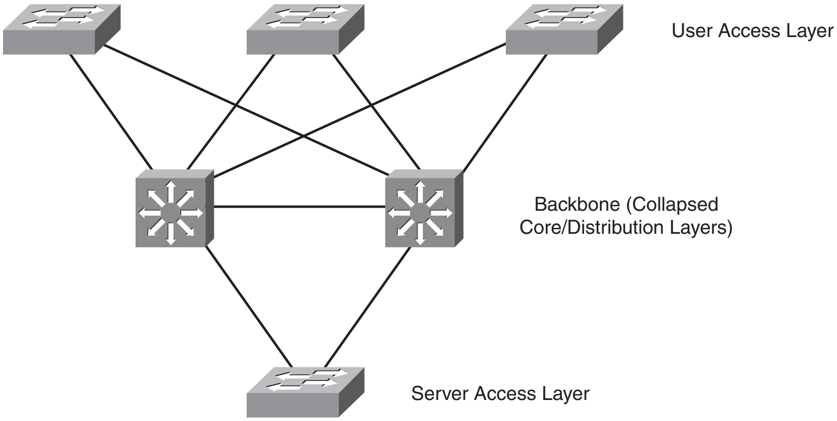
Is a core layer always needed? Without a core layer, the distribution switches must be fully meshed. This becomes more of a problem as a campus network grows larger. A general rule is to add a core when connecting three or more buildings or four or more pairs of building distribution switches. Some benefits of a campus core are:

- Adds a hierarchy to distribution switch connectivity
- Simplifies cabling because a full-mesh between distribution switches is not required
- Reduces routing complexity by summarizing distribution networks

## Small Campus Design

In a small campus, the core and distribution can be combined into one layer. *Small* is defined as fewer than 200 end devices. In very small networks, one multilayer switch might provide the functions of all three layers. Figure 1-1 shows a sample small network with a collapsed core.

**Figure 1-1 Small Campus Network**



## Medium Campus Design

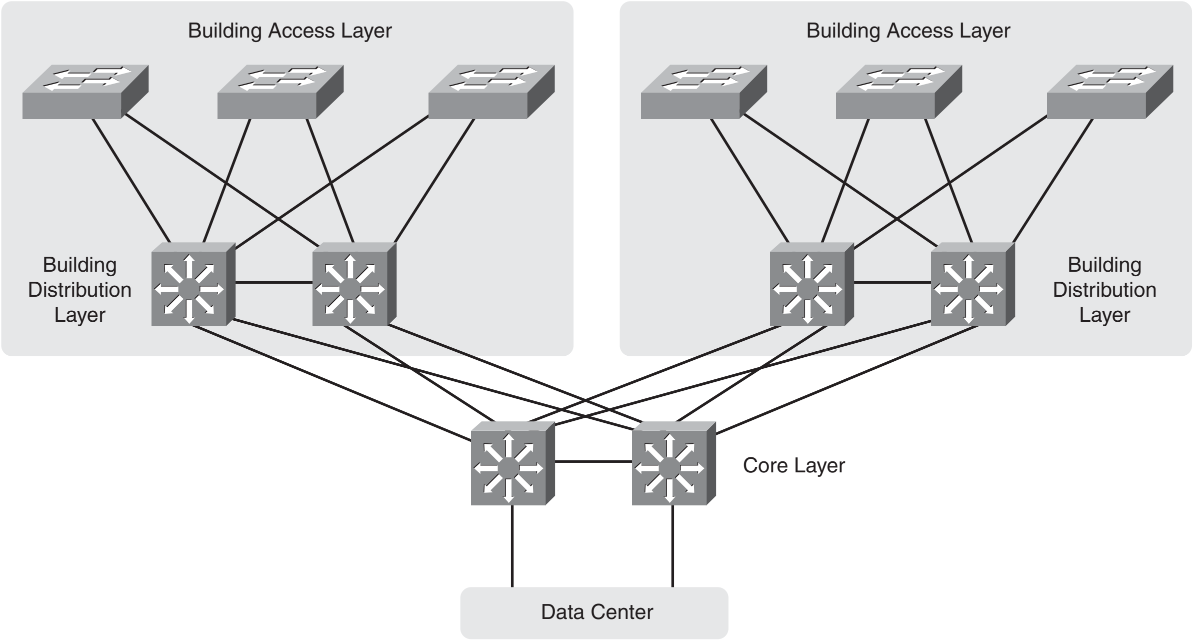
A medium-sized campus, defined as one with between 200 and 1000 end devices, is more likely to have several distribution switches and thus require a core layer. Each building or floor is a campus block with access switches uplinked to redundant multilayer distribution switches. These are then uplinked to redundant core switches, as shown in Figure 1-2.

## Data Center Design

The core layer connects end users to the data center devices. The data center segment of a campus can vary in size from few servers connected to the same switch as users in a small campus, to a separate network with its own three-layer design in a large enterprise. The three layers of a data center model are slightly different:

- **Core layer:** Connects to the campus core. Provides fast switching for traffic into and out of the data center.
- **Aggregation layer:** Provides services such as server load balancing, content switching, SSL off-load, and security through firewalls and IPS.
- **Access layer:** Provides access to the network for servers and storage units. Can be either Layer 2 or Layer 3 switches.

**Figure 1-2 Medium-Sized Campus Network**



## Network Traffic Flow

The need for a core layer and the devices chosen for the core also depend on the type of network traffic and traffic flow patterns. Modern converged networks include different traffic types, each with unique requirements for security, QoS, transmission capacity, and delay. These include

- IP telephony signaling and media
- Core Application traffic, such as Enterprise Resource Programming (ERP), Customer Relationship Management (CRM)
- Multicast multimedia
- Network management
- Application data traffic, such as web pages, email, file transfer, and database transactions
- Scavenger class traffic that requires less-than-best-effort treatment

The different types of applications also have different traffic flow patterns. These might include

- Peer-to-Peer applications such as IP phone calls, video conferencing, file sharing, and instant messaging provides real-time interaction. It might not traverse the core at all, if the users are local to each other. Their network requirements vary, with voice having strict jitter needs and video conferencing using high bandwidth.
- Client-Server applications require access to servers such as email, file storage, and database servers. These servers are typically centralized in a data center, and users require fast, reliable access to them. Server farm access must also be securely controlled to deny unauthorized users.
- Client-Enterprise Edge applications are located on servers at the WAN edge, reachable from outside the company. These can include email and web servers, or e-commerce servers, for example. Access to these servers must be secure and highly available.

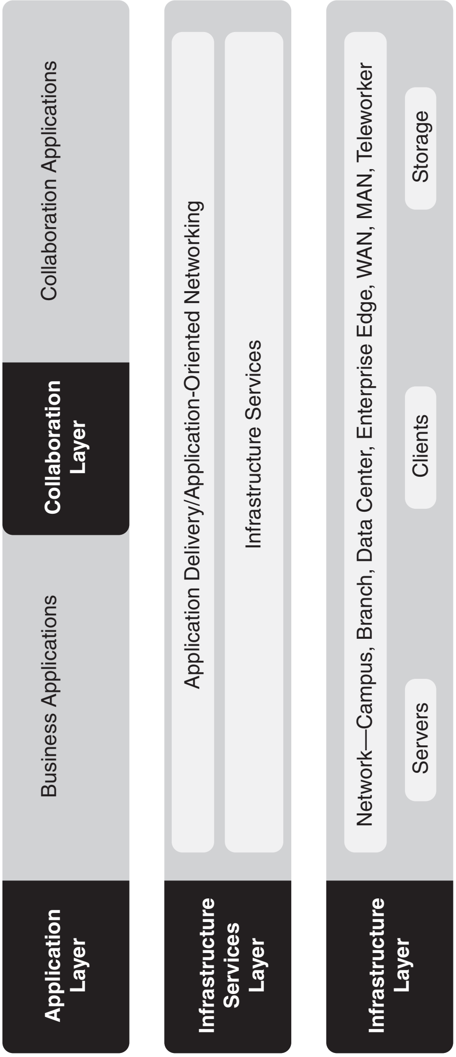
# Service-Oriented Network Architecture

Service-Oriented Network Architecture (SONA) attempts to provide a design framework for a network that can deliver the services and applications businesses need. It acknowledges that the network connects all components of the business and is critical to them. The SONA model integrates network and application functionality cooperatively and enables the network to be smart about how it handles traffic to minimize the footprint of applications.

Figure 1-3 shows how SONA breaks down this functionality into three layers:

- **Network Infrastructure:** Campus, data center, branch, and so on. Networks and their attached end systems (resources such as servers, clients, and storage.) These can be connected anywhere within the network. The goal is to provide anytime/any place connectivity.
- **Interactive Services:** Resources allocated to applications, using the network infrastructure. These include
  - Management
  - Infrastructure services such as security, mobility, voice, compute, storage, and identity
  - Application delivery
  - Virtualization of services and network infrastructure
- **Applications:** Includes business policy and logic. Leverages the interactive services layer to meet business needs. Has two sublayers:
  - Application layer, which defines business applications
  - Collaboration layer, which defines applications such as unified messaging, conferencing, IP telephony, video, instant messaging, and contact centers

**Figure 1-3 SONA Model**


Application Layer
Business Applications
Collaboration Layer
Collaboration Applications
Infrastructure Services Layer
Application Delivery/Application-Oriented Networking
Infrastructure Services
Infrastructure Layer
Network—Campus, Branch, Data Center, Enterprise Edge, WAN, MAN, Teleworker
Servers
Clients
Storage

## Planning a Network Implementation

It is important to use a structured approach to planning and implementing any network changes or new network components. A comprehensive lifecycle approach lowers the total cost of ownership, increases network availability, increases business agility, and provides faster access to applications and services.

The Prepare, Plan, Design, Implement, Operate, and Optimize (PPDIOO) Lifecycle Approach is one structure that can be used. The components are

- **Prepare:** Organizational requirements gathering, high-level architecture, network strategy, business case strategy
- **Plan:** Network requirements gathering, network examination, gap analysis, project plan
- **Design:** Comprehensive, detailed design
- **Implement:** Detailed implementation plan, and implementation following its steps
- **Operate:** Day-to-day network operation and monitoring
- **Optimize:** Proactive network management and fault correction

Network engineers at the CCNP level will likely be involved at the implementation and following phases. They can also participate in the design phase. It is important to create a detailed implementation plan that includes test and verification procedures and a rollback plan. Each step in the implementation plan should include a description, a reference to the design document, detailed implementation and verification instructions, detailed rollback instructions, and the estimated time needed for completion. A complex implementation should be done in sections, with testing at each incremental section.

CHAPTER 2

# VLAN Implementation

VLANs are used to break large campus networks into smaller pieces. The benefit of this is to minimize the amount of broadcast traffic on a logical segment.

## VLAN Overview

A virtual LAN (VLAN) is a logical LAN, or a logical subnet. It defines a broadcast domain. A physical subnet is a group of devices that shares the same physical wire. A logical subnet is a group of switch ports assigned to the same VLAN, regardless of their physical location in a switched network. VLAN membership can be assigned either statically by port, or dynamically by MAC address or username.

Two types of VLANs are

- **End-to-end VLAN:** VLAN members reside on different switches throughout the network. They are used when hosts are assigned to VLANs for policy reasons, rather than physical location. This provides users a consistent policy and access to resources regardless of their location. It also makes troubleshooting more complex because so many switches can carry traffic for a specific VLAN, and broadcasts can traverse many switches. Figure 2-1 shows end-to-end VLANs.
- **Local VLAN:** Hosts are assigned to VLANs based on their location, such as a floor in a building. This design is more scalable and easier to troubleshoot because the traffic flow is more deterministic. It enables more redundancy and minimizes failure domains. It does require a routing function to share resources between VLANs. Figure 2-2 shows an example of local VLANs.

When planning a VLAN structure, consider traffic flows and link sizing. Take into account the entire traffic pattern of applications found in your network. For instance, IP voice media traffic travels directly between phones, but signaling traffic must pass to the Unified Communications Manager. Multicast traffic must communicate back to the routing process and possibly call upon a Rendezvous Point. Various user applications, such as email and Citrix, place different demands on the network.

**Figure 2-1 End-to-End VLANs**

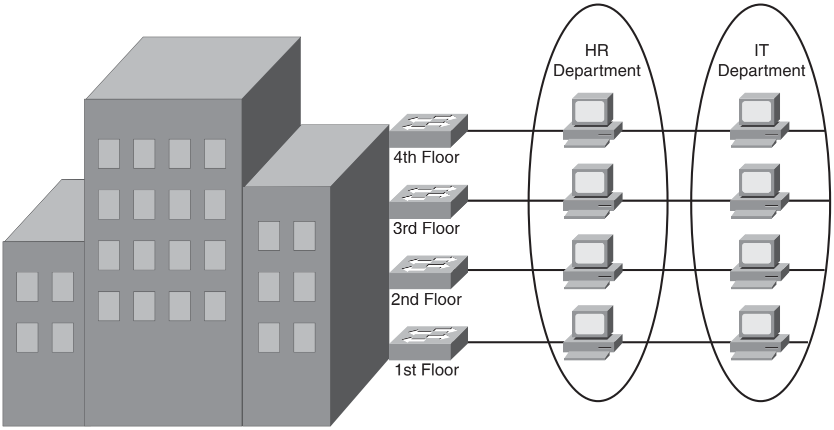
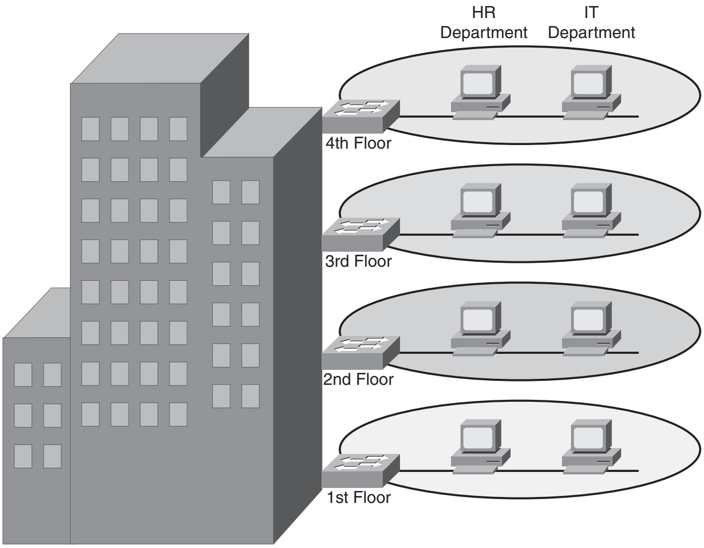


**Figure 2-2 Local VLANs**



Application flow influences link bandwidth. Remember that uplink ports need to handle all hosts communicating concurrently, and although VLANs logically separate traffic, traffic in different VLANs still travels over the same trunk line. Benchmark throughput for critical application and user data during peak hours; then analyze the results for any bottlenecks throughout the layered design.

User access ports are typically Fast Ethernet or faster. Access switches must have the necessary port density and can be either Layer 2 or Layer 3. Ports from user Access to the Distribution layer should be Gigabit Ethernet or better, with an oversubscription ratio of no more than 20:1. Distribution switches should be multilayer or Layer 3. Links from Distribution to the Core should be Gigabit Etherchannel or 10-Gig Ethernet, with an oversubscription of no more than 4:1.

## VLAN Planning

Before beginning a VLAN implementation, you need to determine the following information:

- VLAN numbering, naming and IP addressing scheme
- VLAN placement—local or multiple switches
- Are any trunks necessary and where?
- VTP parameters
- Test and verification plan

## Creating a VLAN and Assigning Ports

VLANs must be created before they can be used. Creating VLANs is easy—in global configuration mode just identify the VLAN number and optionally name it!

```
(config)# vlan 12
(config-vlan)# name MYVLAN
```

Delete a VLAN by using the same command with **no** in front of it. There is no need to include the name when deleting.

When statically assigning ports to VLANs, first make the interface an access port, and then assign the port to a VLAN. At the interface configuration prompt:

```
(config-if)# switchport mode access
(config-if)# switchport access vlan 12
```

## Verifying VLAN Configuration

To see a list of all the VLANs and the ports assigned to them, use the command **show vlan**. To narrow down the information displayed, you can use these keywords after the command: **brief, id**, **vlan-number**, or *name* **vlan-name**:

```
ASW# show vlan brief
VLAN Name  Status  Ports
---- ------------------------------ -------- -------------------
  ----------
1  default  active  Fa0/1, Fa0/2, Fa0/3,
    Fa0/10,Fa0/11,Fa0/12
20  VLAN0020  active  Fa0/5,Fa0/6,Fa0/7
21  VLAN0021  active  Fa0/8,Fa0/9
1002 fddi-default  active
1003 trcrf-default  active
1004 fddinet-default  active
1005 trbrf-default  active
```

Other verification commands include

- **show running-config interface** *interface no*: Use the following to verify the VLAN membership of the port:

  ```
  ASW# show run interface fa0/5
  Building configuration...
  Current configuration 64 bytes
  interface FastEthernet 0/5
   switchport access vlan 20
   switchport mode access
  ```

- **show mac address-*table interface*** *interface-no. vlan vlan no*: Use the following to view MAC addresses learned through that port for the specified VLAN:

  ```
  ASW# show mac address-table interface fa0/1
       Mac Address Table
  ------------------------------------------
  Vlan  Mac Address   Type       Ports
  ----  -----------   ----       -----
  1    0030.b656.7c3d DYNAMIC     Fa0/1
  Total Mac Addresses for this criterion: 1
  ```

- **show interfaces** *interface-no.* **switchport:** Use the following to see detailed information about the port configuration, such as entries in the Administrative Mode and Access Mode VLAN fields:

  ```
  ASW# show interfaces fa0/1 switchport
  Name: Fa0/1
  Switchport: Enabled
  Administrative Mode: dynamic desirable
  Operational Mode: static access
  Administrative Trunking Encapsulation: negotiate
  Operational Trunking Encapsulation: native
  ```

```
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Unknown unicast blocked: false
Unknown multicast blocked: false
Broadcast Suppression Level: 100
Multicast Suppression Level: 100
Unicast Suppression Level: 100
```

# VLAN Trunking

A *trunk* is a link that carries traffic for more than one VLAN. Trunks multiplex traffic from multiple VLANs. They typically connect switches and enable ports on multiple switches to be assigned to the same VLAN.

Two methods of identifying VLANs over trunk links are

- **Inter-Switch Link (ISL):** A Cisco proprietary method that encapsulates the original frame in a header, which contains VLAN information. It is protocol-independent and can identify Cisco Discovery Protocol (CDP) and bridge protocol data unit (BPDU) frames.
- **802.1Q:** Standards-based, tags the frames (inserts a field into the original frame immediately after the source MAC address field), and supports Ethernet and Token Ring networks.

When a frame comes into a switch port, the frame is tagged internally within the switch with the VLAN number of the port. When it reaches the outgoing port, the internal tag is removed. If the exit port is a trunk port, its VLAN is identified in either the ISL encapsulation or the 802.1Q tag. The switch on the other end of the trunk removes the ISL or 802.1Q information, checks the VLAN of the frame, and adds the internal tag. If the exit port is a user port, the original frame is sent out unchanged, making the use of VLANs transparent to the user.

If a nontrunking port receives an ISL-encapsulated packet, the port cannot remove the ISL header. By default, the system installs ISL system CAM entries and drops ISL packets. In special, rare circumstances, these CAM entries are installed for every active VLAN in the switch. To prevent such collisions, enter the **`no-isl-entries enable`** command on switches connected to other switches. If the ISL header and footer cause the MTU size to be exceeded, it might be counted as an error.

If a nontrunking port receives an 802.1Q frame, the source and destination MAC addresses are read, the tag field is ignored, and the frame is switched normally at Layer 2.

## Configuring a Trunk Link

Ports can become trunk ports either by static configuration or dynamic negotiation using Dynamic Trunking Protocol (DTP). A switch port can be in one of five DTP modes:

- **Access:** The port is a user port in a single VLAN.
- **Trunk:** The port negotiates trunking with the port on the other end of the link.
- **Non-negotiate:** The port is a trunk and does not do DTP negotiation with the other side of the link.
- **Dynamic Desirable:** Actively negotiates trunking with the other side of the link. It becomes a trunk if the port on the other switch is set to **trunk**, **dynamic desirable**, or **dynamic auto** mode.
- **Dynamic Auto:** Passively waits to be contacted by the other switch. It becomes a trunk if the other end is set to **trunk** or **dynamic desirable** mode.

Configure a port for trunking at the interface configuration mode:

```
(config-if)#switchport mode {dynamic {auto | desirable} | trunk}
```

If dynamic mode is used, DTP negotiates the trunking state and encapsulation. If trunk mode is used, you must specify encapsulation, and you can disable all DTP negotiation:

```
(config-if)#switchport trunk encapsulation {isl | dot1q | negotiate}
(config-if)# switchport nonnegotiate
```

If you use 802.1Q, specify a native VLAN for the trunk link with the command:

```
(config-if)# switchport trunk native vlan vlan-no
```

Frames from the native VLAN are sent over the trunk link untagged. Native VLAN must match on both sides of the trunk link. VLAN 1 is the default native VLAN for all ports, but best practice is to set the native VLAN to one not assigned to users. This practice also decreases the danger of having a large spanning tree instance in VLAN1.

## VLANs Allowed on the Trunk

By default, a trunk carries traffic for all VLANs. You can change that behavior for a particular trunk link by giving the following command at the interface config mode:

```
switchport trunk allowed vlan vlans
```

Make sure that both sides of a trunk link enable the same VLANs.

### Verifying a Trunk Link

Two commands you can use to verify your trunk configuration are

```
# show running-config
# show interfaces [interface no.] switchport | trunk
```

Using the **trunk** keyword with the **show interfaces** command gives information about the trunk link:

```
# show interfaces fastethernet 0/1 trunk
Port    Mode      Encapsulation Status    Native vlan
Fa0/1  desirable   n-802.1q     trunking   1
Port   Vlans allowed on trunk
Fa0/1   1-150
<further output omitted>
```

## Best Practices for Trunking

- Change the Native VLAN to one not assigned to any users.
- On links that should be trunks, turn off trunking negotiation by setting the mode to **trunk**, specifying the encapsulation type, and adding the **nonnegotiate** command.
- On links that should never be trunks, turn off trunking negotiation by setting the switchport mode to **host**. This sets it as an access port, enables Portfast, and disables EtherChannel negotiation.
- Limit the VLAN traffic carried by the trunk to only those VLANs it needs to carry.

# VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) is a Cisco-proprietary protocol that runs over trunk links and synchronizes the VLAN databases of all switches in the

VTP domain. A VTP domain is an administrative group; all switches within that group must have the same VTP domain name configured, or they do not synchronize databases.

VTP works by using Configuration Revision numbers and VTP advertisements:

- All switches send out VTP advertisements every five minutes or when there is a change to the VLAN database (when a VLAN is created, deleted, or renamed).
- VTP advertisements contain a Configuration Revision number. This number is increased by one for every VLAN change.
- When a switch receives a VTP advertisement, it compares the Configuration Revision number against the one in its VLAN database.
- If the new number is higher, the switch overwrites its database with the new VLAN information and forwards the information to its neighbor switches.
- If the number is the same, the switch ignores the advertisement.
- If the new number is lower, the switch replies with the more up-to-date information contained in its own database.

## VTP Switch Roles

A switch can be a VTP:

- **Server:** The default VTP role. Servers can create, delete, and rename VLANs. They originate both periodic and triggered VTP advertisements and synchronize their databases with other switches in the domain.
- **Client:** Clients cannot make VLAN changes. They originate periodic VTP advertisements and synchronize their databases with other switches in the domain.
- **Transparent:** It can create, delete, and rename VLANs, but its VLANs are only local. It does not originate advertisements or synchronize its database with any other switches. It forwards VTP advertisements out its trunk links, however.

The two versions of VTP are Version 1 and Version 2. To use Version 2, all switches in the domain must be capable of using it. Configure one server for

Version 2, and the information is propagated through VTP. Version 2 has the following added features:

- It supports Token Ring VLANs.
- Transparent switches pass along messages from both versions of VTP.
- Consistency checks are performed only when changes are configured through the CLI or SNMP.

## Configuring VTP

VTP configuration is done at the global config mode. To configure the switch's VTP mode:

```
(config)# vtp {server | client |transparent}
```

To configure the VTP domain name:

```
(config)# vtp domain name
```

To configure a VTP password (all switches in the domain must use the same password):

```
(config)# vtp password password
```

To configure the switch to use VTP Version 2:

```
(config)# vtp version 2
```

## Verifying and Monitoring VTP

To get basic information about the VTP configuration, use `show vtp status`. The example shows the default settings:

```
# show vtp status
VTP Version  : 1
Configuration Revision  : 0
Maximum VLANs supported locally  : 1005
Number of existing VLANs  : 5
VTP Operating Mode  : Server
VTP Domain Name  :
(config)#
VTP Pruning Mode  : Disabled
VTP V2 Mode  : Disabled
VTP Traps Generation  : Disabled
MD5 digest  :
```

## Adding a New Switch to a VTP Domain

Adding a new switch in client mode does not prevent it from propagating its incorrect VLAN information. A server synchronizes to a client if the client has the higher configuration revision number. You must reset the revision number back to 0 on the new switch. To be safe, follow these steps:

1. With the switch disconnected from the network, set it as VTP transparent and delete the vlan.dat file from its flash memory.
2. Set it to a fake VTP domain name and into client mode.
3. Reboot the switch.
4. Configure the correct VTP settings, such as domain, password, mode, and version.
5. Connect the switch to the network, and verify that it receives the correct information.

# EtherChannels

An EtherChannel is a way of combining several physical links between switches into one logical connection. Normally, Spanning Tree blocks redundant links; EtherChannels get around that and enable load balancing across those links. Traffic is balanced between the channel links on the basis of such things as source or destination MAC address or IP address. The EtherChannel load-balancing method is configured at global configuration mode.

```
(config)# port-channel load-balance type
```

A logical interface—called the Port Channel interface—is created. Configuration can be applied to both the logical and physical interfaces.

Some guidelines for EtherChannels follows:

- Interfaces in the channel do not have to be physically next to each other or on the same module.
- All ports must be the same speed and duplex.
- All ports in the bundle should be enabled.
- None of the bundle ports can be a SPAN port.
- Assign an IP address to the logical Port Channel interface, not the physical ones, if using a Layer 3 EtherChannel.

- Put all bundle ports in the same VLAN, or make them all trunks. If they are trunks, they must all carry the same VLANs and use the same trunking mode.
- The configuration you apply to the Port Channel interface affects the entire EtherChannel. The configuration you apply to a physical interface affects only that interface.

## Configuring an EtherChannel

Basically, you should configure the logical interface and then put the physical interfaces into the channel group:

```
(config)# interface port-channel number
![any additional configuration, such as trunking for a Layer 2
 EtherChannel]
```

For a Layer 3 EtherChannel, add the following:

```
(config-if)# no switchport
(config-if)# ip address address mask
```

Then, at each port that is part of the EtherChannel, use the following:

```
(config)# interface { number | range interface - interface}
(config-if)# channel-group number mode {auto | desirable | on}
```

Putting the IP address on the Port Channel interface creates a Layer 3 EtherChannel. Simply putting interfaces into a channel group creates a Layer 2 EtherChannel, and the logical interface is automatically created.

The Cisco proprietary Port Aggregation Protocol (PAgP) dynamically negotiates the formation of a channel. There are three PAgP modes:

- **On:** The port channels without using PAgP negotiation. The port on the other side must also be set to On.
- **Auto:** Responds to PAgP messages but does not initiate them. Port channels if the port on the other end is set to Desirable. This is the default mode.
- **Desirable:** Port actively negotiates channeling status with the interface on the other end of the link. Port channels if the other side is Auto or Desirable.

Link Aggregation Control Protocol (LACP) is an IEEE standard protocol, IEEE 802.3ad, which does the same thing. LACP modes follow:

- **On:** The port channels without using LACP negotiation. The port on the other side must also be set to On.
- **Active:** Port actively negotiates channeling with the port on the other end of the link. A channel forms if the other side is Passive or Active.
- **Passive:** Responds to LACP messages but does not initiate them. A channel forms only if the other end is set to Active.

If you want to use LACP, specify it under the interface and put the interface in either active or passive mode:

```
(config-if)# channel-protocol lacp(config-if)channel-group number
 mode {active/passive}
```

# Verifying an EtherChannel

Some typical commands for verifying include the following:

```
# show running-config interface number
# show interfaces number etherchannel
# show etherchannel number port-channel
# show etherchannel summary
# show etherchannel load-balance
```

# Troubleshooting VLAN Issues

Configuration problems can arise when user traffic must traverse several switches. The following sections list some common configuration errors. But before you begin troubleshooting, create a plan. Check the implementation plan for any changes recently made, and determine likely problem areas.

## Troubleshooting User Connectivity

User connectivity can be affected by several things:

- **Physical connectivity:** Make sure the cable, network adapter, and switch port are good. Check the port’s link LED.
- **Switch configuration:** If you see FCS errors or late collisions, suspect a duplex mismatch. Check configured speed on both sides of the link. Make sure the port is enabled and set as an access port.

- **VLAN configuration:** Make sure the hosts are in the correct VLAN.
- **Allowed VLANs:** Make sure that the user VLAN is allowed on all appropriate trunk links.

## Troubleshooting Trunking

When troubleshooting trunking, make sure that physical layer connectivity is present before moving on to search for configuration problems such as

- Are both sides of the link in the correct trunking mode?
- Is the same trunk encapsulation on both sides?
- If 802.1Q, is the same native VLAN on both sides? Look for CDP messages warning of this error.
- Are the same VLANs permitted on both sides?
- Is a link trunking that should not be?

## Troubleshooting VTP

The following are some common things to check when troubleshooting problems with VTP:

- Make sure you are trunking between the switches. VTP is sent only over trunk links.
- Make sure the domain name matches on both switches. (The name is case sensitive.)
- If the switch is not updating its database, make sure it is not in transparent mode.
- If using passwords, make sure they all match. To remove a password, use `no vtp password`.
- If VLANs are missing, check the Revision number for a possible database overwrite. Also check the number of VLANs in the domain. There might be too many VLANs for VTP to update properly.

CHAPTER 3

# Spanning Tree

Ethernet network design balances two separate imperatives. First, Ethernet has no capacity for detecting circular paths. If such paths exist, traffic loops around and accumulates until new traffic is shut out. (This is called a broadcast storm.) Second, having secondary paths is good preparation for inevitable link failure.

Spanning Tree is a protocol that prevents loop formation by detecting redundant links and disabling them until needed. Designers can therefore build redundant links, and the protocol enables one to pass traffic and keep the other in reserve. When the active link fails, the secondary link is enabled quickly.

## Understanding the Spanning Tree Protocol

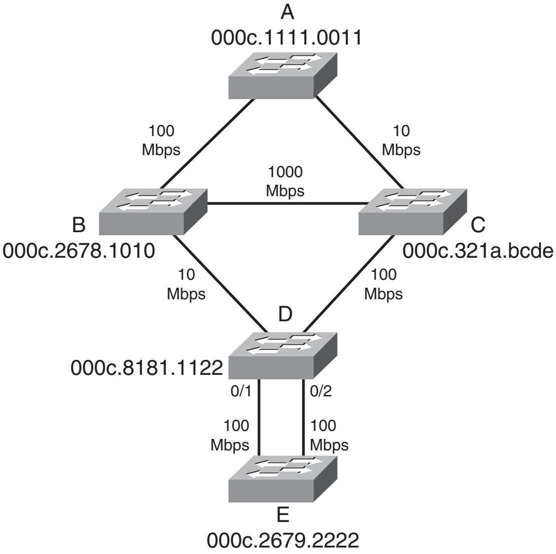
Switches either forward or filter Layer 2 frames. The way they make the forwarding/filtering decision can lead to loops in a network with redundant links. Spanning Tree is a protocol that detects potential loops and breaks them.

A Layer 2 switch is functionally the same thing as a transparent bridge. Transparent bridges:

- Learn MAC (Media Access Control) addresses by looking at the source address of incoming frames. They build a table mapping MAC address to port number.
- Forward broadcasts and multicasts out all ports except the one which they came. (This is called flooding.)
- Forward unknown unicasts out all ports except the one in which they came. An unknown unicast is a message bound for a unicast MAC address that is not in the switch's table of addresses and ports.
- Do not make any changes to the frames as they forward them.

Spanning Tree Protocol (STP) works by selecting a root bridge and then selecting one loop-free path from the root bridge to every other switch. (STP uses the term *bridge* because it was written before there were switches.) Consider the following switched network (see Figure 3-1).

**Figure 3-1 Example Switched Topology**



Spanning Tree must select

- One root bridge
- One root port per nonroot bridge
- One designated port per network segment

## Spanning Tree Election Criteria

Spanning Tree builds paths out from a central point along the fastest available links. It selects paths according to the following criteria:

- Lowest root bridge ID (BID)
- Lowest path cost to the root
- Lowest sender bridge ID
- Lowest sender port ID (PID)

When reading the path selection criteria, remember the following:

- **Bridge ID:** Bridge priority: Bridge MAC address.
- **Bridge priority:** 2-btye value, 0–65,535 (0–0xFFFF).
- **Default priority:** 32,768 (0x8000).
- **Port ID:** Port priority: port number.
- **Port priority:** A 6-bit value, 0–63, default is 32.
- **Path cost:** This is the cumulative value of the cost of each link between the bridge and the root. Cost values were updated in 2000, and you should see only new cost values, but both are given in the following table (see Table 3-1). Old and new switches work together.

**Table 3-1 Spanning Tree Costs**

| Link Speed | Previous IEEE Specification | Current IEEE Specification |
|---|---|---|
| 10 Mb/s | 100 | 100 |
| 100 Mb/s | 10 | 19 |
| 1 Gbps | 1 | 4 |
| 10 Gbps | 1 | 2 |

## STP Election

Spanning Tree builds paths out from a starting point, the “root” of the tree. The first step in selecting paths is to identify this root device. Then each device selects its best path back to the root, according to the criteria laid out in the previous sections (lowest root BID, lowest cost, lowest advertising BID, lowest port ID).

### Root Bridge Election

Looking at Figure 3-1, first select the root bridge. Assume each switch uses the default priority.

- Switch A BID = 80–00–00–0c-11–11–00–11
- Switch B BID = 80–00–00–0c–26–78–10–10
- Switch C BID = 80–00–00–0c-32–1a-bc-de
- Switch D BID = 80–00–00–0c-81–81–11–22
- Switch E BID = 80–00–00–0c–26–79–22–22

Switch A has the lowest BID, so it is the root. Each nonroot switch must now select a root port.

## Root Port Election

The root port is the port that leads back to the root. Continuing with Figure 3-1, when A is acknowledged as the root, the remaining bridges sort out their lowest cost path back to the A:

- **`Switch B:`** Uses the link to A with a cost of 19 (link speed of 100 Mb/s).
- **`Switch C:`** The connected link has a cost of 100 (Ethernet), the link through B has a path cost of 38 (two 100-Mb/s links), and so B is chosen.
- **`Switch D:`** The link through B has a path cost of 119, the path cost through C to A is 119, the path through C then B is 57, so C is chosen.
- **`Switch E:`** The lowest path cost is the same for both ports (76 through D to C to B to A). Next check sender BID—sender for both ports is D so that it does not break the tie. Next check sender Port ID. Assuming default port priority, the PID for 0/1 is lower than the PID for 0/2, so the port on the left is the root port.

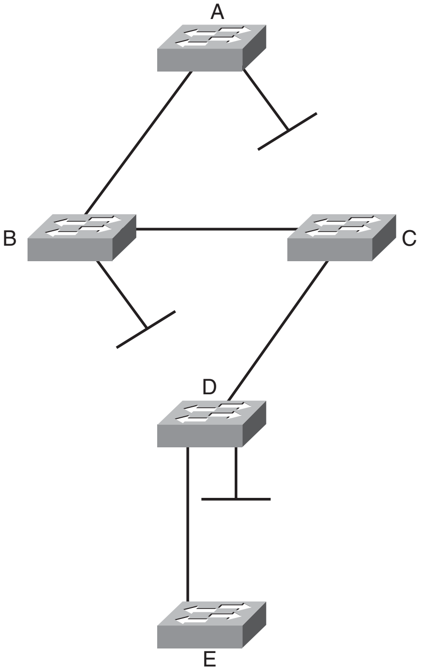
## Designated Port Election

Designated ports are ports that lead away from the root. Obviously, all ports on the root bridge are designated ports (A–B and A–C in Figure 3-1).

- **Segment B–D:** B has the lowest path cost to root (19 versus 119), so it is designated for this segment.
- **Segment C–D:** C has the lowest path cost to the root (100 versus 119), so it is designated for this segment.
- **Segment B–C:** B has the lowest path cost to the root (19 versus 100), so it is designated for this segment.
- **Both segments D–E:** D has the lowest cost to the root (57 versus 76), so it is designated for both segments.

Now the looped topology has been turned into a tree with A at the root. Notice that there are no more redundant links.

**Figure 3-2 Active Topology After Spanning Tree Is Complete**



## Bridge Protocol Data Units

Switches exchange Bridge Protocol Data Units (BPDU). The two types of BPDUs are Configuration and Topology Change Notification(TCN). Configuration BPDUs are sent every two seconds from the root toward the downstream switches. They:

- Are used during an election
- Maintain connectivity between switches
- Send timer information from the root

TCN BPDUs are sent by a downstream switch toward the root when:

- There is a link failure.
- A port starts forwarding, and there is already a designated port.
- The switch receives a TCN from a neighbor.

When a switch receives a TCN BPDU, it acknowledges that with a configuration BPDU that has the TCN Acknowledgment bit set.

When the root bridge receives a TCN, it starts sending configuration BPDUs with the TCN bit set for a period of time equal to max age plus forward delay. Switches that receive this change their MAC table aging time to the Forward Delay time, causing MAC addresses to age faster. The topology change also causes an election of the root bridge, root ports, and designated ports.

Some of the fields in the BPDU include

- **Root bridge ID:** The BID of the current root
- **Sender's root path cost:** The cost to the root
- **Sender's bridge ID:** Sender's priority concatenated to MAC
- **Sender's port ID:** The port number, transmitted as final tie-breaker
- **Hello time:** Two seconds by default
- **Forward Delay:** Fifteen seconds by default
- **Max Age:** Twenty seconds by default

## Spanning Tree Port States

When a port is first activated, it transitions through the following stages shown in Table 3-2.

**Table 3-2 Spanning Tree Port States**

| Port State | Timer | Action |
|---|---|---|
| Blocking | Max Age (20 sec) | Discards frames, does not learn MAC addresses, receives BPDUs |
| Listening | Forward Delay (15 sec) | Discards frames, does not learn MAC addresses, receives BPDUs to determine its role in the network |
| Learning | Forward Delay (15 sec) | Discards frames, does learn MAC addresses, receives and transmits BPDUs |
| Forwarding | | Accepts frames, learns MAC addresses, receives and transmits BPDUs |

## Per-VLAN Spanning-Tree

The IEEE's version of STP assumes one common Spanning-tree instance (and thus one root bridge) regardless of how many VLANs are configured.

With the Cisco Per-VLAN Spanning-Tree (PVST+) there is a different instance of STP for each VLAN. To derive the VLAN BID, the switch picks a different MAC address from its base pool for each VLAN. Each VLAN has its own root bridge, root port, and so on. You can configure these so that data flow is optimized, and traffic load is balanced among the switches by configuring different root bridges for groups of VLANs.

PVST+ is enabled by default on Cisco switches.

## Configuring Spanning Tree

To change the STP priority value, use the following:

```
Switch (config)# spanning-tree vlan vlan_no. priority value
```

To configure a switch as root without manually changing priority values, use the following:

```
Switch (config)# spanning-tree vlan vlan_no. root {primary | secondary}
```

To change the STP port cost for an access port, use the following:

```
Switch(config-if)# spanning-tree cost value
```

To change the STP port cost for a VLAN on a trunk port, use the following:

```
Switch(config-if)# spanning-tree vlan vlan_no. cost value
```

To display STP information for a VLAN, use the following:

```
Switch# show spanning-tree vlan vlan_no.
```

To display the STP information for an interface, use the following:

```
Switch # show spanning-tree interface interface_no. [detail]
```

To verify STP timers, use the following:

```
Switch # show spanning-tree bridge brief
```

## Portfast

Portfast is a Cisco-proprietary enhancement to Spanning Tree that helps speed up network convergence. It is for access (user) ports only. Portfast causes the port to transition directly to forwarding, bypassing the other STP states. Connecting a switch to a Portfast port can cause loops to develop. Configure Portfast on an interface or interface range:

```
(config-if)# spanning-tree portfast
```

It can also be configured globally:

```
(config)# spanning-tree portfast default
```

# Rapid Spanning Tree

Rapid Spanning Tree (RSTP) 802.1w is a standards-based, nonproprietary way of speeding STP convergence. Switch ports exchange an explicit handshake when they transition to forwarding. RSTP describes different port states than regular STP, as shown in Table 3-3.

**Table 3-3 Comparing 802.1d and 802.1w Port States**

| STP Port State | Equivalent RSTP Port State |
|---|---|
| Disabled | Discarding |
| Blocking | Discarding |
| Listening | Discarding |
| Learning | Learning |
| Forwarding | Forwarding |

## RSTP Port Roles

RSTP also defines different Spanning Tree roles for ports:

- **Root port:** The best path to the root (same as STP)
- **Designated port:** Same role as with STP
- **Alternate port:** A backup to the root port
- **Backup port:** A backup to the designated port
- **Disabled port:** Not used in the Spanning Tree
- **Edge port:** Connected only to an end user

## BPDU Differences in RSTP

In regular STP, BPDUs are originated by the root and relayed by each switch. In RSTP, each switch originates BPDUs, whether or not it receives a BPDU on its root port. All eight bits of the BPDU type field are used by RSTP. The TC and TC Ack bits are still used. The other six bits specify the port's role and its RSTP state and are used in the port handshake. The RSTP

BPDU is set to Type 2, Version 2. PVST is done by Rapid PVST+ on Catalyst switches.

## RSTP Fast Convergence

The Rapid Spanning Tree process understands and incorporates topology changes much quicker than the previous version:

- **RSTP uses a mechanism similar to BackboneFast:** When an inferior BPDU is received, the switch accepts it. If the switch has another path to the root, it uses that and informs its downstream switch of the alternative path.
- **Edge ports work the same as Portfast ports:** They automatically transition directly to forwarding.
- **Link type:** If you connect two switches through a point-to-point link and the local port becomes a designated port, it exchanges a handshake with the other port to quickly transition to forwarding. Full-duplex links are assumed to be point-to-point; half-duplex links are assumed to be shared.
- **Backup and alternate ports:** Ports that can transition to forwarding when no BPDUs are received from a neighbor switch (similar to UplinkFast).

If an RSTP switch detects a topology change, it sets a TC timer to twice the hello time and sets the TC bit on all BPDUs sent out its designated and root ports until the timer expires. It also clears the MAC addresses learned on these ports. Only changes to the status of non-Edge ports cause a TC notification.

If an RSTP switch receives a TC BPDU, it clears the MAC addresses on that port and sets the TC bit on all BPDUs sent out its designated and root ports until the TC timer expires. Enable and verify Rapid STP with the commands:

```
Switch(config)# spanning-tree mode rapid-pvst
Switch# show spanning-tree
```

A version of PVST+ is used with Rapid Spanning Tree, called Per-VLAN Rapid Spanning Tree (PVRST+). You should still configure root and secondary root bridges for each VLAN when using RSTP.

# Multiple Spanning Tree

With Multiple Spanning Tree (MST), you can group VLANs and run one instance of Spanning Tree for a group of VLANs. This cuts down on the number of root bridges, root ports, designated ports, and BPDUs in your network. Switches in the same MST Region share the same configuration and VLAN mappings. Configure and verify MST with these commands:

```
(config)# spanning-tree mode mst
(config)# spanning-tree mst configuration
(config-mst)# name region_name
(config-mst)# revision number
(config-mst)# instance number vlan vlan_range
(config-mst)# end
# show spanning-tree mst
```

To be compatible with 802.1Q trunking, which has one common Spanning Tree (CST) for all VLANs, MST runs one instance of an Internal Spanning Tree (IST). The IST appears as one bridge to a CST area and is MST instance number 0. The original MST Spanning Trees (called M-Trees) are active only in the region; they combine at the edge of the CST area to form one.

# Spanning Tree Stability Mechanisms

Spanning Tree has several additional tools for tuning STP to protect the network and keep it operating properly. They include

- PortFast (discussed previously)
- UplinkFast
- BackboneFast
- BPDU Guard
- BPDU Filtering
- Root Guard
- UDLD
- Loop Guard

## UplinkFast

UplinkFast is for speeding convergence when a direct link to an upstream switch fails. The switch identifies backup ports for the root port. (These are called an uplink group.) If the root port fails, one of the ports in the uplink group is unblocked and transitions immediately to forwarding; it bypasses the listening and learning stages. It should be used in wiring closet switches with at least one blocked port.

The command to enable uplinkfast is shown next. Please note that uplinkfast is enabled globally, so the command affects all ports and all VLANs.

```
(config)# spanning-tree uplinkfast
```

## BackboneFast

BackboneFast is used for speeding convergence when a link fails that is not directly connected to the switch. It helps the switch detect indirect failures. If a switch running BackboneFast receives an inferior BPDU from its designated bridge, it knows a link on the path to the root has failed. (An inferior BPDU is one that lists the same switch for the root bridge and designated bridge.)

The switch then tries to find an alternate path to the root by sending a Root Link Query (RLQ) frame out all alternate ports. The root then responds with an RLQ response, and the port receiving this response can transition to forwarding. Alternate ports are determined in this way:

- If the inferior BPDU was received on a blocked port, the root port and any other blocked ports are considered alternates.
- If the inferior BPDU was received on the root port, all blocked ports are considered alternates.
- If the inferior BPDU was received on the root port and there are no blocked ports, the switch assumes it has lost connectivity with the root and advertises itself as root.

Configure this command on all switches in the network:

```
(config)# spanning-tree backbonefast
```

## BPDU Guard

BPDU Guard prevents loops if another switch is attached to a Portfast port. When BPDU Guard is enabled on an interface, it is put into an error-disabled state (basically, shut down) if a BPDU is received on the interface. It can be enabled at either global config mode—in which case it affects all Portfast interfaces—or at interface mode. Portfast does not need to be enabled for it to be configured at a specific interface. The following configuration example shows BPDU guard being enabled and verified.

```
(config)# spanning-tree portfast bpduguard default
(config-if)# spanning-tree bpduguard enable
# show spanning-tree summary totals
```

## BPDU Filtering

BPDU filtering is another way of preventing loops in the network. It also can be enabled either globally or at the interface and functions differently at each. In global config, if a Portfast interface receives any BPDUs, it is taken out of Portfast status. At interface config mode, it prevents the port from sending or receiving BPDUs. The commands are

```
(config)# spanning-tree portfast bpdufilter default
 (config-if)# spanning-tree bpdufilter enable
```

## Root Guard

Root Guard is meant to prevent the wrong switch from becoming the Spanning Tree root. It is enabled on ports other than the root port and on switches other than the root. If a Root Guard port receives a BPDU that might cause it to become a root port, the port is put into “root-inconsistent” state and does not pass traffic through it. If the port stops receiving these BPDUs, it automatically reenables itself. To enable and verify Root Guard use the following commands:

```
(config-if)# spanning-tree guard root
# show spanning-tree inconsistentports
```

## Unidirectional Link Detection

A switch notices when a physical connection is broken by the absence of Layer 1 electrical keepalives. (Ethernet calls this a link beat.) However, sometimes a cable is intact enough to maintain keepalives but not to pass data in both directions. This is a Unidirectional Link. Operating at Layer 2, Unidirectional Link Detection (UDLD) detects a unidirectional link by sending periodic hellos out to the interface. It also uses probes, which must be acknowledged by the device on the other end of the link.

UDLD has two modes: normal and aggressive. In normal mode, the link status is changed to Undetermined State if the hellos are not returned. In aggressive mode, the port is error-disabled if a unidirectional link is found. Aggressive mode is the recommended way to configure UDLD.

To enable UDLD on all fiber-optic interfaces, use the following command:

```
(config)# udld [enable | aggressive]
```

Although this command is given at global config mode, it applies only to fiber ports.

To enable UDLD on nonfiber ports, give the same command at interface config mode.

To control UDLD on a specific fiber port, use the following command:

```
(config-if)# udld port {aggressive | disable}
```

To reenable all interfaces shut by UDLD, use the following:

```
# udld reset
```

To verify UDLD status, use the following:

```
# show udld interface
```

## Loop Guard

Loop Guard prevents loops that might develop if a port that should be blocking inadvertently transitions to the forwarding state. This can happen if the port stops receiving BPDUs (perhaps because of a unidirectional link or a software/configuration problem in its neighbor switch). When one of the ports in a physically redundant topology stops receiving BPDUs, the STP conceives the topology as loop-free. Eventually, the blocking port becomes designated and moves to forwarding state, thus creating a loop. With Loop Guard enabled, an additional check is made.

If no BPDUs are received on a blocked port for a specific length of time, Loop Guard puts that port into "loop inconsistent" blocking state, rather than transitioning to forwarding state. Loop Guard should be enabled on all switch ports that have a chance of becoming root or designated ports. It is most effective when enabled in the entire switched network in conjunction with UDLD.

To enable Loop Guard for all point-to-point links on the switch, use the following command:

```
(config)# spanning-tree loopguard default
```

To enable Loop Guard on a specific interface, use the following:

```
(config-if)# spanning-tree guard loop
```

Loop Guard automatically reenables the port if it starts receiving BPDUs again.

## Troubleshooting STP

Some common things to look for when troubleshooting Spanning Tree Protocol include

- **Duplex mismatch:** When one side of a link is half-duplex and the other is full-duplex. This causes late collisions and FCS errors.
- **Unidirectional link failure:** The link is up but data flows only in one direction. It can cause loops.
- **Frame corruption:** Physical errors on the line cause BPDUs to be lost, and the port incorrectly begins forwarding. This is caused by duplex mismatch, bad cable, or cable too long.
- **Resource errors:** STP is implemented in software, so a switch with an overloaded CPU or memory might neglect some STP duties.
- **Port Fast configuration errors:** Connecting a switch to two ports that have Port Fast enabled. This can cause a loop.
- **STP tuning errors:** Max age or forward delay set too short can cause a loop. A network diameter that is set too low causes BPDUs to be discarded and affects STP convergence.

## Identifying a Bridging Loop

Suspect a loop if you see the following:

- You capture traffic on a link and see the same frames multiple times.
- All users in a bridging domain have connectivity problems at the same time.
- There is abnormally high port utilization.

To remedy a loop quickly, shut redundant ports and then enable them one at a time. Some switches enable debugging of STP to help in diagnosing problems. The following commands are useful for isolating a bridging loop:

**show interfaces**

**show spanning tree**

**show bridge**

**show process cpu**

**debug spanning tree**

**show mac address-table aging-time** *vlan#*

**show spanning-tree vlan** *vlan#* **detail**

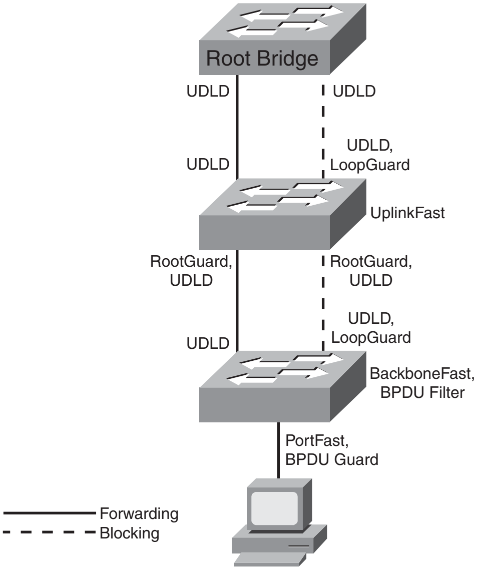
## Spanning-Tree Best Practices

To optimize data flow in the network, design and configure Spanning Tree in the following ways:

- Statically configure switches to be the primary and secondary root bridges by setting priority values.
- Consider which interfaces will become designated and root ports (possibly set port priorities/path cost).
- Tune STP using the tools detailed in this section.
- Enable UDLD aggressive mode on all fiber interfaces.
- Design STP domains that are as simple and contained as possible by using multilayer switches and routed links.
- Use PVRST+ or MST for the fastest convergence times.

Confused by all the acronyms and STP features? Figure 3-3 shows the STP tools you might use in your network and where you might use them.

**Figure 3-3 Example Switched Topology**



# CHAPTER 4

# InterVLAN Routing

VLANs divide the network into smaller broadcast domains but also prohibit communication between domains. To enable communication between those groups–without also passing broadcasts–routing is used.

## InterVLAN Routing Using an External Router

A Layer 2 switch can connect to a router to provide reachability between VLANs. This can be done either via separate physical links for each VLAN or via a trunk link from the switch to the router. A trunk link is most common and this type of setup is frequently called Router on a Stick.

When using a trunk link you must create separate subinterfaces on the router's physical interface—one subinterface for each VLAN plus one for the native VLAN. This can work with any kind of switch and the implementation is straightforward, but the router becomes a single point of failure for all users, and the trunk link might become congested.

The router's configuration would look similar to the following:

```
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.20
 description Voice VLAN
 encapsulation dot1Q 20
 ip address 10.1.20.1 255.255.255.0
!
interface FastEthernet0/1.99
 description Native VLAN
 encapsulation dot1Q 99 native
 ip address 10.1.99.1 255.255.255.0
!
interface FastEthernet0/1.120
 description Data VLAN
 encapsulation dot1Q 120
ip address 10.1.120.1.255.255.255.0
```

# InterVLAN Routing Using Multilayer Switches

A multilayer switch can do both Layer 2 switching and Layer 3 routing between VLANs. This section walks you through the switching process and focuses on order of operations. The order in which things happen is extremely important for two reasons. First, the order of events is good test material. Second, understanding the processing order allows you to evaluate how the various filtering and forwarding mechanisms interact. (Examples include error checking, access-lists, VLAN access-lists, routing, and QoS.)

## The Layer 2 and Layer 3 Forwarding Process

A multilayer switch does Layer 2 forwarding when the destination MAC address is mapped to one of its interfaces. The steps involved in Layer 2 forwarding are as follows:

Input

1. Receive frame
2. Verify frame integrity
3. Apply inbound VLAN ACL (VLAN Access Control List)
4. Look up destination MAC (Media Address Code)

Output

1. Apply outbound VLAN ACL
2. Apply outbound QoS ACL
3. Select output port
4. Place in port queue
5. Rewrite
6. Forward

A multilayer switch does Layer 3 forwarding when the destination MAC address is one of the switch's own addresses. The steps involved in Layer 3 forwarding are as follows:

Input

1. Receive frame.
2. Verify frame integrity.

3. Apply inbound VLAN ACL.
4. Look up destination MAC.

Routing

1. Apply input ACL
2. Switch if entry is in CEF cache
3. Apply output ACL

Output

1. Apply outbound VLAN ACL.
2. Apply outbound QoS ACL.
3. Select output port.
4. Place in interface queue.
5. Rewrite source and destination MAC, IP checksum and frame check sequence, and decrement TTL (Time to Live field in the IP header).
6. Forward.

## Understanding the Switching Table

Multilayer switches use Application Specific Integrated Circuits (ASIC) to forward packets at wire speed. The Content Addressable Memory (CAM) table, used for Layer 2 switching, is created by recording the source MAC address and ingress port of each frame. It contains binary values (0 or 1) and must find an exact match to have a hit.

In comparison, Multilayer Switching (MLS) uses aa Ternary Content Addressable Memory (TCAM) table to store information needed by Layer 3 and higher processing. This might include QoS and ACLs. Values in the TCAM table include ternary values (0, 1, or wildcard). An exact match is not required—the longest match is considered a hit.

## MLS Interfaces

A multilayer switch can have the following types of interfaces:

- **Layer 2 Interface:** Either an access port assigned to a VLAN or a trunk port.

- **Switch Virtual Interface (SVI):** A virtual, software interface for the VLAN itself. Can be either a Layer 2 interface or a Layer 3 interface.
- **Routed Interface:** A physical interface that is not associated with a VLAN and acts like a router port.

## SVI Configuration

A default SVI for VLAN 1 is automatically created in the switch. To create an SVI use the command **`interface`** *`vlan#`*. Configure an IP address on the SVI to make it a Layer 3 interface. SVIs are used to

- Route or fallback bridge between VLANs.
- Provide a default gateway for users in that VLAN.
- Route traffic into or out of its associated VLAN.
- Provide an IP address for connectivity to the switch itself.
- Provide an interface for routing protocols.

An SVI is considered "up" as long as at least one port in its associated VLAN is active and forwarding. If all ports in the VLAN are down, the interface goes down to avoid creating a routing black hole. You might not want the status of a particular port (one not connected to a host) to affect the SVI's status. Some Cisco switches enable you to use the following command on that interface.

```
Switch(config-if)# switchport autostate exclude
```

To configure InterVLAN routing using a Layer 3 SVI, you need to

- Enable IP routing.
- Create the VLANs.
- Create the SVIs.
- Associate an IP address with each SVI.
- Configure a dynamic routing protocol if needed.

```
Switch(config)#ip routing
Switch(config)# vlan 3
Switch(config)# interface vlan 3
Switch(config-if)#ip address 10.3.3.3 255.255.255.0
```

## Routed Switch Port Configuration

To configure an interface as a routed port, you must remove the Layer 2 functionality with the **no switchport** interface command. Then you can add an IP address and configure routing as needed:

```
sw1(config)# int fa 1/0/5
sw1(config-if)# no switchport
sw1(config-if)# ip address 10.5.5.5 255.255.255.0
```

To verify your configuration, use the commands **show ip interface brief**, **show interface**, or **show running-config interface** *int-#*.

# Understanding Switch Forwarding Architectures

Packets entering a router or multilayer switch are handled by one of three types of switching:

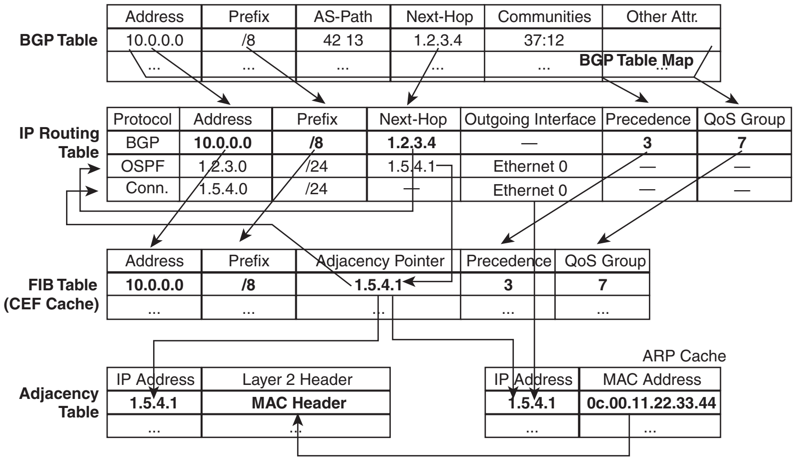
- **Process Switching:** Each packet must be examined by the CPU and handled in software. Slowest method, used in routers only.
- **Fast Switching:** CPU process switches the first packet in each flow, then caches that information, and switches subsequent packets in hardware. Faster than process switching, used in routers and multilayer switches. Also called route caching.
- **Cisco Express Forwarding (CEF):** A table is prebuilt with adjacency information for all destinations in the routing table. Fastest method, is the default for Cisco routers and multilayer switches. Also called topology-based switching.

# CEF Switching

Multilayer Switching (MLS) is a switch feature that enables the switch to route traffic between VLANs and routed interfaces in a highly optimized and efficient manner. Cisco Express Forwarding (CEF) is used to facilitate MLS (see Figure 4-1). Cisco Express Forwarding (CEF) does the following:

- Separates control plane hardware from data plane hardware.
- Controls plane runs in software and builds FIB and adjacency table.
- The data plane uses hardware to forward most IP unicast traffic.
- Uses TCAM table.
- Can be centralized or distributed.

**Figure 4-1 Cisco Express Forwarding**



BGP Table

| Address | Prefix | AS-Path | Next-Hop | Communities | Other Attr. |
|---|---|---|---|---|---|
| 10.0.0.0 | /8 | 42 13 | 1.2.3.4 | 37:12 | |
| ... | ... | ... | ... | ... | ... |

IP Routing Table

| Protocol | Address | Prefix | Next-Hop | Outgoing Interface | Precedence | QoS Group |
|---|---|---|---|---|---|---|
| BGP | **10.0.0.0** | **/8** | **1.2.3.4** | — | **3** | **7** |
| OSPF | 1.2.3.0 | /24 | 1.5.4.1 | Ethernet 0 | — | — |
| Conn. | 1.5.4.0 | /24 | — | Ethernet 0 | — | — |

FIB Table (CEF Cache)

| Address | Prefix | Adjacency Pointer | Precedence | QoS Group |
|---|---|---|---|---|
| **10.0.0.0** | **/8** | **1.5.4.1** | **3** | **7** |
| ... | ... | ... | ... | ... |

Adjacency Table

| IP Address | Layer 2 Header |
|---|---|
| **1.5.4.1** | **MAC Header** |
| ... | ... |

ARP Cache

| IP Address | MAC Address |
|---|---|
| **1.5.4.1** | **0c.00.11.22.33.44** |
| ... | ... |

Not all types of traffic can be handled by CEF. Some types that are *punted* (sent to the processor for handling) are

- Packets with IP header options
- Tunneled traffic
- 802.3 (IPX) or other unsupported encapsulation types
- Packets with an expiring TTL
- Packets that must be fragmented

## Configuring and Troubleshooting CEF

By default, CEF is on and supports per destination load sharing.

To disable CEF

- **4500:** Use `(config)# no ip cef`.
- **3500/3700:** On each interface, use `(config)# no ip route-cache cef`.
- 6550 with policy feature card, distributed FC, and multilayer switch FC: cannot be disabled.

View CEF information with the following:

```
# show interface fastethernet 2/2 | begin L3
```

View switching statistics with the following:

```
# show interface fastethernet 2/2 | include switched
```

View FIB with the following:

```
# show ip cef {interface} {detail}
```

View detailed CEF FIB entry with the following:

```
# show ip cef fastethernet 2/2 10.0.0.1 detail
```

Troubleshoot CEF drops with the following:

```
# show cef drop
```

Troubleshoot CEF adjacencies with the following:

```
# show adjacency
```

CHAPTER 5

# Implementing High Availability

A highly available network is the goal of every network engineer. Having a highly available network makes the job easier because it helps to prevent network outages and minimize downtime.

## Components of High Availability

There are five components to high availability: redundancy, technology, people, processes, and tools. The first two can be obtained through network design; the last three are more difficult to implement and control.

### Redundancy

Redundancy attempts to eliminate single points of failure by providing duplicate devices and links. This costs more, so the added cost must be balanced against the added benefit. Add redundancy where it will have the most impact on availability, in the core of your network, data center, or e-commerce module. Critical WAN or ISP connections are another possible location.

A redundant network has *path diversity* with multiple links between multiple devices. It can have *geographic diversity*, with data centers in multiple sites. Networks frequently have dual core and distribution switches, with dual uplinks to each. Dual WAN providers, with dual WAN edge routers, are commonly used. Companies can design their networks with connections to dual Telco central offices and power substations to achieve additional redundancy.

### Technology

Some of the technologies found in Cisco routers and Layer 3 switches enhance availability by providing routing continuity, fast failure detection to trigger a failover, and fast routing convergence. These include

- Cisco Nonstop Forwarding (NSF)
- Stateful Switchover (SSO)
- Stackwise technology on 3750 switches
- Virtual Switch System (VSS)
- Monitoring tools such as SNMP and Syslog
- IP Service Level Agreement (SLA)

Each of these technologies is discussed in later sections of this chapter.

Some other technological features that enhance availability include server load balancing, firewall stateful failover, and fast routing convergence.

## People

Although the "people" part of high availability is not usually under the control of the network engineer, it is an important part of the equation. The following items should be considered:

- **Staff work habits:** Staff should pay attention to detail, and their work should be reliable and consistent to make troubleshooting easier.
- **Staff skills and technical training:** A knowledgeable staff understands the network technologies and configures devices correctly. A company lab enables failover scenarios to be tested before incorporating them into the network and allows network engineers to practice their skills.
- **Communication and documentation:** There should be good communication between teams responsible for the network, security, servers, and applications. There should also be communication with users. Good documentation, readily available, is critical to understanding how the network is designed and how it should behave during a failure.
- **Sufficient time to accomplish a task:** Not having enough time to accomplish a network-related task leads to important components, such as testing and documentation, being left out. The design target should be a better than just "adequate" network.
- **Align staff with the services they support:** This helps ensure clear lines of responsibility for the different segments of the network. Be sure to include the people responsible for a segment in the planning for its high availability.

## Processes

Companies that build repeatable processes and design templates have more cohesive networks and save time in troubleshooting problems. Process documentation should include configuration change procedures, failover and lab testing procedures, and network implementation procedures. These should be regularly reviewed and improved as part of the PPDIOO process.

A lab that reflects the current production network enables thorough testing and validation of such changes as new configurations and IOS versions and ensures that the staff thoroughly understands network failover processes.

Having a meaningful change control process includes the complete testing of all changes and how they affect failover within the entire network before they are implemented. Changes must be well planned with a roll-back strategy in place. A risk analysis can also help determine if the change is worthwhile.

Network management processes are often overlooked. These should include

- Capacity audits
- IOS version management
- Corporate best-practice design compliance
- Disaster recovery and business continuity plans
- Evaluating the security impact of a proposed change

## Tools

A well-designed, highly available network can have a failure without it being noticed by users. It is important to have tools in place to monitor the network and send alerts when a failover occurs. Monitoring can also help spot problems as they begin to occur, enabling you to be proactive in your network management. There are many third-party tools available for this; some IOS tools are discussed in later sections of this chapter.

Good documentation is a critical tool to have. Good documentation includes up-to-date network diagrams with network addresses, VLAN information, and interface information. Important servers, applications, and services should be noted. Document not only HOW the network is designed, but also WHY it is designed that way.

# Resiliency and High Availability

A highly available network is a resilient network. A resilient network employs various methods to allow it to recover and continue operating in the event of a failure. Resiliency leads to high availability through the following components:

- **Network-level resiliency (the focus of this book):** This includes redundant links and redundant devices, but it doesn't stop there. Those devices must be configured so they fail between devices, or links, quickly.
- **System-level resiliency:** This includes redundancy within the hardware, such as dual power supplies, and cold-standby parts, such as extra stackable switches or switch modules. It also includes features within the hardware that enable fast failover.
- **Network management and monitoring:** You need to detect a failure immediately and be informed of the actions taken automatically to remediate it.

## Network Level Resiliency

Redundant links were discussed in Chapter 2. STP blocks a redundant link by default so that they are in an active/backup configuration. Etherchannels enables multiple links to be active. If a failure occurs they distribute traffic across the remaining links.

Configure your devices for fast convergence to avoid traffic drops when a link fails. RSTP is preferred over 802.1D STP because it provides faster failover. Use routing protocols such as EIGRP that have fast convergence times. You might need to tune the Layer 2 and Layer 3 protocol timers.

For accurate monitoring statistics, it is important that network clocks are synchronized. Use NTP for this. Syslog, SNMP, and IP SLA are some tools that help you monitor and track your network's resiliency. They are discussed in more detail in a future section.

## Fast Failover

When measuring network resiliency, you must consider how long it takes for failover and convergence at all layers of the OSI stack, not just Layers 1–3. Table 5-1 outlines some of the typical convergence times.

**Table 5-1 Convergence Times for Network Components**

| Network Component | Convergence Time |
|---|---|
| Rapid Spanning Tree | Subsecond for minor failures, 1–2 seconds for major failures. |
| Etherchannel | Approximately 1 second to redirect traffic to a different link in the channel. |
| First Hop Redundancy Protocols such as HSRP, VRRP, or GLBP | Default of 10 seconds. Recommended tuning of hello time to 1 second and hold time to 3 second yields a 3 second convergence time. |
| Routing Protocols | Subsecond for OSPF and EIGRP with recommended tuning of timers. |
| Switch Service Modules | Typically 3–5 seconds. Exception is Cisco Application Control Engine (ACE) with 1 second failover in active/active configuration. |
| Computer/Server TCP Stacks | 9-second session teardown for Windows, longer for other OSs. |

# Optimizing Redundancy

You should be aware that redundancy does not always equal resiliency. Too much redundancy can increase the network complexity to a point that it becomes harder to troubleshoot and actually leads to a less-available network. There are too many paths for the data to follow, so it becomes less deterministic. The cost is much higher, also.

## NSF with SSO

Layers 2–4 convergence time is enhanced in Cisco 4500 and 6500 series switches with redundant route processors (RP) by using NSF with SSO. When using this, only one RP is active. The standby RP synchronizes its configuration and dynamic state information (such as CEF, MAC, and FIB tables) with the active RP. When the active RP fails, SSO enables the standby RP to take over immediately. NSF keeps the switch forwarding traffic during the switchover, using the existing route and CEF tables. The goal of NSF with SSO is to prevent routing adjacencies from resetting, which prevents a routing flap. The switchover to the new RP must be completed before routing timers expire, or the router's neighbors will tear down their adjacency and routing will be disrupted.

When the new RP is up, the old routes are marked as stale, and the RP asks its routing peers to refresh them. When routing is converged, it updates the routing and CEF tables on the switch and the linecards.

NSF is supported with EIGRP, OSPF, ISIS, and BGP. An *NSF-capable router* supports NSF; an *NSF-aware router* does not support NSF but understands it and continues forwarding traffic during SSO.

Use NSF with SSO in locations where you do not have a duplicate switch for failover, such as at the user access or Enterprise network edge. Otherwise it can actually cause longer convergence. Routing protocols timers can be tuned very short to provide fast convergence. With SSO, the switchover to the standby RP might not occur before the tuned routing Dead timer expires, and the adjacency would be reset.

## Designing for Redundancy

Figure 5-1 shows where you would typically use redundancy within a campus network. Access switches are either chassis-based with dual Supervisor engines and dual power supplies or are stackable switches. They have redundant, fully meshed links to redundant distribution switches, which, in turn, have redundant links to redundant core switches. Distribution and core switch pairs are connected via a Layer 2 or Layer 3 link. This design minimizes single points of failure and enables the network to recover from a link or switch failure.

## Layer 2 Versus Layer 3 Access Design

You can use a Layer 2 or a Layer 3 access layer. When using L2, VLANs can either be distributed across multiple switches or local to each switch. Figure 5-2 shows L2 access switches with VLAN 10 on both of them. This design is not recommended. The FHRP Active switch and the STP Root must be statically configured as the same switch. STP blocks one uplink per access switch. RSTP helps speed convergence.

There must be a physical link between distribution switches, and it should be a L2 trunk. Without that link, any traffic between switches must go through an access switch. Additionally, failure of one of the access-to-distribution uplinks causes packets to be dropped until the FHRP dead timer expires.

Figure 5-1 **Designing for Redundancy**

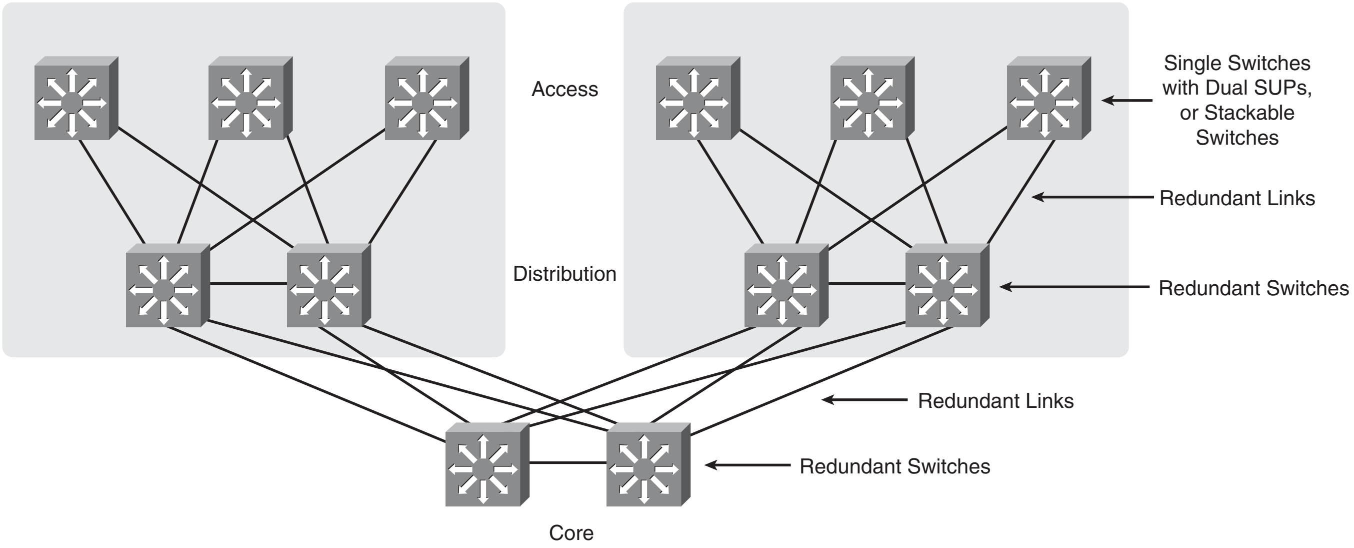


**Figure 5-2 Layer 2 Access Switches with Distributed VLANs**

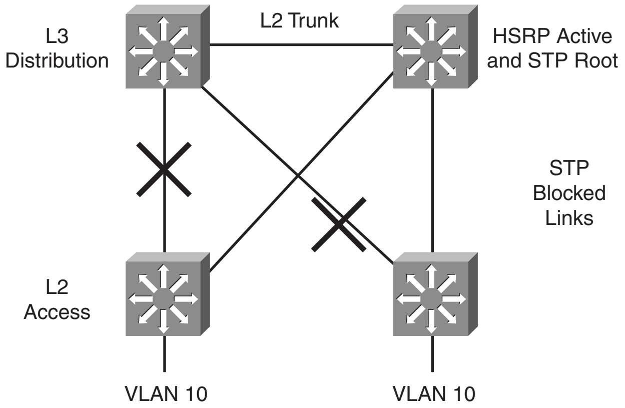
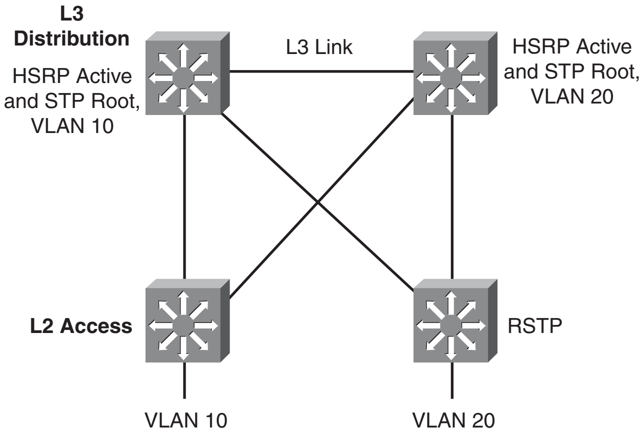


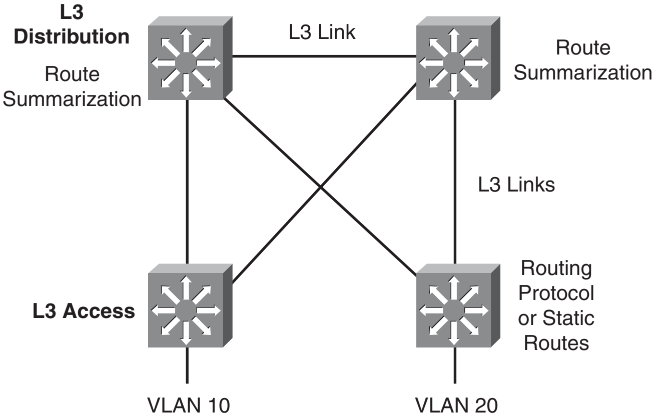
Figure 5-3 shows the recommended design when using L2 access switches. Each VLAN is local to one switch. The FHRP Active and STP Root must still be the same switch. They are still statically configured per VLAN so that traffic flow will be deterministic. Because the link between distribution switches is L3, there are no L2 loops. Thus no links are blocked by STP. However, traffic does not load balance between links because each switch forwards traffic only over the link to its HSRP Active and STP Root switch. RSTP is still used for faster convergence.

**Figure 5-3 Layer 2 Access Switches with Local VLANs**



In Figure 5-4 the access switches are L3. This gives the faster convergence and is easiest to implement. All links between switches are L3. There is no need for HSRP, although STP should still be enabled in case of a misconfiguration. Access switches can load balance traffic across both uplinks. The access switches either run a routing protocol or use static routes. The distribution switches summarize routes for the access VLANs.

**Figure 5-4 Layer 3 Access Switches**



## Using Nonchassis Based Access Switches

Using more than one stand-alone switch, such as the Cisco 3560 or 3750, in an access closet requires special design consideration. You can either daisy-chain the switches or use the Cisco Stackwise technology. When you daisy-chain switches, the top and bottom members of the chain typically uplink to one distribution switch each. You must add a link (or *loopback cable*) between the top and bottom switch. Otherwise, a failure in the link between two access switches might cause return traffic to be blackholed. Alternatively you can configure the link between the distribution switches as an L2 trunk.

Stackwise switches enable you to manage each group of access switches as one. Two stack member switches uplink to the distribution switches. Special cables connect the switches, and you should still connect the top and bottom members of the stack using a Stackwise cable. The link between distribution switches can then be an L3 link without worry of blackholing return traffic.

# Network Management for High Availability

Network administrators use network management tools:

- To verify network performance
- To characterize, or baseline, network performance

- To understand amount and direction of traffic flow within the network
- To troubleshoot network problems

## Syslog

Cisco devices produce system logging (or *syslog*) messages that can be output to the device console, VTY connection, system buffer, or remote syslog server. If sent to a syslog server, messages are sent on UDP port 514. You are probably familiar with the syslog message `%SYS-5-CONFIG_I: Configured from console by console`, for instance. A syslog message always starts with the percent sign and has the following format:

```
%FACILTY-SUBFACILITY-SEVERITY-MNEMONIC: message text
```

Each portion of a syslog message has a specific meaning:

- `FACILITY-SUBFACILITY`: This tells the protocol, module, or process that generated the message. Some examples are SYS for the operating system, OSPF, IF for an interface, and IP.
- `SEVERITY`: A number from 0 to 7 designating the importance of the action reported. The levels are
  - Emergency: 0
  - Alert: 1
  - Critical: 2
  - Error: 3
  - Warning: 4
  - Notice: 5
  - Informational: 6
  - Debugging: 7
- `MNEMONIC`: A code that identifies the action reported.
- A plain-text description of the event that triggered the syslog message.

## SNMP

An SNMP manager collects information from SNMP agents residing on network devices, either through regular polling or by event-generated traps.

The information is stored on the local device in a Management Information Base (MIB). Access to the MIB is controlled by SNMP community strings. Access can be read-only (RO) or read-write(RW).

There are three versions of SNMP. Versions 1 and 2 send the community strings in clear text. They cannot authenticate the source of a message or encrypt a message. Therefore they should be used only for read-only access. SNMPv3 adds three security levels:

- `noAuthNoPriv`: Neither authenticates nor encrypts
- `authNoPriv`: Authenticates the sender but does not encrypt the message
- `authPriv`: Both authenticates the sender and encrypts the message

The following configuration creates a standard access list that allows only traffic sourced from the host at 10.1.1.1. Two community-strings are created, “ccnp” for read-only access and “c1sc0” for read-write access. Read-write access is permitted only from the host specified in access list 1. Next, the SNMP server address is given, along with the command to send traps messages to that server. Because SNMP version 3 is used, the username “admin” is needed:

```
sw1(config)# access-list 1 permit 10.1.1.1
sw1(config)# snmp-server community ccnp ro
sw1(config)# snmp-server community c1sc0 rw 1
sw1(config)# snmp-server host 10.1.1.2 traps admin
```

## IP SLA

IP SLA is a feature that enables a Cisco router or switch to simulate specific types of traffic and send it to a receiver, called a *responder*. IP SLA probes can simulate various types of traffic, such as HTTP, FTP, DHCP, UDP jitter, UDP echo, HTTP, TCP connect, ICMP echo, ICMP path echo, ICMP path jitter, and DNS, and can report statistics such as path jitter. It has highly granular application configuration options such as TCP/UDP port numbers, TOS byte, and IP prefix bits. This is useful for measuring application performance end-to-end across your network. It can also be used to track reachability and then decrement HSRP priority values or bring up secondary links. Additionally, IP SLA can also be used as a measure of reliability and continuous availability. SNMP traps can be generated from events such as connection loss, timeout, roundtrip time threshold, average jitter threshold, one-way packet loss, one-way jitter, and one-way latency.

To enable IP SLA, configure the source to send the required type of data probes. The receiver can be a computer, or it can be another Cisco device. The configuration of a Cisco responder is simple. Use the global **ip sla responder** command. One benefit of using a Cisco device as the responder is that it can add time stamps to help measure latency and jitter. These time stamps take into account the device processing time so that the measurement reflects only network latency.

The configuration of the IP SLA source is more complex. You must create a monitor session, list the traffic type, responder IP address, and any other desired variables such as DSCP value. Then you schedule the probes. Optionally configure tracking using the IP SLA session. The following commands set up an IP SLA session that measures UDP jitter for a voice port. Traffic is sent every 120 seconds, starting when the last command is given and continues until it is manually stopped:

```
sw1(config)#ip sla 1
sw1(config-ip-sla)#udp-jitter 10.1.1.3 65422 codec g729a
sw1(config-ip-sla-jitter)#frequency 120
sw1(config-ip-sla-jitter)#exit
sw1(config)#ip sla schedule 1 life forever start-time now
```

## CHAPTER 6

# First Hop Redundancy

Specifying a default gateway leads to a single point of failure. Proxy Address Resolution Protocol (ARP) is one method for hosts to dynamically discover gateways, but it has issues in a highly available environment. With Proxy ARP:

- Hosts ARP for all destinations, even remote.
- Router responds with its MAC.
- Problem: Slow failover because ARP entries take minutes to timeout.

Instead of making the host responsible for choosing a new gateway, router redundancy protocols enable two or more routers to support a shared MAC address. If the primary router is lost, the backup router assumes control of traffic forwarded to that MAC. This section refers to routers but includes those multilayer switches that can also implement Layer 3 redundancy.

## Hot Standby Router Protocol

Hot Standby Router Protocol (HSRP) is a Cisco proprietary protocol.

With HSRP, two or more devices support a virtual router with a fictitious MAC address and unique IP address. Hosts use this IP address as their default gateway and the MAC address for the Layer 2 header. The virtual router's MAC address is 0000.0c07.AC*xx*, in which *xx* is the HSRP group. Multiple groups (virtual routers) are allowed.

The *Active* router forwards traffic. The *Standby* is backup. The standby monitors periodic hellos (multicast to 224.0.0.2, UDP port 1985) to detect a failure of the active router. On failure, the standby device starts answering messages sent to the IP and MAC addresses of the virtual router.

The active router is chosen because it has the highest HSRP priority (default priority is 100). In case of a tie, the router with the highest configured IP address wins the election. A new router with a higher priority does not cause an election unless it is configured to *preempt*—that is, take over from a lower priority router. Configuring a router to preempt also ensures that the highest priority router regains its active status if it goes down but then comes back online again.

Interface tracking reduces the active router's priority if a specified circuit is down. This enables the standby router to take over even though the active router is still up.

## HSRP States

HSRP devices move between these states:

- **Initial:** HSRP is not running.
- **Learn:** The router does not know the virtual IP address and is waiting to hear from the active router.
- **Listen:** The router knows the IP and MAC of the virtual router, but it is not the active or standby router.
- **Speak:** Router sends periodic HSRP hellos and participates in the election of the active router.
- **Standby:** Router monitors hellos from active router and assumes responsibility if active router fails.
- **Active:** Router forwards packets on behalf of the virtual router.

**Note**

The Learn state is not actually seen in debugs of HSRP.

## Configuring HSRP

To begin configuring HSRP, use the **standby** *group-number* **ip** *virtual-IP-address* command in interface configuration mode. Routers in the same HSRP group must belong to the same subnet/virtual LAN (VLAN.) Give this command under the interface connecting to that subnet or VLAN. For instance, use the following to configure the router as a member of HSRP group 39 with virtual router IP address 10.0.0.1:

```
Router(config-if)# standby 39 ip 10.0.0.1
```

HSRP authentication helps prevent unauthorized routers from seeing user traffic:

```
Router(config-if)# stand 2 authentication md5 key-string cisco
```

Tune HSRP with four options: Priority, Preempt, Timers, and Interface Tracking.

Manually select the active router by configuring its priority higher than the default of 100:

```
Router(config-if)# standby 39 priority 150
```

Along with configuring priority, configure **preempt** to enable a router to take over if the active router has lower priority, as shown in the following commands. This helps lead to a predictable data path through the network. The second command shown delays preemption until the router or switch has fully booted and the routing protocol has converged. Time how long it takes to boot and add 50 percent to get the delay value in seconds:

```
Router(config-if)# standby 39 preempt
Router(config-if)# standby 39 preempt delay minimum 90
```

Speed convergence by changing the hello and hold times. The following sets the hello interval to 2 seconds and the hold time to 3 seconds. They can be set between 1–255 seconds (the default hello is 3 seconds and hold time is 10 seconds):

```
Router(config-if)# standby 39 timers 1 3
```

Tracking an interface can trigger an election if the active router is still up but a critical interface (such as the one to the Internet) is down. In the following, if serial 1/0/0 is down, the router's HSRP priority is decremented by 100 (the default value to decrement is 10):

```
Router(config-if)# standby 39 track s1/0/0 100
```

Another way to track an indirect connection is to use IP SLA (described in Chapter 5). With IP SLA tracking, HSRP can failover to the standby router if any connection on the path to a remote location fails or exceeds link-quality thresholds. The following sample configuration shows how to add tracking an IP SLA session number 5 to an existing HSRP interface configuration:

```
Router(config)#ip sla 5
Router(config-ip-sla)# udp-jitter 172.17.1.2 16000
Router(config)#track 10 rtr 5
Router(config-if)# int fa 1/0/15
Router(config-if)# stand 2 track 10 decrement 50
```

**Note**

The standby router must be configured with the preempt command for it to take control.

Multiple HSRP standby groups can be configured, and the same router can be active for some groups and standby for others by adjusting priorities. You can have a maximum of 255 groups. When using Layer 3 switches, configure the same switch as the primary HSRP router and the Spanning Tree root.

## Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) is similar to HSRP, but it is an open standard (RFC 2338). Two or more devices act as a virtual router. With VRRP, however, the IP address used can be either a virtual one or the actual IP address of the primary router. VRRP is supported only on Cisco 4500 and 6500 series switches.

The VRRP *Master* router forwards traffic. The master is chosen because it owns the real address, or it has the highest priority. (The default is 100.) If a real address is supported, the owner of real address *must* be master. A *Backup* router takes over if the master fails, and there can be multiple backup routers. They monitor periodic hellos multicast by the master to 224.0.0.18, using UDP port 112, to detect a failure of the master router.

Multiple VRRP groups are allowed, just as with HSRP.

Routers in the same VRRP group must belong to the same subnet/VLAN. To enable VRRP, give this command **vrrp** *group-number* **ip** *virtual-IP-address* under the interface connecting to that subnet or VLAN:

```
Router(config-if) # vrrp 39 ip 10.0.0.1
```

Control the master and backup elections by configuring priority values from 1–255. If a master VRRP router is shut down, it advertises a priority of 0. This triggers the backup routers to hold an election without waiting for the master's hellos to time out.

```
Router(config-if)# vrrp 39 priority 175
```

VRRP uses the following timers:

- Advertisement, or hello, interval in seconds. Default is 1 second.
- Master down interval. Equals 3 x advertisement interval plus skew time. Similar to a hold or dead timer.
- Skew time. (256–priority) / 256. This is meant to ensure that the highest priority backup router becomes master because higher priority routers have shorter master down intervals.

To change the timers on the master, use the following command because it is the router that advertises the hellos:

```
Router(config-if)# vrrp 39 timers advertise 5
```

To change the timers on the backup routers, use the following command because they hear the hellos from the master:

```
Router(config-if)# vrrp 39 timers learn
```

VRRP cannot track interfaces but can track IP SLA object groups.

# GLBP

One issue with both HSRP and VRRP is that only the primary router is in use; the others must wait for the primary to fail before they are used. These two protocols use groups to get around that limitation. However, Gateway Load Balancing Protocol (GLBP) enables the simultaneous use of up to four gateways, thus maximizing bandwidth. With GLBP, there is still one virtual IP address. However, each participating router has a virtual MAC address, and different routers' virtual MAC addresses are sent in answer to ARPs for the virtual IP address. GLBP can also use groups up to a maximum of 1024 per physical interface. GLBP is supported only on Cisco 4500 and 6500 series switches.

The load sharing is done in one of three ways:

- **Weighted load balancing:** Traffic is balanced proportional to a configured weight.
- **Host-dependent load balancing:** A given host always uses the same router.
- **Round-robin load balancing:** Each router MAC is used to respond to ARP requests in turn.

GLBP routers elect an Active Virtual Gateway (AVG). It is the only router to respond to ARPs. It uses this capacity to balance the load among the GLBP routers. The highest priority router is the AVG; the highest configured IP address is used in case of a tie.

The actual router used by a host is its Active Virtual Forwarder (AVF). GLBP group members multicast hellos every 3 seconds to IP address 224.0.0.102, UDP port 3222. If one router goes down, another router answers for its MAC address.

Configure GLBP with the interface command *glbp group-number* **ip** *virtual-IP-address,* as shown:

```
Router(config-if)# glbp 39 ip 10.0.0.1
```

To ensure deterministic elections, each router can be configured with a priority. The default priority is 100:

```
Router(config-if)# glbp 39 priority 150
```

Hello and hold (or dead) timers can be configured for each interface with the command **glbp** *group-number* **timers [msec]** *hello-time* **[msec]** *hold-time*. Values are in seconds unless the **msec** keyword is used.

GLBP can also track interfaces just as with HSRP. If a tracked interface goes down, another router answers for the first router's MAC address.

## Planning Router Redundancy Implementation

Before configuring first-hop redundancy, determine which protocol is best in your network. If you have the same VLAN on multiple access switches, use HSRP or VRRP. If you use local VLANs, contained to a single switch, GLBP is an option.

Before configuring HSRP or VRRP on a multilayer switch, determine which switch is the root bridge for each VLAN. The root bridge should be the active HSRP/VRRP router. Determine priorities to be used, and whether you need tracking or timer adjustment.

After your implementation, verify and test. To view the switch's standby status, use the **show standby interface** *interface* command or **show standby brief**. To monitor standby activity, use the **debug standby** command.

# CHAPTER 7

# Campus Network Security

Attention has traditionally been paid to network perimeter security, such as firewall, and to mitigating Layer 3 attacks. However, networks must be protected against Layer 2 attacks, too. These are launched from devices inside the network by either a rogue device or a legitimate device that has been compromised. Rogue devices might be placed maliciously or might just be connected to an access switch by an employee wanting more switch port or wireless access. They include

- Wireless routers or hubs
- Access switches
- Hubs

A switch might become the Spanning Tree root bridge and disrupt user traffic. Use **root guard** and **bpdu guard** commands to prevent this. (Spanning Tree security is discussed later in this chapter.)

The following are four typical types of attacks against a switched network:

- **MAC address:** Based attacks-such as MAC address flooding
- **VLAN-based attacks:** VLAN hopping and attacks against devices on the same VLAN
- **Spoofing attacks:** DHCP spoofing, MAC spoofing, Address Resolution Protocol (ARP) spoofing, and Spanning Tree attacks
- **Attacks against the switch:** Cisco Discovery Protocol (CDP) manipulation, Telnet attacks, and Secure Shell (SSH) attacks

## MAC Address-Based Attacks

Common MAC address-based attacks rely on flooding the CAM table and can be mitigated by using port security and port-based authentication.

## MAC Address Flooding

In a MAC address flooding attack, the attacker fills the switch's Content Addressable Memory (CAM) table with invalid MAC addresses. After the table is full, all traffic with an address not in the table is flooded out all interfaces. This has two bad effects: more traffic on the LAN and more work for the switch. This can also cause the CAM tables of adjacent switches to overflow. Additionally, the intruder's traffic is also flooded, so they have access to more ports than they would normally have. After the attack stops, CAM entries age out and life returns to normal. However, meanwhile the attacker might have captured a significant amount of data.

Port security and port-based authentication can help mitigate MAC address attacks.

## Port Security

Port security limits the number of MAC addresses allowed per port and can also limit which MAC addresses are allowed. Allowed MAC addressed can be manually configured or the switch can sticky learn them. Table 8-1 lists port security commands; these are given at the interface.

**Table 7-1 Port Security Commands**

| Command | Description |
|---|---|
| **switchport port-security** | Enables port security on that interface. |
| **switchport port-security maximum** *value* | Specifies the max MAC addresses allowed on this port. Default is 1. |
| **switchport port-security violation {shutdown \| restrict \| protect}** | Configures the action to be taken when the maximum number is reached and a MAC address not associated with the port attempts to use the port, or when a station whose MAC address is associated with a different port attempt to access this port. Default is shutdown. |
| **switchport port-security mac-address** *mac-address* | Statically associates a specific MAC address with a port. |
| **switchport port-security mac-address sticky** | Enables the switch port to dynamically learn secure MAC addresses. MAC addresses learned through that port, up to the maximum number, if a maximum is configured, are treated as secure MAC addresses. |
| **show port security [interface** *interface* \| **address]** | Verifies port security actions. |

The following commands show how to verify the port security configuration:

```
Switch# show port-security interface fa 1/0/15
Port Security       : Enabled
Port Status        : Secure-Up
Violation Mode       : Shutdown
Aging Time         : 0 mins
Aging Type         : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 2
Total MAC Addresses    : 0
Configured MAC Addresses  : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan  : 0000.0000.0000:0
Security Violation Count  : 0
```

## Port-Based Authentication

802.1x authentication requires a computer (called a client) to be authenticated before it is allowed access to the LAN. This can be combined with port security to enable only authenticated clients with specified MAC addresses to access a port. When a computer connects to a switch port configured for 802.1x authentication, it follows these steps:

**Step 1.** The port is in the *unauthorized* state, allowing only 802.1x EAP over LAN (EAPOL) traffic.

**Step 2.** The client connects to the port. The switch either requests authentication or the client sends an EAPOL frame to begin authentication.

**Step 3.** The switch relays authentication information between the client and a RADIUS server that acts in proxy for the client.

**Step 4.** If authentication succeeds, the port transitions to the *authorized* state, and normal LAN traffic is allowed through it.

Table 7-2 shows commands to configure 802.1x authentication on a switch.

**Table 7-2 Configuring 802.1x Port Authentication**

| Command | Description |
|---|---|
| `(config)#`**`aaa new-model`** | Enables AAA on the switch |
| `(config)#`**`aaa authentication dot1x default group radius`** | Creates a AAA method list that says to use 802.1x authentication by default, using a RADIUS server (configured separately) |
| `(config)#`**`dot1x system-auth-control`** | Globally enables 802.1x authentication on the switch |
| `(config-if)#`**`dot1x port-control [auto \| force-authorized \| force-unauthorized]`** | Enables 802.1x authentication on an interface of the switch and sets default port state |
| **`show dot1x`** | Verifies 802.1x authentication |

# VLAN-Based Attacks

VLAN-based attacks include VLAN hopping, in which a station can access a VLAN other than its own. This can be done with switch spoofing or with 802.1Q double-tagging.

## Switch Spoofing

Switch spoofing involves a station configured to negotiate a trunk link between itself and the switch. By default, switches dynamically negotiate trunking status using Dynamic Trunking Protocol (DTP). If a computer can use DTP to establish a trunk link to the switch, it receives all traffic bound for every VLAN allowed on that trunk. By default, all VLANs are allowed on a trunk.

You can mitigate this by turning off DTP on all ports that should not become trunks, such as most access ports, using the interface command **switchport nonegotiate**. If the port should be an access port, configure it as such with the interface command **switchport mode access** and turn off CDP on that port. Additionally, shut down all unused ports and assign them to an unused VLAN. The commands to do this are

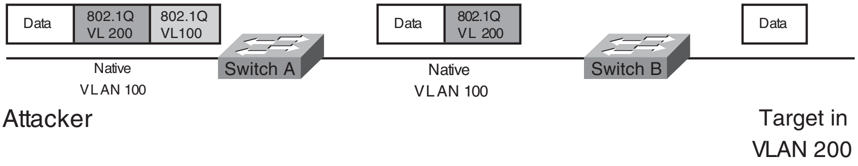
```
Switch(config)# interface interface
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan vlan
Switch(config-if)# shutdown
```

## 802.1Q Double-Tagging

A double-tagging attack is possible because 802.1Q trunking does not tag frames from the native VLAN. In this attack, the attacking computer negotiates a trunk port between itself and the switch and then generates frames with two 802.1Q tags. The first tag matches the native VLAN of the trunk port, and the second matches the VLAN of a host it wants to attack, as shown in Figure 7-1. The first switch in the path strips off the first 802.1Q tag and forwards it to adjacent switches. The next switch forwards the frame based on the VLAN listed in the second tag.

The double-tagging method of a VLAN hopping attack works even if trunk ports are set to off, if the trunk has the same VLAN as the attacker.

**Figure 7-1 VLAN Hopping by 802.1Q Double-Tagging**



Switch A removes the first tag for VLAN 100 because it matches the native VLAN for that link. It forwards the frame out all links with the same native VLAN, including its link to Switch B. Switch B sees the frame come in with an 802.1Q tag for VLAN 200, so it forwards it out the VLAN 200 link to the victim computer.

To mitigate this type of attack, use the same strategies used for switch spoofing. You can also use VLAN access control lists, called *VACLs*, or implement Private VLANs.

## VACLs

Cisco switches support of various kinds of ACLs:

- Traditional Router ACL (RACL)
- QoS ACL
- VACL

VLAN access control lists (VACL) are similar to route-maps because they are composed of statements that contain match and set conditions. In a VACL, the “set” conditions are called “actions.” Actions include **forward**, **drop**, and **redirect**. Like route-maps, VACL statements are numbered for ordering. After configuration, VACLs are applied to traffic to specified VLANs.

The following is a sample VACL that instructs the switch to drop traffic matching ACL 101 (not shown) and forward all other traffic:

```
Switch(config)# vlan access-map Drop101 10
Switch(config-access-map)# match ip address 101
Switch(config-access-map)# action drop
!
Switch(config-access-map)# vlan access-map Drop101 20
Switch(config-access-map)# action forward
!
Switch(config)# vlan filter Drop101 vlan-list 10
```

To view VACL settings, use the commands **show vlan access-map** *vacl_name* or **show vlan filter access-map** *vacl_name*.

## Private VLANs

Private VLANs (PVLAN) enable large companies or service providers to isolate users into separate multiaccess domains. Using a VLAN for each group is not scalable. For instance, the switch's maximum VLANs would limit the number of customers an ISP can have. Each VLAN requires a separate IP subnet, which could also be a limiting factor.

PVLANs divide a VLAN into secondary VLANs, letting you isolate a set of ports from other ports within the same VLAN. There are two types of secondary VLANs:

- **Community VLANs:** Ports can communicate with other ports in the same community VLAN.
- **Isolated VLANs:** Ports cannot communicate with each other.

Ports within a private VLAN can be one of three types:

- **Community:** Communicates with other community ports and with promiscuous ports
- **Isolated:** Communicates only with promiscuous ports
- **Promiscuous:** Communicates with all ports

Table 7-3 shows the commands to configure a primary private VLAN, secondary PVLANs, and their associated ports.

**Table 7-3 Configuring Private VLANs**

| Command | Description |
|---|---|
| **vlan** *vlan-id* | Enters VLAN configuration mode. |
| **private-vlan {community \| isolated \| primary}** | Configures the VLAN as a private VLAN and specifies the type. Repeat this command to configure all primary and secondary VLANs |
| **vlan** *primary-vlan-id* | Enters configuration mode for the primary VLAN. |
| **private-vlan association** *secondary_vlan_list* | Associates secondary VLANs with the primary one. Separate the secondary VLAN numbers with a comma, no spaces. |
| **switchport mode private-vlan {host \|promiscuous}** | Configures a port as either a host port (for community or isolated) or a promiscuous port. |
| **switchport private-vlan host-association** *primary_vlan_ID secondary_vlan_ID* | Associates a host port with its primary and secondary PVLANs |
| **private-vlan mapping** *primary_vlan_ID secondary_vlan_list* | Associates a promiscuous port with its primary and secondary PVLANs. |
| **show interfaces** *interface* **switchport** | Verifies the VLAN configuration. |
| **show interfaces private-vlan mapping** | Verify the private VLAN configuration. |

## Protected Ports

On some lower-end switches, protected ports can provide a simple version of private VLANs. Traffic from a protected port can access only an unprotected port. Traffic between protected ports is blocked. Configure port protection at the interface:

```
Switch(config-if)# port protected
```

# Spoof Attacks

Spoof attacks include DHCP spoofing, MAC address spoofing, and ARP spoofing.

## DHCP Spoofing

A DHCP spoofing attacker listens for DHCP requests and answers them, giving its IP address as the client default gateway. The attacker then becomes a "man-in-the-middle" as all off-net traffic flows through it.

DHCP snooping can prevent DHCP spoofing attacks. When DHCP snooping is enabled, only ports that uplink to an authorized DHCP server are trusted and allowed to send all types of DHCP messages. All other ports on the switch are untrusted and can send only DHCP requests. If a DHCP response (or "offer") is seen on an untrusted port, the port is shut down. The switch can also be configured to send information, such as port ID, using DHCP option 82.

**Note**

DHCP snooping configuration is user impacting because the switch drops all DHCP requests until the ports are configured. You should do this during off hours or during a maintenance window.

Configure DHCP snooping with the following commands, either globally or for a particular VLAN. Configure only individual ports that uplink to DHCP servers as trusted ports.

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping information option
Switch(config)# ip dhcp snooping vlan number number
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip dhcp snooping limit pkts-per-second
Switch# show ip dhcp snooping
```

## IP Source Guard

To extend the protection further, IP Source Guard tracks the IP addresses of the host connected to each port and prevents traffic sourced from another IP address from entering that port. The tracking can be done based on just an IP address or on both IP and MAC addresses.

Enable IP Source Guard for both IP and MAC addresses on host access interfaces with the command **ip verify source port-security**.

## ARP Spoofing

In an ARP spoofing attack, the attacker sends out gratuitous (unsolicited) ARP messages giving the IP address of the local default gateway, with its own MAC address as the Layer 2 address. Local devices overwrite their existing correct ARP information with the incorrect one, and, thus, they forward off-net traffic to the attacker (it becomes a "man-in-the-middle"). If the attacker then forwards it on to the legitimate router, this type of attack might go undetected by the users.

Dynamic ARP Inspection (DAI) can work with DHCP spoofing to stop ARP spoofing. DAI defines trusted and untrusted interfaces. It intercepts ARP messages on untrusted ports and checks them against the IP address/MAC address bindings in the DHCP snooping database. They must match for the switch to forward the traffic. Access ports should be configured as untrusted, and ports that connect to other switches or to a router should be trusted.

Enable DAI on a VLAN, or multiple VLANs, and configure trusted interfaces. You can optionally configure a rate limit or configure which addresses DAI matches against. (The default is IP and MAC address.) The basic commands are

```
Switch(config)# ip arp inspection vlan vlan_id
Switch(config-if)# ip arp inspection trust
```

## Securing Your Switch

Here are some basic security suggestions for network devices:

- Use passwords that are not susceptible to a dictionary attack. Add numbers or substitute numbers and symbols for letters.
- Limit Telnet access using access lists.
- Use SSH instead of Telnet.
- Physically secure access to the device.
- Use banners that warn against unauthorized access.
- Remove unused services, such as finger, the TCP and UDP small servers, service config, and HTTP server.
- Set up and monitor Syslog.
- Disable automatic trunking on all nontrunk ports.
- Disable CDP on ports where it is not needed.

# CHAPTER 8

# Voice and Video in a Campus Network

Voice over IP (VoIP) has become common in the business world, and now Video over IP is becoming more integrated into networks. Neither should be added to a network without advance planning to ensure good voice and video quality. Some benefits of converging voice, video, and data networks include

- **Consolidating network expenses:** Only one wire and one switch port are needed per user. One network to provision and manage.
- **More efficient bandwidth use:** Bandwidth can be used for data when there is not an active voice/video session.
- **Lower telephony costs:** Internal calls use the data network, rather than the PSTN.
- **Innovative services:** Ability to unify a company's various methods of communication.
- **For service providers, the ability to sell new services:** Can lead to increased revenue, flexibility in pricing, and access to new communication devices.

Voice, video, and data have different network requirements. Voice requirements include low bandwidth, little delay, small amounts of jitter (variable delay), small amounts of packet loss, a highly available network, and PoE. Security requirements are about average, but management is highly important.

Video requirements depend on whether it is a one-way stream or an interactive video session. One-way streams use a fairly steady amount of bandwidth but in interactive sessions the bandwidth varies widely. Typical requirements include high bandwidth, little delay, small amounts of jitter, and little packet loss. High availability is not as important, and PoE is not needed. Security and management needs are medium.

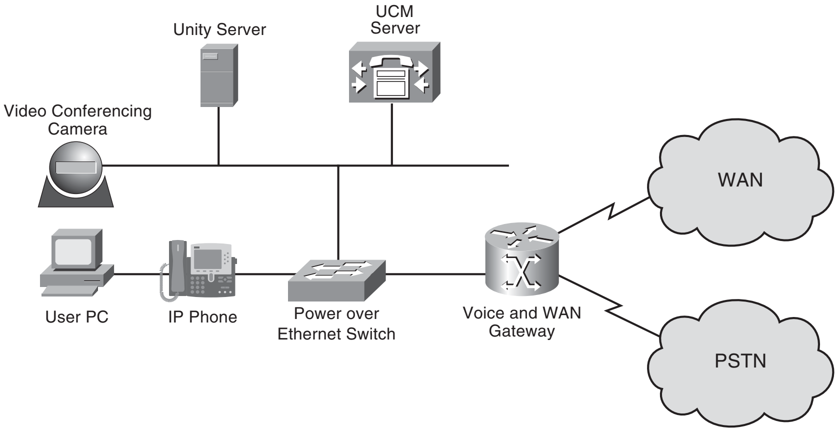
Data requirements typically include high bandwidth, but delay and jitter are not crucial. A highly available network is needed, but PoE is not. Data security should be high, with medium management levels.

## VoIP in a Campus Network

Figure 8-1 shows some components of a VoIP system, which can include the following:

- **IP phones:** Provide voice and applications to the user
- **Cisco Unified Communications Manager (UCM):** Serves as an IP PBX. Registers phones, controls calls
- **Voice gateways:** Translates between PSTN and IP calls and provides backup to the Cisco UCM (IP PBX, or Call Agent)
- **Gatekeepers:** An optional component that can do call admission control, allocate bandwidth for calls, and resolve phone numbers into IP addresses
- **Video conferencing unit:** Allows voice and video in the same phone call
- **Multipoint control unit:** Allows multiple participants to join an audio or video conference call
- **Application server:** Provides services such as Unity voice mail and Presence

**Figure 8-1 Some Components of a VoIP System**



VoIP traffic consists of two types: voice bearer and call control signaling. Voice bearer traffic is carried over the UDP-based Real Time Protocol (RTP). Call control uses one of several different protocols to communicate between the phone and UCM and between the UCM and the voice gateways.

## Preparing the Network for VoIP

When adding voice or video to an existing network, you should examine several things in advance to provide the high level of availability users expect in their phone system:

- **What features are needed?:** Power for IP phones, security for voice calls, and Quality of Service (QoS) to control bandwidth, delay, jitter, and packet loss.
- **The physical plant:** Cabling at least CAT-5.
- **Electrical power for the IP phones:** Use either PoE from Catalyst switch or power inline module, or a power brick.
- **Bandwidth:** Commit no more than 75 percent of bandwidth. Consider all types of traffic: voice, video, and data. Have more than enough bandwidth if possible. Include both voice and call-control traffic in your planning.
- **Network management:** Monitor and proactively manage the network so that it is always available. Need voice VLANs on the switches and DHCP for the phones.
- **High availability-provide redundant hardware and links:** Need uninterruptible power supply (UPS) with auto-restart, monitoring, and four-hour response contract. Might need generator backup. Maintain correct operating temperatures.

### Network and Bandwidth Considerations

The network requirements for VoIP include:

- Maximum delay of 150–200 ms (one-way)
- No more than 1 percent packet loss
- Maximum average jitter of 30 ms
- Bandwidth of 21–106 kbps per call, plus approximately 150 bps per phone for control traffic

A formula to use when calculating bandwidth needed for voice calls is as follows:

(Packet payload + all headers) × Packet rate per second

## Voice VLANs

Cisco switches can be configured to dynamically place IP telephones into a Voice, or auxiliary, VLAN separate from the data VLANs. They can do this even when the phone and PC are physically connected to the same switch port. Voice VLANs enable phones to be dynamically placed in a separate IP subnet from hosts, to have QoS (using 802.1Q/p headers) and security policies applied, and make troubleshooting easier.

Cisco IP phones have a small internal switch that places an 802.1q tag on voice traffic and marks the Class of Service (CoS) bits in the tag. Data traffic is sent untagged over the native VLAN. The switch port does not actually become a trunk and still can be configured as an access port. It is considered a multi-VLAN access port.

### Power over Ethernet (PoE) Switches

IP phones can receive power from PoE switches, eliminating the need for an electrical plug.

Two power standards are the Cisco Inline PoE and the IEEE's 802.3af standard. Both have a method for sensing that a powered device is connected to the port. 802.3af specifies a method for determining the amount of power needed by the device. Cisco devices, when connected to Cisco switches, can additionally use CDP to send that information. Most IP phones require no more than 15 W of power specified in 802.3af, but some new phones, access points, and surveillance cameras require more. The 802.3at standard will specify up to 30 W of power. Some Cisco switches can currently supply up to 20 W.

Because a switch assumes 15.4 W of power until the connected device tells it the amount needed (via CDP), calculate your power budget based on 15.4 W for all devices because if the switch reboots, all ports will ask for 15.4 W until they get the correct information. Non-CDP devices always get allocated 15.4 W.

Cisco PoE switches automatically detect and provide power. To disable this function, or to reenable it, use the interface command **power inline {never | auto}**. To view interfaces and the power allotted to each, use **show power inline** [*interface*].

## QoS for VoIP

QoS gives special treatment to certain traffic at the expense of others. Using QoS in the network has several advantages:

- Prioritizes access to resources so that critical traffic can be served
- Allows good management of network resources
- Allows service to be tailored to network needs
- Allows mission-critical applications to share the network with other data

People sometimes think that there is no need for QoS strategies in a LAN. However, switch ports can experience congestion because of port speed mismatches, many people trying to access the switch backbone, and many people trying to send traffic to the same switch port (such as a server port). Voice and video traffic contends with, and can be affected by, data traffic within both the WAN and the LAN.

### QoS Actions

Three QoS strategies are commonly implemented on interfaces in which traffic enters the switch:

- **Classification:** Distinguishing one type of traffic from another. After traffic is classified, other actions can be performed on it. Some classification methods include access lists, ingress interface, and NBAR.
- **Marking:** At Layer 2, placing an 802.1p CoS value within the 802.1Q frame tag. At Layer 3, setting IP Precedence or Differentiated Services Code Point (DSCP) values in the packet's IP header.
- **Policing:** Determining whether a specific type of traffic is within preset bandwidth levels. If so, it is usually allowed and might be marked. If not, the traffic is typically marked or dropped. CAR and class-based policing are examples of policing techniques.

Other QoS techniques are typically used on outbound interfaces:

- **Traffic shaping and conditioning:** Attempts to send traffic out in a steady stream at a specified rate. Buffers traffic that goes above that rate and sends it when there is less traffic on the line.
- **Queuing:** After traffic is classified and marked, one way it can be given special treatment is to be put into different queues on the interface to be sent out at different rates and times. Some examples include

priority queuing, weighted fair queuing, and custom queuing. The default queuing method for a switch port is FIFO.

- **Dropping:** Normall interface queues accept packets until they are full and then drop everything after that. You can implement prioritized dropping so that less important packets are dropped before more important ones, such as with Weighted Random Early Detection (WRED).

## DSCP Values

Differentiated services provide levels of service based on the value of certain bits in the IP header or the 802.1Q tag. Each hop along the way must be configured to treat the marked traffic the way you want; this is called per-hop behavior (PHB).

In the Layer 2 802.1q tag, you use the three 802.1p bits to set the CoS value. Voice is usually set to 5 and video to 4.

In the Layer 3 IP header, you use the 8-bit ToS field. You can set either IP Precedence using the top 3 bits or Differentiated Services Code Points (DSCP) using the top 6 bits of the field. The bottom 2 bits are set aside for congestion notification. The default DSCP value is 0, which corresponds to best-effort delivery.

The six DSCP bits can be broken down into two sections: The first 3 bits define the DiffServ Assured Forwarding (AF) class, and the next 2 bits define the drop probability within that class. The sixth bit is 0 and unused. AF classes 1–4 are defined, and within each class, 1 is low drop probability, 2 is medium, and 3 is high (meaning that traffic is more likely to get dropped if there is congestion). These are shown in Table 8-1.

**Table 8-1 DSCP Assured Forwarding Values**

| | Low Drop | Medium Drop | High Drop |
|---|---|---|---|
| Class 1 | AF11 | AF12 | AF13 |
| Class 2 | AF21 | AF22 | AF23 |
| Class 3 | AF31 | AF32 | AF33 |
| Class 4 | AF41 | AF42 | AF43 |

Voice bearer traffic uses an Expedited Forwarding value of DSCP 46 to give it higher priority within the network.

### Trust Boundaries

When IP traffic comes in already marked, the switch has some options about how to handle it. It can

- Trust the DSCP value in the incoming packet, if present.
- Trust the IP Precedence value in the incoming packet, if present.
- Trust the CoS value in the incoming frame, if present.
- Classify the traffic based on an IP access control list or a MAC address access control list.

Mark traffic for QoS as close to the source as possible. If the source is an IP telephone, it can mark its own traffic. If not, the building access module switch can do the marking. If those are not under your control, you might need to mark at the distribution layer. Classifying and marking slows traffic flow, so do not do it at the core. All devices along the path should then be configured to trust the marking and provide a level of service based on it. The place where trusted marking is done is called the trust boundary.

## Configuring VoIP Support on a Switch

Before implementing VoIP, plan the following:

1. **PoE:** Ensure that enough power is available for all phones, with a UPS backup.
2. **Voice VLAN:** Determine the number of VLANs needed and the associated IP subnets. Add DHCP scopes for the phones, and add the phone networks to the routing protocol.
3. **QoS:** Decide which marking and queues will be used. Implement AutoQoS and then tune as needed.
4. **Fast Convergency:** To enhance high availability, tune the routing and HSRP/VRRP/GLBP timers.
5. **Test Plan:** Test the voice implementation thoroughly before converting users to it. Check that both the phone and PC get the correct IP addresses, that the phone registers with the UCM, and that calls to and from the phone succeed.

### Manual Configuration

To associate a voice VLAN with a switch port, use the following:

```
Switch(config-if)# switchport voice vlan vlan-ID
```

To configure an IOS switch to trust the markings on traffic entering an interface, use the following:

```
Switch(config-if)# mls qos trust {dscp | cos}
```

To configure the switch to trust the traffic markings only if a Cisco phone is connected, use the following:

```
Switch(config-if)# mls qos trust device cisco-phone
```

To set a COS value for frames coming from a PC attached to the phone, use the following:

```
Switch(config-if)# switchport priority extend cos cos-value
```

To verify the interface parameters, use the following:

```
Switch(config-if)# show interfaces interface switchport
```

To verify the QoS parameters on an interface, use the following:

```
Switch(config-if)# show mls qos interface interface
```

## Using AutoQoS

When AutoQoS is enabled, the switch configures its interfaces based on a best-practices template. AutoQoS has the following benefits:

- Automatic discovery and classification of network applications.
- Creates QoS policies for those applications.
- Configures the switch to support Cisco IP phones and network applications. Manual configuration can also be done afterward.
- Sets up SNMP traps for network reporting.
- Configures consistently across your network when used on all routers and switches.

CDP must be enabled for AutoQoS to function properly with Cisco IP phones.

AutoQoS commands for switches running Native IOS are shown in Table 8-3.

**Table 8-3 AutoQoS Commands for IOS**

| Command | Description |
|---|---|
| `(config-if)#`**`auto qos voip trust`** | Configures the port to trust the COS on all traffic entering the port. |
| `(config-if)#`**`auto qos voip cisco-phone`** | Configures the port to trust traffic markings only if a Cisco phone is connected to the port. Requires that CDP be enabled. |
| `#`**`show auto qos [interface`** *`interface`*`]` | Shows the AutoQoS configuration. Does not show any manual QoS configuration: Use **show run** to see that. |

# Video over IP

Video traffic roughly falls into one of three categories: many-to-many, many-to-few, and few-to-many.

Many-to-many includes interactive video, such as Telepresence, Webex, desktop video conferencing, and other peer-to-peer video and collaboration applications. The data flow is client-to-client, or MCU-to-client. Bandwidth needs for high definition video vary during the session but are high-up to 12 Mb/s per location, with compression.

Many-to-few sessions represent IP surveillance cameras. The video flow is from the camera source to a storage location, from storage to a client, or from the source to a client. These typically require up to 4 Mb/s of bandwidth per camera.

Few-to-many describes the typical streaming video, either from an internal company source or an Internet source. It also applies to digital signage media. This is the most predictable of all video streams and users typically tolerate less-than-perfect quality. Traffic flows are from storage-to-client or from server-to-client.

## QoS Requirements for Video

Video traffic should be compressed because of its high bandwidth needs, but this causes a lot of variation in network traffic. A picture that does not change much can compress well, resulting in fairly low bandwidth use. But when there are a lot of changes in the picture, such as when someone moves or shares a new document, compression does not work as well, which results in high bandwidth use. In contrast, voice traffic is fairly steady.

Video should be placed in its own queue and might be prioritized depending on company requirements. Consider placing interactive and streaming video into different queues. Aim to provide no more than 200 ms of latency for most video applications.

Make sure that there is sufficient bandwidth in the network before adding video applications.

# CHAPTER 9

# Wireless LANs in a Campus Network

Wireless LANs (WLAN) transmit and receive data using radio or infrared signals, sent through an access point (AP), and are not usually required to have radio frequency (RF) licenses. WLANs are local to a building or a campus and are an extension of the wired network.

## Cisco Unified Wireless Network

The Cisco Unified Wireless Network concept has five components that work together to create a complete network, from client devices to network infrastructure, to network applications. Cisco has equipment appropriate to each component. Table 9-1 lists components and equipment.

**Table 9-1 Cisco Unified Wireless Network Components**

| Component | Description and Device |
|---|---|
| Client devices | Cisco client and Cisco compatible third-party vendor clients |
| Mobility platform | APs and bridges using LWAPP |
| Network unification | Leverages existing wired network. Includes WLAN controllers and switch and router modules |
| Network management | Visualize and secure the WLAN. WCS for location tracking, RF management, wireless IPS, and WLC management |
| Mobility services | Applications such as wireless IP phones, location appliances, and RF firewalls |

Cisco wireless IP phones have the same features as Cisco wired IP phones and can use LEAP for authentication.

The Cisco Compatible Extensions Program tests other vendors' devices for compatibility with Cisco wireless products. Using products certified by this program ensures full functionality of Cisco enhancements and proprietary extensions.

# Characteristics of Wireless LANs

WLANs function similarly to Ethernet LANs with the access point providing connectivity to the rest of the network as would a switch. The physical layer is radio waves, rather than wires. IEEE 802.11standard defines the physical and data link specifications, including the use of MAC addresses. The same protocols (such as IP) and applications (such as IPsec) can run over both wired and wireless LANs.

The following lists some characteristics of wireless LANs and the data transmitted over wireless networks.

- WLANs use Carrier Sense Multi-Access/Collision Avoidance (CSMA/CA).
- Wireless data is half-duplex. CSMA/CA uses Request to Send (RTS) and Clear to Send (CTS) messages to avoid collisions.
- Radio waves have unique potential issues. They are susceptible to interference, multipath distortion, and noise. Their coverage area can be blocked by building features, such as elevators. The signal might reach outside the building and lead to privacy issues.
- WLAN hosts have no physical network connection. They are often mobile and often battery-powered. The wireless network design must accommodate this.
- WLANs must adhere to each country’s RF standards.

## Service Set Identifiers (SSID)

An SSID maps to a VLAN and can be used to segment users into groups requiring different security or QoS treatment. SSIDs can be broadcast by the access point or statically configured on the client, but the client must have the same SSID as the AP to register with it. SSID names are case sensitive. When multiple SSIDs/VLANs are used on an AP, the wired connection back to the network must be a trunk to carry all the VLANs.

## WLAN Topologies

The use of wireless products falls into three categories:

- Client access, which allows mobile users to access the wired LAN resources

- Wireless connections between buildings
- Wireless mesh

Wireless connections can be made in *ad-hoc* mode or *infrastructure* mode. Ad-hoc mode (or Independent Basic Service Set [IBSS]) is simply a group of computers talking wirelessly to each other with no access point (AP). It is limited in range and functionality. Infrastructure mode's BSS uses one AP to connect clients. The range of the AP's signal, called its microcell, must encompass all clients. The Extended Service Set (ESS) uses multiple APs with overlapping microcells to cover all clients. Microcells should overlap by 10–15 percent for data and 15–20 percent for voice traffic. Each AP should use a different channel. "Pico" cells, with even smaller coverage areas, can also be used.

Workgroup bridges connect to devices without a wireless network interface card (NIC) to allow their access to the wireless network.

Wireless mesh networks can span large distances because only the edge APs connect to the wired network. The intermediate APs connect wirelessly to multiple other APs and act as repeaters for them. Each AP has multiple paths through the wireless network. The Adaptive Wireless Path (AWP) protocol runs between APs to determine the best path to the wired network. APs choose backup paths if the best path fails.

## Client Connectivity

Clients associate with an access point as follows:

Access points send out beacons announcing information such as SSID, unless configured not to.

**Step 1.** The client sends a probe request and listens for beacons and probe responses.

**Step 2.** The AP sends a probe response.

**Step 3.** The client initiates an association to the AP. 802.1x authentication, and any other security information is sent to the AP.

**Step 4.** The AP accepts the association. SSID and MAC address information is exchanged.

**Step 5.** The AP adds the client's MAC address to its association table.

Clients can roam between APs, but the APs must be configured with the same SSIDs/VLANs and security settings. Layer 2 roaming is done between APs on the same subnet and managed by the switches using a multicast protocol: Inter-Access Point Protocol (IAPP). Layer 3 roaming is done between APs on different subnets and is managed by the wireless LAN controllers. The switch connected to the AP updates its MAC address table when a client roams.

Short roaming times are needed for VoIP to reduce delay. A client will attempt to roam (or associate with another AP) when

- It misses too many beacons from the AP.
- The data rate is reduced.
- The maximum data retry count is exceeded.
- It is configured to search for another AP at regular intervals.

# Cisco Wireless Network Components

Cisco supports two types of wireless solutions: one using autonomous access points, and one using lightweight (or “dumb”) access points in combination with WLAN controllers. The wired network infrastructure is the same for both types: switches and routers.

Access points can receive their power from Power over Ethernet (PoE) switches, routers with PoE switch modules, or midspan power injectors, thus alleviating the need for electrical outlets near them. APs require up to 15 W of power, so plan your power budget accordingly.

## Autonomous (Stand-alone) APs

Autonomous APs run Cisco IOS, are programmed individually, and act independently. They can be centrally managed with the CiscoWorks Wireless LAN Solution Engine (WLSE), can use Cisco Secure Access Control Server (ACS) for RADIUS and TACAS+ authentication, and Wireless Domain Services (WDS) for RF management. Redundancy consists of multiple APs.

### Network Design for Autonomous APs

When using stand-alone APs, the traffic flow is from client to AP to connected switch, and from there into the rest of the network. Plan the SSIDs and VLANs that will be on each AP, keeping in mind any roaming that users might do. Autonomous APs support Layer 2 roaming only, so SSIDs and VLAN must be statically configured on every AP in which a user might roam. Make sure to include a management VLAN on the AP.

Ensure that the AP has a power source, either a PoE switch or a power injector. Configure the switch interface connected to the AP as a trunk if the AP has multiple VLANs.

## Lightweight Access Points

Lightweight APs divide the 802.11 processing between the AP and a Cisco Wireless LAN Controller (WLC). This is sometimes called "split MAC," because they split the functions of the MAC layer, Layer 2. Their management components also include the Wireless Control System (WCS) and a location-tracking appliance. Redundancy consists of multiple WLCs. The AP handles real-time processes, and the WLC handles processes such as:

- Authentication
- Client association/mobility management
- Security management
- QoS policies
- VLAN tagging
- Forwarding of user traffic

The Lightweight Access Point Protocol (LWAPP) supports the split MAC function in traffic between a lightweight AP and its controller. LWAPP uses AES-encrypted control messages and encapsulates, but does not encrypt, data traffic.

Controllers and APs can also use a new IETF-standard protocol to communicate with each other: the Control and Provisioning of Wireless Access Points (CAPWAP) protocol. CAPWAP operates very much like LWAPP.

Both LWAPP and CAPWAP operate over UDP. The controller does not have to be in the same broadcast domain and IP subnet, just IP reachable. Lightweight APs follows this process to discover their controller:

**Step 1.** The AP requests a DHCP address. The DHCP response includes the management IP address of one or more WLCs.

**Step 2.** The AP sends an LWAPP or CAPWAP Discovery Request message to each WLC.

**Step 3.** The WLCs respond with an LWAPP or CAPWAP Discovery Response that includes the number of APs currently associated to it.

**Step 4.** The AP sends a Join Request to the WLC with the fewest APs associated to it.

**Step 5.** The WLC responds with a Join Response message; the AP and the controller mutually authenticate each other and derive encryption keys to be used with future control messages. The WLC then configures the AP with settings, such as SSIDs, channels, security settings, and 802.11 parameters.

### Network Design for Lightweight APs

When using lightweight APs the traffic flow is from the AP, through the network, to the controller, and from there out to the rest of the network. User traffic is tunneled between the AP and the controller. Make sure that the AP and controller have Layer 3 connectivity.

The controller placement can be distributed, with a controller in each building or at each site, if no roaming between buildings is needed. A centralized design, with redundant controllers placed together, such as in a data center, simplifies management and increases user mobility.

SSIDs and VLANs must be planned, just as with an autonomous AP. But the configuration is done on the controller. Clients are placed into VLANs based either on the controller they connect to or an authentication process. The management VLAN is mapped to the controller. Controllers support both Layer 2 and Layer 3 roaming.

The link between a lightweight AP and the switch is an access port, assigned to a VLAN. The link between the controller and its connected switch is a trunk link. Controllers with several switch links can create an Etherchannel to the switch to increase bandwidth. Link aggregation is recommended for the 4400 series and is required on the WiSM and the 3750G integrated controllers.

Ensure that the AP has a power source, either a PoE switch or a power injector.

## Wireless LAN Controllers

Cisco WLAN controllers can be either an appliance, a module, or integrated into a 3750G switch. In the appliance line, the 5500 series is meant for large deployments and, as of this writing, supports up to 250 APs. The 4400 series is for medium-sized deployments and supports from 12 APs to 100 APs. The 2100 series is for small deployments and supports from 6 APs to 25 APs.

The WLAN controller integrated into a Cisco 3750G switch can support up to 25 APs per switch,or 100 per switch stack. The Wireless Services Module (WiSM) can be installed into Cisco 6500 and 7600 series switches for large deployments that need support for up to 300 APs. Cisco ISR routers have a WLAN controller module that can support up to 25 APs for small deployments.

### Hybrid Remote Edge Access Point (H-REAP)

Wireless controllers need not be in the same physical location as their associated APs. However, having an AP and its controller separated by a WAN link can lead to some inefficiencies and problems. Two clients in the remote location that need to connect would have their traffic tunneled over the WAN to the controller and back again. Additionally, the AP would lose functionality if the WAN were down.

H-REAP addresses these problems:

- **Connected mode:** When the controller is reachable, the AP transmits user authentication to the controller. It sends traffic in specified WLANs (usually local traffic) to its local switch, however, rather than tunneling it back to the controller. The connection from the AP to switch needs to be a trunk link if the AP handles multiple VLANs. Traffic bound to remote networks is still tunneled over the WAN to the controller.
- **Disconnected mode:** When the controller is not reachable, the AP authenticates clients itself. It still sends client to its connected switch, but of course remote locations will not be reachable if the WAN is down.

H-REAP is configured at the controller for any APs that operate in this mode.

## Integrating Wireless into the LAN

This section covers configuring your switches for wireless APs and controllers, and planning your installation.

## Switch Configuration

When the switch port connects to a stand-alone AP, configure it as an access port if the AP has only one VLAN and a trunk port if it has multiple VLANs. Trust CoS if the link is a trunk. Set the trunk native VLAN to the AP's management VLAN. Prioritize voice if you use wireless phones.

When the switch port connects to a controller-based AP, the port should be an access port. The port should be placed into the management VLAN because it is used for traffic between the AP and the controller. Trust DSCP on the port. If using wireless IPT, also set up QoS to prioritize voice.

The switch port connecting to a WLAN controller should be configured as a trunk link. Limit the trunk to wireless and management VLANs. Trust CoS and prioritize voice if you use wireless IP phones.

Links to a 4400 series controller might be aggregated into a Layer 2 Etherchannel. The 4400 cannot negotiate aggregation, so it is important to set the channel-group mode to "On". Otherwise, the configuration is the same as with any other Etherchannel. Configure the channel as a trunk, allow only the management and wireless VLANs, and trust CoS.

The WiSM requires a separate VLAN for its management. This VLAN should be assigned only to the module's service port and should not be used outside of the switch. Assign the VLAN to the service port with the global command **wism service-vlan** *vlan*. Assign an IP address to the VLAN interface; this IP address is used to communicate with the WiSM. The WiSM contains eight logical ports that connect to the switch fabric in two Etherchannel bundles. It also contains two separate controllers. Bundle configuration is done at each controller, using the **wism module** *slot#* **controller** *controller#* set of global commands.

## Planning for a Wireless Implementation

In planning a wireless implementation, first gather requirements. Some questions to ask include the following:

- How many APs and where will they be installed?
- Stand-alone or controller-based?
- If controller-based, where will the controllers be located?
- Is PoE available?
- What VLANs and SSIDs will be used?
- What are the bandwidth requirements?

- What are the QoS requirements?
- Do you need security such as ACLs or Radius server?
- Do you need UPS for controllers?

When the requirements are gathered, create an implementation plan with details such as:

- Total needs, from the requirements that were previously gathered
- Any changes needed to the network design
- Any additional equipment needed
- Implementation steps
- Testing plan

The test plan might include checking that the AP and its clients get a DHCP address, that the AP is reachable from a management station, that clients can reach the network and Internet, and that the controller can reach the Radius server if used. To troubleshoot problems with wireless connectivity, review the steps for an AP to register with a WLC and a client with an AP.

# CHAPTER 1

# Maintenance

Maintenance might seem separate from the process of troubleshooting but imagine it as the other side of the same coin. Any device that is well maintained will be more reliable, suffers fewer problems, and will be easier and quicker to repair. Network owners, such as businesses and governments, want computer systems that are consistently available. Good troubleshooting technique minimizes the length of time of an outage, but good maintenance technique reduces outages.

You must select the appropriate tools and techniques for the network you maintain, based on law, company policy, and your experience. You need to understand, whichever elements you incorporate into your strategy, that a structured approach to maintenance is a key part of reducing unplanned outages.

**NOTE**

TSHOOT doesn't assume a specific approach to maintenance. Organizations might produce documentation and monitor their networks in unique ways. TSHOOT focuses on understanding the general practices that are used to successfully maintain a network.

## Methodology

Network maintenance involves many different kinds of tasks, such as

- Installing new equipment
- Adjusting settings to support new service
- Securing the network
- Restoring service
- Backing up configs
- Planning new or upgraded service

- Building redundancy and disaster recovery
- Documentation
- Responding to user complaints

Many activities are reactive, and it is easy for interrupt-driven issues to monopolize your time. Defining a preventative maintenance schedule can help you avoid “firefighting.” Taking a more structured approach—as opposed for waiting for the phone to ring—can also help you recognize problems earlier and respond to them more efficiently. A broader perspective toward the network also provides an opportunity to align costs with the organization’s goals and budget effectively.

Several generic maintenance frameworks are available. Some organizations embrace a specific methodology, but many organizations pick, choose, and customize pieces that fit their environment. The important point is to have a documented approach to maintenance. If your organization doesn’t have a documented strategy, you might want to research some of these models.

- IT Infrastructure Library (ITIL)
- FCAPS
- Telecommunications Management Network (TMN)
- Cisco Lifecycle Services/PPDIOO
- Microsoft Operations Framework

After you choose a specific model, map the model onto processes you can use to maintain the network and then select the tools that you use.

## Common Tasks

Although organizations that own networks have different expectations, the management of every network still includes some basic components. Planning and accomplishing these tasks repetitively and competently is a key to successful network management.

Some common tasks include

- Adds, moves, and changes
- Compiling documentation
- Preparing for disaster
- Capacity planning/utilization monitoring

- Troubleshooting
- Proactive scheduled maintenance
- Rollback plans for each change
- Lab testing in a controlled environment before each change is put into production to minimize risk

Preventative maintenance is the process of anticipating potential sources of failure and dealing with the problem before it occurs. It is probably not possible to anticipate every source of failure, but careful thought might help you identify candidates. One technique to identify issues is to look at prior records of trouble, such as trouble tickets, ISP records, network monitoring systems, or purchase records. Use this information to categorize and rank the experience of your network.

Organizations are typically willing to accept small periods of scheduled downtime to offset the probability of long periods of unscheduled downtime. Using the data collected from your experience, consider the steps that can be taken during this window of time. Operating systems can be patched or upgraded to more stable and secure versions. Redundancy can be tested to ensure smooth failover. Additionally, normal business changes (such as new circuits) can be accomplished during this period to minimize disruption.

Most large organizations use a system of change controls to enforce a thought-out approach to configuration changes. Change control involves producing a document that describes the change to be made, who will make it, when the change will be made, and who will be affected. A well-written change control document will also have some notes about how the new configuration can be "backed out" if something goes wrong. This change control is then approved by management.

Change control systems help the business balance the need to update network components and configurations against the risk of changes. Change control systems also protect the network administrator—if each change is well thought out and thoroughly communicated, the business has the opportunity to accept the risks inherent in change.

Documentation reduces troubleshooting time and smoothes project communication as networks are changed and upgraded. Although time consuming, it is impossible to over emphasize the importance of accurate and up-to-date documentation. Well-maintained documentation includes details such as

- Configuration templates or standards
- Configuration history

- Equipment inventory (including serial number and support contract information)
- Circuit inventory (including circuit ID and service provider contact)
- IP address assignment
- Network drawings
- Communication plan
- Out-of-band communication details
- Expected traffic patterns

Templates can be a fill-in-the-blanks version of a complete configuration or can be *snippets* that show how your organization handles specific issues, such as IPsec tunnels. Either way, templates provide an opportunity for consistency and enable technicians to more quickly move from interpreting to troubleshooting. Consider, for instance, access-lists and how easily they might be confused. Access-list 100 might be typically related to permitting SNMP to certain destinations but on some devices is used to filtering traffic on the public interface. Understanding the ramifications of confusion in this example, it is easy to see the benefit of standardizing things such as labels. (And in this case, it is probably best to use named access-lists, not numbered.)

The documentation for the communication plan should include contact information for internal IT and management contacts, and vendor and service provider information. The plan should also specify who should be contacted, in what circumstances, and how often. For instance, should a technician update the business contract or the Network Operations Center? Is there a proscribed after-action review?

Often the individual documentation elements are combined, such as IP addresses and circuit IDs on the Network diagram, or simplified, such as a TFTP server directory to keep configuration history.

Documentation should also include a disaster recovery plan. Disasters come in many sizes, so it pays to consider several cases. If the problem is related to a single piece of equipment, consider Cisco SmartNet maintenance as a way to guarantee backup hardware is onsite quickly. Even in the case where a spare is procured, you need a backup of the configuration and IOS. If getting a spare involves a service contract, you probably also need the serial number. Someone onsite needs a console cable and a laptop with a serial port. Larger disasters, such as a fire, might require replacing equipment from memory. It’s a good idea to also have a record of the installed cards and licenses. Finally, consider the staff at the site. Is there someone there who can be talked through copying a config or do you need a technician to go to the site?

A final common piece to managing the network is to have some form of network monitoring. Network monitors take many forms, from simple no-frills systems to complex central management. These systems are available from a variety of vendors and through open source. Regardless of which system you use, you need to pull data showing utilization, availability, performance, and errors. The system should alert the staff through emails or SMS messages so that you are aware of problems before the phone rings.

After the monitoring system is in place, you need to periodically characterize performance as a snapshot. A *snapshot* describes the expected performance of a system and enables you to compare later performance and recognize change. For instance, changes in jitter or in dropped packets might indicate that a WAN link is oversubscribed. In addition, a functional baseline for performance metrics serves as a critical diagnostic tool for security breaches and zero-day attacks and worms. Without thorough knowledge of typical behavior on a given network, aberrant traffic analyses become a subjective art.

# Tools

Most network administrators have a variety of tools in their toolbag. Some of the basic tools include a configuration history, device logs, and documentation. As the number of devices maintained grows, tools that collect data about the performance of the network and tools that collect user issues become increasingly important.

## Configurations

A configuration history is built by saving the device configuration to a central point periodically or after each change. IOS supports a variety of different remote targets. FTP and TFTP are commonly used because implementations are bundled with many operating systems, and free open-source versions are readily available.

```
Blackburn-rtr01# copy run ?
  archive:        Copy to archive: file system
  flash:          Copy to flash: file system
  ftp:            Copy to ftp: file system
  http:           Copy to http: file system
  https:          Copy to https: file system
  idconf          Load an IDConf configuration file
  null:           Copy to null: file system
  nvram:          Copy to nvram: file system
  pram:           Copy to pram: file system
  rcp:            Copy to rcp: file system
  running-config  Update (merge with) current system configuration
  scp:            Copy to scp: file system
  slot0:          Copy to slot0: file system
  startup-config  Copy to startup configuration
  syslog:         Copy to syslog: file system
  system:         Copy to system: file system
  tftp:           Copy to tftp: file system
  tmpsys:         Copy to tmpsys: file system
  xmodem:         Copy to xmodem: file system
  ymodem:         Copy to ymodem: file system
```

One way to build a configuration history is to save your configuration after each change. Saving the file with the date attached makes it easy to sort later, and adding a .txt makes it easy for Windows-based machines to open the file. In the following example, the TFTP server has a directory for each site and the configuration is saved with the date:

```
Blackburn-rtr01# copy run tftp
Address or name of remote host []? 192.168.255.10
Destination filename [blackburn-rtr01-confg]? blackburn/blackburn-
  rtr01-09-08-25.txt
!!
820 bytes copied in 2.628 secs (312 bytes/sec)
```

Logging events and alerts to Syslog is another important tool. Syslog is a facility that receives alerts from network equipment and stores them in a common log. Again, many version of syslog are available. Events are logged based on a severity scale, from zero to seven. Choosing a logging level tells the router to transmit events at that level and lower. To set up syslog support on an IOS device, the logging keyword is used, as shown here:

```
Blackburn-rtr01(config)# logging trap ?
  <0-7>          Logging severity level
  alerts         Immediate action needed           (severity=1)
  critical       Critical conditions               (severity=2)
  debugging      Debugging messages                (severity=7)
  emergencies    System is unusable                (severity=0)
  errors         Error conditions                  (severity=3)
  informational  Informational messages            (severity=6)
  notifications  Normal but significant conditions (severity=5)
  warnings       Warning conditions                (severity=4)
  <cr>
```

```
Blackburn-rtr01(config)# logging on
Blackburn-rtr01(config)# logging 192.168.255.10
Blackburn-rtr01(config)# logging trap informational
```

As the rate of log entries grows (because there are more devices or because the sensitivity is changed), finding the appropriate information in the logs becomes more cumbersome. One way to make it easier to tie events together in the log is to have accurate time on each device so that log entries have a consistent time. Time stamps become vital in forensics and post mortems, where sequence and patterns of events evolve into chains of evidence.

Time is synchronized on network devices using the network time protocol (NTP). Setting up NTP is straightforward; specify the NTP server with the command **ntp server** <*ip address*>. Time servers are organized by *stratums*, where stratum 1 clocks are super precise atomic clocks, stratum 2 devices get their time from stratum 1, stratum 3 devices ask stratum 2, and so on. Public stratum-1 devices are listed on the Internet; it is considered a courtesy that each organization has a minimal number of connections to a stratum-1 device and that other clocks in the organization pull from these stratum-2 devices.

Another time-related logging issue to consider is time zone. Will your organization log using local time zones, the time zone of headquarters, or set all devices to GMT? The following example demonstrates the time zone set to GMT, logging set, and the router set to use a remote NTP server:

```
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
ntp server 192.168.1.1
clock timezone GMT 0 0
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
```

Cisco IOS supports an Archive and Restore feature that makes maintaining a configuration history and logs easier. The archive function maintains a current copy of the configuration and a set of previous configurations. The archive can be maintained within the router or at an accessible URL. The restore function enables the router to smoothly revert to any of the saved configurations.

Setting up the archive function involves going into the archive configuration mode. The path command specifies a backup location, and time-period is used to periodically backup the configuration. If write-memory is specified, an archive copy will be made whenever the configuration is saved. Archive copies have a version number, such as “-1” on the end. This version number is reset with each router reset, so it would be hard to use this as a long-term archive. The path can include $h for the hostname and $t for time, so it is possible to time stamp each saved file. Using the time stamp is impractical with a Windows TFTP server, however, because the time stamp includes colons. In the next example the filename is *hostname*.txt and results in Blackburn-rtr01 saving files named Blackburn-rtr01.txt-1 and Blackburn-rtr01.txt-2. The example is set to back up at the maximum periodic interval, so most backups happen because the administrator saves the configuration:

```
archive
 path tftp://192.168.255.10/$h.txt
 write-memory
 time-period 525600
```

The router uses a standard name structure for all saved files, counting up to 14 and then cycling back to 1. This is hard to use as a complete configuration history. One possible solution is to save the archive to flash and to have administrators save to TFTP periodically (which automatically updates the flash archive). The periodic backup could be set to run once a week, just in case someone forgot to "copy run start":

```
archive
 path flash://$h
 write-memory
 time-period 10080
```

Archive can help troubleshoot in two ways. First, archive can compare differences between different versions of the config: archive config differences. Second, Archive can also be used to supplement syslog with all commands executed on the router. In archive configuration mode, enter log config mode. **logging enable** turns on command capture; **hidekeys** prevents logging passwords. Normally the log of commands is kept in memory on the router, but **Notify syslog** exports the commands to syslog. This configuration is shown here:

```
archive
 path flash://$h
 write-memory
 time-period 10080
 log config
  logging enable
  hidekeys
  notify syslog
```

To review the archive files, use the command **show archive**:

```
Blackburn-rtr01# show archive
The next archive file will be named tftp://192.168.255.10/Blackburn-
 rtr01-7
 Archive #  Name
   0
   1      tftp://192.168.255.10/Blackburn-rtr01-1
   2      tftp://192.168.255.10/Blackburn-rtr01-2
   3      tftp://192.168.255.10/Blackburn-rtr01-3
   4      tftp://192.168.255.10/Blackburn-rtr01-4
   5      tftp://192.168.255.10/Blackburn-rtr01-5
   6      tftp://192.168.255.10/Blackburn-rtr01-6 <- Most Recent
   7
   8
   9
   10
   11
   12
   13
   14
```

Finally, the archiving function adds the ability to restore to a previous configuration. Replacing an old configuration with **copy tftp run** results in the tftp file being merged into the running configuration whereas **copy tftp start** results in a complete replacement but requires a restart.

An archive can be restored with the **configure replace** command. The router compares the running configuration against the archive and builds and applies a list of commands necessary to match the archive. This method avoids reapplying existing commands or rebooting to make the migration:

```
Router# configure replace tftp://192.168.255.10/blackburn-rtr01-5
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Loading blackburn-rtr01-5 from 192.168.255.10 (via FastEthernet0/0): !
```

One trick when working with a remote router is to use "**reload in 5**" to schedule a reload. If a command inadvertently breaks the connection, the router reboots to the last saved configuration. If everything works, reload cancel prevents the reboot. The same functionality is available with **configure replace** *filename* **time** but avoids the reboot. Avoid the rollback by confirming the change is working with **configure confirm**.

## Other Tools

Documentation is a huge part of troubleshooting, and there are many tools that you can use to compile documentation. One of the key things to understand about documentation is that it must be easy and quick to update, or it will quickly grow stale. Microsoft Visio is a common way to show connectivity. A database or spreadsheet is frequently used to track inventory. You can use a ticketing system to list issues and gather trending data. Wikis are a more recent innovation that enables the network staff to produce and edit documentation.

There isn't a definitive way to produce documentation; the important part is to have documentation that is useful in the troubleshooting process. Ideally, the documentation should also feed directly into the disaster recovery process as well, so it should include part numbers, serial numbers, service contracts, and a variety of information that isn't strictly part of the "network" description.

Cisco has a variety of web-based tools that are helpful. The Dynamic Configuration tool is useful in planning hardware configurations; this tool can verify compatibility and build a parts list to help you plan a project. The Feature Navigator verifies that a specific feature is in a particular version of IOS. The Power Calculator calculates the required power supply for PoE installations. Many other tools are available through CCO, so it's worth spending some time understanding the width of the offering.

A final category of tools to consider are the network performance monitoring tools. Typically, monitoring and performance tracking in a small organization is accomplished with a phone—people call when they have problems. As an organization grows, however, it becomes more and more important to recognize problems before they occur. This same information can be used to budget hardware and circuit upgrades.

Monitoring tools typically use SNMP, Netflow, pings, and Syslog data to compile statistics about the current and historical behavior of the network. Typically, networks are monitored for capacity usage, availability, delay, and CPU and memory utilization. Solarwinds, nGenius, OPNET Net Doctor, SP Guru, and WhatsUpGold each make products that fulfill these functions, and MRTG is a similar open source project.

Remember to plan a monitoring system around the service level agreements (SLA) in the environment. Service providers typically offer some performance guarantees, such as minimizing unplanned downtime or minimizing jitter. The business might insist that IT also support SLAs internally. The Network monitoring system should provide information to back up both types of SLAs. Cisco has built in a SLA monitoring tool that can make availability statistics known and monitored for critical links and servers. This is called SLA Monitor and is customizable for MPLS, link utilization, RTT, and others. It is quite useful for critical traffic real-time monitoring and notification. Frequently these statistics are run as a continuous background process between CE nodes between sites, if remote connectivity between critical traffic endpoints is a business driver.

CHAPTER 2

# Troubleshooting Methodology

The responsibilities of a network administrator boil down to four essential measurements: Maximize performance and availability; minimize cost, and time-to-repair.

This chapter focuses on minimizing time-to-repair. The time it takes to restore functionality is predicated on two things: preparation and technique. The previous chapter spoke about the elements of preparation, such as documentation and scheduled preventative maintenance. This chapter focuses on the techniques that you can apply to minimize downtime.

Each of the troubleshooting practices described in this chapter assume that good documentation exists and that appropriate tools are available. Troubleshooting is much more frustrating and time consuming when the necessary preparation isn't accomplished.

**NOTE**

The Cisco Troubleshooting test doesn't assume a specific approach. Many approaches and different approaches might be successful in specific situations. The test does advocate a structured approach to troubleshooting, based on the scientific method.

## Principles

The scientific method is commonly described as a six-step process:

1. Define the problem.
2. Gather information.
3. Hypothesize.
4. Test hypothesis.
5. Analyze test.
6. Interpret results and, if necessary, generate a new hypothesis.

The first step—problem description—is usually accomplished when a user reports a problem. The initial problem description tends to be vague or overly general. (“The Internet is down!”) A troubleshooter’s initial response should therefore be to gather more information and build a more specific description. You can determine symptoms by talking to the user, by personal observation, or by referring to management systems such as Netflow, Syslog, and SNMP monitors.

When you have an adequate description of the problem, you can form a hypothesis. A hypothesis is a hypothetical potential problem whose symptoms would be similar. The hypothesis should commonly suggest a way to prove or disprove itself. For instance, if you suspect that the WAN connection is down, looking at the interface status or pinging a remote device would test that theory.

Test results will either support or refute a theory. A single test result can’t prove a theory but just support it. For example, ping might be used to test a WAN connection. A ping timeout cannot, by itself, be considered definitive. The target might be shut down or have a firewall that drops ICMP. Test results should be confirmed through a number of different lines of evidence. If the tests contradict the hypothesis, start over with a new theory.

After a hypothesis is accepted as a reasonable explanation, you can take action to fix the problem. Of course, any action is another type of test. If the action doesn’t fix the problem, simply develop a new hypothesis and repeat the process.

## Structured Troubleshooting

The term *structured troubleshooting* describes any systematic way of collecting information, forming a hypothesis, and testing. In a structured approach, each unsuccessful test rules out entire classes of possible solutions and gracefully suggests the next hypothesis. An unstructured—random—approach usually takes much longer and is less likely to be successful.

A number of techniques have been used successfully, their common feature being a rigorous and thoughtful approach that collects data and analyzes data:

- **Top down:** Start at the OSI application layer and drill down.
- **Bottom up:** Start with the OSI physical layer and work up.
- **Divide and conquer:** Start at the network layer and follow the evidence, developing specific tests of each hypothesis.
- **Follow path:** Consider the “packets perspective” and examine the devices and processes it encounters moving through the network. Understand the order of operations within each device to do this.

- **Spot difference:** Compare the configuration to an older version or to that of a similar device. Diff and WinDiff are tools that make this comparison easy.
- **Move the problem:** Swap components to see if the problem moves with a device.

**Figure 2-1  The OSI Model**

| | |
|---|---|
| 7 | Application |
| 6 | Presentation |
| 5 | Session |
| 4 | Transport |
| 3 | Network |
| 2 | Data Link |
| 1 | Physical |

There isn't a single "best method," although a given technician might find one more intuitive or more suitable for a given problem. It's a good idea to be familiar with each technique and to change approaches if necessary.

Two troubleshooting tactics need special mention. Most technicians build up a reservoir of experience, which gives them an intuition about the solution to a given problem. This can be incredibly impressive when it works; the trick is to not let this become a series of random stabs when it doesn't work.

Networkers also look for things that happened at about the same time, on the theory that the similar timing implies causation. This thinking is a logical error: *post hoc ergo proctor hoc*. Sometimes this does provide a clue, but large networks have many things happening contemporaneously every second. This troubleshooting method can easily provide a false lead.

# The Troubleshooting Method

Troubleshooting a network falls into a series of steps that mirror the scientific method.

The first step in troubleshooting is to define the problem. Some users, for instance, might report that "The Internet is down," when what they mean is "My e-mail is taking a long time to download." Some users over-generalize or exaggerate for effect, but most users lack the technical sophistication to tell which symptoms are relevant. Always start the troubleshooting process by gathering a detailed description of the problem. Ask questions to gather details, such as the names and locations of affected devices. One good way to gather details is to ask about how the problem can be duplicated. (“So, if I browse the web I’ll see this problem?”)

After defining the problem, gather information about the problem. What is the scope? What other devices or locations are affected? When did it start? How can you test the problem?

As information is gathered, one or more theories might begin to form. Develop tests that confirm or refute the theories, and work to find the root cause. Tests can be as simple as pings or as complex as implementing a configuration change; the tests should be aimed at separating valid theories.

When the testing process is complete, take a moment to consider the results. Do the results suggest a configuration or hardware change? Is the problem resolved? If not, reconsider the problem description and the original hypothesis. Either the problem was not completely and accurately described, or the hypothesis was incorrect and needs to be revisited.

When the problem is resolved, take some time to consider the changes. The state of the network and the problem resolution need to be communicated, and documentation might need to be updated. Past these obvious steps, consider whether the problem found can be in other parts of the network. If the problem were in the configuration, think through the configuration template used in your network and determine if the fix needs to be repeated preemptively on other devices.

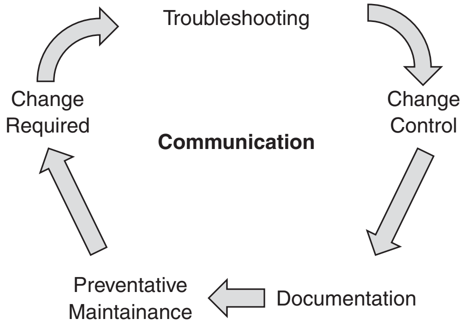
Each organization has its own specific methods for working through the break/fix cycle. The important points here are to work logically and methodically, and to view each problem as an opportunity to perfect the larger network.

# Integrating Troubleshooting into Maintenance

Every interaction with the network is an opportunity to learn. Smart organizations capture information learned to solve similar problems and to help understand the network in the future. Change control and documentation are the two principal ways that feedback from network changes is incorporated into the maintenance cycle, as shown in Figure 2-2.

Preventative maintenance is ongoing, but changing conditions or reported problems create the need to make a change. Troubleshooting identifies the corrective action to upgrade or repair the network. Throughout these processes, a regular communication with end users is critical to understand the problem and to gather feedback on the solution. Communication with end users, within the team, and with management is pervasive throughout the cycle.

**Figure 2-2 Maintenance Cycle**



Change control is a process found in many organizations with large networks. The change-control process is a formal communication process for requesting and receiving permission. Change control provides an opportunity for management and peers to be aware and consent to the proposed change. The change process encourages the network technician to take a deliberate and thoughtful approach. Finally, the change process creates a record of the change that can be incorporated in documentation.

After a change is made and an issue is resolved, updating documentation must be seen as a part of the clean-up process. Most organizations have records including IPs, inventory, configurations, and topology; changes need to be added to these records. If the change is sufficiently broad, it might also need to be incorporated into standards and templates so that other devices can be preemptively upgraded. As records and standards change, team members need to be educated on the changes.

A *baseline* is a reading of the critical parameters of the network (such as latency and utilization) over a period of time. The baseline serves as a record of normal behavior to help identify how performance has changed. Updating baseline information is part of the documentation process.

**NOTE**

A number of tools can compile baseline data and monitor the network continuously. Cisco Works, HP Openview, What's Up? and SolarWinds are examples of commercial applications. Cacti and MRTG are two well-known Open Source versions.

Think about troubleshooting as a holistic process. Approach each issue with a rational evidence-based philosophy, make thoughtful changes, and communicate with all the invested groups often.

# CHAPTER 3

# Troubleshooting Tools

Cisco IOS has a number of ways to extract data about the state of the machine. Understanding the capabilities of the operating system and how to use them effectively can reduce time-to-repair and the stress of a network outage.

## IOS Filtering Tools

Most of the commands for pulling information from a router are familiar to anyone with Cisco IOS experience. Many people are not familiar with the filtering techniques that enable a troubleshooter to quickly focus.

Some of these filters are command-specific. Consider **show ip route**, which is a familiar command. When used, this command shows a complete routing table (as shown here):

```
Foard-rtr01# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
 level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
 static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.100.1.1 to network 0.0.0.0

     172.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
B       172.136.157.12/30 [20/0] via 10.1.254.246, 2d01h
S       172.99.120.2/31 [1/0] via 10.100.254.240
B       172.139.78.232/30 [20/0] via 10.1.254.246, 2d01h
B       172.136.88.20/30 [20/0] via 172.176.128.25, 5w2d
B       172.136.41.104/30 [20/0] via 172.176.128.25, 5w2d
B       172.137.230.128/30 [20/0] via 172.176.128.25, 6d18h
B       172.139.83.100/30 [20/0] via 172.176.128.25, 1w5d
     172.16.0.0/32 is subnetted, 1 subnets
S       172.16.201.141 [1/0] via 10.100.254.240
```

```
     192.168.0.0/30 is subnetted, 6 subnets
B       192.168.26.52 [20/0] via 10.1.254.246, 2d01h
B       192.168.241.236 [20/0] via 172.176.128.25, 5w2d
…
```

The output for this command can continue over many pages of information. One way to summarize this information is to ask for a summary using **show ip route summary**.

```
Foard-rtr01# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           19          1216        2888
static          4           22          1664        3952
bgp 65100       19          385         25856       62428
  External: 382 Internal: 22 Local: 0
internal        45                                  52740
Total           68          426         28736       122008
Removing Queue Size 0
```

A second routing table filtering option is to ask for a selection of routes. Specifying an address, mask, and the keyword **longer-prefixes** asks for anything that matches the prefix or any routes contained within the prefix. The following example shows all the more-specific routes contained within the 10.1.254.0/24 block:

```
Foard-rtr01# show ip route 10.1.254.0 255.255.255.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
 level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
 static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.100.254.240 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 241 subnets, 12 masks
C       10.1.254.244/30 is directly connected, Multilink31
C       10.1.254.246/32 is directly connected, Multilink31
B       10.1.254.252/30 [200/0] via 10.100.1.2, 1d09h
C       10.1.254.232/30 is directly connected, Multilink42
C       10.1.254.234/32 is directly connected, Multilink42
```

The options for filtering available for a given **show** command vary, so it's a good idea to spend some time with the question mark and understand the options available for areas of focus in your organization.

Generic filters can also be applied to all **show** commands. **Show process cpu**, which might be used to look for runaway processes, can be used as an example. First, an example portion of output is shown:

```
Foard-rtr01# show process cpu
CPU utilization for five seconds: 14%/13%; one minute: 14%; five
 minutes: 14%
 PID Runtime(ms)   Invoked   uSecs   5Sec   1Min   5Min TTY Process
   1         292      6405      45  0.00%  0.00%  0.00%   0 Chunk Manager
   2         296    639947       0  0.00%  0.00%  0.00%   0 Load Meter
   3           0         1       0  0.00%  0.00%  0.00%   0 chkpt message ha
   4           0         1       0  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   5     1600592    326740    4898  0.00%  0.04%  0.00%   0 Check heaps
   6        2016     28869      69  0.00%  0.00%  0.00%   0 Pool Manager
   7           0         2       0  0.00%  0.00%  0.00%   0 Timers
   8           0         2       0  0.00%  0.00%  0.00%   0 ATM AutoVC Perio
   9           0         2       0  0.00%  0.00%  0.00%   0 ATM VC Auto Crea
  10           0     53330       0  0.00%  0.00%  0.00%   0 IPC Dynamic Cach
  11           0         1       0  0.00%  0.00%  0.00%   0 IPC Zone Manager
  12          20   3199682       0  0.00%  0.00%  0.00%   0 IPC Periodic Tim
  13          12   3199682       0  0.00%  0.00%  0.00%   0 IPC Deferred Por
  14           0         4       0  0.00%  0.00%  0.00%   0 IPC Seat Manager
  15           0         1       0  0.00%  0.00%  0.00%   0 IPC BackPressure
  16      387428  57716731       6  0.07%  0.01%  0.00%   0 EnvMon
…
```

The pipe (|) character is used to filter output by passing it through logic such as include, exclude, begin, and section. Output is matched against a regular expression.

Following is a table of common regular expression characters.

| Character | Usage | Example |
|---|---|---|
| ^ | Begins with | ^Fast matches lines that begin with FastEthernet. |
| $ | Ends with | FastEthernet0/0$ matches lines that end with FastEthernet0/0. |
| . | Any character | Ethernet./. matches Ethernet 0/0, FastEthernet0/1, and Ethernet ?. |

| Character | Usage | Example |
|---|---|---|
| \| | Or | FastEthernet 0/0\|1 matches either FastEthernet0/0 and FastEthernet0/1. |
| _ | Matches beginning, end, or braces | _Ethernet_ matches any line that includes the word "Ethernet." |

A show command piped to include will display any line of output that matches the regular expression. In the following example, the pipe is used to look for any line that includes the text "IP Input".

```
Foard-rtr01# show process cpu | include IP Input
  87     2755292  47045037         58  0.07%  0.07%  0.07%   0 IP
 Input
```

The running configuration is another place to see piping work. In the following example, piping to begin starts the output at the telnet ports. This is a lot easier that using the space key to work through a large configuration:

```
Foard-rtr01# show running-configuration | begin vty
line vty 0 4
 exec-timeout 20 0
 password 7 0401001C02010D4106
 logging synchronous
 transport input ssh
 transport output telnet ssh
line vty 5 15
 exec-timeout 20 0
 password 7 0401001C02010D4106
 logging synchronous
 transport input ssh
 transport output telnet ssh
!
ntp source Loopback0
ntp master 5
ntp update-calendar
ntp server 172.31.55.2
ntp peer 10.1.1.123 key 1 source Loopback0
end
```

In the preceding example, piping to begin also includes all the text after the part of interest. Piping to **section** shows the indented commands under a line that matches the regular expression. In the following example, the sections found under the keyword vty are shown:

```
Foard-rtr01# show running-config | section vty
line vty 0 4
 exec-timeout 20 0
 password 7 045C021302284D4906
 logging synchronous
 transport input ssh
 transport output telnet ssh
line vty 5 15
 exec-timeout 20 0
 password 7 14101B1E010D2B2C2B
 logging synchronous
 transport input ssh
 transport output telnet ssh
```

### NOTE

Piping output can be a great way to focus on relevant details, but **show running-configuration | section** is a lot to type, particularly repeatedly. The alias command can make this easier. In configuration mode, create a shortened version of a command as shown next.

```
rtr01(config)# alias exec srs show running-configuration | section
```

### Note

Now "**srs**" is the shortened version of the long and cumbersome command. Type **srs vty** to see the same output as the example.

The pipe symbol is also used as an OR within a regular expression, as shown in the next examples. Normally, **show ip interface brief** summarizes all the interfaces found on a router. Some routers have a large number of interfaces, making even this simplified display cumbersome. In the following text, some of the interfaces are grouped into multilinks and others are turned off. Finding the detail you need is complicated by the long and confusing output:

```
Foard-rtr01# show ip interface brief
Interface             IP-Address    OK? Method Status             Protocol
FastEthernet0/0       10.87.1.1     YES NVRAM  up                    up
FastEthernet0/0.2     10.76.2.2     YES NVRAM  up                    up
FastEthernet0/0.3     10.76.3.2     YES NVRAM  up                    up
FastEthernet0/0.4     10.76.4.2     YES NVRAM  up                    up
FastEthernet0/0.5     10.76.5.2     YES NVRAM  up                    up
FastEthernet0/0.6     10.76.6.2     YES NVRAM  up                    up
FastEthernet0/0.7     10.76.7.2     YES NVRAM  up                    up
FastEthernet0/0.8     10.76.8.2     YES NVRAM  up                    up
FastEthernet0/0.12    10.76.12.2    YES NVRAM  up                    up
FastEthernet0/0.120   10.76.12.130  YES NVRAM  up                    up
FastEthernet0/0.1000  10.76.0.2     YES NVRAM  up                    up
FastEthernet0/1       unassigned    YES NVRAM  administratively down down
GigabitEthernet0/1    unassigned    YES NVRAM  administratively down down
FastEthernet0/2       unassigned    YES NVRAM  administratively down down
GigabitEthernet0/2    unassigned    YES NVRAM  administratively down down
GigabitEthernet0/3    unassigned    YES NVRAM  administratively down down
Serial1/0             unassigned    YES NVRAM  administratively down down
Serial1/0.402         unassigned    YES unset  administratively down down
Serial1/0.404         10.1.254.237  YES NVRAM  administratively down down
Serial1/1             unassigned    YES NVRAM  administratively down down
Serial1/2             unassigned    YES NVRAM  administratively down down
Serial1/3             unassigned    YES NVRAM  administratively down down
Serial1/4             unassigned    YES NVRAM  administratively down down
Serial1/5             unassigned    YES NVRAM  administratively down down
Serial1/6             unassigned    YES NVRAM  administratively down down
Serial1/7             unassigned    YES NVRAM  administratively down down
Serial2/0:0           unassigned    YES NVRAM  up                    up
Serial2/1:0           unassigned    YES NVRAM  up                    up
Serial2/2:0           unassigned    YES NVRAM  up                    up
Serial3/0             unassigned    YES NVRAM  up                    up
Serial3/0.100         172.16.128.26 YES NVRAM  up                    up
Serial3/1             unassigned    YES NVRAM  down                  down
Serial4/0:0           unassigned    YES NVRAM  down                  down
Serial4/1:0           unassigned    YES NVRAM  down                  down
Serial4/2:0           unassigned    YES NVRAM  down                  down
Serial4/3:0           unassigned    YES NVRAM  down                  down
Serial4/4:0           unassigned    YES NVRAM  up                    up
Serial4/5:0           unassigned    YES NVRAM  up                    up
Serial4/6:0           unassigned    YES NVRAM  up                    up
Serial4/7:0           unassigned    YES NVRAM  up                    up
Serial6/0:0           unassigned    YES NVRAM  down                  down
Serial6/1:0           unassigned    YES NVRAM  down                  down
Serial6/2:0           unassigned    YES NVRAM  down                  down
Serial6/3:0           unassigned    YES NVRAM  down                  down
```

```
Serial6/4:0          unassigned    YES NVRAM  up                    up
Serial6/5:0          unassigned    YES NVRAM  up                    up
Serial6/6:0          unassigned    YES NVRAM  up                    up
Serial6/7:0          unassigned    YES NVRAM  up                    up
SSLVPN-VIF0          unassigned    NO  unset  up                    up
Multilink20          10.1.254.249  YES NVRAM  down                  down
Multilink31          10.1.254.245  YES NVRAM  up                    up
Multilink42          10.1.254.233  YES NVRAM  up                    up
Loopback0            10.1.1.1      YES NVRAM  up                    up
Loopback1            10.254.253.94 YES NVRAM  up                    up
```

To condense the output to the active parts, the following example pipes the output to exclude any lines with the words “unassigned” or “administratively.” Notice how much this simplifies the display:

```
Foard-rtr01# show ip interface brief | exclude unassigned|administra-
  tively
Interface                  IP-Address     OK? Method Status      Protocol
FastEthernet0/0            10.87.1.1      YES NVRAM  up          up
FastEthernet0/0.2          10.76.2.2      YES NVRAM  up          up
FastEthernet0/0.3          10.76.3.2      YES NVRAM  up          up
FastEthernet0/0.4          10.76.4.2      YES NVRAM  up          up
FastEthernet0/0.5          10.76.5.2      YES NVRAM  up          up
FastEthernet0/0.6          10.76.6.2      YES NVRAM  up          up
FastEthernet0/0.7          10.76.7.2      YES NVRAM  up          up
FastEthernet0/0.8          10.76.8.2      YES NVRAM  up          up
FastEthernet0/0.12         10.76.12.2     YES NVRAM  up          up
FastEthernet0/0.120        10.76.12.130   YES NVRAM  up          up
FastEthernet0/0.1000       10.76.0.2      YES NVRAM  up          up
Serial3/0.100              172.176.128.26  YES NVRAM  up         up
Multilink20                10.1.254.249    YES NVRAM  down       down
Multilink31                10.1.254.245    YES NVRAM  up         up
Multilink42                10.1.254.233    YES NVRAM  up         up
Loopback0                  10.1.1.1        YES NVRAM  up         up
Loopback1                  10.254.253.94   YES NVRAM  up         up
```

### NOTE

The **alias** command can make this easier. In configuration mode, create a shortened version of a command as shown here.

```
Router(config)# alias exec ii show ip interface brief | exclude unas-
  signed|administratively
```

Now **ii** is the shortened version of the long and cumbersome command.

A second example shows the OR capability by piping the output of **show process cpu** to include lines that start with CPU or include the words **IP Input**:

```
Foard-rtr01# show process cpu | inc ^CPU|IP Input
CPU utilization for five seconds: 14%/13%; one minute: 14%; five
  minutes: 14%
   87     2755772  47054573         58  0.07%  0.07%  0.07%   0 IP
  Input
```

## Output Redirection

In addition to filtering output, IOS also enables **show** command output to be redirected. Redirecting output enables an administrator to collect information for archiving or to share with other troubleshooters and save it as a text file.

Output can be piped to a file using either redirect or tee. Redirect just creates the file, whereas tee also displays the content in session. Any filesystem supported by that router is supported, so output can be pointed at flash, tftp, ftp, http, and other destinations.

The syntax to use this function is

```
Show command | redirect file
Show command | tee file
```

The next examples show the running configuration being piped to TFTP. In the first example, the output is redirected. The second example tees the output so that it builds the TFTP file and displays on screen.

```
Foard-rtr01# show running-configuration | redirect tftp://tftp/
  Foard-rtr01-shrun.txt
Translating "tftp"...domain server (10.186.2.30) [OK]

Foard-rtr01# show running-configuration | tee tftp://tftp/
  Foard-rtr01-shrun.txt
!
Building configuration...

Current configuration : 22291 bytes
…
```

## IOS Troubleshooting Tools

Ping and traceroute are the most obvious tools available in IOS to test the network.

TSHOOT

Ping tests connectivity and is so commonly used that even end users are passingly familiar with it. A ping response shows that a working path between two end points exists. End systems sometimes have firewalls that prevent response, but generally ping is a reasonable first test of network connectivity:

```
Foard-rtr01# ping 10.186.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.186.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
```

Exclamation marks show a response, but there is a lot of information besides the most obvious part. First, pay attention to the pattern of the response. Alternating success and failure (!.!.!) is a classic sign of a load balancing problem, where one path succeeds and the other fails. Second, pay attention to the response time. Many applications depend on quick response. Voice, for instance, assumes a round-trip time of less than 150 ms. The response time can also clue the troubleshooter to utilization issues. If the response time is much larger than usual that might indicate a heavy traffic load and queuing. If you notice that the minimum and maximum times vary widely, this could also be a sign of queuing because of a heavy load.

Ping can do a lot more than that simple test, however. Privileged mode supports an extended ping that enables every aspect of ping to be controlled. This opens up many more tests that can be accomplished with the humble command.

The following example below an extended ping. Notice that the command **ping**—with no destination specified—is entered in privileged mode. The example sends five pings of 100 bytes, then five of 200 bytes, continuing to 1500 byte pings. The DF bit (do not fragment) is set. A similar ping might be used if you suspect that an intermediate link didn't support the same size MTU as the source and destination. A more detailed explanation of the command is found after the example:

```
Foard-rtr01# ping
Protocol [ip]:
Target IP address: 10.186.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: loopback0
Type of service [0]:
```

```
Set DF bit in IP header? [no]: y
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]: y
Sweep min size [36]: 100
Sweep max size [18024]: 1500
Sweep interval [1]: 100
Type escape sequence to abort.
Sending 75, [100..1500]-byte ICMP Echos to 10.186.1.1, timeout is 2
  seconds:
Packet sent with a source address of 10.1.1.1
Packet sent with the DF bit set
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  !!
!!!!!
Success rate is 100 percent (75/75), round-trip min/avg/max =
  8/10/12 ms
```

Remember that defaults are shown in square brackets. Selecting all the defaults is similar to a normal ping.

Sometimes testing involves repeatedly pinging (for instance, when you believe that an interface is flapping up and down). An extended ping with a repeat count of 99999 can be used to interactively test the network over a period of time.

Pings can be set to different packet sizes through the Datagram Size variable. The router can automate testing a range of sizes. To do so, use the extended commands and choose to sweep a range of sizes.

If a router is asked to forward a packet that is larger than the MTU of the transmitting link, the router normally breaks the packet into smaller pieces. Setting the DF bit instructs receiving routers to discard the traffic rather than fragment it.

Using different size packets and setting the DF bit allows testing MTU. When the MTU limit is reached, all subsequent pings will be dropped.

Another nice testing technique is to change the source interface. Pings are normally sourced from the transmitting interface. Using an internal interface as the source shows that the receiving device and the intermediate routers understand how to route back to that prefix.

A final idea is to try different Type of Service settings. Many networks now carry voice, video, and prioritized data. Voice is commonly set to ToS 5, so pinging using ToS 5 enables a peek into how the QoS settings are functioning.

Like ping, there is an extended version of traceroute. It has a few of the same capabilities, with one other significant testing ability. Traceroute in IOS uses UDP, and extended traceroute enables setting the UDP port. This can be used to test application performance for applications that use UDP, such as voice. This is important when trying to diagnose the affects of firewalls and access-lists.

An example extended traceroute is shown next. The only choice specified in the example is to use UDP port 16000:

```
Newton-rtr01# traceroute
Protocol [ip]:
Target IP address: 10.200.1.1
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]: 16000
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 10.200.1.1
```

In the same way the UDP port connectivity can be probed with traceroute, telnet can be used to test TCP ports. Telnet does not offer many options, but by changing the target port, different network services can be tested. The following examples show that email and the web server respond on the appropriate ports:

```
Foard-rtr01# telnet www.example.com 25
Translating "www.example.com"...domain server (10.1.2.2) [OK]
Trying www.example.com (172.16.0.25, 25)... Open
220 www.example.com ESMTP Postfix

Foard-rtr01# telnet www.example.com 80
<ctrl-c>
HTTP/1.1 400 Bad Request
Content-Type: text/html
Date: Fri, 4 Sep 2009 17:14:29 GMT
Connection: close
Content-Length: 35

<h1>Bad Request (Invalid Verb)</h1>
```

# Hardware Diagnostics

The commands examined so far have dealt with network issues, but sometimes the problem is within the IOS device. Several commands describe the functional state of an IOS device.

If network hardware is suspected, a good place to start troubleshooting is to understand the external environment. The **show environment all** command displays information about the temperature within the device and the state of the power supplies. Especially when troubleshooting remotely it is easy to forget power and air conditioning, but problems in either area can lead to device malfunction:

```
Foard-rtr01# sh environment all
Power Supplies:
        Power Supply 1 is AC Power Supply. Unit is on.
        Power Supply 2 is AC Power Supply. Unit is on.

Temperature readings:
        NPE Inlet        measured at 25C/77F
        NPE Outlet       measured at 27C/80F
        I/O Cont Inlet   measured at 25C/77F
        I/O Cont Outlet  measured at 28C/82F
        CPU Die          measured at 43C/109F

Voltage readings:
        +3.30 V       measured at +3.30 V
        +1.50 V       measured at +1.49 V
        +2.50 V       measured at +2.50 V
        +1.80 V       measured at +1.79 V
        +1.20 V       measured at +1.20 V
        VDD_CPU       measured at +1.28 V
        VDD_MEM       measured at +2.50 V
        VTT           measured at +1.25 V
        +3.45 V       measured at +3.43 V
        -11.95        measured at -12.17 V
        +5.15 V       measured at +4.96 V
        +12.15 V      measured at +12.18 V

Envm stats saved 0 time(s) since reload
```

A complete and accurate inventory is another part of troubleshooting. Of course, this information is much more useful if obtained before a problem occurs and connectivity drops! By comparing the inventory to previous inventories, it is possible to recognize differences (caused, presumably, by hardware failure). If the organization has a Cisco SmartNet maintenance contract, the serial number and part-number information is necessary to obtain spares:

```
Foard-rtr01# show inventory
NAME: "Chassis", DESCR: "Cisco 7206VXR, 6-slot chassis"
PID: CISCO7206VXR      , VID:    , SN: 24323096

NAME: "NPE-G2 0", DESCR: "Cisco 7200 Series Network Processing
 Engine NPE-G2"
PID: NPE-G2            , VID: V03 , SN: JAS1456B4EC

NAME: "disk2", DESCR: "256MB Compact Flash Disk for NPE-G2"
PID: MEM-NPE-G2-FLD256 , VID:    , SN:

NAME: "module 0", DESCR: "I/O Dual FastEthernet Controller"
PID: C7200-I/O-2FE/E   , VID:    , SN: 21753008

NAME: "disk0", DESCR: "Cisco 7200 I/O PCMCIA Flash Disk, 48M"
PID: MEM-I/O-FLD48M    , VID:    , SN:

NAME: "disk1", DESCR: "Cisco 7200 I/O PCMCIA Flash Disk, 48M"
PID: MEM-I/O-FLD48M    , VID:    , SN:

NAME: "module 1", DESCR: "Serial"
PID: PA-8T-V35=        , VID:    , SN: 49010448

NAME: "module 2", DESCR: "4 port, software configurable Multichannel
 T1/E1 with TDM Port Adapter"
PID: PA-MCX-4TE1         , VID:    , SN: JAS1680Y0EM

NAME: "module 3", DESCR: "Enhanced 2 port T3/E3 clear channel PA"
PID: PA-2T3/E3-EC        , VID: V01 , SN: JAS249200K5

NAME: "module 4", DESCR: "8 port, software configurable Multichannel
 T1/E1 without TDM Port Adapter"
PID: PA-MC-8TE1+         , VID:    , SN: JAS1689A2MM

NAME: "module 6", DESCR: "8 port, software configurable Multichannel
 T1/E1 without TDM Port Adapter"
PID: PA-MC-8TE1+         , VID:    , SN: JAS1689A2BV

NAME: "Power Supply 1", DESCR: "Cisco 7200 AC Power Supply"
PID: PWR-7200-AC       , VID:    , SN:

NAME: "Power Supply 2", DESCR: "Cisco 7200 AC Power Supply"
PID: PWR-7200-AC       , VID:    , SN:
```

A lack of memory can also cause a network issue. The **show memory** command displays the state of memory on a device; focus on the Free column to determine if enough is available. Another sign of memory issues is %SYS-2-MALLOCFAIL messages:

```
Foard-rtr01# show memory
                Head    Total(b)     Used(b)     Free(b)   Lowest(b)
 Largest(b)
Processor    6319860   818832732    74864300   743968432   742841100
 727580236
      I/O   38000000    67108864    11964260    55144604    55137712
 54643068
Transient   37000000    16777216       58244    16718972    16226680
 16718696
…
```

Hardware issues can also manifest themselves on the interfaces. **Show controller** can show some information about the interface—serial interfaces in particular report things such as cable information here. **Show interface** (shown next) displays a good deal of information about the state of the interface. In particular, pay attention to four measurements:

- **Input queue drops:** Signify that the router had more traffic than it could process. Some amount of drops is excusable, but drops could be related to CPU oversaturation. Double-check the processor with the **show processes cpu** command.
- **Output queue drops:** Usually mean that the line is congested.
- **Input errors:** These errors show duplex errors, interface problems, and CRC errors.
- **Output errors:** Usually related to duplex issues.

```
Foard-rtr01# show interface
FastEthernet0/0 is up, line protocol is up
  Hardware is i82543 (Livengood), address is 000a.f3f7.9808 (bia
 000a.f3f7.9808)
  Description: enter port #
  Internet address is 10.100.1.1/16
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 32/255, rxload 14/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  1., loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 5517000 bits/sec, 2571 packets/sec
  5 minute output rate 12927000 bits/sec, 2550 packets/sec
     1326060749 packets input, 711066620 bytes
     Received 45468700 broadcasts, 0 runts, 0 giants, 0 throttles
     148 input errors, 0 CRC, 0 frame, 0 overrun, 148 ignored
     0 watchdog
     0 input packets with dribble condition detected
     1191821108 packets output, 2981100223 bytes, 0 underruns
     2 output errors, 0 collisions, 4 interface resets
     5634739 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     2 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

# Working with External Tools

The IOS troubleshooting capabilities are supplemented by external network management tools. Cisco IOS devices support these tools and in many cases supply detailed information to the management system. This section describes the methods used to coordinate with these tools.

## Packet Sniffing

Packet capture from a laptop or specialized device enables low-level vision into the exact traffic flowing over a link. Capturing traffic can show errors and underlying protocol traffic. The issue with packet capture is that switches do not forward all traffic out all ports, so it is difficult to find a port from which to see all traffic.

SPAN (Switched Port Analyzer) is a tool within IOS switches to direct copies of packets to a capture port. SPAN is configured by identifying a source port or VLAN from which traffic should be copied. SPAN is then pointed to an output port, to which a packet capture tool is attached. SPAN can capture traffic on a switch and output to a trunked VLAN. A second switch can then capture the VLAN and output it to a port. This configuration is called remote SPAN (RSPAN).

The generic configuration of SPAN is

```
Monitor session [session number] [source|destination]
 [interface|vlan]
```

The following example shows the configuration used when suspicious device is on port F0/1 and a packet capture tool is plugged into port F0/24:

```
Monitor session 1 source interface f0/1
Monitor session 1 destination interface f0/24
```

Router IP Traffic Export (RITE) is similar to SPAN but used by routers to capture traffic to a monitoring port. The following example demonstrates capturing ten percent of the interesting traffic on f0/1 and exporting it to a device with a given MAC:

```
(config)# ip traffic-export profile rite
(config-rite)# interface FastEthernet 0/1
(config-rite)# bidirectional
(config-rite)# mac-address 00a.8aab.90a0
(config-rite)# incoming access-list my_acl
(config-rite)# outgoing sample one-in-every 10
(config)# interface FastEthernet0/0
(config-if)# ip traffic-export apply rite
```

RITE can also be used to export the traffic to a file on the router. From there it can be copied off for inspection on a PC:

```
traffic-export interface fastethernet0/0 copy tftp:
```

## Netflow

Netflow collects summaries of traffic information and transmits the summary to a Netflow collector. Netflow is enabled on each monitored interface. Netflow supports a version 5 and version 9; this should be set to match the requirements of your network management system. Finally, Netflow exports information to a target IP address. The commands to accomplish these actions are shown here:

```
(config-if)# ip flow ingress
(config)# ip flow-export version [5|9]
(config)# ip flow-export destination [ip-address]
```

In addition to using a monitoring system to track Netflow, an administrator can also peek into the current flows using **show ip cache flow**.

## SNMP and EEM

SNMP is another monitoring protocol. Whereas Netflow tracks traffic, SNMP can monitor any type of event or statistic from the device. SNMP is supported by most network monitoring systems. The router also has a tool to react to events through embedded event manager (EEM).

SNMP is set up by identifying a server and listing the events to be monitored. If **snmp-server enable traps** is used without specifying specific events, all traps are monitored:

```
(config)# snmp-server host [ip-address]
(config)# snmp-server enable traps
```

EEM enables custom reactions to events and acts as a supplement to SNMP. Events can be triggered by any SNMP event and actions can include (among others) SNMP, Syslog, IOS commands, and email messages.

A simple example EEM applet is shown next. This applet logs a Syslog message and outputs a message to the console in reaction to an administrator entering configuration mode:

```
Event manager applet CONFIG-STARTED
Event cli pattern "configure terminal" sync on skip no occurs 1
Action 1.0 syslog priority critical msg "Configuration mode was
  entered"
Action 2.0 syslog priority informational msg "Change control poli-
  cies apply.  Authorized access only."
```

**NOTE**

EEM applets are starting to appear on the Internet, both at Cisco.com and at other sites.

# CHAPTER 4

# Troubleshooting Switches

Ethernet is ubiquitous in campus networks and Data Centers. Movement to consolidate networks has collapsed storage and virtualization, and telephony has put more traffic on Ethernet. Maintaining this critical infrastructure involves understanding the component pieces: Spanning Tree, VLANs, InterVLaN routing, and gateway redundancy.

Poor forwarding performance on switches is usually associated with cabling and port problems, duplex mismatch, or TCAM issues.

Problems at the physical layer can be seen from **show interface**, **show interface counters** and **show interface counters errors**. Look for the following errors:

- **Align-Err, runts:** Alignment errors are usually associated with cabling, NICs, or duplex mismatch.
- **FCS-Err:** Frame Check Sequence errors are usually associated with a cabling issue.
- **Xmit-Err:** The transmission buffers are full. Commonly associated with switching a faster link to a slower link.
- **Undersize, Giants:** Suspect the transmitting NIC.
- **Single-Col, Multi-Col, Late-Col, Excess-Col:** Collisions are a sign of duplex mismatch.

An example of these commands is shown here.

```
Newton-Sw01# show interface fastethernet1/1
FastEthernet1/2 is up, line protocol is up (connected)
  Hardware is C6k 100Mb 802.3, address is 001c.58c8.ac92 (bia
 001c.58c8.ac92)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:43, output hang never
  Last clearing of "show interface" counters 6w5d
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output
 drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  1 minute input rate 0 bits/sec, 0 packets/sec
  1 minute output rate 7000 bits/sec, 9 packets/sec
     4182737 packets input, 719363170 bytes, 0 no buffer
     Received 5970 broadcasts (174 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     45957071 packets output, 19815895675 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out

Newton-Sw01# sh interface counters

Port            InOctets   InUcastPkts   InMcastPkts   InBcastPkts
Fa1/1            6658590         73024            27            95
Fa1/2          719363238       4176768           174          5796
…
Newton-Sw01# sh interface counters errors

Port   Align-Err    FCS-Err    Xmit-Err     Rcv-Err UnderSize OutDiscards
Fa1/1          0        309           0         309         0           0
Fa1/2          0          0           0           0         0           0
…
```

Duplex mismatch is a common cause of forwarding problems. Half-duplex is unusual in modern networks, so duplex mismatch usually occur when one port is set to auto and the other to full. Setting everything to auto is Cisco's recommendation.

### NOTE

When speed and duplex are auto, Cisco switches also support auto-MDIX. (The switch will adjust the port to be straight through or crossover as needed.)

```
Interface f0/0
  Mdix auto
```

# Spanning Tree

Redundancy is a common technique to increase availability in computer networks. Ethernet redundancy would look like multiple core switches and multiple paths between workgroup switches and the core. Of course, multiple paths mean loops, and Ethernet lacks a mechanism for dealing with loops.

Spanning Tree is a protocol that detects potential loops and breaks them:

1. Each switch advertises Bridge Protocol Data Units (BPDU) that periodically announces name (bridge ID), current root, and cost to the root. Each switch starts believing it is the root.
2. If a switch receives a BPDU with a different root, it compares roots. If the received BPDU has a lower root, the switch changes root and recalculates cost to the root. The port that received the superior BPDU is the **root port**—the port that leads to the root. Other ports are **designated ports**—ports leading away from the root.

   Each link has a cost based on its speed, as shown in the following table.

   | Link Speed | Cost |
   |---|---|
   | Ethernet | 100 |
   | Fast Ethernet | 19 |
   | Gigabit Ethernet | 4 |
   | Ten Gigabit Ethernet | 2 |

3. If a switch receives two BPDUs with the same root but different costs, it uses the lower cost port. The port with the higher cost is blocked (it filters all traffic except BPDUs) to prevent a loop. Blocked ports are also called **non-designated**.

At the end of the process there will be one root bridge. Each nonroot switch will have one root port.

Spanning tree status can be seen using the **show spanning-tree [vlan vland-id]** command, as shown here:

```
Newton-Sw01# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    8192
             Address     001d.4664.7d01
             Cost        4
             Port        641 (GigabitEthernet6/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     001d.46c8.ac01
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 300

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- ----------------------
Fa1/2            Desg FWD 19        128.2    Edge P2p
Fa1/3            Desg FWD 19        128.3    Edge P2p
Fa1/4            Desg FWD 19        128.4    Edge P2p
Fa1/5            Desg FWD 19        128.5    Edge P2p
Fa1/7            Desg FWD 19        128.7    Edge P2p
Fa1/9            Desg FWD 19        128.9    Edge P2p
Fa1/10           Desg FWD 19        128.10   Edge P2p
Fa1/11           Desg FWD 19        128.11   Edge P2p
Fa1/12           Desg FWD 19        128.12   Edge P2p
…
```

The details of received BPDUs can be seen using **show spanning-tree interface [interface] detail**. This command shows root status, cost, and timers:

```
Newton-Sw01# show spanning-tree vlan 1 detail

 VLAN0001 is executing the rstp compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 001d.4664.cc01
  Configured hello time 2, max age 20, forward delay 15, tranmsit
 hold-count 6
  Current root has priority 8192, address 001d.4632.6c01
  Root port is 641 (GigabitEthernet6/1), cost of root path is 4
  Topology change flag not set, detected flag not set
  Number of topology changes 119 last change occurred 25w6d ago
          from GigabitEthernet6/1
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 300

 Port 2 (FastEthernet1/2) of VLAN0001 is designated forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.2.
   Designated root has priority 8192, address 001d.4664.ec01
   Designated bridge has priority 32768, address 001d.4664.cc01
   Designated port id is 128.2, designated path cost 4
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode
   Link type is point-to-point by default
   Bpdu guard is enabled
   Root guard is enabled on the port
   BPDU: sent 9120, received 0
…
```

Before spanning-tree, loops meant that traffic would cycle continuously. Over a short time traffic would accrete in the loop until it consumed all capacity. This is called a broadcast storm. *Broadcast storms* are still a real danger, but spanning tree has mitigated this almost entirely. The danger today is that—through protocol failure or administrative misprogramming—when a broadcast storm forms, few administrators have seen it before and know how to deal with it.

A broadcast storm can be diagnosed when the switches become saturated with traffic. All the traffic lights will be solid, the switch will be slow to respond, and users will complain about network speed.

The only fix for a broadcast storm is to break the loop. If the switches are accessible, it might be possible to fix spanning-tree. Otherwise, the administrator must manually remove redundant links.

As previously stated, the purpose of spanning-tree is to select one root path and filter all others. When there are multiple links between two switches it seems intuitive that, rather than turn one off, the switches should use all the links together. This is possible using Etherchannel.

Etherchannel logically combines several physical links between switches and spanning tree treats the bundle as a single port. Up to eight physical lines may be combined in this way.

Etherchannel failures cluster into three groups:

- All ports must be identical (speed, duplex, access or trunk, VLAN). If Etherchannel will not form, look for inconsistencies between ports.
- Both switches must either be configured or a link aggregation protocol (LACP or PAgP) must be used. If only one side is configured for Etherchannel, look for Etherchannel ports that are error-disabled.
- The channel might form, but traffic might still be traveling predominately over a single link. This is because traffic is statistically multiplexed using a three-bit hash. This means that the traffic is split over eight paths, and an etherchannel of three links will split the load in a

2:1:1 ratio. Fix this by using 2, 4, or 8 links. Second, the hash uses a user-selectable Ethernet or IP field. If all traffic comes from a single source and the switch is hashing on source MAC, it will not multiplex. Fix this by selecting a different hashing method.

# VLANs

Virtual LANs are logical broadcast domains, administratively assembled from component ports on the switches in the network. Switches are interconnected by Ethernet lines that use 802.1q, a shim header inserted in the Ethernet frame. 802.1q adds a two-byte shim, 12 bits of which are used to identify the VLAN and three bits of which are used to specify Layer-2 class of service. (This is called the 802.1p subfield.)

When troubleshooting VLaN switching issues, concentrate on three types of failure:

- **Wiring issues:** Cabling issues, power outage, or bad switch ports
- **Switch issues:** Software bugs, hardware bugs, loops, and ARP issues
- **Logic issues:** Misconfigured VLANs, VTP, trunks, and native VLAN mismatch

Troubleshooting switches often involves using these tables to understand the path traffic takes through the switch. Two commands can help identify the path taken:

- **Show platform forward:** Displays forwarding info from TCAM
- **Traceroute mac:** Shows intermediate MACs from source to destination

Switches keep several mapping tables. Each of these tables is shown in the following table, as well as the IOS command to examine the table.

| Table | IOS Command |
|---|---|
| MAC Address Table: Maps MAC addresses to ports | **Show mac-address** |
| VLAN assignments: Maps VLANs to ports<br>**Show interface switchport** | **Show vlan** |
| VLAN Database: Maps names to VLANs | **Show vlan** |
| Trunk assignments<br>**Show interface trunk**<br>**Show etherchannel** | **Show interface switchport** |

## Switched Virtual Interfaces and InterVLAN routing

Routing between VLANs can be accomplished on a Layer 3 switch or on a router. Troubleshooting the control plane (the Layer 3 structures) is identical between the two. This means that OSPF runs identically on the two platforms.

The data plane (the structures and hardware that handle frame forwarding) is different between routers and Layer 3 switches. In both cases, **show ip cef** shows the cef forwarding table, and **show adjacency** shows the Layer 2 headers used in forwarding.

Catalyst 3560, 3750, and 4500 switches can also use **show platform** to see detailed forwarding information.

Catalyst 6500 switches display forwarding details using **show mls cef** commands.

Another difference between routers and Layer 3 switches, in the context of troubleshooting intervlan routing, is the concept of an SVI (Switched Virtual Interface).

Routers forward traffic between ports using Layer 3 information.

Layer 3 switches can have multiple ports in the same vlan and pass traffic between them using MAC information. Layer 3 switches also support SVIs (these look like interface vlan 1) that act as virtual layer-3 ports for a VLAN. Finally, a switch can treat a port as a separate routed port.

From a troubleshooting perspective, routed ports do not run switching protocols like Spanning Tree or Etherchannel. SVIs, on the other hand, are extremely stable. An SVI changes only state to down when all the VLAN ports are down.

## First-Hop Redundancy

Hosts are configured with a default gateway—a router address that will pass traffic off the local subnet. The problem is that router failures strand the hosts. The solution is first-hop redundancy protocols, which enable two routers to cooperatively support a single IP, which can then be given to hosts as a default gateway.

There are three first-hop redundancy protocols:

- HSRP is an older Cisco proprietary protocol. One router is the active and one is the standby. The routers pass keepalives that enable the standby to recognize failure of the primary router.

- VRRP is an open standard but is otherwise similar to HSRP. Because HSRP works, many organizations have continued to use HSRP.
- GLBP is an open standard, but it enables simultaneous load balancing over as many as four gateways.

Because HSRP is the most common, this section focuses on HSRP. The general configuration and troubleshooting strategy applies well to VRRP and GLBP, however.

HSRP is configured under the interface using **standby** commands. Routers in the same HSRP group share a Mac and IP, so standby is used to identify the group and virtual IP.

By default, each HSRP speaker has a priority of 100. The speaker with the highest priority is the active router. If a new router starts however, HSRP does not change the active router until the failure of the active router. To change this so that the higher priority is instantly recognized, use the **preempt** command. An HSRP snippet is shown here to illustrate the configuration:

```
Interface f0/0
Ip address 10.1.1.2 255.255.255.0
Standby 2 ip 10.1.1.1
Standby 2 priority 120
Standby 2 preempt
```

Verify the HSRP state of a router using **show standby**, which summarizes this information to a table (an example is shown next). To see detailed information on HSRP, such as timers and virtual MAC, use **show standby interface**:

```
Maiden-rtr01# show standby
GigabitEthernet0/1 - Group 135
  State is Active
    23 state changes, last state change 25w6d
  Virtual IP address is 135.159.64.1
  Active virtual MAC address is 0000.0c07.ac87
    Local virtual MAC address is 0000.0c07.ac87 (v1 default)
  Hello time 5 sec, hold time 20 sec
    Next hello sent in 0.284 secs
  Preemption enabled
  Active router is local
  Standby router is unknown
  Priority 150 (configured 150)
  Group name is "hsrp-Gi0/1-135" (default)
Richardson-rtr01# show standby interface gi0/1
```

```
Global           Confg: 0000
Gi0/1 If hw      BCM1125 Internal MAC (27), State 0x210040
Gi0/1 If hw      Confg: 0000
Gi0/1 If hw      Flags: 0000
Gi0/1 If sw      Confg: 0000
Gi0/1 If sw      Flags: 0000
Gi0/1 Grp 135    Confg: 0072, IP_PRI, PRIORITY, PREEMPT, TIMERS
Gi0/1 Grp 135    Flags: 0000

HSRP virtual IP Hash Table (global)
103 172.25.96.1    Gi0/1      Grp 135

HSRP MAC Address Table
43 Gi0/1 0000.0c07.ac87
    Gi0/1 Grp 135
```

**show standby brief** is mirrored with **show vrrp brief** and **show glbp brief**. Similarly, **show standby interface** and **debug standby** have equivalents for the other first-hop redundancy protocols.

# CHAPTER 5

# Troubleshooting Routing

This section reviews troubleshooting for common routing protocols. A more theoretical explanation of the working of the protocols is available in the *BSCI Quick Reference Guide*.

## Network Layer Connectivity

Routers use three tables to make routing decisions: the routing table, ARP table, and CEF mappings

The routing table is visible using **show ip route**. Each entry in the routing table has an output interface or next hop. Packets are routed per the routing table, matching the longest prefix match first and then by other metrics determined by that IGP's algorithm.

When a determination of the next hop has been made, the router needs to turn this information into a destination Layer 2 address. For this purpose, mapping tables are maintained that match Layer 2 and Layer 3 addresses. The ARP table (**show ip arp**) and the frame-relay map (**show frame-relay map**) are examples of this.

Cisco Express Forwarding (CEF) is the common switching method found on most Cisco gear. CEF combines information from the routing table and the various mapping tables to optimize routing and to optimize the construction of new Layer 2 headers. CEF entries may be viewed using **show ip cef** and associated commands.

## Routing Protocols

Routing protocols are mechanisms that enable routers to share information about the structure of the network. Regardless of the protocol, troubleshooting routing protocol issues have some basic logic that is true for any routing protocol. Troubleshooting routing issues always starts with looking at the routing table. Use **ping** to test connectivity, **show ip route** to inspect the routing table to see if the route is present, and **traceroute** to inspect how traffic is forwarding. **show ip protocols** displays information about the current routing protocols, such as autonomous system and timer values.

Troubleshooting routing issues can be summarized by answering three basic questions:

1. Is the correct route advertised?
2. Is the correct route communicated?
3. Is there a more desirable path (lower AD or longer prefix length)?

# EIGRP

After determining that there is a routing problem in EIGRP using the routing table or ping, follow the three basic steps to troubleshooting.

EIGRP stores information in three tables that can be interrogated.

| Table | Command |
|---|---|
| Interface table: Lists EIGRP-enabled interfaces | **Show ip eigrp interface** |
| Neighbor table: Lists discovered neighbors | **Show ip eigrp neighbors** |
| Topology table: Complete list of received EIGRP routes | **Show ip eigrp topology** |

## Is the Correct Route Advertised?

Verify that the router attached to the destination subnet is advertising the route. There are several ways to see the advertised subnets; two good ways are either direct interrogation of the running configuration using **show running-config | section eigrp** or by reviewing the protocol settings using **show ip protocol** (shown here):

```
Hickory-rtr01# show ip protocol
Routing Protocol is "eigrp 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 100, bgp 65096
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    GigabitEthernet0/1
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.4.254           90      00:39:11
    10.1.4.253           90      00:38:55
  Distance: internal 90 external 170
```

EIGRP also advertises only subnets of interfaces that match a network statement.  **show ip protocol** provides the matching network statements.

## Is the Correct Route Communicated?

EIGRP shares only routes with neighbors—devices with which it has exchanged hellos. Verify that connected devices are neighbors using **show ip eigrp neighbors**. **debug ip eigrp packets** should show hellos and updates if devices are connected, and **debug ip eigrp** should show details about the contained routing information communicated.

EIGRP neighborship requires bidirectional communication, authentication, that the AS be the same, and that timers are close to the same. EIGRP also sends only hellos over interfaces that match a network statement. If a router hasn't identified a link as an EIGRP link in this way, it will not send hellos and it will not form neighborship. EIGRP values, such as timers, and a list of EIGRP interfaces is available through **show ip eigrp interfaces**:

```
Hickory-rtr01# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address          Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                     (sec)         (ms)       Cnt Num
1   10.1.4.253      Gi0/0             14 2w0d        1   200  0  1797
0   10.1.4.254      Gi0/0             14 2w0d        1   200  0  729
Hickory-rtr01# show ip eigrp interface
IP-EIGRP interfaces for process 100

                  Xmit Queue   Mean   Pacing Time   Multicast  Pending
Interface  Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer Routes
Gi0/0        2        0/0        1        0/1           50        0
Lo0          0        0/0        0        0/1            0        0
```

If the devices are neighbors, routes could be blocked using distribution lists or route-maps. Distribution lists would be listed in **show ip protocol**.

## Is There a More Desirable Path?

Finally, if the route is not in the routing table, use **show ip eigrp topology** to see if the route is known to EIGRP. It could be that the route is known, but there is a more desirable path. **show ip route** shows only the selected EIGRP route. To see all known EIGRP routes, use **show ip eigrp topology**.

# OSPF

Three OSPF tables can be reviewed in troubleshooting. A fourth—the Routing Information Base—is used to store SPF calculations but is largely unavailable to the administrator.

| Table | Command |
|---|---|
| Interface table: Lists OSPF-enabled interfaces | **Show ip ospf interface** |
| Neighbor table: Lists discovered neighbors | **Show ip ospf neighbors** |
| Link State Database: LSAs received | **Show ip ospf database** |

If a routing problem exists in OSPF, follow the same basic steps to troubleshooting.

## Is the Correct Route Advertised?

Verify that the router attached to the destination subnet is advertising the route. Advertised subnets are visible using either **show running-config | section ospf** or by reviewing **show ip protocol**.

OSPF also limits advertisements to the subnets of interfaces that match network statements. **show ip protocol** provides the matching network statements. **show ip ospf statistics** can also help by showing how often SPF is running, potentially showing network instability.

## Is the Correct Route Communicated?

OSPF shares routes with neighbors. Verify that connected devices are neighbors using **show ip ospf neighbors**. **show ip ospf database** displays the link state information. **debug ip ospf adj** should show issues preventing neighborship.

OSPF neighborship requires six parameters to agree:

- Bidirectional communication.
- Equal timer values.

- Matching AS number.
- Routers must agree on the type of their common area.
- Routers must agree on the prefix of their common subnet.
- Authentication, if used, must agree on type and password.

OSPF sends only Hellos over interfaces that match a network statement. If a link does not match a network entry, no Hellos will be transmitted and no neighbors will form over the link. OSPF protocol values can be seen using **show ip ospf interfaces**.

If the devices are neighbors, routes could be blocked at boundary routers using distribution lists or route-maps. Distribution lists would be listed in **show ip protocol**.

## Is There a More Desirable Path?

It is possible that OSPF has chosen an unexpected path to a destination. It could also be that routes from other routing protocols are present with a lower administrative distance or that an intermediate system has a static route. Checking routing tables along the expected path is the best way to reveal this.

# BGP

BGP maintains two tables outside of the routing table, one for neighbors and one for BGP routing information.

| Table | Command |
|---|---|
| Neighbor table: Lists neighbors | **Show ip bgp neighbors** |
| BGP table: Contains all received BGP prefixes and associated attributes, as well as showing the BGP best path | **Show ip bgp** |

BGP troubleshooting can also follow the three basic steps.

## Is the Correct Route Advertised?

Verify that the router attached to the destination subnet is advertising the route. This can be seen from the running configuration (**show running-config | section bgp**) or the BGP table (**show ip bgp**—self-originated routes have a next hop of 0.0.0.0).

BGP advertises only explicitly identified prefixes for which there is a matching route from another source (like a connected route).

## Is the Correct Route Communicated?

BGP communicates prefixes with administratively defined neighbors. Verify that defined neighbors are reachable using ping and that they are neighbors by reviewing **show ip bgp neighbors**. A partial output from this is shown next—**show ip bgp neighbors** includes considerable detail. **debug ip bgp updates** should show hellos and advertisements, and **debug ip bgp** should show details about the contained routing information being communicated:

```
Hickory-rtr01# show ip bgp neighbor
BGP neighbor is 10.1.255.5,  remote AS 4800, external link
  BGP version 4, remote router ID 59.43.0.71
  BGP state = Established, up for 2w0d
  Last read 00:00:15, last write 00:00:17, hold time is 90,
 keepalive interval i                      s 30 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                2       1162
    Keepalives:         40808      40817
    Route Refresh:          0          0
    Total:              40811      41980
  Default minimum time between advertisement runs is 30 seconds
…
```

BGP neighborship requires bidirectional communication, authentication, and that the AS match the expected AS. BGP values, such as timers and AS, are available through **show ip bgp**.

If the devices are neighbors, routes could be blocked using distribution lists or route-maps. Distribution lists would be listed in **show ip protocol**.

## Is There a More Desirable Path?

If the route is not in the routing table, use **show ip bgp** to see if the route is known and valid. Routes can be invalidated if the BGP next hop is unreachable; if so routing to this address must be recursively troubleshot. The following partial example shows several routes that are **valid** and **best**, shown by the preceding ***>**.

```
ahk-rtr01# sh ip bgp
BGP table version is 17312, local router ID is 10.254.254.12
Status codes: s suppressed, d damped, h history, * valid, > best, i
  - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          182.225.207.13             0 65000 65097 i
*> 10.43.0.0/24     182.225.207.13             0 65000 65086 65042 i
*> 10.43.0.0/22     182.225.207.13             0 65000 65086 65042 i
*> 10.45.128.0/24   182.225.207.13             0 65000 65100 65044 i
*> 10.49.0.0/22     182.225.207.13             0 65000 65086 65300 i
*> 10.61.0.0/16     182.225.207.13             0 65000 65060 i
*> 10.63.0.0/20     182.225.207.13             0 65000 65062 i
*> 10.65.0.0/19     182.225.207.13             0 65000 65064 i
*> 10.71.0.0/16     182.225.207.13             0 65000 65086 65302 i
*> 10.87.0.0/16     182.225.207.13             0 65000 65086 i
…
```

## Route Redistribution

Organization sometimes must support more than one routing protocol. For example, a business might use EIGRP within a campus and BGP over the MPLS WAN. Routing information is passed between the protocols using redistribution. Redistributed routes are treated as external in the receiving protocol.

Redistribution extracts routes from the routing table, so only routes that appear in the routing table will be exported. If routes are not present, confirm the routes are present in the routing table at the redistribution point. You need to identify and understand the interaction of all redistribution points. Creating a routing loop through multiple redistribution points is quite possible.

Because routing protocols use different metrics, redistributed routes lose routing information. Distance Vector routing protocols, including EIGRP, assume that the metric for imported routes should be infinity unless another value is specified. When redistributing into EIGRP, a default metric must be set or no routes will be imported! OSPF will import only classful routes unless **redistribute subnets** is used, so this is also a point to review in troubleshooting.

In addition to protocol specific commands, **debug ip routing** can show routes as they are added or withdrawn from the routing table.

If **ip route profile** is added to the config, the **show ip route profile** command shows routing table changes over consecutive 5-second intervals. This is particularly helpful to show that routes are flapping—being added and withdrawn continuously.

## Router Performance

Routing protocol performance can be symptomatic of general router problems. Routing protocol problems can be seen if the router CPU is overburdened or memory is fully utilized.

Transient events, such as SNMP communication or a heavy traffic load, can temporarily spike the CPU. High CPU utilization is a concern when it becomes on-going. Signs of CPU oversubscription include dropped packets, increased latency, slow response to telnet and console, and when the router skips routing updates.

**Show process cpu** can identify processes that are consuming CPU cycles. The ARP Input process consumes more cycles if the router has to generate a large number of ARPs, for instance in response to malicious traffic. Net Background is used to manage buffer space. IP Background is used whenever an interface changes state, utilization here could indicate a flapping interface.

**Show process cpu history** displays the overall utilization as a bar graph. This is a nifty way to see if the current load is an aberration or the norm.

A second general router issue is the router switching mode. There are three common modes:

- Process switching uses the CPU to process each packet. Process switching is CPU-intensive and reduces throughput and increases jitter. It is turned on by using **no ip route-cache**.
- Fast switching uses the CPU to process an initial packet but then caches the result. It is less CPU-intensive, but utilization still tracks the traffic load. It is turned on using **ip route-cache**, and the cache can be reviewed using **show ip cache**.
- Cisco Express Forwarding (CEF) is the default switching mode. CEF is resilient to traffic load. It is turned on using **ip cef**, and CEF entries can be seen by using **show ip cef** and **show adjacency**. CEF is required for some IOS features, such as NBAR, WRED, and AutoQoS.

The interface switching mode is shown from the **show ip interface** command.

A third general router issue is router memory utilization. Memory is overused when there is no available system memory or when the memory is too fragmented to be useful.

One easy, but not pleasant, way to see a memory problem is to load a version of IOS that requires more RAM than is present on the router. Memory can also be depleted by a memory leak—a bug that assigns memory to processes but does not clean up when the process is complete. Memory leaks can be recognized over time using **show memory allocating-process totals** and **show memory dead** and by researching known bugs within CCO. If found, the only solution is to move to a known good version of IOS.

Memory leaks sometimes appear on interfaces as buffer leaks. Buffer leaks can be seen using **show interface**, where the "input queue" shows buffer utilization. **Show buffer** also shows a buffer leak, here by looking at the number of free buffers.

Finally, memory leaks are sometimes seen in BGP, which is a heavy consumer of memory in the best of times, so a memory leak here can quickly bloom into a larger issue. **show process memory | include bgp** shows the memory utilization of the four BGP processes. **show diag** can be used to evaluate memory used on the line cards.

CHAPTER 6

# Troubleshooting Security Features

Network security has been seen as a separate function, but security has evolved to be a pervasive element. Routers are both potential targets for attacks and platforms that can offer security services.

Network devices have three types of functions and traffic, all of which are affected by security concerns:

- Management plane: The functions involved in management, such as device access, configuration, and telemetry.
- Control plane: The functions spoken between network devices, such as routing protocols.
- Data plane: Packet forwarding functionality.

Security for the management plane means controlling all the means of accessing the device and making configuration changes. Common security steps for various protocols include

- Console: Physically secure access to the device and set reasonable time-outs. Use password protected modems for out-of-band access, and control authentication centrally with RADIUS or TACACS+ to regularly change passwords.
- Telnet/SSH: Limit use of telnet because it transmits usernames and passwords in the clear. Limit telnet access using access-lists to predefined IPs. Use SSH instead.
- HTTP/HTTPS/SNMP: Centralize authentication and limit access to predefined IPs. Disable if not used.

Many control plane protocols, such as EIGRP, OSPF, HSRP, and GLBP, include peer authentication based on MD5 hashing. Vulnerabilities in ARP and DHCP can be addressed with switch capabilities to inspect and deal with maliciousness. DHCP snooping observes responses to ensure they come from the server, whereas Dynamic ARP Inspection looks for and blocks spoofed ARP responses. Likewise, spanning-tree protection is available based on an understanding of the topology using technologies such as root guard and BPDU guard. The router can also protect against maliciousness by performing reverse path checking—making sure that packets arrive on the interface that would be used to route the reply.

The data plane is secured by controlling access, visibility, and flow. Keeping unauthorized users off the network is the role of network access control and 802.1x. Encryption and VLANs can be used to isolate traffic and prevent interception. Finally, traffic flows can be limited and inspected using access-list, flexible packet matching, IOS Firewall, and Intrusion Prevention Systems. IP source tracker allows for an easier, scalable solution to tracking DoS attacks compared to the traditional ACL. Zone-based security firewalls permit you to get granular in inspection and well-defined interface-based zone pairings to specify what traffic is permitted.

The IOS Firewall is easy to set up. An access-list is used to block all nonapproved traffic. Context-based access control(CBAC) is then used to modify the access-list, as replies to all outbound connections are allowed:

```
Ip access-list extended block
  Deny ip any any
Ip inspect name CBACInt f0/0
  Ip access-group block in
  Ip inspect CBAC out
```

# Troubleshooting Security Features

The key issue with security features is that they limit traffic to create a security policy. This can work against the natural flow of troubleshooting, where the focus is on allowing communication. The issue is to recognize how the security policy compares to troubleshooting steps and to always work within the organizations change control system.

Troubleshooting the management plane, specifically authentication, can be tricky because it is possible to lock yourself out. The best approach is to have a backup plan to access the router—out-of-band access, a user to reset power, or a second authentication method. If no one is onsite, use the **reload in 10** command to schedule a reboot in 10 minutes before beginning work. It is also a good idea to allow local authentication (shown next) so that if access-list changes block access to RADIUS or TACACS+ there is still a way to login:

```
Aaa authentication default group tacacs+ local
Username brent password denise
```

SNMP uses UDP 161, and access-list blocking can be tested using extended traceroute on that port. SNMP can also be set up with access-lists and authentication to control access. Temporarily lifting these might also provide insight into any problems.

Troubleshooting the control plane comes down to neighbors. If a routing protocol doesn't see a directly connected peer, the problem is either a protocol issue or a firewalling issue. To verify that protocol traffic is passing, consider using **debug** to witness hellos (**debug ip eigrp packets**), or use the router as a protocol analyzer by using **debug ip packet** ***access-list***. (The access list limits **debug** to just the traffic of interest.) The following example shows this done to analyze BGP traffic:

```
(config)# Ip access-list 101 permit tcp any any eq 179
Debug ip packet 101
```

The data plane includes support for user applications. Testing access can be accomplished with traceroute and telnet. Traffic is usually controlled using access-lists, so another way to troubleshoot connections is to log access-list matches. Access-list logging forces traffic to be processor switched and should be used in a limited manner. (Matches can be limited by narrowly crafting permit statements or though the established keyword, for instance.). ACL matches are forwarded to Syslog with this option, so used sparingly it is a good way to understand which line in the access-list is disposing of traffic. To set up logging, add the keyword log onto a ACL line. To see the denied traffic at the end of a list, for instance, add the following line to your ACL:

```
Deny ip any any log
```

# INDEX

## NUMBERS

**3DES, 119**

**802.1Q**
- double-tagging attacks, 195
- VLAN identification, 141

## A

**ABR (Area Border Routers), OSPF, 33**

**Access layer (Hierarchical Design Model), 2, 129**

**Access mode (DTP), 142**

**ACK (Acknowledgement) packages (EIGRP), 15**

**ACL (Access Control Lists)**
- crypto ACL, 89, 123
- interface ACL, 124
- VACL, 195-196

**Active (OpenSent) state (BGP peering), 75**

**Active mode (LACP), 148**

**Active Process Enhancement (EIGRP), 30**

**AD (Administrative Distance)**
- EIGRP, 16-17
- routing and, 9, 56-57

**address pools, mobile network connectivity implementation plans, 96**

**address translation, IPv4 and IPv6 integration, 113**

**ADSL (Asymmetrical DSL), branch office implementation plans, 86**

**AES (Advanced Encryption Standard), 120**

**AH (Authentication Headers), IPsec, 117**

**air conditioning, troubleshooting, 246**

**alert logging, maintenance, 224**

**Align-Err (alignment errors), troubleshooting, 252**

**AP (Access Points)**
- autonomous (stand-alone) AP, 213
- CAPWAP, 214
- H-REAP, 216
- lightweight AP, 214-215
- LWAPP, 214

**application servers, VoIP, 201**

**Archive and Restore feature, maintenance, 225-227**

**ARP (Address Resolution Protocol)**
- DAI, 199
- spoofing attacks, 198

**AS (autonomous systems), BGP, 69-70**

**ASBR (Autonomous System Boundry Routers), OSPF, 33**

**asymmetric key algorithms, IPsec encryption, 120**

**authentication**
- BGP, 79
- EIGRP, 27-28
- IPsec, 119
- mobile network connectivity, 93
- OSPF, 46-47
- port-based authentication, campus networks, 193-194
- security, 270

**Auto mode (PAgP), 147**

**auto-MDIX, 253**

**autonomous (stand-alone) AP (Access Points), WLAN, 213**

**AutoQoS, VoIP, 207**

## B

**backbone area (OSPF), 32**

**backbone routers (OSPF), 33**

**BackboneFast, STP, 160**

**backups, DSL connections, 85-89**

**bandwidth**
- EIGRP over WAN, 24
- mobile worker connectivity requirements, 93
- VoIP requirements, 202

**baselines (maintenance), 234**

**BDR (backup DR) routers (OSPF), 41**

**BGP (Border Gateway Protocol), 10**
- AS, 69-70
- authentication, 79
- configuring, 74-75
- databases, 70-71
- EBGP, 71
- filtering routes, 78

IBGP, 71
implementation plans, 67
*dual-homed ISP connections, 68*
*multihomed ISP connections, 69*
*route options, 68*
*single-homed ISP connections, 68*
keepalive messages, 71
MP-BGP, IPv6 routing, 109
next-hop selection, 72-73
notification messages, 71
open messages, 71
overview of, 69-70
path selection
*attributes, 76*
*influencing path selection, 78*
*route maps, 78*
*selection criteria, 77-78*
synchronization rule, 73
troubleshooting, 265-266, 269
update messages, 71
verifying, 80-82

**BPDU (Bridge Protocol Data Units), 154-155**
BPDU Guard, STP, 160, 191
filtering, STP, 161
RSTP, 157

**Branch Architecture component (Cisco Enterprise Architecture), 7**

**branch offices. *See also* mobile network connectivity**
design considerations
*large branch offices, 84*
*medium-sized branch offices, 84*
*small branch offices, 83*
implementation plans
*backup DSL connections, 85-87, 89*
*dynamic routing over GRE tunnels, 90-91*
*floating static routes, 90*
*EIGRP load sharing, 92*
*IPsec VPN, 89*
*verifying exisiting services, 85*

**bridging**
DSL, 87
bridging loops, identifying, 163

**broadcast networks, OSPF, 40**

**broadcast storms, troubleshooting, 256**

## C

**Campus Architecture component (Cisco Enterprise Architecture), 7**

**campus networks**
hierarchical design model
*access layer, 129*
*core layer, 130-133*
*distribution layer, 129*
network implementation plans, 135
security, 191
*802.1Q double-tagging attacks, 195*
*ARP spoofing attacks, 198*
*DHCP spoofing attacks, 198*
*IP Source Guard, 198*
*MAC address flooding attacks, 192*
*port security, 192-193*
*port-based authentication, 193-194*
*protected ports, 197*
*PVLAN, 196-197*
*switch spoofing attacks, 194*
*switches, 199*
*VACL, 195-196*
SONA, 134
Video over IP, 208
VoIP, 200
*bandwidth requirements, 202*
*configuring VoIP support on switches, 206-207*
*network requirements, 202*
*PoE switches, 203*
*QoS, 204-207*
*system components, 201*
*system requirements, 202*
*Voice VLAN, 203*
WLAN
*autonomous (stand-alone) AP, 213*
*characteristics of, 211*
*Cisco Unified Wireless Networks, 210*
*client connectivity, 212*
*controllers, 215*
*H-REAP, 216*
*LAN integration, 217-218*
*lightweight AP, 214-215*
*SSID, 211*
*topologies of, 211-212*

**CAPWAP (Control and Provisioning of Wireless Access Points), 214**

**CEF (Cisco Express Forwarding), 170-172, 261, 268**

**change control, maintenance, 221, 234**

**Cisco Easy VPN (Virtual Private Networks), mobile network connectivity implementation plans, 95-97**

**Cisco Enterprise Architecture, complex networks, 7**

**Cisco Lifestyle Services, routing implementation plans, 12**
**Cisco Unified Wireless Networks, 210**
**Client role (VTP), 144**
**collisions, troubleshooting, 252**
**complex networks**
Cisco Enterprise Architecture, 7
Enterprise Composite Model, 2-3
Hierarchical Design Model, 1-2
IIN, 8
QoS, 7
routing
*AD, 9*
*BGP, 10*
*building routing tables, 11*
*choosing routes, 11*
*EIGRP, 10*
*implementation plans, 12-13*
*OSPF, 10*
SONA, 9
virtualization, 8
**Configuration BPDU (Bridge Protocol Data Units), 154**
**configuration histories, maintenance, 223-224**
**Connect state (BGP peering), 75**
**connection security, mobile worker connectivity, 93**
**Control plane (networks), security, 270-272**
**Core layer (Hierarchical Design Model), 2, 130**
data centers, 131
medium campus networks, 131
network traffic flows, 133
small campus networks, 131
**corporate security, mobile worker connectivity, 93**
**crypto ACL (Access Control Lists), 89, 123**
**crypto maps**
IPsec crypto maps, 89, 124
mobile network connectivity implementation plans, 97
**customizing**
EIGRP configurations
*passive-interface command, 25*
*summarization, 26*
*unicast neighbors, 25*
OSPF configurations
*authentication, 46-47*
*default routes, 44*
*NSSA, 44*
*passive-interface command, 43*
*stub areas, 44*
*summarization, 43*
*Totally Stubby Areas, 44*
*virtual links, 45*

## D

**DAD (Duplicate Address Discovery), NDP, 103**
**DAI (Dynamic ARP Inspection), 199**
**data centers**
core layer, 131
Data Center component (Cisco Enterprise Architecture), 7
**Data plane (networks), security, 270-272**
**databases (BGP), 70-71**
**DBD (Database Description) packets (OSPF), 36**
**defining the problem (troubleshooting methodology), 232**
**deny statements, controlling route redistribution, 50**
**DES (Data Encryption Standard), 119**
**designated port elections (STP), 153**
**Desirable mode (PAgP), 147**
**DHCP (Dynamic Host Configuration Protocol)**
snooping, 270
spoofing attacks, 198
**diagnostics (hardware), 246, 248-249**
**Diffie-Hellman Key Exchange protocol, IPsec encryption, 120**
**distribute lists, 54**
**Distribution layer (Hierarchical Design Model), 2, 129**
**DMVPN (Dynamic Multipoint Virtual Private Networks), 90**
**documentation**
HA, 175
maintenance, 221-222, 228
routing implementation documentation, creating, 13
updating (troubleshooting methodology), 233
**double-tagging attacks (802.1Q), 195**
**downtime, maintenance and, 221**
**DR (Designated Routers), OSPF, 41**

**DSL (Digital Subscriber Lines)**
- ADSL, 86
- backup connections, 85-89
- branch office implementation plans, 85-89
- bridging, 87
- G.SHDSL, 86
- HDSL, 86
- PPPoA, 87
- PPPoE, 87
- SDSL, 86
- troubleshooting, 89
- VDSL, 86
- verifying, 89

**DTP (Dynamic Trunking Protocol), 142**

**DUAL (Diffusing Update Algorithm), EIGRP, 17-18**

**dual-homed ISP connections, BGP, 68**

**duplex mismatches, troubleshooting, 252-253**

**Dynamic Auto mode (DTP), 142**

**Dynamic Configuration tool, maintenance, 228**

**Dynamic Desirable mode (DTP), 142**

**Dynamic NAT-PT (Network Address Translation-Protocol Translation), IPv4 and IPv6 integration, 113**

**dynamic routing over GRE tunnels, branch office implementation plans, 90-91**

## E

**Easy VPN, mobile network connectivity implementation plans, 95-97**

**EBGP (External Border Gateway Protocol), 71**

**EEM (Embedded Event Manager), 251**

**EGP (Exterior Gateway Protocol), AS, 69**

**EIGRP (Enhanced Interior Gateway Routing Protocol), 10, 14**
- ACK packages, 15
- authentication, 27-28
- configuring, 20
- customizing configurations
  - *passive-interface command, 25*
  - *summarization, 26*
  - *unicast neighbors, 25*
- default routes, creating, 20-21
- EIGRP across WAN
  - *bandwidth limitations, 24*
  - *EIGRP over EoMPLS, 21-22*
  - *EIGRP over Frame Relay, 23-24*
  - *EIGRP over MPLS, 22*
- Hello packages, 15
- implementation plans, 19-20
- IPv6 routing, 106-107
- load balancing, 26
- load sharing, branch office implementation plans, 92
- neighbor discovery process, 15-16
- Query packages, 15
- Reply packages, 15
- route exchange process, 15-16
- route selection
  - *AD, 16-17*
  - *DUAL, 17-18*
  - *EIGRP metric, 16-17*
  - *example of, 18*
  - *FD, 16-17*
- scalability, 29-30
- troubleshooting, 21, 262-264
- Update packages, 15
- verifying, 21

**encryption, IPsec, 119-120**

**end-to-end VLAN (Virtual Local Area Networks), 137**

**Enterprise Campus section (Enterprise Composite Model), 3**

**Enterprise Composite Model, complex networks, 2-3**

**Enterprise Edge section (Enterprise Composite Model), 3**

**EoMPLS (Ethernet over MPLS), EIGRP over EoMPLS, 21-22**

**ESP (Encapsulating Security Payload), IPsec, 118**

**Established state (BGP peering), 75**

**EtherChannels, 146**
- configuring, 147-148
- failures, troubleshooting, 256
- verifying, 148

**Ethernet, PoE switches, 203**

**event logging, maintenance, 224**

**Excess-Col, troubleshooting, 252**

## F

**fast failover, HA, 176-177**

**fast switching, 170, 268**

**FCAPS (Fault, Configuration, Accounting, Performance, and Security), routing implementation plans, 12**

**FCS-Err (Frame Check Sequence errors), troubleshooting, 252**

**FD (Feasible Distance), EIGRP, 16-17**

**Feature Navigator, maintenance, 228**

**filtering, IOS filtering tools, 235-236**
- common regular expressions, 237
- generic filters, 237
- piping output, 237-242

**firewalls (IOS), 271**

**first hop redundancy, 258-260**
- GLBP, 189-190
- HSRP, 185-187, 190
- router implementation plans, 190
- VRRP, 188-190

**floating static routes, branch office implementation plans, 90**

**Frame Relay, EIGRP over Frame Relay, 23-24**

## G

**G.SHDSL (Symmetric High-speed DSL), branch office implementation plans, 86**

**gatekeepers, VoIP, 201**

**gateways (voice), VoIP, 201**

**gathering information (troubleshooting methodology), 233**

**GET VPN (Group Encrypted Transport Virtual Private Networks), 90**

**giant errors, troubleshooting, 252**

**GLBP (Gateway Load Balancing Protocol), 189-190, 259**

**graceful shutdown, EIGRP, 30**

**GRE (Generic Routing Encapsulation), 90**
- dynamic routing over GRE tunnels, branch office implementation plans, 90-91
- IPsec, 125-126
- IPv6 over IPv4 tunneling, 111

## H

**H-REAP (Hybrid Remote Edge Access Points), WLAN, 216**

**HA (High Availability)**
- documentation, 175
- monitoring, 175
- network management
  - *IP SLA, 183-184*
  - *SNMP, 182*
  - *Syslog, 182*
- people in, 174
- processes in, 175
- redundancy, 173
  - *designing for redundancy, 178*
  - *layer 2 access design, 178-180*
  - *layer 3 access design, 178-180*
  - *nonchassis based access switches, 181*
  - *NSF with SSO, 177-178*
- resiliency
  - *fast failover, 176-177*
  - *network level resiliency, 176*
- technology in, 173
- tools in, 175

**hardware diagnostics, 246-249**

**HDSL (High-data-rate DSL), branch office implementation plans, 86**

**Hello packages (EIGRP), 15**

**Hello packets (OSPF), 36**

**Hierarchical Design Model**
- access layer, 129
- complex networks, 1-2
- core layer, 130
  - *data centers, 131*
  - *medium campus networks, 131*
  - *network traffic flows, 133*
  - *small campus networks, 131*
- distribution layer, 129

**host addressing (IPv6), 101**
- manual IP address assignments, 102
- manual network assignments, 102-103
- renumbering, 104
- stateless autoconfiguration, 103

**HSRP (Hot Standby Routing Protocol), 185-187, 190, 258-260**

**HTTP (Hypertext Transfer Protocol), security, 270**

**HTTPS (Hypertext Transfer Protocol Secure), 270**

## I

**IBGP (Internal Border Gateway Protocol), 71**

**Idle state (BGP peering), 75**

**IGP (Interior Gateway Protocols), AS, 69**

**IIN (Intelligent Information Networks), 8-9**

**IKE (Internet Key Exchange), IPsec VPN, 121**

**information, gathering (troubleshooting methodology), 233**

**input errors, 248**

**input queue drops, 248**

**interdomain routing, 69**

**interface ACL (Access Control Lists), 124**

**internal routers, OSPF, 33**

**interVLAN routing**

CEF switching, 170-172

external routers, 166

fast switching, 170

multilayer switches

*Layer 2 forwarding process, 167-168*

*Layer 2 interfaces, 168*

*layer 3 forwarding process, 167-168*

*routed interfaces, 169-170*

*SVI, 169*

*switching tables, 168*

process switching, 170

troubleshooting, 258

**inventories, troubleshooting, 246, 248**

**IOS commands**

IPsec GRE tunnels, configuring, 126

site-to-site IPsec VPN, configuring, 121

*crypto ACL, 123*

*crypto maps, 124*

*interface ACL, 124*

*ISAKMP policies, 122*

*transform sets, 123*

**IOS filtering tools, 235-236**

common regular expressions, 237

generic filters, 237

piping output, 237-242

**IOS Firewall, configuring, 271**

**IP (Internet Protocol)**

predefined IP, security, 270

IP phones, VOIP, 201

IP Source Guard, spoofing attacks, 198

**IP SLA (Internet Protocol Service Level Agreements)**

HA network management, 183-184

path control, 62-64

**IPsec (IPSecurity)**

AH, 117

authentication, 119

encryption, 119-120

ESP, 118

GRE, 125-126

key management, 120

mobile network connectivity implementation plans, 96

PKI, 120

Transport mode, 118

Tunnel mode, 118-119

IPsec VPN

*branch office implementation plans, 89*

*crypto ACL, 89, 123*

*crypto maps, 89, 124*

*interface ACL, 124*

*ISAKMP policies, 122*

*monitoring, 125*

*SA, 121*

*site-to-site VPN via IOS, 121-124*

*transform sets, 123*

*troubleshooting, 125*

*verifying, mobile network connectivity implementation plans, 97*

**IPv4 (Internet Protocol version 4), IPv6 integration, 110**

address translation, 113

Dynamic NAT-PT, 113

Static NAT-PT, 113

tunneling IPv6 over IPv4, 111-112

**IPv6 (Internet Protocol version 6)**

addressing, 99

*host addressing, 101-103*

*renumbering, 104*

*simplifying addresses, 100*

*special addresses, 101*

features of, 99

headers, format of, 127

IPv4 integration, 110

*address translation, 113*

*Dynamic NAT-PT, 113*

*Static NAT-PT, 113*

*tunneling IPv6 over IPv4, 111-112*

multiaccess links, 115

P2MP links, 115

P2P links, 114

routing

*EIGRP, 106-107*

*MP-BGP, 109*

*OSPFv3, 107-108*

*RIPng, 105, 109*

*static routing, 105*

**ISAKMP (Internet Security Association and Key Management Protocol), 121-122**

**ISATAP (Intra-Site Automatic Tunnel Addressing Protocol), IPv6 over IPv4 tunneling, 112**

**ISL (Inter-Switch Link), VLAN identification, 141**

**ISP (Internet Service Providers), BGP, 68-69**

**ITIL (IT Infrastructure Library), routing implementation plans, 12**

## J - K - L

**keepalive messages (BGP), 71**

**LACP (Link Aggregation Control Protocol), 148**

**LAN (Local Area Networks), 210.** ***See also*** **VLAN; WLAN**

**large branch offices, design considerations, 84**

**Late-Col, troubleshooting, 252**

**Layer 2 switches, 150**

**lightweight AP (access points), WLAN, 214-215**

**links**
- IPv6 links, 114-115
- UDLD, 161
- virtual links, OSPF configurations, 45
- VLAN trunking links, 142-143

**load balancing (EIGRP), 26**

**load sharing (EIGRP), branch office implementation plans, 92**

**local VLAN (Virtual Local Area Networks), 137**

**logging events/alerts, maintenance, 224**

**loops**
- bridging loops, identifying, 163
- Loop Guard, STP, 162

**LSA (Link State Advertisements), 34-35**

**LSAck (Link State Acknowledgment) packets (OSPF), 36**

**LSDB (Link-State Databases), overload protection, 34**

**LSR (Link State Request) packets (OSPF), 36**

**LSU (Link State Update) packets (OSPF), 36**

**LWAPP (Lightweight Access Point Protocol), 214**

## M

**MAC addresses, flooding attacks, 192**

**maintenance.** ***See also*** **troubleshooting**
- alert logging, 224
- Archive and Restore feature, 225-227
- baselines, 234
- change control, 221, 234
- common tasks, 220-221
- configuration histories, 223-224
- documentation, 221-222, 228
- downtime, 221
- Dynamic Configuration tool, 228
- event logging, 224
- Feature Navigator, 228
- framework models, 220
- methodology of, 219-220
- monitoring tools, 228
- network monitoring, maintenance, 223
- performance tracking, 228
- Power Calculator, 228
- preventive maintenance, 221, 233
- Syslog support, 224
- time stamps, 225
- troubleshooting, integrating with, 233-234

**Management plane (networks), troubleshooting security, 270-271**

**medium campus networks, core layer, 131**

**medium-sized branch offices, design considerations, 84**

**memory**
- BGP memory leaks, 269
- memory leaks, 269
- routing, 269
- troubleshooting, 248, 269

**mobile network connectivity**
- components of, 94
- implementation plans, 95
  - *address pools, 96*
  - *allowing IPsec traffic, 96*
  - *crypto maps, 97*
  - *tuning NAT, 98*
  - *verifying IPsec VPN, 97*
  - *VPN subnet routing, 97*
- managing, 93
- QoS, 93
- user authentication, 93

**mobile worker connectivity, 93.** ***See also*** **branch offices**

**monitoring**
- HA, 175
- IPsec VPN, 125
- monitoring tools, 228
- networks, maintenance, 223
- VTP, 145

**MP-BGP (Multiprotocol Border Gateway Protocol), IPv6 routing, 109**

**MPLS (Multiprotocol Label Switching)**
- EIGRP over MPLS, 22
- EoMPLS, EIGRP over EoMPLS, 21-22
- OSPF over Layer 2 MPLS, 42
- OSPF over Layer 3 MPLS, 42

**MST (Multiple Spanning Tree), fast convergence, 159**

**Multi-Col, troubleshooting, 252**

**multiaccess links (IPv6), 115**

**multihomed ISP connections, BGP, 69**

**multipoint control units, VoIP, 201**

## N

**NAT (Network Address Translation)**
- Dynamic NAT-PT, IPv4 and IPv6 integration, 113
- mobile network connectivity implementation plans, 98
- NAT Traversal, 119
- Static NAT-PT, IPv4 and IPv6 integration, 113

**NBMA (Nonbroadcast Multiaccess) networks, OSPF, 40-42**

**NDP (Neighbor Discovery Protocol), 103**

**neighbor databases (BGP), 70**

**neighbor discovery, NDP, 103**

**Netflow, 250**

**network command (BGP), 74**

**network layer connectivity, troubleshooting, 261**

**network level resiliency, HA, 176**

**networks**
- monitoring, maintenance, 223
- network implementation plans, 135
- security, troubleshooting, 270-272
- traffic flows, core layer, 133

**Non-negotiate mode (DTP), 142**

**notification messages (BGP), 71**

**NS (Neighbor Solicitation) multicasts, IPv6, 115**

**NSF (Non-Stop Forwarding), HA redundancy, 177-178**

**NSSA (Not-so-Stubby Areas), OSPF, 32, 44**

**NTP (Network Time Protocol), 225**

## O

**OER (Optimized Edge Routing), path control, 66**

**offset-lists, path control, 61**

**On mode**
- LACP, 148
- PAgP, 147

**open messages (BGP), 71**

**OpenConfirm state (BGP peering), 75**

**OpenSent (Active) state (BGP peering), 75**

**optimizing routing**
- AD, 56-57
- distribute lists, 54
- multiple routing protocols, 54-55
- passive-interface command, 54
- prefix lists, 52-53
- route maps, 48-52
- route redistribution, 58-59
- seed metrics, 56
- updates, 48

**OSI troubleshooting method, 231-232**

**OSPF (Open Shortest Path First), 10**
- ABR, 33
- ASBR, 33
- backbone area, 32
- backbone routers, 33
- BDR routers, 41
- broadcast networks, 40
- configuring, 38
- customizing configurations
  - *authentication, 46-47*
  - *default routes, 44*
  - *NSSA, 44*
  - *passive-interface command, 43*
  - *stub areas, 44*
  - *summarization, 43*
  - *Totally Stubby Areas, 44*
  - *virtual links, 45*
- DBD packets, 36
- DR routers, 41
- Hello packets, 36
- implementation plans, 37
- internal routers, 33
- link costs, 33
- LSA, 34-35
- LSAck packets, 36
- LSR packets, 36
- LSU packets, 36
- MPLS, 42
- NBMA networks, 40, 42
- neighbor discovery process, 37
- neighbor relationships, 36
- network structure, 31-32
- NSSA, 32, 44
- OSPF metric, 33
- OSPFv3, IPv6 routing, 107-108
- P2MNP networks, 40
- P2MP networks, 40
- P2P networks, 40
- regular area, 32
- RID, 39
- route exchange process, 37
- stub areas, 32
- Totally NSSA, 32

Totally Stubby Areas, 32, 44
troubleshooting, 39-40, 264-265
verifying, 39-40

**output errors, 248**

**output queue drops, 248**

**output redirection, 242**

## P

**P2MNP (Point-to-Multipoint Nonbroadcast), OSPF, 40**

**P2MP (Point-to-Multipoint)**
IPv6 links, 115
OSPF, 40

**P2P (Point-to-Point)**
IPv6 links, 114
OSPF, 40

**packet sniffing**
RITE, 250
SPAN, 249

**PAgP (Port Aggregation Protocol), 147**

**Passive mode (LACP), 148**

**passive-interface command**
EIGRP configurations, 25
OSPF configurations, 43
routing optimization, 54

**path control**
IOS IP SLA, 62-64
OER, 66
offset-lists, 61
policy-based routing, 64-66
VRF, 66

**peering (BGP), 75**

**performance tracking, maintenance, 228**

**permit statements, controlling route redistribution, 50**

**ping tests, 243-244**

**PKI (public key infrastructure), IPsec, 120**

**PoE (Power over Ethernet) switches, 203**

**policy-based routing**
path control, 64-66
route maps, 51

**Portfast, 156**

**Power Calculator, maintenance, 228**

**power supplies, troubleshooting, 246**

**PPDIOO (Prepare, Plan, Design, Implement, Operate and Optimize) Lifecycle Approach (network implementation plans), 135**

**PPPoA (Point-to-Point Protocol over ATM), DSL, 87**

**PPPoE (Point-to-Point Protocol over Ethernet), DSL, 87**

**predefined IP (Internet Protocol), security, 270**

**prefix lists, 52-53, 78**

**preventive maintenance, 221, 233**

**problems, defining (troubleshooting methodology), 232**

**process switching, 170, 268**

**Proxy ARP (Address Resolution Protocol), 185**

**PVLAN (Private VLAN), 196-197**

**PVST+ (Per-VLAN Spanning Tree)**
RSTP, 158
STP, 155

## Q - R

**QoS (Quality of Service)**
AutoQoS, VoIP, 207
complex networks, 7
mobile network connectivity, 93
Video over IP, 208
VoIP
*AutoQoS, 207*
*classification strategy, 204*
*dropping, 205*
*DSCP values, 205*
*marking strategy, 204*
*policing strategy, 204*
*queuing, 204*
*traffic shaping/conditioning, 204*
*trust boundaries, 206*

**Query packages (EIGRP), 15**

**queue drops, 248**

**redirecting output, 242**

**redistributing routes, 267**

**redundancy**
first hop redundancy
*GLBP, 189-190*
*HSRP, 185-187, 190*
*router implementation plans, 190*
*VRRP, 188-190*
HA, 173
*designing for redundancy, 178*
*fast NSF with SSO, 177-178*
*Layer 2 access design, 178-180*
*Layer 3 access design, 178-180*
*nonchassis based access switches, 181*

**regular area (OSPF), 32**

**Reply packages (EIGRP), 15**

**resiliency, HA, 176-177**

**restoring data, Archive and Restore feature, 225-227**

**reviewing results (troubleshooting methodology), 233**

**RIB (Routing Information Base) databases (BGP), 70**

**RID (Router ID), OSPF, 39**

**RIPng (RIP next generation), IPv6 routing, 105, 109**

**RITE (Router IP Traffic Export), 250**

**root bridge elections (STP), 152**

**Root Guard, STP, 161, 191**

**root port elections (STP), 153**

**route maps, 48**

- BGP path selection, 78
- match conditions, 49-50
- policy-based routing, 51
- route redistribution, controlling, 50-51
- set conditions, 49-50
- syntax of, 49
- tagging routes via, 52

**router discovery (NDP), 103**

**routing**

- AD, 9, 16-17
- BGP, 10
  - *AS, 69-70*
  - *authentication, 79*
  - *BGP (RIB) databases, 70*
  - *configuring, 74-75*
  - *dual-homed ISP connections, 68*
  - *EBGP, 71*
  - *filtering routes, 78*
  - *IBGP, 71*
  - *keepalive messages, 71*
  - *multihomed ISP connections, 69*
  - *neighbor databases, 70*
  - *network command, 74*
  - *next-hop selection, 72*
- multiaccess networks, 73
  - *implementation plans, 67-68*
  - *notification messages, 71*
  - *open messages, 71*
  - *overview of, 69-70*
  - *path selection, 76-78*
  - *peering, 75*
  - *route options, 68*
  - *routing tables, 71*
  - *single-homed ISP connections, 68*
  - *synchronization rule, 73*
  - *update messages, 71*
  - *verifying, 80-82*
- choosing routes, 11
- EIGRP, 10, 14
  - *ACK packages, 15*
  - *Active Process Enhancement, 30*
  - *authentication, 27-28*
  - *configuring, 20*
  - *creating default routes, 20-21*
  - *customizing configurations, 25-26*
  - *EIGRP across WAN, 21-24*
  - *EIGRP over EoMPLS, 21-22*
  - *EIGRP over Frame Relay, 23-24*
  - *EIGRP over MPLS, 22*
  - *graceful shutdown, 30*
  - *Hello packages, 15*
  - *implementation plans, 19-20*
  - *load balancing, 26*
  - *neighbor discovery process, 15-16*
  - *passive-interface command, 25*
  - *Query packages, 15*
  - *Reply packages, 15*
  - *route exchange process, 15-16*
  - *route selection, 16-18*
  - *scalability, 29-30*
  - *stub routing, 29*
  - *summarization, 26*
  - *troubleshooting, 21*
  - *unicast neighbors, 25*
  - *Update packages, 15*
  - *verifying, 21*
- FD, EIGRP, 16-17
- first-hop redundancy, 258-260
- floating static routes, branch office implementation plans, 90
- HSRP, 185-187, 190
- implementation plans, 12-13
- interdomain routing, 69
- interVLAN routing
  - *CEF switching, 170-172*
  - *external routers, 166*
  - *fast switching, 170*
  - *multilayer switches, 167-170*
  - *process switching, 170*
  - *troubleshooting, 258*
- IPv6
  - *EIGRP, 106-107*
  - *IPv4 integration, 110-113*
  - *MP-BGP, 109*
  - *multiaccess links, 115*
  - *OSPFv3, 107-108*
  - *point-to-multipoint links, 115*
  - *point-to-point links, 114*
  - *RIPng, 105, 109*
  - *static routing, 105*

optimizing
*AD, 56-57*
*controlling updates, 48*
*distribute lists, 54*
*multiple routing protocols, 54-55*
*passive-interface command, 54*
*prefix lists, 52-53*
*route maps, 48-52*
*route redistribution, 58-59*
*seed metrics, 56*
OSPF, 10, 31
*ABR, 33*
*ASBR, 33*
*authentication, 46-47*
*backbone area, 32*
*backbone routers, 33*
*BDR routers, 41*
*broadcast networks, 40*
*configuring, 38*
*customizing configurations, 43-47*
*DBD packets, 36*
*default routes, 44*
*DR routers, 41*
*Hello packets, 36*
*implementation plans, 37*
*internal routers, 33*
*link costs, 33*
*LSA, 34-35*
*LSAck packets, 36*
*LSR packets, 36*
*LSU packets, 36*
*NBMA networks, 40-42*
*neighbor discovery process, 37*
*neighbor relationships, 36*
*network structure, 31-32*
*NSSA, 32, 44*
*OSPF metric, 33*
*OSPF over Layer 2 MPLS, 42*
*OSPF over Layer 3 MPLS, 42*
*P2MNP networks, 40*
*P2MP networks, 40*
*P2P networks, 40*
*passive-interface command, 43*
*regular area, 32*
*RID, 39*
*route exchange process, 37*
*stub areas, 32, 44*
*summarization, 43*
*Totally NSSA, 32*
*Totally Stubby Areas, 44*
*troubleshooting, 39-40*
*verifying, 39-40*
*virtual links, 45*
path control
*IOS IP SLA, 62-64*
*OER, 66*
*offset-lists, 61*
*policy-based routing, 64-66*
*VRF, 66*
policy-based routing
*path control, 64-66*
*route maps, 51*
redistributing, 267
redundancy, router implementation plans, 190
route maps, 48
*BGP path selection, 78*
*match conditions, 49-50*
*policy-based routing, 51*
*route redistribution, controlling, 50-51*
*set conditions, 49-50*
*syntax of, 49*
*tagging routes via, 52*
route redistribution
*configuring via multiple routing protocols, 55*
*controlling via route maps, 50-51*
*implementation plans, 58-59*
*match conditions, 49*
routing tables
*BGP, 71*
*building, 11*
static routing, IPv6, 105
stub routing, EIGRP, 29
troubleshooting
*BGP, 265-266, 269*
*EIGRP, 262-264*
*memory, 269*
*network layer connectivity, 261*
*OSPF, 264-265*
*performance, 268*
*route redistribution, 267*
*routing protocols, 261*
VRRP, 188-190

**RSA, 120**

**RSTP (Rapid Spanning Tree Protocol), 157-158**

**runts (alignment errors), troubleshooting, 252**

## S

**SA (Security Associations), 121**

**scalability, EIGRP, 29-30**

**schema poisoning, 9**

**SDSL (Symmetrical DSL), branch office implementation plans, 86**

**security**
authentication, 270

campus networks, 191
*802.1Q double-tagging attacks, 195*
*ARP spoofing attacks, 198*
*DHCP spoofing attacks, 198*
*IP Source Guard, 198*
*MAC address flooding attacks, 192*
*port security, 192-193*
*port-based authentication, 193-194*
*protected ports, 197*
*PVLAN, 196-197*
*switch spoofing attacks, 194*
*switches, 199*
*VACL, 195-196*
firewalls (IOS), 271
HTTP, 270
HTTPS, 270
IPsec
*AH, 117*
*authentication, 119*
*configuring site-to-site VPN via IOS, 121-124*
*encryption, 119-120*
*ESP, 118*
*establishing VPN, 121*
*GRE, 125-126*
*key management, 120*
*monitoring IPsec VPN, 125*
*PKI, 120*
*Transport mode, 118*
*troubleshooting IPsec VPN, 125*
*Tunnel mode, 118-119*
mobile worker connectivity, 93
networks, troubleshooting, 270-272
predefined IP, 270
SNMP, 270-272
SSH, 270
switches, 199
telnet, 270

**seed metrics, routing optimization, 56**

**Server role (VTP), 144**

**Service Provider Edge section (Enterprise Composite Model), 3-5**

**simplifying IPv6 addresses, 100**

**Single-Col, troubleshooting, 252**

**single-homed ISP connections, BGP, 68**

**SLA (Service Level Agreements)**
monitoring tools, 228
SLA Monitor, 229

**small branch offices, design considerations, 83**

**small campus networks, core layer, 131**

**snapshots, network monitoring, 223**

**SNMP (Simple Network Management Protocol), 250**
HA network management, 182
security, 270-272

**SONA (Service-Oriented Network Architecture), 9, 134**

**source interfaces, changing, 244**

**SPAN (Switched Port Analyzer), 249**

**Spanning Tree protocol, 254-256**

**spoofing attacks**
ARP spoofing attacks, 198
DHCP spoofing attacks, 198
IP Source Guard, 198

**SSH (Secure Shell), security, 270**

**SSID (Service Set Identifiers), WLAN, 211**

**SSO (Single Sign-On), NSF with SSO, 177-178**

**stand-alone (autonomous) AP (Access Points), 213**

**Static NAT-PT (Network Address Translation - Protocol Translation), IPv4 and IPv6 integration, 113**

**static routing, IPv6, 105**

**STP (Spanning Tree Protocol), 150**
BackboneFast, 160
best practices, 164-165
BPDU, 154-155
*BPDU Guard, 160, 191*
*filtering, 161*
*RSTP, 157*
bridging loops, identifying, 163
configuring, 156
elections
*criteria for, 151*
*designated port elections, 153*
*root bridge elections, 152*
*root port elections, 153*
Loop Guard, 162
MST, 159
path costs, 152
port states, 155-157
Portafast, 156
PVST+, 155, 158
Root Guard, 161, 191
RSTP, 157-158
troubleshooting, 163
UDLD, 161
UplinkFast, 159

**structured troubleshooting, 231-232**

**stub areas (OSPF), 32, 44**

**stub routing (EIGRP), 29**

**summarization**
  EIGRP, 26
  OSPF configurations, 43
**switched virtual interfaces, troubleshooting, 258**
**switches**
  BPDU, 154-157
  CEF, 261, 268
  fast switching, 268
  Layer 2 switches, 150
  PoE switches, 203
  process switching, 268
  security, 199
  spoofing attacks, 194
  troubleshooting, 252
    *auto-MDIX, 253*
    *Etherchannel failures, 256*
    *first-hop redundancy, 258-260*
    *Spanning Tree protocol, 254-256*
    *VLAN, 257-258*
  VoIP support, configuring, 206-207
  WLAN configurations, 217
**symmetric key algorithms, IPsec encryption, 119**
**synchronization rule (BGP), 73**
**Syslog**
  HA network management, 182
  maintenance, 224

## T

**TCN BPDU (Topology Change Notification Bridge Protocol Data Units), 154**
**Teleworker Architecture component (Cisco Enterprise Architecture), 7**
**telnet, 245, 270**
**temperature readings, troubleshooting, 246**
**templates, maintenance, 222**
**testing**
  ping tests, 243-244
  source interfaces, changing, 244
  telnet, 245
  testing process (troubleshooting methodology), 233
  traceroute, 245
  Type of Service settings, 244
**time stamps, maintenance, 225**
**time zones, maintenance, 225**
**TMN (Telecommunications Management Networks), routing implementation plans, 12**
**Totally NSSA (Not-So-Stubby Area), OSPF, 32**
**Totally Stubby Areas (OSPF), 32, 44**
**traceroute, 245**
**traffic flows (network), core layer, 133**
**Transform Sets, 121-123**
**transparent bridges, 150**
**Transparent role (VTP), 144**
**Transport mode (IPsec), 118**
**troubleshooting. *See also* maintenance**
  air conditioning, 246
  Align-Err, 252
  broadcast storms, 256
  CEF switching, 172
  collisions, 252
  DSL connections, 89
  duplex mismatches, 252-253
  EEM, 251
  EIGRP, 21
  Excess-Col, 252
  FCS-Err, 252
  giant errors, 252
  hardware diagnostics, 246-249
  holistic process, troubleshooting as, 234
  inventories, 246-248
  IOS filtering tools, 235-236
    *common regular expressions, 237*
    *generic filters, 237*
    *piping output, 237-242*
  IPsec VPN, 125
  Late-Col, 252
  maintenance, integrating with, 233-234
  memory, 248
  methodology of
    *defining the problem, 232*
    *gathering information, 233*
    *OSI troubleshooting method, 231-232*
    *principles of, 230-231*
    *reviewing results, 233*
    *structured troubleshooting, 231-232*
    *testing process, 233*
  Multi-Col, 252
  Netflow, 250
  networks, security, 270-272
  OSPF, 39-40, 108
  output redirection, 242
  packet sniffing, 249-250
  ping tests, 243-244
  power supplies, 246
  routing
    *BGP, 265-266, 269*
    *EIGRP, 262-264*
    *memory, 269*
    *network layer connectivity, 261*
    *OSPF, 264-265*
    *performance, 268*

*redistributing, 267*
*routing protocols, 261*
runts, 252
security, 270-272
Single-Col, 252
SNMP, 250
source interfaces, changing, 244
STP, 163
switched virtual interfaces, 258
switches, 252
*auto-MDIX, 253*
*CEF, 268*
*Etherchannel failures, 256*
*fast switching, 268*
*first-hop redundancy, 258-260*
*process switching, 268*
*Spanning Tree protocol, 254-256*
*VLAN, 257-258*
telnet, 245
temperature readings, 246
traceroute, 245
Type of Service settings, 244
undersize errors, 252
VLAN, 257
*interVLAN routing, 258*
*trunking, 149*
*user connectivity, 148-149*
*VTP, 149*
voltage, 246
Xmit-Err, 252

**Trunk mode (DTP), 142**

**trunks (VLAN)**
allowed VLAN, 143
best practices, 143
troubleshooting, 149
trunk links, 142
verifying links, 143
VLAN identification, 141
VTP, 143
*adding switches to VTP domains, 146*
*configuring, 145*
*monitoring, 145*
*switch roles, 144*
*troubleshooting, 149*
*verifying, 145*

**Tunnel mode (IPsec), 118-119**

**tunneling**
DMVPN, 90
GET VPN, 90
GRE, 90-91, 111
IPv6 over IPv4
*6to4 tunnels, 112*
*GRE tunnels, 111*
*IPv4-compatibile IPv6 tunnels, 112*
*ISATAP tunnels, 112*
*manual tunnels, 111*
ISATAP, IPv6 over IPv4 tunneling, 112
VTI, 90

**Type of Service settings, testing, 244**

## U

**UCM (Unified Communications Manager), VoIP, 201**

**UDLD (Unidirectional Link Detection), STP, 161**

**undersize errors, troubleshooting, 252**

**updates**
documentation/processes (troubleshooting methodology), 233
routing updates, controlling, 48
update messages (BGP), 71
Update packages (EIGRP), 15

**UplinkFast, STP, 159**

**user authentication, mobile network connectivity, 93**

## V

**VACL (VLAN access control lists), 195-196**

**VDSL (Very-high-rate DSL), branch office implementation plans, 86**

**Video over IP. *See also* VoIP**
campus networks, 208
video conferencing units, 201

**virtual interfaces (switched), troubleshooting, 258**

**virtual links, OSPF configurations, 45**

**virtualization, complex networks, 8**

**VLAN (Virtual Local Area Networks). *See also* LAN**
802.1Q double-tagging attacks, 195
creating, 139
end-to-end VLAN, 137
EtherChannels, 146
*configuring, 147-148*
*verifying, 148*
implementation plans, 139
interVLAN routing
*CEF switching, 170-172*
*external routers, 166*
*fast switching, 170*
*multilayer switches, 167-170*
*process switching, 170*
local VLAN, 137
port assignments, 139
protected ports, 197
PVLAN, 196-197

PVST+
*RSTP, 158*
*STP, 155*
switch spoofing attacks, 194
troubleshooting, 257
*interVLAN routing, 258*
*trunking, 149*
*user connectivity, 148-149*
*VTP, 149*
trunking
*allowed VLAN, 143*
*best practices, 143*
*configuring trunk links, 142*
*troubleshooting, 149*
*verifying links, 143*
*VLAN identification, 141*
*VTP, 143-146, 149*
VACL, 195-196
verifying configurations, 139-140
Voice VLAN, 203
VTP, 143
*adding switches to VTP domains, 146*
*configuring, 145*
*monitoring, 145*
*switch roles, 144*
*troubleshooting, 149*
*verifying, 145*
WLAN, 210

**voice gateways, 201**

**Voice VLAN (Virtual Local Area Networks), 203**

**VoIP (Voice over IP), campus networks, 200. *See also* Video over IP**
bandwidth requirements, 202
network requirements, 202
PoE switches, 203
QoS, 204-207
system components, 201
system requirements, 202
Voice VLAN, 203
VoIP support on switches, 206-207

**voltage, troubleshooting, 246**

**VPN (Virtual Private Networks)**
Cisco Easy VPN, mobile network connectivity implementation plans, 95-97
DMVPN, 90
Easy VPN, mobile network connectivity implementation plans, 95-97
GET VPN, 90
IPsec VPN
*branch office implementation plans, 89*
*crypto ACL, 89, 123*
*crypto maps, 89, 124*
*interface ACL, 124*
*ISAKMP policies, 122*
*monitoring, 125*
*SA, 121*
*site-to-site VPN via IOS, 121-124*
*transform sets, 123*
*troubleshooting, 125*
*verifying, mobile network connectivity implementation plans, 97*
subnet routing, mobile network connectivity implementation plans, 97

**VRF (Virtual Routing and Forwarding), path control, 66**

**VRRP (Virtual Router Redundancy Protocol), 188-190, 259**

**VTI (Virtual Tunnel Interface), 90**

**VTP (VLAN Trunking Protocol), 143**
configuring, 145
monitoring, 145
switch roles, 144
switches, adding to VTP domains, 146
troubleshooting, 149
verifying, 145

## W

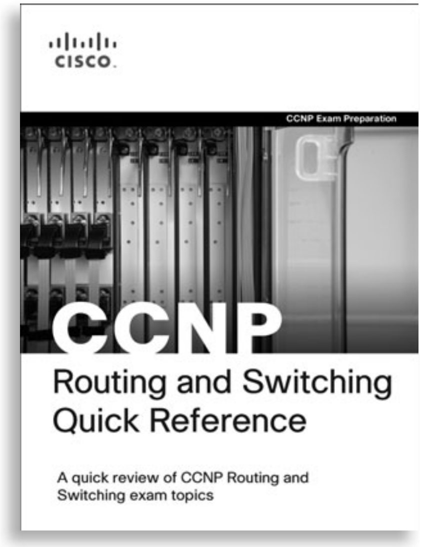
**WAN (Wide Area Networks)**
EIGRP over WAN
*bandwidth limitations, 24*
*EIGRP over EoMPLS, 21-22*
*EIGRP over Frame Relay, 23-24*
*EIGRP over MPLS, 22*
WAN component (Cisco Enterprise Architecture), 7

**WLAN (Wireless Local Area Networks). *See also* LAN**
autonomous (stand-alone) AP, 213
characteristics of, 211
Cisco Unified Wireless Networks, 210
client connectivity, 212
controllers, 215
H-REAP, 216
LAN integration, 217-218
lightweight AP, 214-215
SSID, 211
topologies of, 211-212

## X - Y - Z

**Xmit-Err, troubleshooting, 252**

**XML (Extensible Markup Language), schema poisoning, 9**



CISCO.

CCNP Exam Preparation

CCNP

Routing and Switching Quick Reference

A quick review of CCNP Routing and Switching exam topics