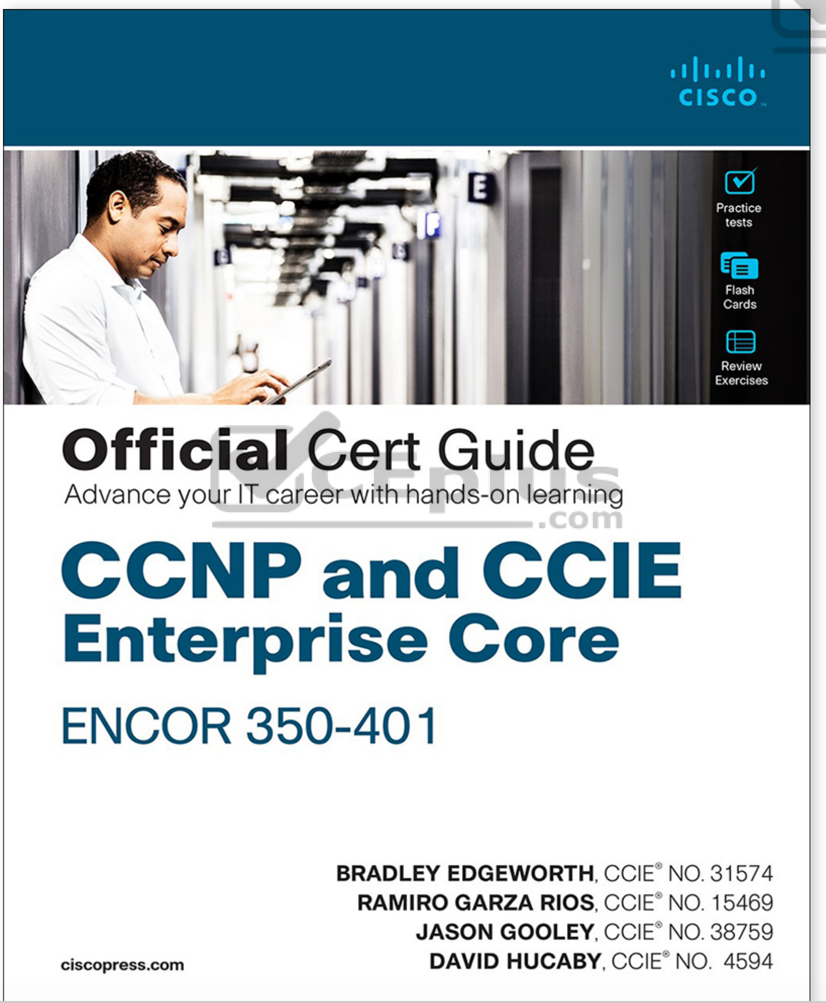


Official Cert Guide

Advance your IT career with hands-on learning

CCNP and CCIE Enterprise Core

ENCOR 350-401

BRADLEY EDGEWORTH, CCIE® NO. 31574
RAMIRO GARZA RIOS, CCIE® NO. 15469
JASON GOOLEY, CCIE® NO. 38759
DAVID HUCABY, CCIE® NO. 4594

ciscopress.com

CCNP and CCIE Enterprise Core ENCOR 300-401 Official Cert Guide

Brad Edgeworth, CCIE No. 31574

Ramiro Garza Rios, CCIE No. 15469

David Hucaby, CCIE No. 4594

Jason Gooley, CCIE No. 38759

CISCO.

CCNP and CCIE Enterprise Core ENCOR 300-401 Official Cert Guide

Brad Edgeworth, Ramiro Garza Rios, David Hucaby, Jason Gooley

Published by:
Cisco Press

ScoutAutomatedPrintCode

Library of Congress Control Number: 2019951592

ISBN-13: 978-1-58714-523-0

ISBN-10: 1-58714-523-5

Warning and Disclaimer

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

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

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

Trademark Acknowledgments

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

Special Sales

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

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

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

Feedback Information

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

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

We greatly appreciate your assistance.

Editor-in-Chief
Mark Taub

Alliances Manager, Cisco Press

Arezou Gol

Product Line Manager

Brett Bartow

Managing Editor

Sandra Schroeder

Development Editor

Ellie Bru

Senior Project Editor

Tonya Simpson

Copy Editor

Kitty Wilson

Technical Editor(s)

Richard Furr, Denise Fishburne, Dmitry Figol, Patrick Croak

Editorial Assistant

Cindy Teeters

Cover Designer

Chuti Prasertsith

Composition

Table of Contents

Introduction

Part I: Forwarding

Chapter 1. Packet Forwarding

“Do I Know This Already?” Quiz

Foundation Topics

Network Device Communication

Forwarding Architectures

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Part II: Layer 2

Chapter 2. Spanning Tree Protocol

“Do I Know This Already?” Quiz

Foundation Topics

Spanning Tree Protocol Fundamentals

Rapid Spanning Tree Protocol

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Chapter 3. Advanced STP Tuning

“Do I Know This Already?” Quiz

Foundation Topics

STP Topology Tuning

Additional STP Protection Mechanisms

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Chapter 4. Multiple Spanning Tree Protocol

“Do I Know This Already?” Quiz

Foundation Topics

Multiple Spanning Tree Protocol

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Chapter 5. VLAN Trunks and EtherChannel Bundles

“Do I Know This Already?” Quiz

Foundation Topics

Dynamic Trunking Protocol

EtherChannel Bundle

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Part III: Routing

Chapter 6. IP Routing Essentials

"Do I Know This Already?" Quiz

Foundation Topics

Routing Protocol Overview

Path Selection

Static Routing

Virtual Routing and Forwarding

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Chapter 7. EIGRP

"Do I Know This Already?" Quiz

Foundation Topics

EIGRP Fundamentals

Path Metric Calculation

Failure Detection and Timers

Route Summarization

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

References in This Chapter

Chapter 8. OSPF

“Do I Know This Already?” Quiz

Foundation Topics

OSPF Fundamentals

OSPF Configuration

Default Route Advertisement

Common OSPF Optimizations

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Chapter 9. Advanced OSPF

“Do I Know This Already?” Quiz

Foundation Topics

Areas

Link-State Announcements

Discontiguous Networks

OSPF Path Selection

Summarization of Routes

Route Filtering

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Chapter 10. OSPFv3

“Do I Know This Already?” Quiz

Foundation Topics

OSPFv3 Fundamentals

OSPFv3 Configuration

IPv4 Support in OSPFv3

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Chapter 11. BGP

"Do I Know This Already?" Quiz

Foundation Topics

BGP Fundamentals

Basic BGP Configuration

Route Summarization

Multiprotocol BGP for IPv6

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Chapter 12. Advanced BGP

“Do I Know This Already?” Quiz

Foundation Topics

BGP Multihoming

Conditional Matching

Route Maps

BGP Route Filtering and Manipulation

BGP Communities

Understanding BGP Path Selection

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

References in This Chapter

Chapter 13. Multicast

“Do I Know This Already?” Quiz

Foundation Topics

Multicast Fundamentals

Multicast Addressing

Internet Group Management Protocol

Protocol Independent Multicast

Rendezvous Points

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

References In This Chapter

Part IV: Services

Chapter 14. QoS

"Do I Know This Already?" Quiz

Foundation Topics

The Need for QoS

QoS Models

Classification and Marking

Policing and Shaping

Congestion Management and Avoidance

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

References In This Chapter

Chapter 15. IP Services

"Do I Know This Already?" Quiz

Foundation Topics

First-Hop Redundancy Protocol

Network Address Translation

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Part V: Overlay

Chapter 16. Overlay Tunnels

"Do I Know This Already?" Quiz

Foundation Topics

Generic Routing Encapsulation (GRE) Tunnels

IPsec Fundamentals

Cisco Location/ID Separation Protocol (LISP)

Virtual Extensible Local Area Network (VXLAN)

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Part VI: Wireless

Chapter 17. Wireless Signals and Modulation

“Do I Know This Already?” Quiz

Foundation Topics

Understanding Basic Wireless Theory

Carrying Data Over an RF Signal

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 18. Wireless Infrastructure

"Do I Know This Already?" Quiz

Foundation Topics

Wireless LAN Topologies

Pairing Lightweight APs and WLCs

Leveraging Antennas for Wireless Coverage

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 19. Understanding Wireless Roaming and Location Services

"Do I Know This Already?" Quiz

Foundation Topics

Roaming Overview

Roaming Between Centralized Controllers

Locating Devices in a Wireless Network

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 20. Authenticating Wireless Clients

“Do I Know This Already?” Quiz

Foundation Topics

Open Authentication

Authenticating with Pre-Shared Key

Authenticating with EAP

Authenticating with WebAuth

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 21. Troubleshooting Wireless Connectivity

“Do I Know This Already?” Quiz

Foundation Topics

Troubleshooting Client Connectivity from the WLC

Troubleshooting Connectivity Problems at the AP

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Part VII: Architecture

Chapter 22. Enterprise Network Architecture

“Do I Know This Already?” Quiz

Foundation Topics

Hierarchical LAN Design Model

Enterprise Network Architecture Options

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 23. Fabric Technologies

“Do I Know This Already?” Quiz

Foundation Topics

Software-Defined Access (SD-Access)

Software-Defined WAN (SD-WAN)

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 24. Network Assurance

Do I Know This Already?

Foundation Topics

Network Diagnostic Tools

Debugging

NetFlow and Flexible NetFlow

Switched Port Analyzer (SPAN) Technologies

IP SLA

Cisco DNA Center Assurance

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Part VIII: Security

Chapter 25. Secure Network Access Control

“Do I Know This Already?” Quiz

Foundation Topics

Network Security Design for Threat Defense

Next-Generation Endpoint Security

Network Access Control (NAC)

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 26. Network Device Access Control and Infrastructure Security

“Do I Know This Already?” Quiz

Foundation Topics

Access Control Lists (ACLs)

Terminal Lines and Password Protection

Authentication, Authorization, and Accounting (AAA)

Zone-Based Firewall (ZBFW)

Control Plane Policing (CoPP)

Device Hardening

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Use the Command Reference to Check Your Memory

Part IX: SDN

Chapter 27. Virtualization

“Do I Know This Already?” Quiz

Foundation Topics

Server Virtualization

Network Functions Virtualization

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 28. Foundational Network Programmability Concepts

“Do I Know This Already?” Quiz

Foundation Topics

Command-Line Interface

Application Programming Interface

Data Models and Supporting Protocols

Cisco DevNet

GitHub

Basic Python Components and Scripts

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

References in This Chapter

Chapter 29. Introduction to Automation Tools

“Do I Know This Already?” Quiz

Foundation Topics

Embedded Event Manager

Agent-Based Automation Tools

Exam Preparation Tasks

Review All Key Topics

Complete Tables and Lists from Memory

Define Key Terms

Chapter 30. Final Preparation

Getting Ready

Tools for Final Preparation

Suggested Plan for Final Review/Study

Part I: Forwarding

Chapter 1. Packet Forwarding

This chapter covers the following subjects:

• **Network Device Communication:** This section explains how switches forward traffic from a Layer 2 perspective and routers forward traffic from a Layer 3 perspective.

• **Forwarding Architectures:** This section examines the mechanisms used in routers and switches to forward network traffic.

This chapter provides a review of basic network fundamentals and then dives deeper into the technical concepts related to how network traffic is forwarded through a router or switch architecture.

"DO I KNOW THIS ALREADY?" QUIZ

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

Table 1-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topic Section	Questions
Network Device Communication	1–4
Forwarding Architectures	5–7

1. Forwarding of network traffic from a Layer 2 perspective uses what information?

a. Source IP address

b. Destination IP address

c. Source MAC address

d. Destination MAC address

e. Data protocol

2. What type of network device helps reduce the size of a collision domain?

a. Hub

b. Switch

c. Load balancer

d. Router

3. Forwarding of network traffic from a Layer 3 perspective uses what information?

a. Source IP address

b. Destination IP address

c. Source MAC address

d. Destination MAC address

e. Data protocol

4. What type of network device helps reduce the size of a broadcast domain?

a. Hub

b. Switch

c. Load balancer

d. Router

5. The __________ can be directly correlated to the MAC address table.

a. Adjacency table

b. CAM

c. TCAM

d. Routing table

6. A ____________ forwarding architecture provides increased port density and forwarding scalability.

a. Centralized

b. Clustered

c. Software

d. Distributed

7. CEF is composed of which components? (Choose two.)

a. Routing Information Base

b. Forwarding Information Base

c. Label Information Base

d. Adjacency table

e. MAC address table

Answers to the “Do I Know This Already?” quiz:

1. D

2. B

3. B

4. D

5. B

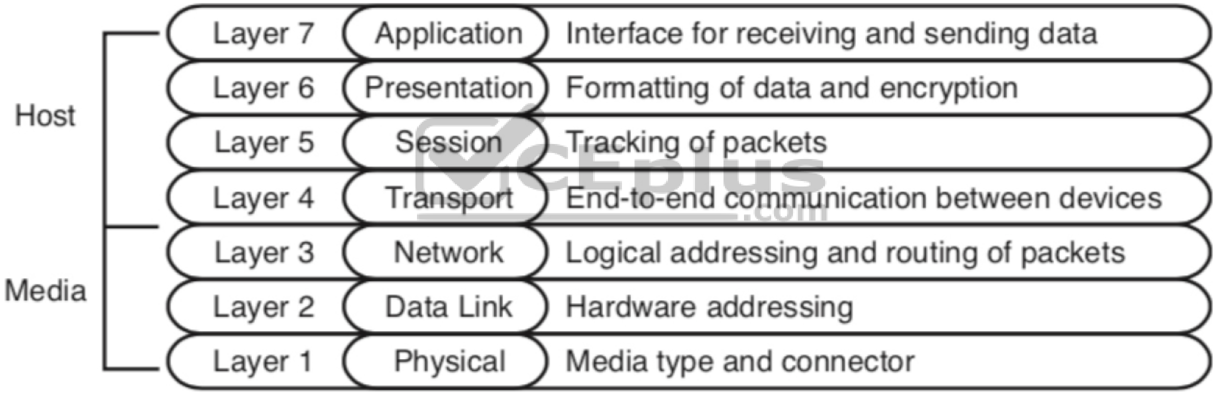
6. D

7. B, D

FOUNDATION TOPICS

NETWORK DEVICE COMMUNICATION

The primary function of a network is to provide connectivity between devices. There used to be a variety of network protocols that were device specific or preferred; today, almost everything is based on *Transmission Control Protocol/Internet Protocol (TCP/IP)*. It is important to note that TCP/IP is based on the conceptual *Open Systems Interconnection (OSI)* model that is composed of seven layers. Each layer describes a specific function, and a layer can be modified or changed without requiring changes to the layer above or below it. The OSI model, which provides a structured approach for compatibility between vendors, is illustrated in Figure 1-1.



Host	Layer 7	Application	Interface for receiving and sending data
	Layer 6	Presentation	Formatting of data and encryption
	Layer 5	Session	Tracking of packets
	Layer 4	Transport	End-to-end communication between devices
Media	Layer 3	Network	Logical addressing and routing of packets
	Layer 2	Data Link	Hardware addressing
	Layer 1	Physical	Media type and connector

Figure 1-1 OSI Model

When you think about the flow of data, most network traffic involves communication of data between applications. The applications generate data at Layer 7, and the device/host sends data down the OSI model. As the data moves down the OSI model, it is encapsulated or modified as needed.

At Layer 3, the device/host decides whether the data needs to be sent to another application on the same device, and it would then start to move the data up the stack. Or, if the data needs to be sent to a different device, the device/host continues processing down the OSI model toward Layer 1. Layer 1 is responsible for transmitting the information on to the media (for example, cable, fiber, radio waves). On the receiving side, data starts at Layer 1, then moves to Layer 2, and so on, until it has moved completely up to the Layer 7 and on to the receiving application.

This chapter reinforces concepts related to how a network device forwards traffic from either a Layer 2 or a Layer 3 perspective. The first Layer 2 network devices were bridges or switches, and Layer 3 devices were strictly routers. As technology advanced, the development of faster physical media required the ability to forward packets in hardware through ASICs. As ASIC functionality continued to develop, multilayer switches (MLSs) were invented to forward Layer 2 traffic in hardware as if they were switches; however, they can also perform other functions, such as routing packets, from a Layer 3 perspective.

Layer 2 Forwarding

The second layer of the OSI model, the data link layer, handles addressing beneath the IP protocol stack so that communication is directed between hosts. Network packets include Layer 2 addressing with unique source and destination addresses for segments. Ethernet commonly uses *media access control (MAC)* addresses, and other data link layer protocols such as Frame Relay use an entirely different method of Layer 2 addressing.

The focus of the Enterprise Core exam is on Ethernet and wireless technologies, both of which use MAC addresses for *Layer 2* addressing. This book focuses on the MAC address for Layer 2 forwarding.

Note

A MAC address is a 48-bit address that is split across six octets and notated in hexadecimal. The first three octets are assigned to a device manufacturer, known as the organizationally unique identifier (OUI), and the manufacturer is responsible for ensuring that the last three octets are unique. A device listens for network traffic that contains its MAC address as the packet's destination MAC address before moving the packet up the OSI stack for Layer 3 for processing.

Network broadcasts with MAC address FF:FF:FF:FF:FF:FF are the exception to the rule and will always be processed by all network devices on the same network segment. Broadcasts are not typically forwarded beyond a Layer 3 boundary.

Collision Domains

The Ethernet protocol first used technologies like Thinnet (10BASE-2) and Thicknet (10BASE-5), which connected all the network devices using the same cable and T connectors. This caused problems when two devices tried to talk at the same time because the transmit cable shared the same segment with other devices, and the communication become garbled if two devices talked at the same time. Ethernet devices use *Carrier Sense Multiple Access/Collision Detect (CSMA/CD)* to ensure that only one device talks at time in a *collision domain*. If a device detects that another device is transmitting data, it delays transmitting packets until the cable is quiet. This means devices can only transmit or receive data at one time (that is, operate at half-duplex).

Key Topic

As more devices are added to a cable, the less efficient the network becomes as devices wait until there is not any communication. All of the devices are in the same collision domain. Network hubs proliferate the problem because they add port density while repeating traffic, thereby increasing the size of the collision domain. Network hubs do not have any intelligence in them to direct network traffic; they simply repeat traffic out of every port.

Network switches enhance scalability and stability in a network through the creation of virtual channels. A switch maintains a table that associates a host’s *Media Access Control (MAC)* Ethernet addresses to the port that sourced the network traffic. Instead of flooding all traffic out of every switch port, a switch uses the local *MAC address table* to forward network traffic only to the destination switch port associated with where the destination MAC is attached. This drastically reduces the size of the collision domain between the devices and enables the devices to transmit and receive data at the same time (that is, operate at full duplex).

Figure 1-2 demonstrates the collision domains on a hub versus on a switch. Both of these topologies show the same three PCs, as well as the same cabling. On the left, the PCs are connected to a network hub. Communication between PC-A and PC-B is received by PC-C's NIC, too, because all three devices are in the same collision domain. PC-C must process the frame—in the process consuming resources—and then it discards the packet after determining that the destination MAC address does not belong to it. In addition, PC-C has to wait until the PC-A/PC-B conversation finishes before it can transmit data. On the right, the PCs are connected to a network switch. Communication between PC-A and PC-B are split into two collision domains. The switch can connect the two collision domains by using information from the MAC address table.

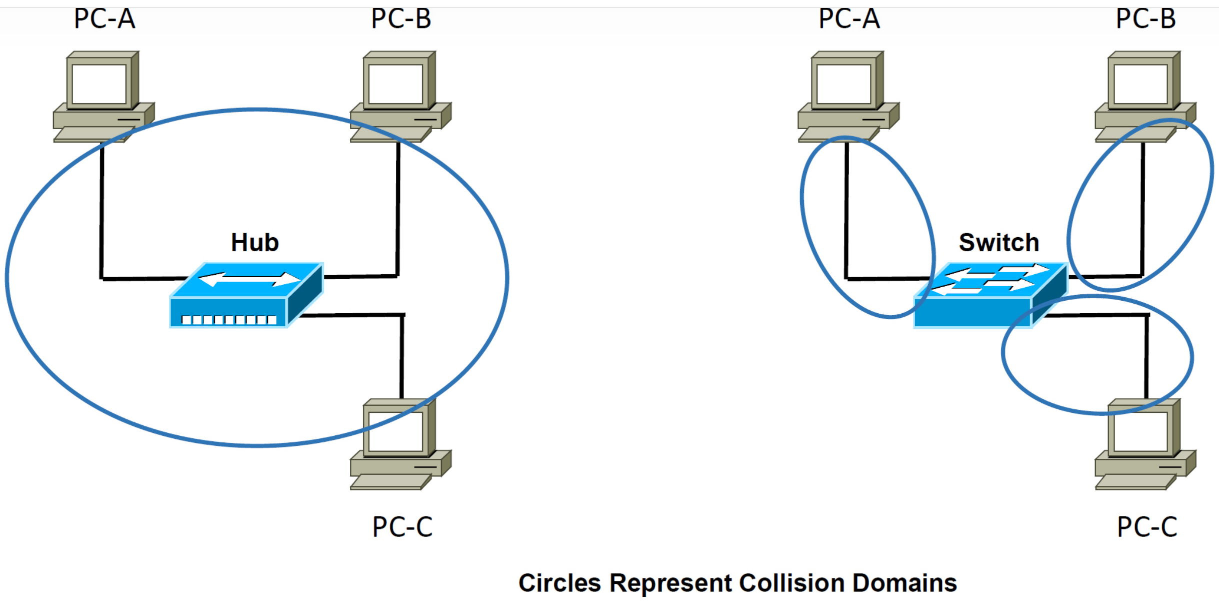
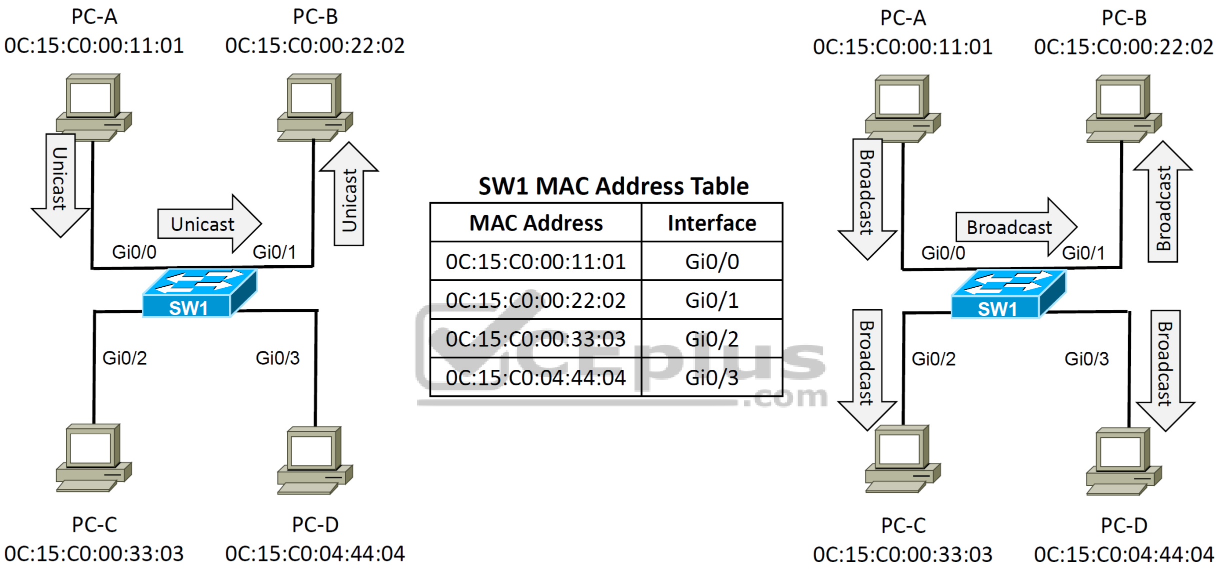


Figure 1-2 Collision Domain on Hubs Versus Switches

When a packet contains a destination MAC address that is not in the switch’s MAC address table, the switch forwards the packet out of every switch port. This is known as *unknown unicast flooding* because the destination MAC address is not known.

Broadcast traffic is network traffic intended for every host on the LAN and is forwarded out of every switch port interface. This is disruptive as it diminishes the efficiencies of a network switch compared to those of a hub because it causes communication between network devices to stop due to CSMA/CD. Network broadcasts do not cross Layer 3 boundaries (that is, from one subnet to another subnet). All devices that reside in the same Layer 2 segment are considered to be in the same *broadcast domain*.

Figure 1-3 displays SW1’s MAC address table, which correlates the local PCs to the appropriate switch port. In the scenario on the left, PC-A is transmitting unicast traffic to PC-B. SW1 does not transmit data out of the Gi0/2 or Gi0/3 interface (which could potentially disrupt any network transmissions between those PCs. In the scenario on the right, PC-A is transmitting broadcast network traffic out all active switch ports.



SW1 MAC Address Table

MAC Address	Interface
0C:15:C0:00:11:01	Gi0/0
0C:15:C0:00:22:02	Gi0/1
0C:15:C0:00:33:03	Gi0/2
0C:15:C0:04:44:04	Gi0/3

Figure 1-3 Unicast and Broadcast Traffic Patterns

Note

The terms *network device* and *host* are considered interchangeable in this text.

Virtual LANs

Adding a router between LAN segments helps shrink broadcast domains and provides for optimal network communication. Host placement on a LAN segment varies because of network addressing. Poor host network assignment can lead to inefficient use of hardware as some switch ports could go as unused.

Key Topic

Virtual LANs (VLANs) provide logical segmentation by creating multiple broadcast domains on the same network switch. VLANs provide higher utilization of switch ports because a port can be associated to the necessary broadcast domain, and multiple broadcast domains can reside on the same switch. Network devices in one VLAN cannot communicate with devices in a different VLAN via traditional Layer 2 or broadcast traffic.

VLANs are defined in the Institute of Electrical and Electronic Engineers (IEEE) 802.1Q standard, which states that 32 bits are added to the packet header in the following fields:

- **Tag protocol identifier (TPID):** This 16-bit is field set to 0x8100 to identify the packet as an 802.1Q packet.

- **Priority code point (PCP):** This 3-bit field indicates a class of service (CoS) as part of Layer 2 quality of service (QoS) between switches.

- **Drop Eligible Indicator (DEI):** This 1-bit field indicates whether the packet can be dropped when there is bandwidth contention.

- **VLAN identifier (VID):** This 12-bit field specifies the VLAN associated with a network packet.

Figure 1-4 displays the VLAN packet structure.

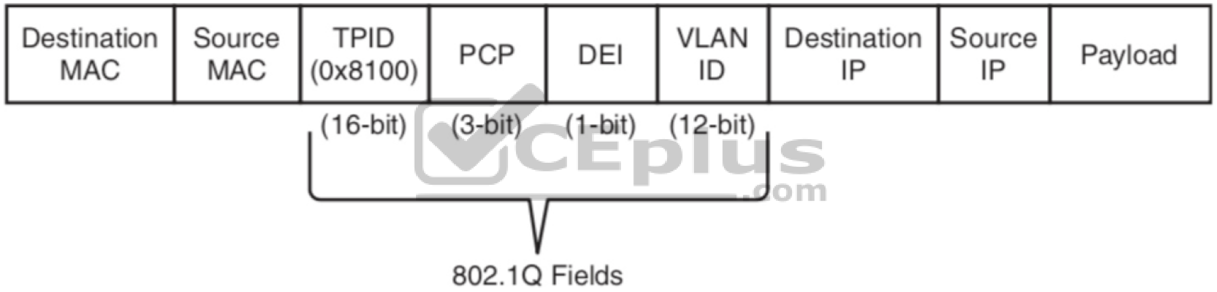


Figure 1-4 VLAN Packet Structure

The VLAN identifier has only 12 bits, which provides 4094 unique VLANs. Catalyst switches use the following logic for VLAN identifiers:

- VLAN 0 is reserved for 802.1P traffic and cannot be modified or deleted.

- VLAN 1 is the default VLAN and cannot be modified or deleted.

- VLANs 2 to 1001 are in the normal VLAN range and can be added, deleted, or modified as necessary.

- VLANS 1002 to 1005 are reserved and cannot be deleted.

- VLANs 1006 to 3967 and 4048 to 4093 are in the extended VLAN range and can be added, deleted, or modified as necessary.

VLANs are created by using the global configuration command **vlan** *vlan-id*. A friendly name (32 characters) is associated with a VLAN through the VLAN submode configuration command **name** *vlanname*. The VLAN is not created until the command-line interface (CLI) has been moved back to the global configuration context or a different VLAN identifier.

Example 1-1 demonstrates the creation of VLAN 10 (PCs), VLAN 20 (Phones), and VLAN 99 (Guest) on SW1.

Example 1-1 Creating a VLAN

```
SW1# configure term
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# vlan 10
SW1(config-vlan)# name PCs
SW1(config-vlan)# vlan 20
SW1(config-vlan)# name Phones
SW1(config-vlan)# vlan 99
SW1(config-vlan)# name Guest
```

VLANs and their port assignment are verified with the **show vlan** [{**brief** | **id** *vlan-id* | **name** *vlanname* | **summary**}] command, as demonstrated in Example 1-2. Notice that the output is split into four main sections: VLAN-to-port assignments, system MTU, SPAN sessions, and private VLANs.

Example 1-2 Viewing VLAN Assignments to Port Mapping

```
SW1# show vlan
! Traditional and common VLANs will be listed in this section. Th
! associated to these VLANs are displayed to the right.
VLAN Name                            Status    Ports
---- -------------------------------- --------- ----------------
1    default                          active    Gi1/0/1, Gi1/0/2,
                                                Gi1/0/4, Gi1/0/5,
                                                Gi1/0/10, Gi1/0/1
                                                Gi1/0/18, Gi1/0/1
                                                Gi1/0/21, Gi1/0/2
                                                Gi1/1/1, Gi1/1/2,
                                                Te1/1/4
10   PCs                              active    Gi1/0/7, Gi1/0/8,
                                                Gi1/0/12, Gi1/0/1
20   Phones                           active    Gi1/0/14
99   Guest                            active    Gi1/0/15, Gi1/0/1
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
!  This section displays the system wide MTU setting for all 1Gbp
!  interface
```

```
VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode
---- ----- ---------- ----- ------ ------ -------- ---- --------
VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode
---- ----- ---------- ----- ------ ------ -------- ---- --------
1    enet  100001     1500  -      -      -        -    -
10   enet  100010     1500  -      -      -        -    -
20   enet  100020     1500  -      -      -        -    -
99   enet  100099     1500  -      -      -        -    -
1002 fddi  101002     1500  -      -      -        -    -
1003 tr    101003     1500  -      -      -        -    -
1004 fdnet 101004     1500  -      -      -        ieee -
1005 trnet 101005     1500  -      -      -        ibm  -
!  If a Remote SPAN VLAN is configured, it will be displayed in t
!  Remote SPAN VLANs was explained in Chapter 24
Remote SPAN VLANs
------------------------------------------------------------------

!  If Private VLANs are configured, they will be displayed in thi
!  Private VLANs are outside of the scope of this book, but more
!  can be found at http://www.cisco.com
Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------
```

The optional **show vlan** keywords provide the following benefits:

- **brief:** Displays only the relevant port-to-VLAN mappings.

- **summary:** Displays a count of VLANS, VLANs participating in VTP, and VLANs that are in the extended VLAN range.

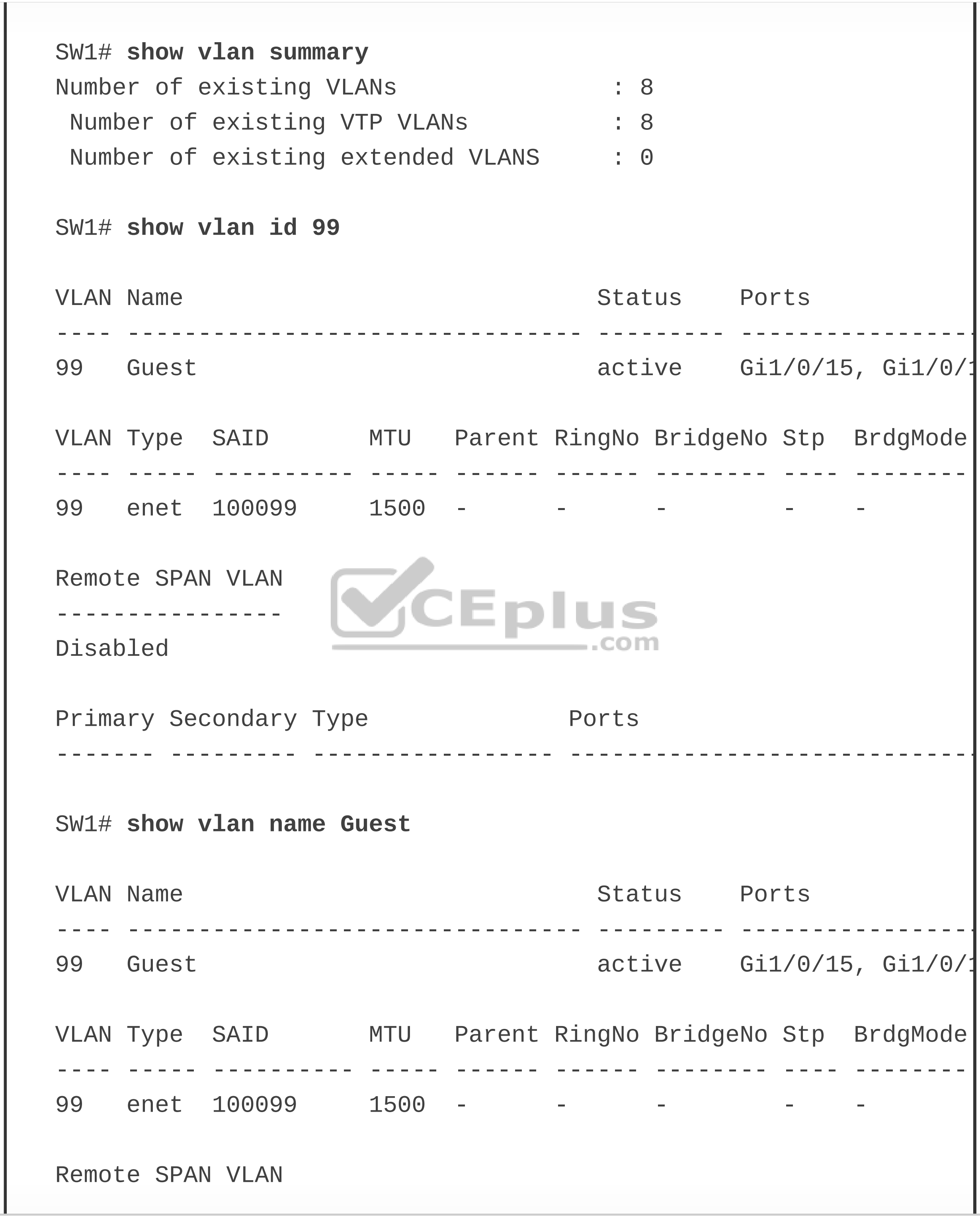
- **id** *vlan-id***:** Displays all the output from the original command but filtered to only the VLAN number that is specified.

- **name** *vlanname***:** Displays all the output from the original command but filtered to only the VLAN name that is specified.

Example 1-3 shows the use of the optional keywords. Notice that the output from the optional keywords **id** *vlan-id* is the same as the output from **name** *vlanname*.

Example 1-3 Using the Optional **show vlan** Keywords

```
SW1# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -----------------
1    default                          active    Gi1/0/1, Gi1/0/2,
                                                Gi1/0/4, Gi1/0/5,
                                                Gi1/0/10, Gi1/0/1
                                                Gi1/0/18, Gi1/0/1
                                                Gi1/0/21, Gi1/0/2
                                                Gi1/1/1, Gi1/1/2,
                                                Te1/1/4
10   PCs                              active    Gi1/0/7, Gi1/0/8,
                                                Gi1/0/12, Gi1/0/1
20   Phones                           active    Gi1/0/14
99   Guest                            active    Gi1/0/15, Gi1/0/1
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```



```
SW1# show vlan summary
Number of existing VLANs                : 8
 Number of existing VTP VLANs           : 8
 Number of existing extended VLANS      : 0

SW1# show vlan id 99

VLAN Name                             Status    Ports
---- -------------------------------- --------- ----------------
99   Guest                            active    Gi1/0/15, Gi1/0/1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode
---- ----- ---------- ----- ------ ------ -------- ---- --------
99   enet  100099     1500  -      -      -        -    -

Remote SPAN VLAN
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ----------------------------


SW1# show vlan name Guest

VLAN Name                             Status    Ports
---- -------------------------------- --------- ----------------
99   Guest                            active    Gi1/0/15, Gi1/0/1

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode
---- ----- ---------- ----- ------ ------ -------- ---- --------
99   enet  100099     1500  -      -      -        -    -

Remote SPAN VLAN
```

```
----------------
Disabled

Primary Secondary Type              Ports
------- --------- ----------------- ----------------------------
```

Key Topic

Access Ports

Access ports are the fundamental building blocks of a managed switch. An access port is assigned to only one VLAN. It carries traffic from the specified VLAN to the device connected to it or from the device to other devices on the same VLAN on that switch. The 802.1Q tags are not included on packets transmitted or received on access ports.

Catalyst switches place switch ports as Layer 2 access ports for VLAN 1 by default. The port can be manually configured as an access port with the command **switchport mode access**. A specific VLAN is associated to the port with the command **switchport access {vlan** *vlan-id* | **name** *name***}**. The ability to set VLANs to an access port by name was recently added with newer code but is stored in numeric form in the configuration.

Example 1-4 demonstrates the configuration of switch ports Gi1/0/15 and Gi1/0/16 as access ports in VLAN 99 for Guests. Notice that the final configuration is stored as numbers for both ports, even though different commands are issued.

Example 1-4 Configuring an Access Port

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# vlan 99
SW1(config-vlan)# name Guests
SW1(config-vlan)# interface gi1/0/15
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 99
SW1(config-if)# interface gi1/0/16
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan name Guest

SW1# show running-config | begin interface GigabitEthernet1/0/15
interface GigabitEthernet1/0/15
 switchport access vlan 99
 switchport mode access
!
interface GigabitEthernet1/0/16
 switchport access vlan 99
 switchport mode access
```

Key Topic

Trunk Ports

Trunk ports can carry multiple VLANs. Trunk ports are typically used when multiple VLANs need connectivity between a switch and another switch, router, or firewall and using only one port. Upon receipt of the packet on the remote trunk link, the headers are examined, traffic is associated to the proper VLAN, then the 802.1Q headers are removed, and traffic is forwarded to the next port, based on MAC address for that VLAN.

> **Note**
>
> Thanks to the introduction of virtualization, some servers run a hypervisor for the operating system and contain a virtualized switch with different VLANs. These servers provide connectivity via a trunk port as well.

Trunk ports are statically defined on Catalyst switches with the interface command **switchport mode trunk**. Example 1-5 displays Gi1/0/2 and Gi1/0/3 being converted to a trunk port.

Example 1-5 Configuring a Trunk Port

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface gi1/0/2
SW1(config-if)# switchport mode trunk
SW1(config-if)# interface gi1/0/3
SW1(config-if)# switchport mode trunk
```

The command **show interfaces trunk** provides a lot of valuable information in several sections for troubleshooting connectivity between network devices:

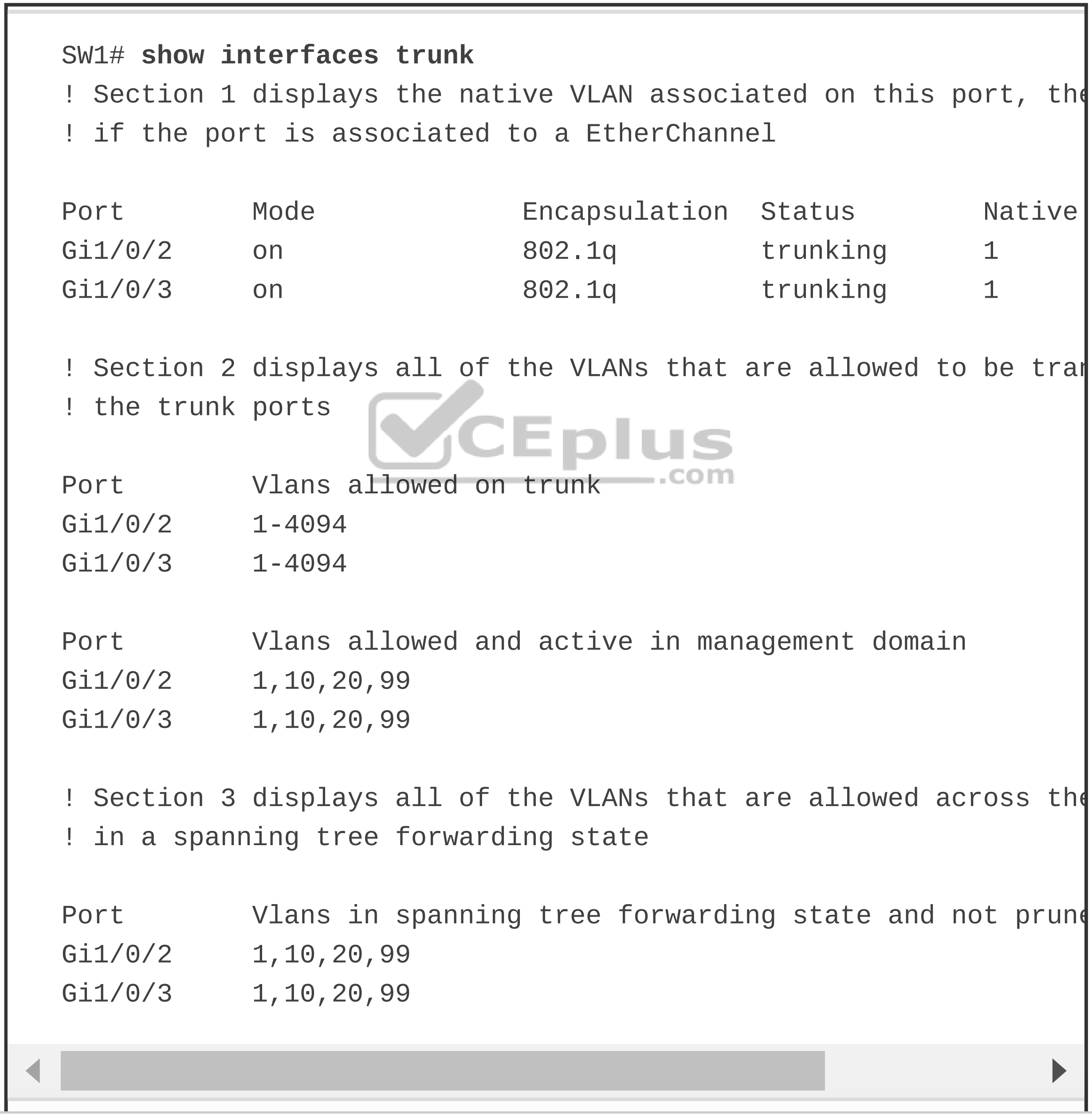
- The first section lists all the interfaces that are trunk ports, the status, the association to an EtherChannel, and whether a VLAN is a native VLAN. Native VLANs are explained in the next section. EtherChannel is explained in Chapter 5, "VLAN Trunks and EtherChannel Bundles."

- The second section of the output displays the list of VLANs that are allowed on the trunk port. Traffic can be minimized on trunk ports to restrict VLANs to specific switches, thereby restricting broadcast traffic, too. Other use cases involve a form of load balancing between network links where select VLANs are allowed on one trunk link, while a different set of VLANs are allowed on a different trunk port.

- The third section displays the VLANs that are in a forwarding state on the switch. Ports that are in blocking state are not listed in this section.

Example 1-6 demonstrates the use of the **show interfaces trunk** command with an explanation of each section.

Example 1-6 Verifying Trunk Port Status



```
SW1# show interfaces trunk
! Section 1 displays the native VLAN associated on this port, the
! if the port is associated to a EtherChannel

Port        Mode             Encapsulation  Status        Native
Gi1/0/2     on               802.1q         trunking      1
Gi1/0/3     on               802.1q         trunking      1

! Section 2 displays all of the VLANs that are allowed to be tran
! the trunk ports

Port        Vlans allowed on trunk
Gi1/0/2     1-4094
Gi1/0/3     1-4094

Port        Vlans allowed and active in management domain
Gi1/0/2     1,10,20,99
Gi1/0/3     1,10,20,99

! Section 3 displays all of the VLANs that are allowed across the
! in a spanning tree forwarding state

Port        Vlans in spanning tree forwarding state and not prune
Gi1/0/2     1,10,20,99
Gi1/0/3     1,10,20,99
```

Native VLANs

In the 802.1Q standard, any traffic that is advertised or received on a trunk port without the 802.1Q VLAN tag is associated to the *native VLAN*. The default native VLAN is VLAN 1. This means that when a switch has two access ports configured as access ports and associated to VLAN 10—that is, a host attached to a trunk port with a native VLAN set to 10—the host could talk to the devices connected to the access ports.

The native VLAN should match on both trunk ports, or traffic can change VLANs unintentionally. While connectivity between hosts is feasible (assuming that they are on the different VLAN numbers), this causes confusion for most network engineers and is not a best practice.

A native VLAN is a port-specific configuration and is changed with the interface command **switchport trunk native vlan** *vlan-id*.

Note

All switch control plane traffic is advertised using VLAN 1. The Cisco security hardening guidelines recommend changing the native VLAN to something other than VLAN 1. More specifically, it should be set to a VLAN that is not used at all (that is, has no hosts attached to it).

Allowed VLANs

As stated earlier, VLANs can be restricted from certain trunk ports as a method of traffic engineering. This can cause problems if traffic between two hosts is expected to traverse a trunk link and the VLAN is not allowed to traverse that trunk port. The interface command **switchport trunk allowed vlan** *vlan-ids* specifies the VLANs that are allowed to traverse the link. Example 1-7 displays sample a configuration for limiting the VLANs that can cross the Gi1/0/2 trunk port for VLANs 1, 10, 20, and 99.

Example 1-7 Viewing the VLANs That Are Allowed on a Trunk Link

```
SW1# show run interface gi1/0/1
interface GigabitEthernet1/0/1
 switchport trunk allowed vlan 1,10,20,99
 switchport mode trunk
```

The full command syntax **switchport trunk allowed {***vlan-ids* | **all** | **none** | **add** *vlan-ids* | **remove** *vlan-ids* | **except** *vlan-ids***}** provides a lot of power in a single command. The optional keyword **all** allows for all VLANs, while **none** removes all VLANS from the trunk link. The **add** keyword adds additional VLANs to those already listed, and the **remove** keyword removes the specified VLAN from the VLANs already identified for that trunk link.

> **Note**
>
> When scripting configuration changes, it is best to use the **add** and **remove** keywords as they are more prescriptive. A common mistake is to use the **switchport trunk allowed vlan** *vlan-ids* command to list only the VLAN that is being added. This results in the current list being overwritten, causing traffic loss for the VLANs that were omitted.

Layer 2 Diagnostic Commands

The information in the “Layer 2 Forwarding” section, earlier in this chapter, provides a brief primer on the operations of a switch. The following sections provide some common diagnostic commands that are used in the daily administration, operation, and troubleshooting of a network.

MAC Address Table

The MAC address table is responsible for identifying the switch ports and VLANs with which a device is associated. A switch builds the MAC address table by examining the source MAC address for traffic that it receives. This information is then maintained to shrink the collision domain (point-to-point communication between devices and switches) by reducing the amount of unknown unicast flooding.

The MAC address table is displayed with the command **show mac address-table [address** *mac-address* | **dynamic** | **vlan** *vlan-id***]**. The optional keywords with this command provide the following benefits:

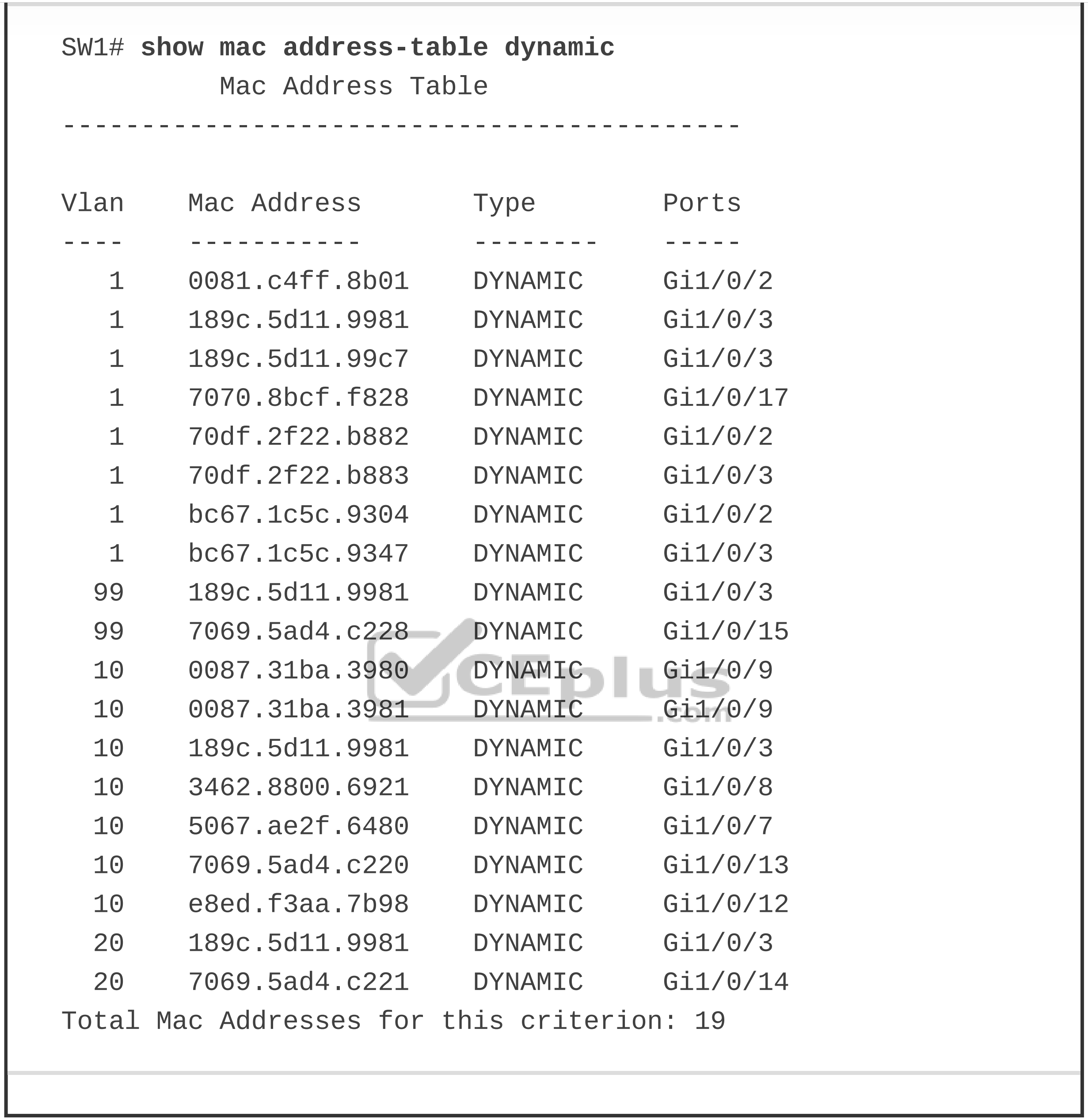
- **address** *mac-address***:** Displays entries that match the explicit MAC address. This command could be beneficial on switches with hundreds of ports.

- **dynamic:** Displays entries that are dynamically learned and are not statically set or burned in on the switch.

- **vlan** *vlan-id***:** Displays entries that matches the specified VLAN.

Example 1-8 shows the MAC address table on a Catalyst. The command in this example displays the VLAN, MAC address, type, and port that the MAC address is connected to. Notice that port Gi1/0/3 has multiple entries, which indicates that this port is connected to a switch.

Example 1-8 Viewing the MAC Address Table



```
SW1# show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0081.c4ff.8b01    DYNAMIC     Gi1/0/2
   1    189c.5d11.9981    DYNAMIC     Gi1/0/3
   1    189c.5d11.99c7    DYNAMIC     Gi1/0/3
   1    7070.8bcf.f828    DYNAMIC     Gi1/0/17
   1    70df.2f22.b882    DYNAMIC     Gi1/0/2
   1    70df.2f22.b883    DYNAMIC     Gi1/0/3
   1    bc67.1c5c.9304    DYNAMIC     Gi1/0/2
   1    bc67.1c5c.9347    DYNAMIC     Gi1/0/3
  99    189c.5d11.9981    DYNAMIC     Gi1/0/3
  99    7069.5ad4.c228    DYNAMIC     Gi1/0/15
  10    0087.31ba.3980    DYNAMIC     Gi1/0/9
  10    0087.31ba.3981    DYNAMIC     Gi1/0/9
  10    189c.5d11.9981    DYNAMIC     Gi1/0/3
  10    3462.8800.6921    DYNAMIC     Gi1/0/8
  10    5067.ae2f.6480    DYNAMIC     Gi1/0/7
  10    7069.5ad4.c220    DYNAMIC     Gi1/0/13
  10    e8ed.f3aa.7b98    DYNAMIC     Gi1/0/12
  20    189c.5d11.9981    DYNAMIC     Gi1/0/3
  20    7069.5ad4.c221    DYNAMIC     Gi1/0/14
Total Mac Addresses for this criterion: 19
```

Note

Troubleshooting network traffic problems from a Layer 2 perspective involves locating the source and destination device and port; this can be done by examining the MAC address table. If multiple MAC addresses appear on the same port, you know that a switch, hub, or server with a virtual switch is connected to that switch port. Connecting to the switch may be required to identify the port that a specific network device is attached to.

Some older technologies (such as load balancing) require a static MAC address entry in the MAC address table to prevent unknown broadcast flooding. The command **mac address-table static mac-address vlan** *vlan-id* {**drop** | **interface** *interface-id*} adds a manual entry with the ability to associate it to a specific switch port or to drop traffic upon receipt.

The command **clear mac address-table dynamic [**{**address** *mac-address* | **interface** *interface-id* | **vlan** *vlan-id*}**]** flushes the MAC address table for the entire switch. Using the optional keywords can flush the MAC address table for a specific MAC address, switch port, or interface.

Key Topic

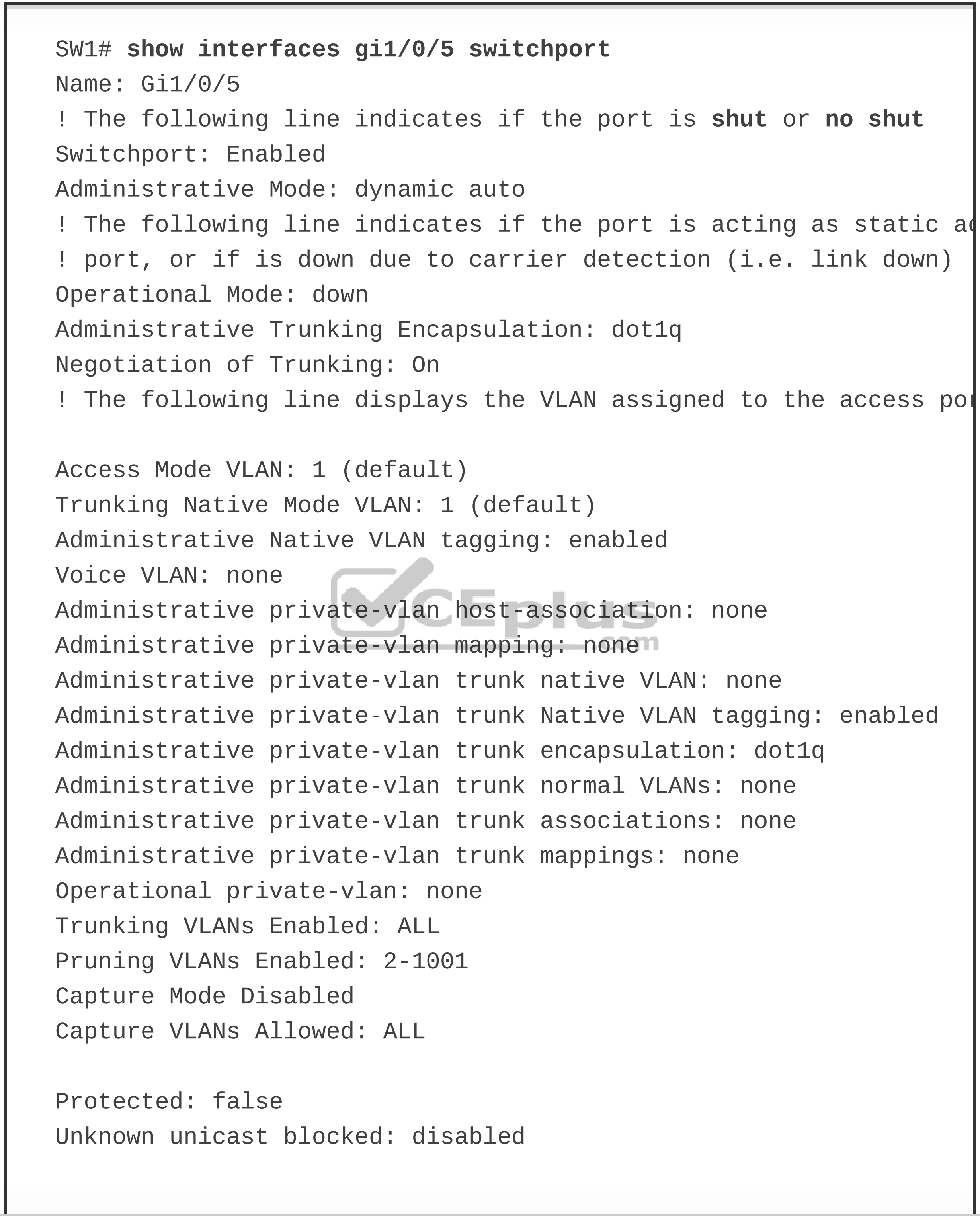
The MAC address table resides in *content addressable memory (CAM)*. The CAM uses high-speed memory that is faster than typical computer RAM due to its search techniques. The CAM table provides a binary result for any query of 0 for true or 1 for false. The CAM is used with other functions to analyze and forward packets very quickly. Switches are built with large CAM to accommodate all the Layer 2 hosts for which they must maintain forwarding tables.

Switch Port Status

Examining the configuration for a switch port can be useful; however, some commands stored elsewhere in the configuration preempt the configuration set on the interface. The command **show interfaces** *interface-id* **switchport** provides all the relevant information for a switch port's status. The command **show interfaces switchport** displays the same information for all ports on the switch.

Example 1-9 shows the output from the **show interfaces gi1/0/5 switchport** command on SW1. The key fields to examine at this time are the switch port state, operational mode, and access mode VLAN.

Example 1-9 Viewing the Switch Port Status



```
SW1# show interfaces gi1/0/5 switchport
Name: Gi1/0/5
! The following line indicates if the port is shut or no shut
Switchport: Enabled
Administrative Mode: dynamic auto
! The following line indicates if the port is acting as static ac
! port, or if is down due to carrier detection (i.e. link down)
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
! The following line displays the VLAN assigned to the access por

Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
```

```
   Unknown multicast blocked: disabled
   Appliance trust: none
```

Interface Status

The command **show interface status** is another useful command for viewing the status of switch ports in a very condensed and simplified manner. Example 1-10 demonstrates the use of this command and includes following fields in the output:

- **Port:** Displays the interface ID or port channel.

- **Name:** Displays the configured interface description.

- **Status:** Displays *connected* for links where a connection was detected and established to bring up the link. Displays *notconnect* for when a link is not detected and *err-disabled* when an error has been detected and the switch has disabled the ability to forward traffic out of that port.

- **VLAN:** Displays the VLAN number assigned for access ports. Trunk links appear as *trunk*, and ports configured as Layer 3 interfaces display *routed*.

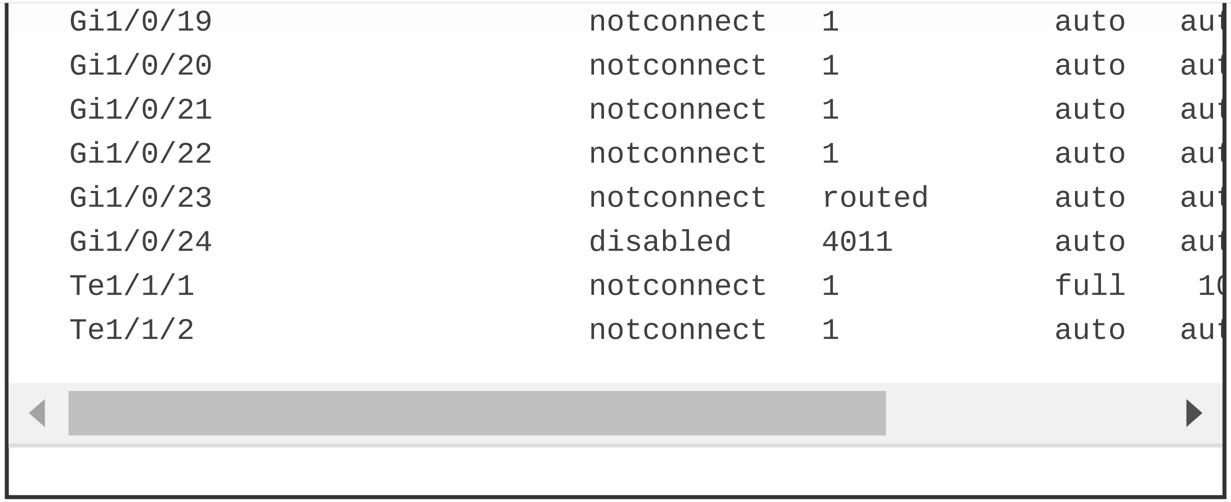
- **Duplex:** Displays the duplex of the port. If the duplex auto-negotiated, it is prefixed by *a-*.

- **Speed:** Displays the speed of the port. If the port speed was auto-negotiated, it is prefixed by *a*-.

- **Type:** Displays the type of interface for the switch port. If it is a fixed RJ-45 copper port, it includes TX in the description (for example, 10/100/1000BASE-TX). Small form-factor pluggable (SFP)–based ports are listed with the SFP model if there is a driver for it in the software; otherwise, it says *unknown*.

Example 1-10 Viewing Overall Interface Status

```
SW1# show interface status

Port       Name               Status       Vlan       Duplex  Spee
Gi1/0/1                       notconnect   1            auto   aut
Gi1/0/2    SW-2 Gi1/0/1       connected    trunk      a-full a-100
Gi1/0/3    SW-3 Gi1/0/1       connected    trunk      a-full a-100
Gi1/0/4                       notconnect   1            auto   aut
Gi1/0/5                       notconnect   1            auto   aut
Gi1/0/6                       notconnect   1            auto   aut
Gi1/0/7    Cube13.C           connected    10         a-full a-100
Gi1/0/8    Cube11.F           connected    10         a-full a-100
Gi1/0/9    Cube10.A           connected    10         a-full  a-10
Gi1/0/10                      notconnect   1            auto   aut
Gi1/0/11                      notconnect   1            auto   aut
Gi1/0/12   Cube14.D Phone     connected    10         a-full a-100
Gi1/0/13   R1-G0/0/0          connected    10         a-full a-100
Gi1/0/14   R2-G0/0/1          connected    20         a-full a-100
Gi1/0/15   R3-G0/1/0          connected    99         a-full a-100
Gi1/0/16   R4-G0/1/1          connected    99         a-full a-100
Gi1/0/17                      connected    1          a-full a-100
Gi1/0/18                      notconnect   1            auto   aut
```



```
Gi1/0/19                      notconnect   1            auto   aut
Gi1/0/20                      notconnect   1            auto   aut
Gi1/0/21                      notconnect   1            auto   aut
Gi1/0/22                      notconnect   1            auto   aut
Gi1/0/23                      notconnect   routed       auto   aut
Gi1/0/24                      disabled     4011         auto   aut
Te1/1/1                       notconnect   1            full    10
Te1/1/2                       notconnect   1            auto   aut
```

Layer 3 Forwarding

Now that we have looked at the mechanisms of a switch and how it forwards Layer 2 traffic, let’s review the process for forwarding a packet from a Layer 3 perspective. Recall that all traffic starts at Layer 7 and works its way down to Layer 1, so some of the *Layer 3 forwarding* logic occurs before Layer 2 forwarding. There are two main methodologies for Layer 3 forwarding:

- Forwarding traffic to devices on the same subnet

- Forwarding traffic to devices on a different subnet

The following sections explain these two methodologies.

Local Network Forwarding

Two devices that reside on the same subnet communicate locally. As the data is encapsulated with its IP address, the device detects that the destination is on the same network. However, the device still needs to encapsulate the Layer 2 information (that is, the source and destination MAC addresses) to the packet. It knows its own MAC address but does not initially know the destination's MAC address.

Key Topic

An *Address Resolution Protocol (ARP)* table provides a method of mapping Layer 3 IP addresses to Layer 2 MAC addresses by storing the IP address of a host and its corresponding MAC address. The device then uses the ARP table to add the appropriate Layer 2 headers to the data packet before sending it down to Layer 2 for processing and forwarding.

For example, an IP host that needs to perform address resolution for another IP host connected by Ethernet can send an ARP request using the LAN broadcast address, and it then waits for an ARP reply from the IP host. The ARP reply includes the required Layer 2 physical MAC address information.

The ARP table contains entries for remote devices that it has communicated with recently and that are on the same IP network segment. It does not contain entries for devices on a remote network but does contain the ARP entry for the IP address of the next hop to reach the remote network. If communication has not occurred with a host after a length of time, the entry becomes stale and is removed from the local ARP table.

If an entry does not exist in the local ARP table, the device broadcasts an ARP request to the entire Layer 2 switching segment. The ARP request strictly asks that whoever owns the IP address in the ARP request reply. All hosts in the Layer 2 segment receive the response, but only the device with the matching IP address should respond to the request.

The response is unicast and includes the MAC and IP addresses of the requestor. The device then updates its local ARP table upon receipt of the ARP reply, adds the appropriate Layer 2 headers, and sends the original data packet down to Layer 2 for processing and forwarding.

Note

The ARP table can be viewed with the command **show ip arp** [*mac-address* | *ip-address* | **vlan** *vlan-id* | *interface-id***]**. The optional keywords make it possible to filter the information.

Packet Routing

Packets must be routed when two devices are on different networks. As the data is encapsulated with its IP address, a device detects that its destination is on a different network and must be routed. The device checks its local routing table to identify its next-hop IP address, which may be learned in one of several ways:

- From a static route entry, it can get the destination network, subnet mask, and next-hop IP address.

- A default-gateway is a simplified static default route that just asks for the local next-hop IP address for all network traffic.

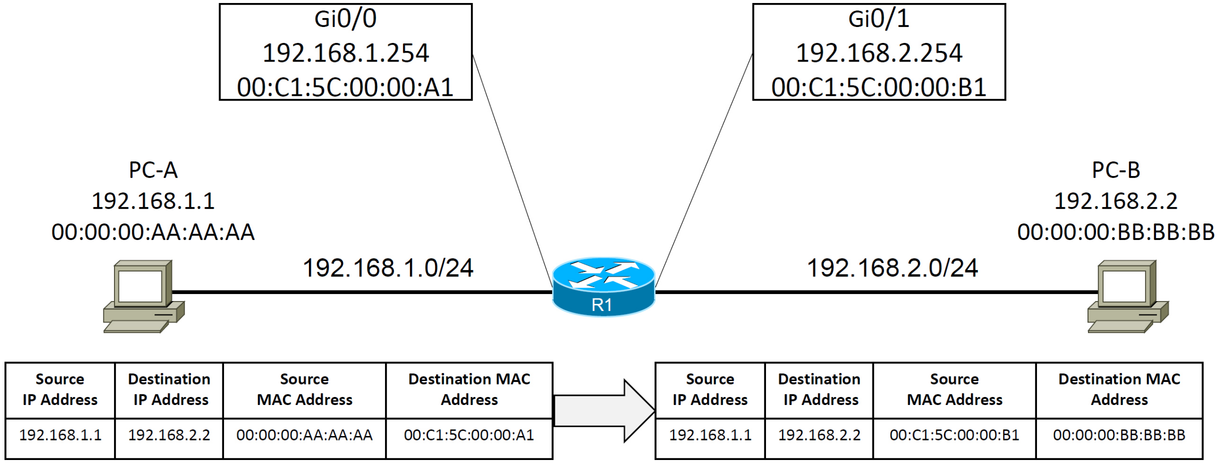
- Routes can be learned from routing protocols.

Key Topic

The source device must add the appropriate Layer 2 headers (source and destination MAC addresses), but the destination MAC address is needed for the next-hop IP address. The device looks for the next-hop IP addresses entry in the ARP table and uses the MAC address from the next-hop IP address’s entry as the destination MAC address. The next step is to send the data packet down to Layer 2 for processing and forwarding.

The next router receives the packet based on the destination MAC address, analyzes the destination IP address, locates the appropriate network entry in its routing table, identifies the outbound interface, and then finds the MAC address for the destination device (or the MAC address for the next-hop address if it needs to be routed further). The router then modifies the source MAC address to the MAC address of the router’s outbound interface and modifies the destination MAC address to the MAC address for the destination device (or next-hop router).

Figure 1-5 illustrates the concept, with PC-A sending a packet to PC-B through an Ethernet connection to R1. PC-A sends the packet to R1’s MAC address, 00:C1:5C:00:00:A1. R1 receives the packet, removes the Layer 2 information, and looks for a route to the 192.168.2.2 address. R1 identifies that connectivity to the 192.168.2.2 IP address is through Gigabit Ethernet 0/1. R1 adds the Layer 2 source address by using its Gigabit Ethernet 0/1 MAC address 00:C1:5C:00:00:B1 and the destination address 00:00:00:BB:BB:BB for PC-B.



Source IP Address	Destination IP Address	Source MAC Address	Destination MAC Address
192.168.1.1	192.168.2.2	00:00:00:AA:AA:AA	00:C1:5C:00:00:A1

Source IP Address	Destination IP Address	Source MAC Address	Destination MAC Address
192.168.1.1	192.168.2.2	00:C1:5C:00:00:B1	00:00:00:BB:BB:BB

Figure 1-5 Layer 2 Addressing Rewrite

Note

This process continues on and on as needed to get the packet from the source device to the destination device.

IP Address Assignment

TCP/IP has become the standard protocol for most networks. Initially it was used with IPv4 and 32-bit network addresses. The number of devices using public IP addresses has increased at an exponential rate and depleted the number of publicly available IP addresses. To deal with the increase in the number of addresses, a second standard, called IPv6, was developed in 1998; it provides 128 bits for addressing. Technologies and mechanisms have been created to allow IPv4 and IPv6 networks to communicate with each other. With either version, an IP address must be assigned to an interface for a router or multilayer switch to route packets.

Key Topic

IPv4 addresses are assigned with the interface configuration command **ip address** *ip-address subnet-mask*. An interface with a configured IP address and that is in an *up* state injects the associated network into the router’s routing table (*Routing Information Base [RIB]*). Connected networks or routes have an *administrative distance (AD)* of zero. It is not possible for any other routing protocol to preempt a connected route in the RIB.

It is possible to attach multiple IPv4 networks to the same interface by attaching a secondary IPv4 address to the same interface with the command **ip address** *ip-address subnet-mask* **secondary**.

IPv6 addresses are assigned with the interface configuration command **ipv6 address** *ipv6-address/prefix-length*. This command can be repeated multiple times to add multiple IPv6 addresses to the same interface.

Example 1-11 demonstrates the configuration of IP addresses on routed interfaces. A routed interface is basically any interface on a router. Notice that a second IPv4 address requires the use of the **secondary** keyword; the **ipv6 address** command can be used multiple times to configure multiple IPv6 addresses.

Example 1-11 Assigning IP Addresses to Routed Interfaces

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gi0/0/0
R1(config-if)# ip address 10.10.10.254 255.255
R1(config-if)# ip address 172.16.10.254 255.255.255.0 secondary
R1(config-if)# ipv6 address 2001:db8:10::254/64
```

```
R1(config-if)# ipv6 address 2001:DB8:10:172::254/64
R1(config-if)# interface gi0/0/1
R1(config-if)# ip address 10.20.20.254 255.255.255.0
R1(config-if)# ip address 172.16.20.254 255.255.255.0 secondary
R1(config-if)# ipv6 address 2001:db8:20::254/64
R1(config-if)# ipv6 address 2001:db8:20:172::254/64
```

Routed Subinterfaces

In the past, there might have been times when multiple VLANs on a switch required routing, and there were not enough physical router ports to accommodate all those VLANs. It is possible to overcome this issue by configuring the switch’s interface as a trunk port and creating logical subinterfaces on a router. A subinterface is created by appending a period and a numeric value after the period. Then the VLAN needs to be associated with the subinterface with the command **encapsulation dot1q** *vlan-id*.

Example 1-12 demonstrates the configuration of two subinterfaces on R2. The subinterface number does not have to match the VLAN ID, but if it does, it helps with operational support.

Example 1-12 Configuring Routed Subinterfaces

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config-if)# int g0/0/1.10
```

```
R2(config-subif)# encapsulation dot1Q 10
R2(config-subif)# ip address 10.10.10.2 255.255.255.0
R2(config-subif)# ipv6 address 2001:db8:10::2/64
R2(config-subif)# int g0/0/1.99
R2(config-subif)# encapsulation dot1Q 99
R2(config-subif)# ip address 10.20.20.2 255.255.255.0
R2(config-subif)# ipv6 address 2001:db8:20::2/64
```

Switched Virtual Interfaces

With Catalyst switches it is possible to assign an IP address to a *switched virtual interface (SVI)*, also known as a *VLAN interface*. An SVI is configured by defining the VLAN on the switch and then defining the VLAN interface with the command **interface vlan** *vlan-id*. The switch must have an interface associated to that VLAN in an *up* state for the SVI to be in an *up* state. If the switch is a multilayer switch, the SVIs can be used for routing packets between VLANs without the need of an external router.

Example 1-13 demonstrates the configuration of the SVI for VLANs 10 and 99.

Example 1-13 Creating a Switched Virtual Interface (SVI)

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface Vlan 10
SW1(config-if)# ip address 10.10.10.1 255.255.255.0
SW1(config-if)# ipv6 address 2001:db8:10::1/64
SW1(config-if)# no shutdown
```

```
SW1(config-if)# interface vlan 99
SW1(config-if)# ip address 10.99.99.1 255.255.255.0
SW1(config-if)# ipv6 address 2001:db8:99::1/64
SW1(config-if)# no shutdown
```

Routed Switch Ports

Some network designs include a point-to-point link between switches for routing. For example, when a switch needs to connect to a router, some network engineers would build out a transit VLAN (for example, VLAN 2001), associate the port connecting to the router to VLAN 2001, and then build an SVI for VLAN 2001. There is always the potential that VLAN 2001 could exist elsewhere in the Layer 2 realm or that spanning tree could impact the topology.

Instead, the multilayer switch port can be converted from a Layer 2 switch port to a routed switch port with the interface configuration command **no switchport**. Then the IP address can be assigned to it. Example 1-14 demonstrates port Gi1/0/14 being converted from a Layer 2 switch port to a routed switch port and then having an IP address assigned to it.

Example 1-14 Configuring a Routed Switch Port

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int gi1/0/14
SW1(config-if)# no switchport
SW1(config-if)# ip address 10.20.20.1 255.255.255.0
```

```
SW1(config-if)# ipv6 address 2001:db8:20::1/64
SW1(config-if)# no shutdown
```

Verification of IP Addresses

IPv4 addresses can be viewed with the command **show ip interface [brief** | *interface-id* | **vlan** *vlan-id***]**. This command’s output contains a lot of useful information, such as MTU, DHCP relay, ACLs, and the primary IP address. The optional **brief** keyword displays the output in a condensed format. However, on devices with large port counts, using the CLI parser and adding an additional **| exclude** field (for example, **unassigned**) yields a streamlined view of interfaces that are configured with IP addresses.

Example 1-15 shows the **show ip interface brief** command used with and without the CLI parser. Notice the drastic reduction in unnecessary data that is presented.

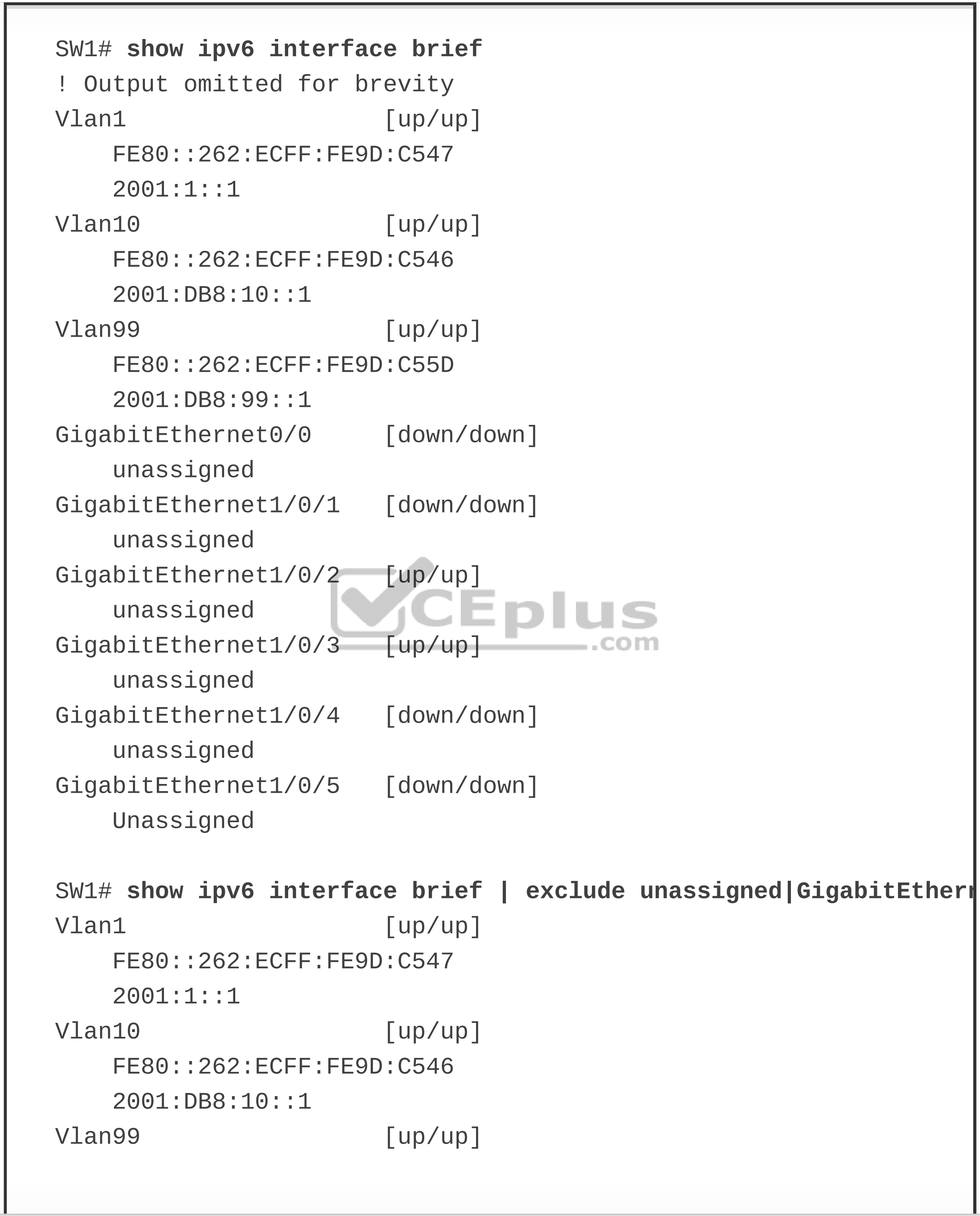
Example 1-15 Viewing Device IPv4 Addresses

```
SW1# show ip interface brief
Interface              IP-Address      OK? Method Status
Vlan1                  unassigned      YES manual up
Vlan10                 10.10.10.1      YES manual up
Vlan99                 10.99.99.1      YES manual up
GigabitEthernet0/0     unassigned      YES unset  down
GigabitEthernet1/0/1   unassigned      YES unset  down
GigabitEthernet1/0/2   unassigned      YES unset  up
```

```
GigabitEthernet1/0/3   unassigned      YES unset  up
GigabitEthernet1/0/4   unassigned      YES unset  down
GigabitEthernet1/0/5   unassigned      YES unset  down
GigabitEthernet1/0/6   unassigned      YES unset  down
GigabitEthernet1/0/7   unassigned      YES unset  up
GigabitEthernet1/0/8   unassigned      YES unset  up
GigabitEthernet1/0/9   unassigned      YES unset  up
GigabitEthernet1/0/10  unassigned      YES unset  down
GigabitEthernet1/0/11  unassigned      YES unset  down
GigabitEthernet1/0/12  unassigned      YES unset  down
GigabitEthernet1/0/13  unassigned      YES unset  up
GigabitEthernet1/0/14  10.20.20.1      YES manual up
GigabitEthernet1/0/15  unassigned      YES unset  up
GigabitEthernet1/0/16  unassigned      YES unset  up
GigabitEthernet1/0/17  unassigned      YES unset  down

SW1# show ip interface brief | exclude unassigned
Interface              IP-Address      OK? Method Status
Vlan10                 10.10.10.1      YES manual up
Vlan99                 10.99.99.1      YES manual up
GigabitEthernet1/0/14  10.20.20.1      YES manual up
GigabitEthernet1/0/23  192.168.1.1     YES manual down
```

The same information can be viewed for IPv6 addresses with the command **show ipv6 interface** [**brief** | *interface-id* | **vlan** *vlan-id*]. Just as with IPv4 addresses, a CLI parser can be used to reduce the information to what is relevant, as demonstrated in Example 1-16.

Example 1-16 Viewing Device IPv6 Addresses



```
SW1# show ipv6 interface brief
! Output omitted for brevity
Vlan1                    [up/up]
    FE80::262:ECFF:FE9D:C547
    2001:1::1
Vlan10                   [up/up]
    FE80::262:ECFF:FE9D:C546
    2001:DB8:10::1
Vlan99                   [up/up]
    FE80::262:ECFF:FE9D:C55D
    2001:DB8:99::1
GigabitEthernet0/0       [down/down]
    unassigned
GigabitEthernet1/0/1     [down/down]
    unassigned
GigabitEthernet1/0/2     [up/up]
    unassigned
GigabitEthernet1/0/3     [up/up]
    unassigned
GigabitEthernet1/0/4     [down/down]
    unassigned
GigabitEthernet1/0/5     [down/down]
    Unassigned

SW1# show ipv6 interface brief | exclude unassigned|GigabitEther
Vlan1                    [up/up]
    FE80::262:ECFF:FE9D:C547
    2001:1::1
Vlan10                   [up/up]
    FE80::262:ECFF:FE9D:C546
    2001:DB8:10::1
Vlan99                   [up/up]
```

```
      FE80::262:ECFF:FE9D:C55D
      2001:DB8:99::1
```

FORWARDING ARCHITECTURES

The first Cisco routers would receive a packet, remove the Layer 2 information, and verify that the route existed for the destination IP address. If a matching route could not be found, the packet was dropped. If a matching route was found, the router would identify and add new Layer 2 header information to the packet.

Advancements in technologies have streamlined the process so that routers do not remove and add the Layer 2 addressing but simply rewrite the addresses. IP packet switching or IP packet forwarding is a faster process for receiving an IP packet on an input interface and making a decision about whether to forward the packet to an output interface or drop it. This process is simple and streamlined so that a router can forward large numbers of packets.

When the first Cisco routers were developed, they used a mechanism called process switching to switch the packets through the routers. As network devices evolved, Cisco created *fast switching* and Cisco Express Forwarding (CEF) to optimize the switching process for the routers to be able to handle larger packet volumes.

Key Topic

Process Switching

Process switching, also referred to as *software switching* or *slow path*, is a switching mechanism in which the general-purpose CPU on a router is in charge of packet switching. In IOS, the ip_input process runs on the general-purpose CPU for processing incoming IP packets. Process switching is the fallback for CEF because it is dedicated to processing punted IP packets when they cannot be switched by CEF.

The types of packets that require software handling include the following:

- Packets sourced or destined to the router (using control traffic or routing protocols)

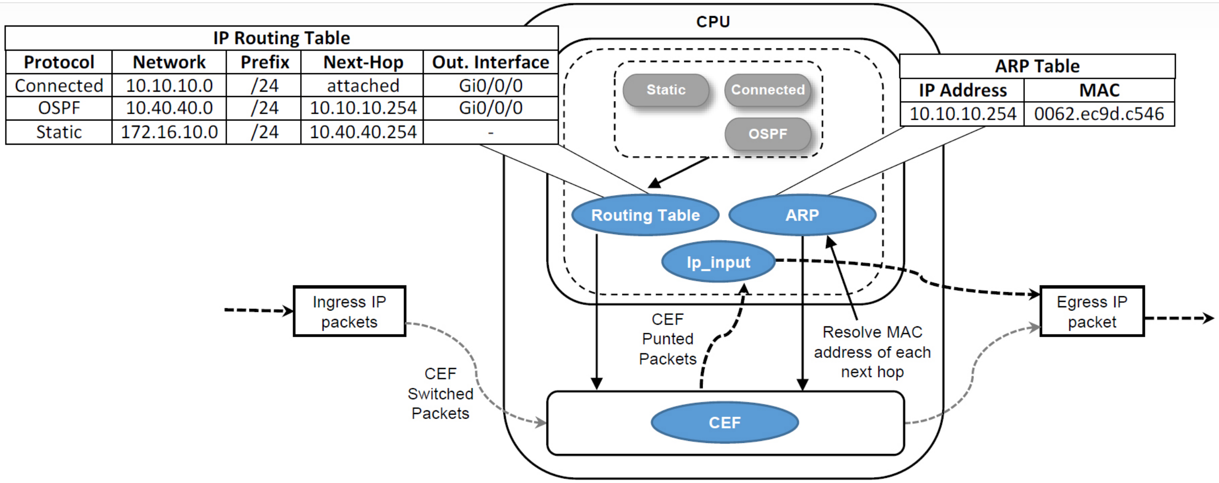
- Packets that are too complex for the hardware to handle (that is, IP packets with IP options)

- Packets that require extra information that is not currently known (for example, ARP)

Note

Software switching is significantly slower than switching done in hardware. The NetIO process is designed to handle a very small percentage of traffic handled by the system. Packets are hardware switched whenever possible.

Figure 1-6 illustrates how a packet that cannot be CEF switched is punted to the CPU for processing. The *ip_input* process consults the routing table and ARP table to obtain the next-hop router's IP address, outgoing interface, and MAC address. It then overwrites the destination MAC address of the packet with the next-hop router's MAC address, overwrites the source MAC address with the MAC address of the outgoing Layer 3 interface, decrements the IP time-to-live (TTL) field, recomputes the IP header checksum, and finally delivers the packet to the next-hop router.



IP Routing Table

Protocol	Network	Prefix	Next-Hop	Out. Interface
Connected	10.10.10.0	/24	attached	Gi0/0/0
OSPF	10.40.40.0	/24	10.10.10.254	Gi0/0/0
Static	172.16.10.0	/24	10.40.40.254	-

ARP Table

IP Address	MAC
10.10.10.254	0062.ec9d.c546

Figure 1-6 Process Switching

The routing table, also known as the *Routing Information Base (RIB)*, is built from information obtained from dynamic routing protocols and directly connected and static routes. The ARP table is built from information obtained from the ARP protocol.

Key Topic

Cisco Express Forwarding

Cisco Express Forwarding (CEF) is a Cisco proprietary switching mechanism developed to keep up with the demands of evolving network infrastructures. It has been the default switching mechanism on most Cisco platforms that do all their packet switching using the general-purpose CPU (software-based routers) since the 1990s, and it is the default switching mechanism used by all Cisco platforms that use specialized application-specific integrated circuits (ASICs) and network processing units (NPUs) for high packet throughput (hardware-based routers).

The general-purpose CPUs on software-based and hardware-based routers are similar and perform all the same functions; the difference is that on software-based routers, the general-purpose CPU is in charge of all operations, including CEF switching (software CEF), and the hardware-based routers do CEF switching using forwarding engines that are implemented in specialized ASICs, ternary content addressable memory (TCAM), and NPUs (hardware CEF). Forwarding engines provide the packet switching, forwarding, and route lookup capability to routers.

Key Topic

Ternary Content Addressable Memory

A switch's *ternary content addressable memory (TCAM)* allows for the matching and evaluation of a packet on more than one field. TCAM is an extension of the CAM architecture but enhanced to allow for upper-layer processing such as identifying the Layer 2/3 source/destination addresses, protocol, QoS markings, and so on. TCAM provides more flexibility in searching than does CAM, which is binary. A TCAM search provides three results: 0 for true, 1 false, and X for do not care, which is a ternary combination.

The TCAM entries are stored in Value, Mask, and Result (VMR) format. The value indicates the fields that should be searched, such as the IP address and protocol fields. The mask indicates the field that is of interest and that should be queried. The result indicates the action that should be taken with a match on the value and mask. Multiple actions can be selected besides allowing or dropping traffic, but tasks like redirecting a flow to a QoS policer or specifying a pointer to a different entry in the routing table are possible.

Most switches contain multiple TCAM entries so that inbound/outbound security, QoS, and Layer 2 and Layer 3 forwarding decisions occur all at once. TCAM operates in hardware, providing faster processing and scalability than process switching. This allows for some features like ACLs to process at the same speed regardless of whether there are 10 entries or 500.

Centralized Forwarding

Given the low cost of a general-purpose CPUs, the price of software-based routers is becoming more affordable, but at the expense of total packet throughput.

When a route processor (RP) engine is equipped with a forwarding engine so that it can make all the packet switching decisions, this is known as a *centralized forwarding architecture*. If the line cards are equipped with forwarding engines so that they can make packet switching decision without intervention of the RP, this is known as a *distributed forwarding architecture*.

For a centralized forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the forwarding engine on the RP. The forwarding engine examines the packet's headers and determines that the packet will be sent out a port on the egress line card and forwards the packet to the egress line card to be forwarded.

Distributed Forwarding

For a distributed forwarding architecture, when a packet is received on the ingress line card, it is transmitted to the local forwarding engine. The forwarding engine performs a packet lookup, and if it determines that the outbound interface is local, it forwards the packet out a local interface. If the outbound interface is located on a different line card, the packet is sent across the switch fabric, also known as the backplane, directly to the egress line card, bypassing the RP.

Figure 1-7 shows the difference between centralized and distributed forwarding architectures.

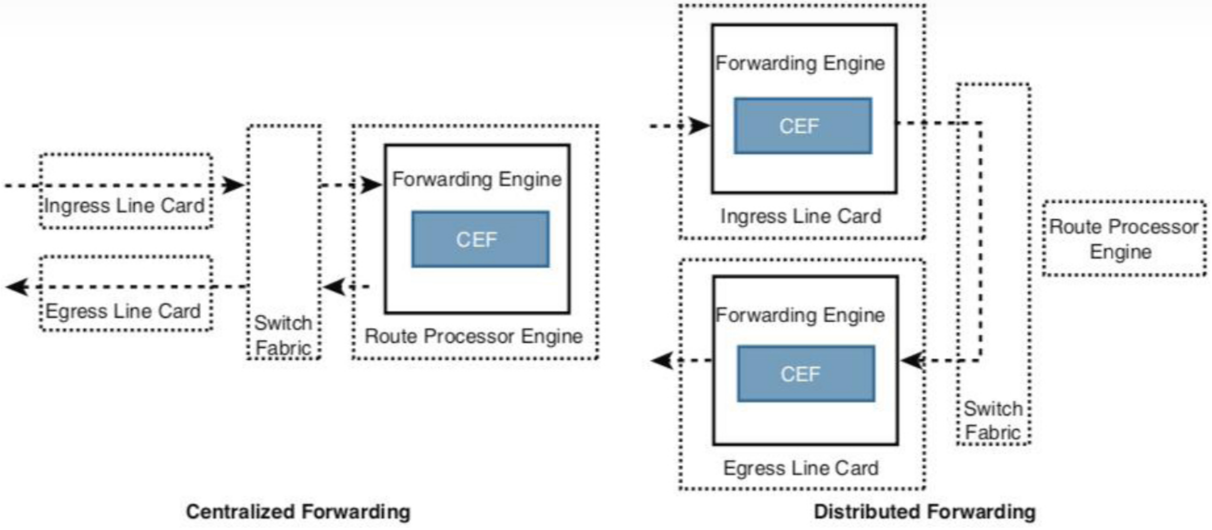


Figure 1-7 Centralized Versus Distributed Forwarding Architectures

Key Topic

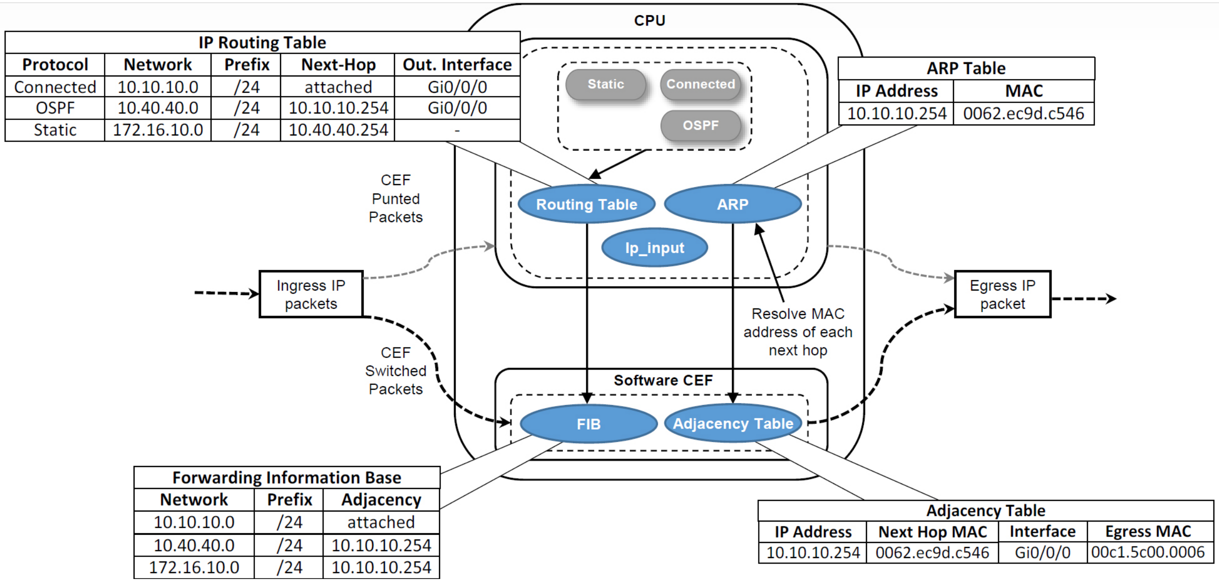
Software CEF

Software CEF, also known as the *software Forwarding Information Base*, consists of the following components:

- **Forwarding Information Base:** The FIB is built directly from the routing table and contains the next-hop IP address for each destination in the network. It keeps a mirror image of the forwarding information contained in the IP routing table. When a routing or topology change occurs in the network, the IP routing table is updated, and these changes are reflected in the FIB. CEF uses the FIB to make IP destination prefix-based switching decisions.

- **Adjacency table:** The adjacency table, also known as the Adjacency Information Base (AIB), contains the directly connected next-hop IP addresses and their corresponding next-hop MAC addresses, as well as the egress interface's MAC address. The adjacency table is populated with data from the ARP table or other Layer 2 protocol tables.

Figure 1-8 illustrates how the CEF table is built from the routing table. First, the FIB is built from the routing table. The 172.16.10.0/24 prefix is a static route with a next hop of 10.40.40.254, which is dependent upon the 10.40.40.0/24 prefix learned via OSPF. The adjacency pointer in the FIB for the 172.16.10.0/24 entry is exactly the same IP address OSPF uses for the 10.40.40.0/24 prefix (10.10.10.254). The adjacency table is then built using the ARP table and cross-referencing the MAC address with the MAC address table to identify the outbound interface.



IP Routing Table

Protocol	Network	Prefix	Next-Hop	Out. Interface
Connected	10.10.10.0	/24	attached	Gi0/0/0
OSPF	10.40.40.0	/24	10.10.10.254	Gi0/0/0
Static	172.16.10.0	/24	10.40.40.254	-

ARP Table

IP Address	MAC
10.10.10.254	0062.ec9d.c546

Forwarding Information Base

Network	Prefix	Adjacency
10.10.10.0	/24	attached
10.40.40.0	/24	10.10.10.254
172.16.10.0	/24	10.10.10.254

Adjacency Table

IP Address	Next Hop MAC	Interface	Egress MAC
10.10.10.254	0062.ec9d.c546	Gi0/0/0	00c1.5c00.0006

Figure 1-8 CEF Switching

Upon receipt of an IP packet, the FIB is checked for a valid entry. If an entry is missing, it is a “glean” adjacency in CEF, which means the packet should go to the CPU because CEF is unable to handle it. Valid FIB entries continue processing by checking the adjacency table for each packet’s destination IP address. Missing adjacency entries invoke the ARP process. Once ARP is resolved, the complete CEF entry can be created.

As part of the packet forwarding process, the packet’s headers are rewritten. The router overwrites the destination MAC address of a packet with the next-hop router's MAC address from the adjacency table, overwrites the source MAC address with the MAC address of the outgoing Layer 3 interface, decrements the IP time-to-live (TTL) field, recomputes the IP header checksum, and finally delivers the packet to the next-hop router.

Note

Packets processed by the CPU are typically subject to a rate limiter when an invalid or incomplete adjacency exists to prevent the starving of CPU cycles from other essential processes.

Note

The TTL is a Layer 3 loop prevention mechanism that reduces a packet's TTL field by 1 for every Layer 3 hop. If a router receives a packet with a TTL of 0, the packet is discarded.

Hardware CEF

The ASICs in hardware-based routers are expensive to design, produce, and troubleshoot. ASICs allow for very high packet rates, but the trade-off is that they are limited in their functionality because they are hardwired to perform specific tasks. There routers are equipped with NPUs that are designed to overcome the inflexibility of ASICs. Unlike ASICs, NPUs are programmable, and their firmware can be changed with relative ease.

The main advantage of the distributed forwarding architectures is that the packet throughput performance is greatly improved by offloading the packet switching responsibilities to the line cards. Packet switching in distributed architecture platforms is done via distributed CEF (dCEF), which is a mechanism in which the CEF data structures are downloaded to forwarding ASICs and the CPUs of all line cards so that they can participate in packet switching; this allows for the switching to be done at the distributed level, thus increasing the packet throughput of the router.

Note

Software CEF in hardware-based platforms is not used to do packet switching as in software-based platforms; instead, it is used to program the hardware CEF.

Stateful Switchover

Routers specifically designed for high availability include hardware redundancy, such as dual power supplies and route processors (RPs). An RP is responsible for learning the network topology and building the route table (RIB). An RP failure can trigger routing protocol adjacencies to reset, resulting in packet loss and network instability. During an RP failure, it may be more desirable to hide the failure and allow the router to continue forwarding packets using the previously programmed CEF table entries rather than temporarily drop packets while waiting for the secondary RP to reestablish the routing protocol adjacencies and rebuild the forwarding table.

Stateful switchover (SSO) is a redundancy feature that allows a Cisco router with two RPs to synchronize router configuration and control plane state information. The process of mirroring information between RPs is referred to as *checkpointing*. SSO-enabled routers always checkpoint line card operation and Layer 2 protocol states. During a switchover, the standby RP immediately takes control and prevents basic problems such as interface link flaps. However, Layer 3 packet forwarding is disrupted without additional configuration. The RP switchover triggers a routing protocol adjacency flap that clears the route table. When the routing table is cleared, the CEF entries are purged, and traffic is no longer routed until the network topology is relearned and the forwarding table is reprogrammed. Enabling nonstop forwarding (NSF) or nonstop routing (NSR) high availability capabilities informs the router(s) to maintain the CEF entries for a short duration and continue forwarding packets through an RP failure until the control plane recovers.

Key Topic

SDM Templates

The capacity of MAC addresses that a switch needs compared to the number of routes that it holds depends on where it is deployed in the network. The memory used for TCAM tables is limited and statically allocated during the bootup sequence of the switch. When a section of a hardware resource is full, all processing overflow is sent to the CPU, which seriously impacts the performance of the switch.

The allocation ratios between the various TCAM tables are stored and can be modified with Switching Database Manager (SDM) templates. Multiple Cisco switches exist, and the SDM template can be configured on Catalyst 9000 switches with the global configuration command **sdm prefer** {**vlan** | **advanced**}. The switch must then be restarted with the **reload** command.

Note

Every switch in a switch stack must be configured with the same SDM template.

Table 1-2 shows the approximate number of resources available per template. This could vary based on the switch platform or software version in use. These numbers are typical for Layer 2 and IPv4 features. Some features, such as IPv6, use twice the entry size, which means only half as many entries can be created.

Table 1-2 Approximate Number of Feature Resources Allowed by Templates

Resource	Advanced	VLAN
Number of VLANs	4094	4094
Unicast MAC addresses	32,000	32,000
Overflow unicast MAC addresses	512	512
IGMP groups and multicast routes	4000	4000
Overflow IGMP groups and multicast routes	512	512
[lb] Directly connected routes	16,000	16,000
[lb] Indirectly connected IP hosts	7000	7000
Policy-based routing access control entry (ACE)s	1024	0
QoS classification ACEs	3000	3000
Security ACEs	3000	3000
NetFlow ACEs	1024	1024
Input Microflow policer ACEs	256,000	0
Output Microflow policer ACEs	256,000	0
FSPAN ACEs	256	256
Control Plane Entries	512	512

The current SDM template can viewed with the command **show sdm prefer**, as demonstrated in Example 1-17.

Example 1-17 Viewing the Current SDM Template

```
SW1# show sdm prefer
Showing SDM Template Info
```

```
This is the Advanced (high scale) template.
  Number of VLANs:                                 4094
  Unicast MAC addresses:                           32768
  Overflow Unicast MAC addresses:                  512
  IGMP and Multicast groups:                       4096
  Overflow IGMP and Multicast groups:              512
  Directly connected routes:                       16384
  Indirect routes:                                 7168
  Security Access Control Entries:                 3072
  QoS Access Control Entries:                      2560
  Policy Based Routing ACEs:                       1024
  Netflow ACEs:                                    768
  Wireless Input Microflow policer ACEs:           256
  Wireless Output Microflow policer ACEs:          256
  Flow SPAN ACEs:                                  256
  Tunnels:                                         256
  Control Plane Entries:                           512
  Input Netflow flows:                             8192
  Output Netflow flows:                            16384
  SGT/DGT and MPLS VPN entries:                    3840
  SGT/DGT and MPLS VPN Overflow entries:           512
These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
```

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 1-3 lists these key topics and the page number on which each is found.

Key Topic

Table 1-3 Key Topics for Chapter 1

Key Topic Element	Description	Page
Paragraph	Collision domain	
Paragraph	Virtual LANs (VLANs)	
Section	Access ports	
Section	Trunk ports	
Paragraph	Content addressable memory	
Paragraph	Address resolution protocol (ARP)	
Paragraph	Packet Routing	
Paragraph	IP address assignment	
Section	Process switching	
Section	Cisco Express Forwarding (CEF)	
Section	Ternary content addressable memory	
Section	Software CEF	
Section	SDM templates	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

access port

Address Resolution Protocol (ARP)

broadcast domain

Cisco Express Forwarding (CEF)

collision domain

content addressable memory (CAM)

Layer 2 forwarding

Layer 3 forwarding

Forwarding Information Base (FIB)

MAC address table

native VLAN

process switching

Routing Information Base (RIB)

trunk port

ternary content addressable memory (TCAM)

virtual LAN (VLAN)

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 1-4 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 1-4 Command Reference

Task	Command Syntax
Define a VLAN	**vlan** *vlan-id* **name** *vlanname*
Configure an interface as a trunk port	**switchport mode trunk**
Configure an interface as an access port assigned to a specific VLAN	**switchport mode access** **switchport access** {**vlan** *vlan-id* | **name** *name*}
Configure a static MAC address entry	**mac address-table static mac-address vlan** *vlan-id* **interface** *interface-id*
Clear MAC addresses from the MAC address table	**clear mac address-table dynamic** [{**address** *mac-address* | **interface** *interface-id* | **vlan** *vlan-id*}]
Assign an IPv4 address to an interface	**ip address** *ip-address subnet-mask*
Assign a secondary IPv4 address to an interface	**ip address** *ip-address subnet-mask* **secondary**
Assign an IPv6 address to an interface	**ipv6 address** *ipv6-address/prefix-length*
Modify the SDM database	**sdm prefer** {**vlan** | **advanced**}
Display the interfaces that are configured as a trunk port and all the VLANs that they permit	**show interfaces trunk**
Display the list of VLANs and their associated ports	**show vlan** [{**brief** | **id** *vlan-id* | **name** *vlanname* | **summary**}]
Display the MAC address table for a switch	**show mac address-table** [**address** *mac-address* | **dynamic** | **vlan** *vlan-id*]
Display the current interface state, including duplex, speed, and link state	**show interfaces**
Display the Layer 2 configuration information for a specific switchport	**show interfaces** *interface-id* **switchport**
Display the ARP table	**show ip arp** [*mac-address* | *ip-address* | **vlan** *vlan-id* | *interface-id*].
Displays the IP interface table	**show ip interface** [**brief** | *interface-id* | **vlan** *vlan-id*]
Display the IPv6 interface table	**show ipv6 interface** [**brief** | *interface-id* | **vlan** *vlan-id*]

Part II: Layer 2

Chapter 2. Spanning Tree Protocol

This chapter covers the following subjects:

- **Spanning Tree Protocol Fundamentals:** This section provides an overview of how switches become aware of other switches and prevent forwarding loops.

- **Rapid Spanning Tree Protocol:** This section examines the improvements made to STP for faster convergence.

A good network design provides redundancy in devices and network links (that is, paths). The simplest solution involves adding a second link between switches to overcome a network link failure or ensuring that a switch is connected to at least two other switches in a topology. However, such topologies cause problems when a switch must forward broadcasts or when unknown unicast flooding occurs. Network broadcasts forward in a continuous loop until the link becomes saturated, and the switch is forced to drop packets. In addition, the MAC address table must constantly change ports as the packets make loops. The packets continue to loop around the topology because there is not a time-to-live (TTL) mechanism for Layer 2 forwarding. The switch CPU utilization increases, as does memory consumption, which could result in the crashing of the switch.

This chapter explains how switches prevent forwarding loops while allowing for redundant links with the use of Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP). Two other chapters also explain STP-related topics:

- **Chapter 3, "Advanced STP Tuning":** Covers advanced STP topics such as BPDU guard and BPDU filter.

- **Chapter 4, "Multiple Spanning Tree Protocol":** Covers Multiple Spanning Tree Protocol.

"DO I KNOW THIS ALREADY?" QUIZ

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

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

Foundation Topic Section	Questions
Spanning Tree Protocol Fundamentals	1–6
Rapid Spanning Tree Protocol	7–9

1. How many different BPDU types are there?

a. One

b. Two

c. Three

d. Fourth

2. What attributes are used to elect a root bridge?

a. Switch port priority

b. Bridge priority

c. Switch serial number

d. Path cost

3. The original 802.1D specification assigns what value to a 1 Gbps interface?

a. 1

b. 2

c. 4

d. 19

4. All of the ports on a root bridge are assigned what role?

a. Root port

b. Designated port

c. Superior port

d. Master port

5. Using default settings, how long does a port stay in the listening state?

a. 2 seconds

b. 5 seconds

c. 10 seconds

d. 15 seconds

6. Upon receipt of a configuration BPDU with the topology change flag set, how do the downstream switches react?

a. By moving all ports to a blocking state on all switches

b. By flushing out all MAC addresses from the MAC address table

c. By temporarily moving all non-root ports to a listening state

d. By flushing out all old MAC addresses from the MAC address table

e. By updating the Topology Change version flag on the local switch database

7. Which of the following is not an RSTP port state?

a. Blocking

b. Listening

c. Learning

d. Forwarding

8. True or false: In a large Layer 2 switch topology, the infrastructure must fully converge before any packets can be forwarded.

a. True

b. False

9. True or false: In a large Layer 2 switch topology that is running RSTP, the infrastructure must fully converge before any packets can be forwarded.

a. True

b. False

Answers to the “Do I Know This Already?” quiz:

1. B

2. B

3. C

4. B

5. D

6. D

7. A, B

8. B

9. B

FOUNDATION TOPICS

SPANNING TREE PROTOCOL FUNDAMENTALS

Spanning Tree Protocol (STP) enables switches to become aware of other switches through the advertisement and receipt of bridge protocol data units (BPDUs). STP builds a Layer 2 loop-free topology in an environment by temporarily blocking traffic on redundant ports. STP operates by selecting a specific switch as the master switch and running a tree-based algorithm to identify which redundant ports should not forward traffic.

STP has multiple iterations:

- 802.1D, which is the original specification

- Per-VLAN Spanning Tree (PVST)
- Per-VLAN Spanning Tree Plus (PVST+)
- 802.1W Rapid Spanning Tree Protocol (RSTP)
- 802.1S Multiple Spanning Tree Protocol (MST)

Catalyst switches now operate in PVST+, RSTP, and MST modes. All three of these modes are backward compatible with 802.1D.

IEEE 802.1D STP

The original version of STP comes from the IEEE 802.1D standards and provides support for ensuring a loop-free topology for one VLAN. This topic is vital to understand as a foundation for Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MST).

802.1D Port States

In the 802.1D STP protocol, every port transitions through the following states:

- **Disabled:** The port is in an administratively off position (that is, shut down).

- **Blocking:** The switch port is enabled, but the port is not forwarding any traffic to ensure that a loop is not created. The switch does not modify the MAC address table. It can only receive BPDUs from other switches.

- **Listening:** The switch port has transitioned from a blocking state and can now send or receive BPDUs. It cannot forward any other network traffic. The duration of the state correlates to the STP forwarding time. The next port state is learning.

- **Learning:** The switch port can now modify the MAC address table with any network traffic that it receives. The switch still does not forward any other network traffic besides BPDUs. The duration of the state correlates to the STP forwarding time. The next port state is forwarding.

- **Forwarding:** The switch port can forward all network traffic and can update the MAC address table as expected. This is the final state for a switch port to forward network traffic.

- **Broken:** The switch has detected a configuration or an operational problem on a port that can have major effects. The port discards packets as long as the problem continues to exist.

> **Note**
>
> The entire 802.1D STP initialization time takes about 30 seconds for a port to enter the forwarding state using default timers.

802.1D Port Types

Key Topic

The 802.1D STP standard defines the following three port types:

- **Root port (RP):** A network port that connects to the root bridge or an upstream switch in the spanning-tree topology. There should be only one root port per VLAN on a switch.

- **Designated port (DP):** A network port that receives and forwards BPDU frames to other switches. Designated ports provide connectivity to downstream devices and switches. There should be only one active designated port on a link.

- **Blocking port:** A network that is not forwarding traffic because of STP calculations.

Key Topic

STP Key Terminology

Several key terms are related to STP:

- **Root bridge:** The root bridge is the most important switch in the Layer 2 topology. All ports are in a forwarding state. This switch is considered the top of the spanning tree for all path calculations by other switches. All ports on the root bridge are categorized as designated ports.

- **Bridge protocol data unit (BPDU):** This network packet is used for network switches to identify a hierarchy and notify of changes in the topology. A BPDU uses the destination MAC address 01:80:c2:00:00:00. There are two types of BPDUs:

- **Configuration BPDU:** This type of BPDU is used to identify the root bridge, root ports, designated ports, and blocking ports. The configuration BPDU consists of the following fields: STP type, root path cost, root bridge identifier, local bridge identifier, max age, hello time, and forward delay.

- **Topology change notification (TCN) BPDU:** This type of BPDU is used to communicate changes in the Layer 2 topology to other switches. This is explained in greater detail later in the chapter.

- **Root path cost:** This is the combined cost for a specific path toward the root switch.

- **System priority:** This 4-bit value indicates the preference for a switch to be root bridge. The default value is 32,768.

- **System ID extension:** This 12-bit value indicates the VLAN that the BPDU correlates to. The system priority and system ID extension are combined as part of the switch's identification of the root bridge.

- **Root bridge identifier:** This is a combination of the root bridge system MAC address, system ID extension, and system priority of the root bridge.

- **Local bridge identifier:** This is a combination of the local switch's bridge system MAC address, system ID extension, and system priority of the root bridge.

- **Max age:** This is the maximum length of time that passes before a bridge port saves its BPDU information. The default value is 20 seconds, but the value can be configured with the command **spanning-tree vlan** *vlan-id* **max-age** *maxage*. If a switch loses contact with the BPDU's source, it assumes that the BPDU information is still valid for the duration of the Max Age timer.

- **Hello time:** This is the time that a BPDU is advertised out of a port. The default value is 2 seconds, but the value can be configured to 1 to 10 seconds with the command **spanning-tree vlan** *vlan-id* **hello-time** *hello-time*.

- **Forward delay:** This is the amount of time that a port stays in a listening and learning state. The default value is 15 seconds, but the value can be changed to a value of 15 to 30 seconds with the command **spanning-tree vlan** *vlan-id* **forward-time** *forward-time*.

Note

STP was defined before modern switches existed. The devices that originally used STP were known as bridges. Switches perform the same role at a higher speed and scale while essentially bridging Layer 2 traffic. The terms *bridge* and *switch* are interchangeable in this context.

Spanning Tree Path Cost

The interface STP cost is an essential component for root path calculation because the root path is found based on the cumulative interface STP cost to reach the root bridge. The interface STP cost was originally stored as a 16-bit value with a reference value of 20 Gbps. As switches have developed with higher-speed interfaces, 10 Gbps might not be enough. Another method, called *long mode*, uses a 32-bit value and uses a reference speed of 20 Tbps. The original method, known as *short mode*, is the default mode.

Table 2-2 displays a list of interface speeds and the correlating interface STP costs.

Table 2-2 Default Interface STP Port Costs

Link Speed	Short-Mode STP Cost	Long-Mode STP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2,000
20 Gbps	1	1,000
100 Gbps	1	200
1 Tbps	1	20
10 Tbps	1	2

Devices can be configured with the long-mode interface cost with the command **spanning-tree pathcost method long**. The entire Layer 2 topology should use the same setting for every device in the environment to ensure a consistent topology. Before enabling this setting in an environment, it is important to conduct an audit to ensure that the setting will work.

Building the STP Topology

This section focuses on the logic switches use to build an STP topology. Figure 2-1 shows the simple topology used here to demonstrate some important spanning tree concepts. The configurations on all the switches do not include any customizations for STP, and the focus is primarily on VLAN 1, but VLANs 10, 20, and 99 also exist in the topology. SW1 has been identified as the root bridge, and the RP, DP, and blocking ports have been identified visually to assist in the following sections.

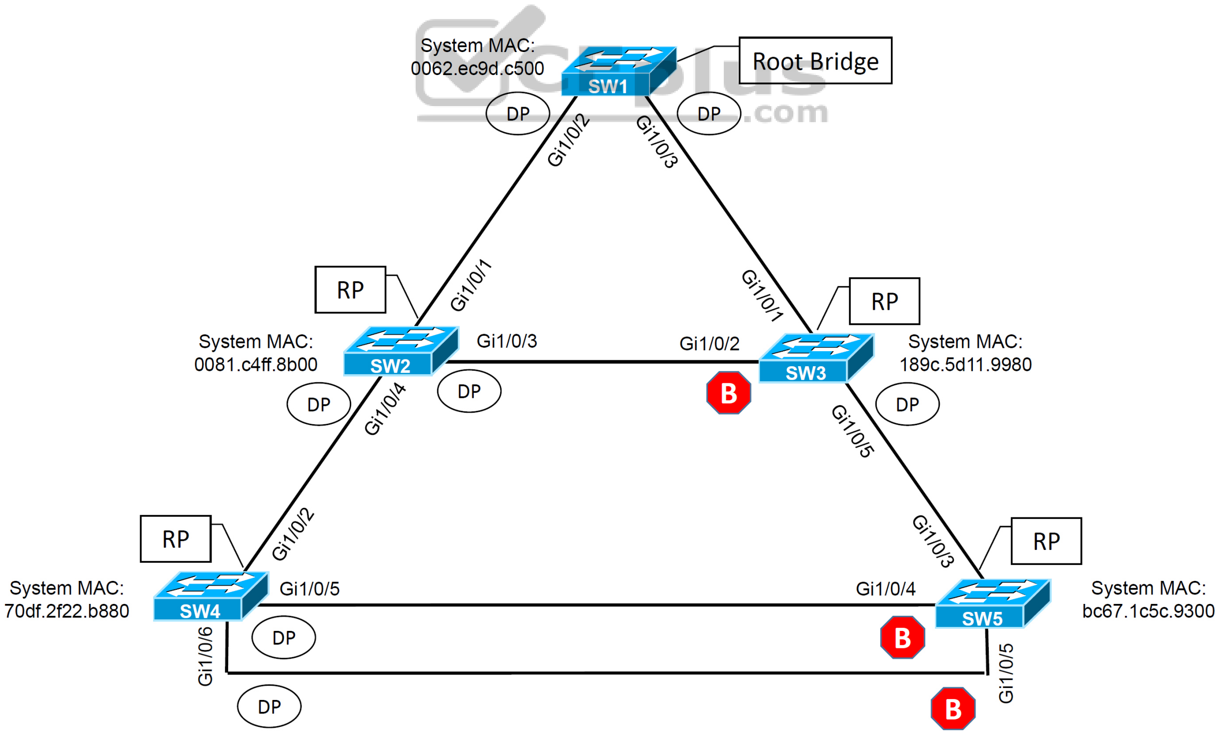


Figure 2-1 Basic STP Topology

Key Topic

Root Bridge Election

The first step with STP is to identify the root bridge. As a switch initializes, it assumes that it is the root bridge and uses the local bridge identifier as the root bridge identifier. It then listens to its neighbor’s configuration BPDU and does the following:

- If the neighbor’s configuration BPDU is inferior to its own BPDU, the switch ignores that BPDU.

- If the neighbor’s configuration BPDU is preferred to its own BPDU, the switch updates its BPDUs to include the new root bridge identifier along with a new root path cost that correlates to the total path cost to reach the new root bridge. This process continues until all switches in a topology have identified the root bridge switch.

STP deems a switch more preferable if the priority in the bridge identifier is lower than the priority of the other switch’s configuration BPDUs. If the priority is the same, then the switch prefers the BPDU with the lower system MAC.

Note

Generally, older switches have a lower MAC address and are considered more preferable. Configuration changes can be made for optimizing placement of the root switch in a Layer 2 topology.

In Figure 2-1, SW1 can be identified as the root bridge because its system MAC address (0062.ec9d.c500) is the lowest in the topology. This is further verified by using the command **show spanning-tree root** to display the root bridge. Example 2-1 demonstrates this command being executed on SW1. The output includes the VLAN number, root bridge identifier, root path cost, hello time, max age time, and forwarding delay. Because SW1 is the root bridge, all ports are designated ports, so the Root Port field is empty. This is one way to verify that the connected switch is the root bridge for the VLAN.

Example 2-1 Verifying the STP Root Bridge

```
SW1# show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                     Root ID        Cost    Time  Age Dly  Ro
---------------- -------------------- --------- ----- --- ---  --
VLAN0001         32769 0062.ec9d.c500         0     2  20  15
VLAN0010         32778 0062.ec9d.c500         0     2  20  15
```

```
VLAN0020         32788 0062.ec9d.c500         0    2   20  15
VLAN0099         32867 0062.ec9d.c500         0    2   20  15
```

In Example 2-1, notice that the root bridge priority on SW1 for VLAN 1 is 32,769 and not 32,768. The priority in the configuration BPDU packets is actually the priority plus the value of the *sys-id-ext* (which is the VLAN number). You can confirm this by looking at VLAN 10, which has a priority of 32,778, which is 10 higher than 32,768.

The advertised root path cost is always the value calculated on the local switch. As the BPDU is received, the local root path cost is the advertised root path cost plus the local interface port cost. The root path cost is always zero on the root bridge. Figure 2-2 illustrates the root path cost as SW1 advertises the configuration BPDUs toward SW3 and then SW3's configuration BPDUs toward SW5.

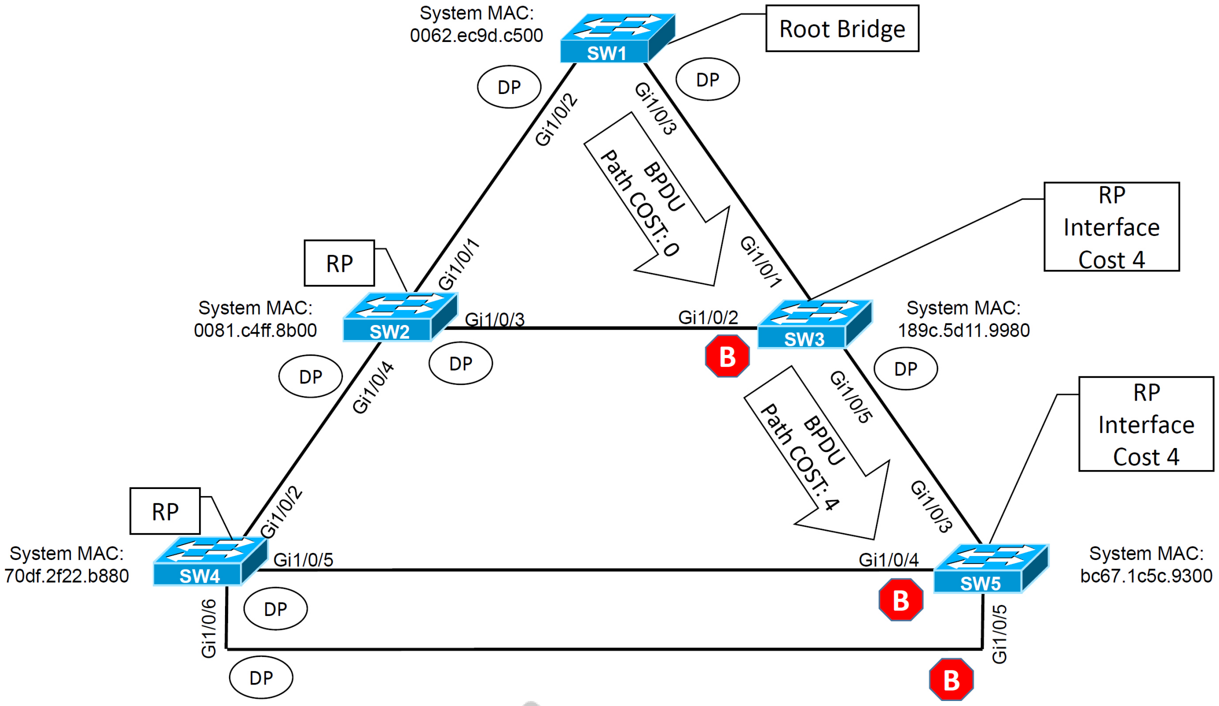


Figure 2-2 STP Path Cost Advertisements

Example 2-2 shows the output of the **show spanning-tree root** command run on SW2 and SW3. The Root ID field is exactly the same as for SW1, but the root path cost has changed to 4 because both switches must use the 1 Gbps link to reach SW1. Gi1/0/1 has been identified on both switches as the root port.

Example 2-2 Identifying the Root Ports

```
SW2# show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  R
---------------- -------------------- --------- ----- --- ---  -
VLAN0001         32769 0062.ec9d.c500         4    2   20  15  G
```

```
VLAN0010         32778 0062.ec9d.c500         4    2   20  15   Gi
VLAN0020         32788 0062.ec9d.c500         4    2   20  15   Gi
VLAN0099         32867 0062.ec9d.c500         4    2   20  15   Gi

SW3# show spanning-tree root
                                         Root    Hello Max Fwd
Vlan                   Root ID           Cost    Time  Age Dly  Ro
---------------- -------------------- --------- ----- --- ---  --
VLAN0001         32769 0062.ec9d.c500         4    2   20  15   Gi
VLAN0010         32778 0062.ec9d.c500         4    2   20  15   Gi
VLAN0020         32788 0062.ec9d.c500         4    2   20  15   Gi
VLAN0099         32867 0062.ec9d.c500         4    2   20  15   Gi
```

Key Topic

Locating Root Ports

After the switches have identified the root bridge, they must determine their root port (RP). The root bridge continues to advertise configuration BPDUs out all of its ports. The switch compares the BPDU information to identify the RP. The RP is selected using the following logic (where the next criterion is used in the event of a tie):

1. The interface associated to lowest path cost is more preferred.

2. The interface associated to the lowest system priority of the advertising switch is preferred next.

3. The interface associated to the lowest system MAC address of the advertising switch is preferred next.

4. When multiple links are associated to the same switch, the lowest port priority from the advertising router is preferred.

5. When multiple links are associated to the same switch, the lower port number from the advertising router is preferred.

Example 2-3 shows the output of running the command **show spanning-tree root** on SW4 and SW5. The Root ID field is exactly the same as on SW1, SW2, and SW3 in Examples 2-1 and 2-2. However, the root path cost has changed to 8 because both switches (SW4 and SW5) must traverse two 1 Gbps link to reach SW1. Gi1/0/2 was identified as the RP for SW4, and Gi1/0/3 was identified as the RP for SW5.

Example 2-3 Identifying the Root Ports on SW4 and SW5

```
SW4# show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                   Root ID          Cost    Time  Age Dly  Ro
---------------- -------------------- --------- ----- --- ---  --
VLAN0001         32769 0062.ec9d.c500         8    2   20  15  Gi
```

```
VLAN0010         32778 0062.ec9d.c500         8    2   20  15   Gi
VLAN0020         32788 0062.ec9d.c500         8    2   20  15   Gi
VLAN0099         32867 0062.ec9d.c500         8    2   20  15   Gi

SW5# show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                  Root ID           Cost    Time  Age Dly   Ro
---------------- -------------------- --------- ----- --- ---   --
VLAN0001         32769 0062.ec9d.c500         8    2   20  15   Gi
VLAN0010         32778 0062.ec9d.c500         8    2   20  15   Gi
VLAN0020         32788 0062.ec9d.c500         8    2   20  15   Gi
VLAN0099         32867 0062.ec9d.c500         8    2   20  15   Gi
```

The root bridge can be identified for a specific VLAN through the use of the command **show spanning-tree root** and examination of the CDP or LLDP neighbor information to identify the host name of the RP switch. The process can be repeated until the root bridge is located.

Locating Blocked Designated Switch Ports

Now that the root bridge and RPs have been identified, all other ports are considered designated ports. However, if two non-root switches are connected to each other on their designated ports, one of those switch ports must be set to a blocking state to prevent a forwarding loop. In our sample topology, this would apply to the following links:

SW2 Gi1/0/3 ← → SW3 Gi1/0/2

SW4 Gi1/0/5 ←→ SW5 Gi1/0/4

SW4 Gi1/0/6 ←→ SW5 Gi1/0/5

The logic to calculate which ports should be blocked between two non-root switches is as follows:

1. The interface is a designated port and must not be considered an RP.

2. The switch with the lower path cost to the root bridge forwards packets, and the one with the higher path cost blocks. If they tie, they move on to the next step.

3. The system priority of the local switch is compared to the system priority of the remote switch. The local port is moved to a blocking state if the remote system priority is lower than that of the local switch. If they tie, they move on to the next step.

4. The system MAC address of the local switch is compared to the system priority of the remote switch. The local designated port is moved to a blocking state if the remote system MAC address is lower than that of the local switch. If the links are connected to the same switch, they move on to the next step.

All three links (SW2 Gi1/0/3 ←→ SW3 Gi1/0/2, SW4 Gi1/0/5 ←→ SW5 Gi1/0/4, and SW4 Gi1/0/6 ←→ SW5 Gi1/0/5) would use step 4 of the process just listed to identify which port moves to a blocking state. SW3's Gi1/0/2, SW5’s Gi1/0/5, and SW5’s Gi1/0/6 ports would all transition to a blocking state because he MAC addresses are lower for SW2 and SW4.

The command **show spanning-tree** [**vlan** *vlan-id*] provides useful information for locating a port’s STP state. Example 2-4 shows this command being used to show SW1’s STP information for VLAN 1. The first portion of the output displays the relevant root bridge’s information, which is followed by the local bridge’s information. The associated interface’s STP port cost, port priority, and port type are displayed as well. All of SW1’s ports are designated ports (Desg) because SW1 is the root bridge.

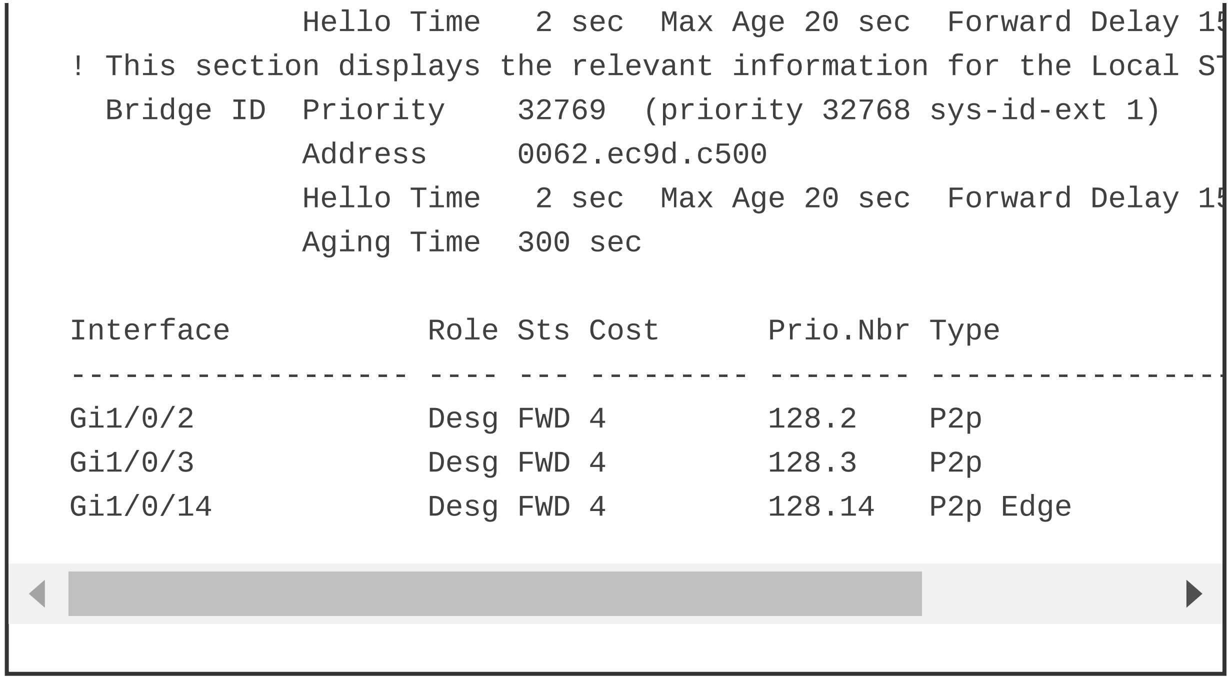
These port types are expected on Catalyst switches:

- **Point-to-point (P2P):** This port type connects with another network device (PC or RSTP switch).

- **P2P edge:** This port type specifies that portfast is enabled on this port.

Example 2-4 Viewing SW1’s STP Information

```
SW1# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
! This section displays the relevant information for the STP root
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             This bridge is the root
```



```
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
! This section displays the relevant information for the Local ST
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----------------
Gi1/0/2             Desg FWD 4         128.2    P2p
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/14            Desg FWD 4         128.14   P2p Edge
```

Note

If the Type field includes *TYPE_Inc -, this indicates a port configuration mismatch between this Catalyst switch and the switch it is connected to. Common issues are the port type being incorrect and the port mode (access versus trunk) being misconfigured.

Example 2-5 shows the STP topology for SW2 and SW3. Notice that in the first root bridge section, the output provides the total root path cost and the port on the switch that is identified as the RP.

All the ports on SW2 are in a forwarding state, but port Gi1/0/2 on SW3 is in a blocking (BLK) state. Specifically, SW3's Gi1/0/2 port has been designated as an alternate port to reach the root in the event that the Gi1/0/1 connection fails.

The reason that SW3's Gi1/0/2 port rather than SW2's Gi1/0/3 port was placed into a blocking state is that SW2's system MAC address (0081.c4ff.8b00) is lower than SW3's system MAC address (189c.5d11.9980). This can be deduced by looking at the system MAC addresses in the output and confirmed by the topology in Figure 2-1.

Example 2-5 Verifying the Root and Blocking Ports for a VLAN

```
SW2# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0081.c4ff.8b00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root FWD 4         128.1    P2p
```

```
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/4             Desg FWD 4         128.4    P2p

SW3# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
! This section displays the relevant information for the STP roo
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 1
! This section displays the relevant information for the Local S
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     189c.5d11.9980
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 1
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/2             Altn BLK 4         128.2    P2p
Gi1/0/5             Desg FWD 4         128.5    P2p
```

Verification of VLANS on Trunk Links

All the interfaces that participate in a VLAN are listed in the output of the command **show spanning-tree**. Using this command can be a daunting task for trunk ports that carry multiple VLANs. The output includes the STP state for every VLAN on an interface for every switch interface. The command **show spanning-tree interface** *interface-id* [**detail**] drastically reduces the output to the STP state for only the specified interface. The optional **detail** keyword provides information on port cost, port priority, number of transitions, link type, and count of BPDUs sent or received for every VLAN supported on that interface. Example 2-6 demonstrates the use of both iterations of the command.

If a VLAN is missing on a trunk port, you can check the trunk port configuration for accuracy. Trunk port configuration is covered in more detail in Chapter 5, "VLAN Trunks and EtherChannel Bundles." A common problem is that a VLAN may be missing from the allowed VLANs list for that trunk interface.

Example 2-6 Viewing VLANs Participating with STP on an Interface

```
SW3# show spanning-tree interface gi1/0/1

Vlan                Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
VLAN0001            Root FWD 4         128.1    P2p
VLAN0010            Root FWD 4         128.1    P2p
VLAN0020            Root FWD 4         128.1    P2p
VLAN0099            Root FWD 4         128.1    P2p

SW3# show spanning-tree interface gi1/0/1 detail
! Output omitted for brevity
 Port 1 (GigabitEthernet1/0/1) of VLAN0001 is root forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.1.
   Designated root has priority 32769, address 0062.ec9d.c500
   Designated bridge has priority 32769, address 0062.ec9d.c500
```

```
     Designated port id is 128.3, designated path cost 0
     Timers: message age 16, forward delay 0, hold 0
     Number of transitions to forwarding state: 1
     Link type is point-to-point by default
     BPDU: sent 15, received 45908

   Port 1 (GigabitEthernet1/0/1) of VLAN0010 is root forwarding
     Port path cost 4, Port priority 128, Port Identifier 128.1.
     Designated root has priority 32778, address 0062.ec9d.c500
     Designated bridge has priority 32778, address 0062.ec9d.c500
     Designated port id is 128.3, designated path cost 0
     Timers: message age 15, forward delay 0, hold 0
     Number of transitions to forwarding state: 1
     Link type is point-to-point by default
   MAC  BPDU: sent 15, received 22957
  ..
```

Key Topic

STP Topology Changes

In a stable Layer 2 topology, configuration BPDUs always flow from the root bridge toward the edge switches. However, changes in the topology (for example, switch failure, link failure, or links becoming active) have an impact to all the switches in the Layer 2 topology.

The switch that detects a link status change sends a topology change notification (TCN) BPDU toward the root bridge, out its RP. If an upstream switch receives the TCN, it sends out an acknowledgement and forwards the TCN out its RP to the root bridge.

Upon receipt of the TCN, the root bridge creates a new configuration BPDU with the Topology Change flag set, and it is then flooded to all the switches. When a switch receives a configuration BPDU with the Topology Change flag set, all switches change their MAC address timer to the forwarding delay timer (with a default of 15 seconds). This flushes out MAC addresses for devices that have not communicated in that 15-second window but maintains MAC addresses for devices that are actively communicating.

Flushing the MAC address table prevents a switch from sending traffic to a host that is no longer reachable by that port. However, a side effect of flushing the MAC address table is that it temporarily increases the unknown unicast flooding while it is rebuilt. Remember that this can impact hosts because of their CSMA/CD behavior. The MAC address timer is then reset to normal (300 seconds by default) after the second configuration BPDU is received.

TCNs are generated on a VLAN basis, so the impact of TCNs directly correlates to the number of hosts in a VLAN. As the number of hosts increase, the more likely TCN generation is to occur and the more hosts that are impacted by the broadcasts. Topology changes should be checked as part of the troubleshooting process. Chapter 3 describes mechanisms such as portfast that modify this behavior and reduce the generation of TCNs.

Topology changes are seen with the command **show spanning-tree** [**vlan** *vlan-id*] **detail** on a switch bridge. The output of this command shows the topology change count and time since the last change has occurred. A sudden or continuous increase in TCNs indicates a potential problem and should be investigated further for flapping ports or events on a connected switch.

Example 2-7 displays the output of the **show spanning-tree vlan 10 detail** command. Notice that it includes the time since the last TCN was detected and the interface from which the TCN originated.

Example 2-7 Viewing a Detailed Version of Spanning Tree State

```
SW1# show spanning-tree vlan 10 detail

 VLAN0010 is executing the rstp compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, sysid 10, address 0062.ec
  Configured hello time 2, max age 20, forward delay 15, transmit
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 42 last change occurred 01:02:09 ago
          from GigabitEthernet1/0/2
  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
```

The process of determining why TCNs are occurring involves checking a port to see whether it is connected to a host or to another switch. If it is connected to another switch, you need to connect to that switch and repeat the process of examining the STP details. You might need to examine CDP tables or your network documentation. You can execute the **show spanning-tree** [**vlan** *vlan-id*] **detail** command again to find the last switch in the topology to identify the problematic port.

Converging with Direct Link Failures

When a switch loses power or reboots, or when a cable is removed from a port, the Layer 1 signaling places the port into a down state, which can notify other processes, such as STP. STP considers such an event a direct link failure and can react in one of three ways, depending upon the topology. This section explains each of these three possible scenarios with a simple three-switch topology where SW1 is the root switch.

Direct Link Failure Scenario 1

In the first scenario, the link between SW2 and SW3 fails. SW2’s Gi1/0/3 port is the DP, and SW3’s Gi1/0/2 port is in a blocking state. Because SW3’s Gi1/0/2 port is already in a blocking state, there is no impact to traffic between the two switches as they both transmit data through SW1. Both SW2 and SW3 will advertise a TCN toward the root switch, which results in the Layer 2 topology flushing its MAC address table.

Direct Link Failure Scenario 2

In the second scenario, the link between SW1 and SW3 fails. Network traffic from SW1 or SW2 toward SW3 is impacted because SW3's Gi1/0/2 port is in a blocking state. Figure 2-3 illustrates the failure scenario and events that occur to stabilize the STP topology:

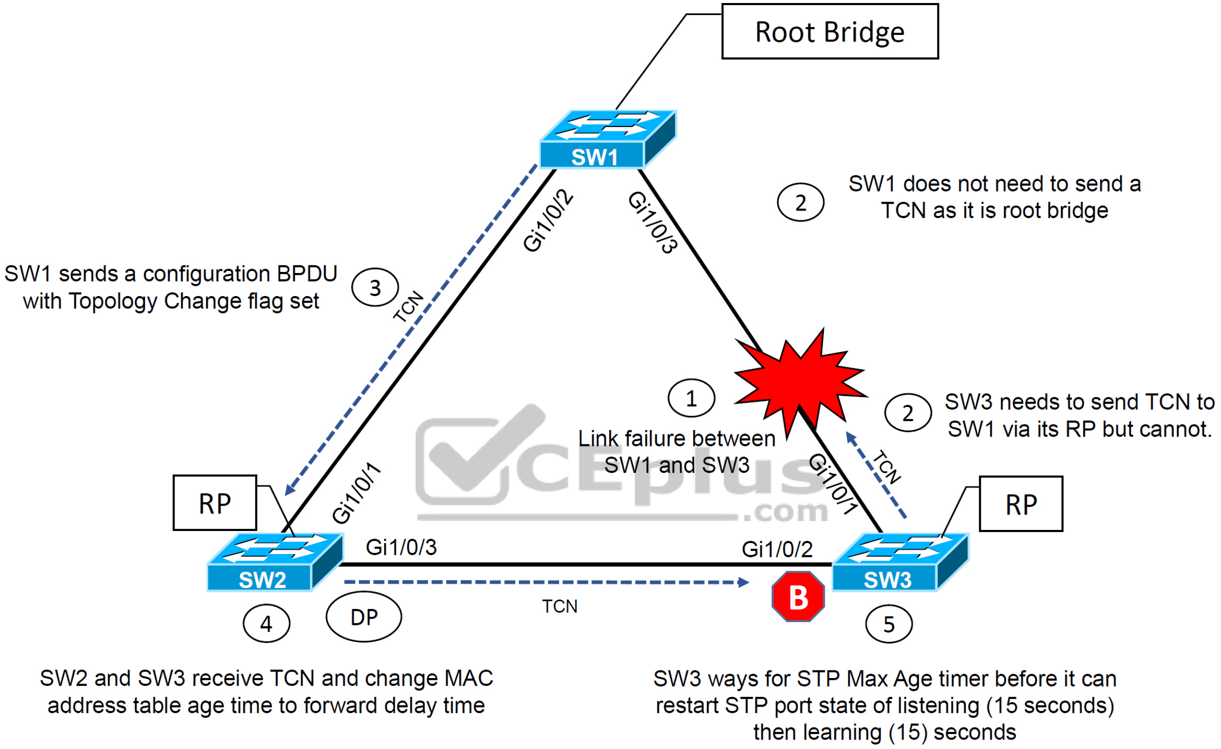

Root Bridge
SW1
Gi1/0/2
Gi1/0/3
2
SW1 does not need to send a TCN as it is root bridge
SW1 sends a configuration BPDU with Topology Change flag set
3
TCN
1
Link failure between SW1 and SW3
2
SW3 needs to send TCN to SW1 via its RP but cannot.
TCN
Gi1/0/1
Gi1/0/1
RP
RP
SW2
Gi1/0/3
Gi1/0/2
SW3
B
TCN
4
DP
5
SW2 and SW3 receive TCN and change MAC address table age time to forward delay time
SW3 ways for STP Max Age timer before it can restart STP port state of listening (15 seconds) then learning (15) seconds

Figure 2-3 Convergence with Direct Link Failure Between SW1 and SW3

Phase 1. SW1 detects a link failure on its Gi1/0/3 interface. SW3 detects a link failure on its Gi1/0/1 interface.

Phase 2. Normally SW1 would generate a TCN flag out its root port, but it is the root bridge, so it does not. SW1 would advertise a TCN if it were not the root bridge.

SW3 removes its best BPDU received from SW1 on its Gi1/0/1 interface because it is now in a down state. At this point, SW3 would attempt to send a TCN toward the root switch to notify it of a topology change; however, its root port is down.

Phase 3. SW1 advertises a configuration BPDU with the Topology Change flag out of all its ports. This BPDU is received and relayed to all switches in the environment.

Note

If other switches were connected to SW1, they would receive a configuration BPDU with the Topology Change flag set as well. These packets have an impact for all switches in the same Layer 2 domain.

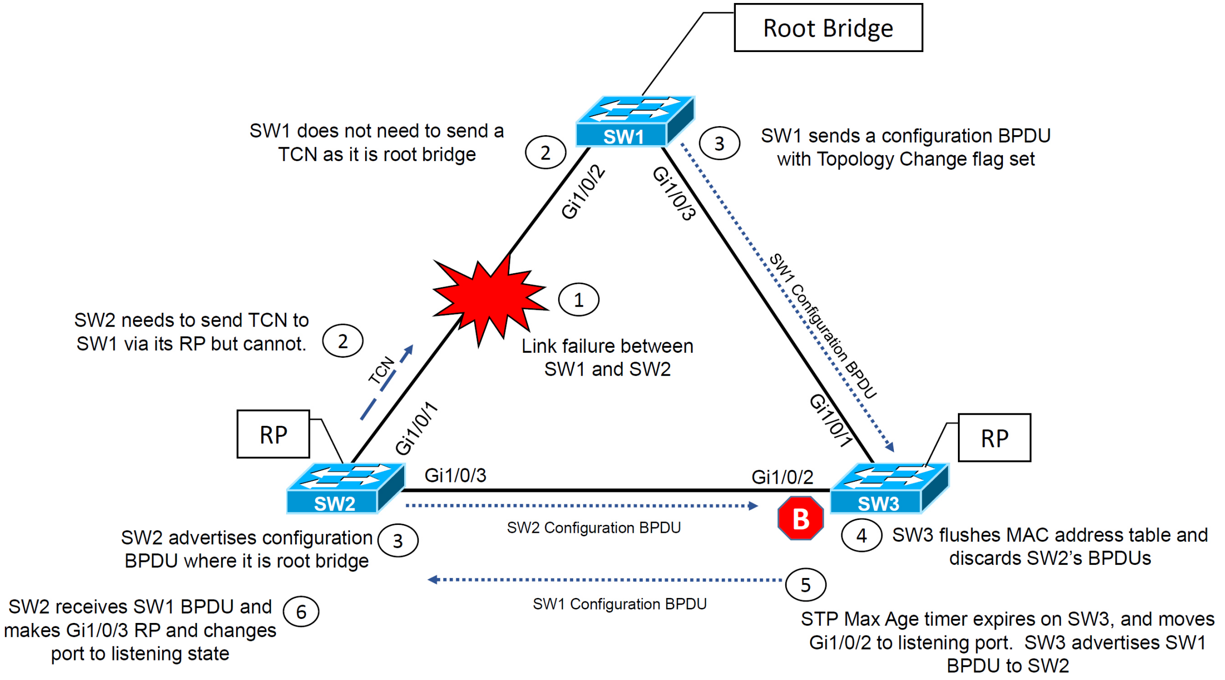
Phase 4. SW2 and SW3 receive the configuration BPDU with the Topology Change flag. These switches then reduce the MAC address age timer to the forward delay timer to flush out older MAC entries. In this phase, SW2 does not know what changed in the topology.

Phase 5. SW3 must wait until it hears from the root bridge again or the Max Age timer expires before it can reset the port state and start to listen for BPDUs on the Gi1/0/2 interface (which was in the blocking state previously).

The total convergence time for SW3 is 30 seconds: 15 seconds for the listening state and 15 seconds for the learning state before SW3's Gi1/0/2 can be made the RP.

Direct Link Failure Scenario 3

In the third scenario, the link between SW1 and SW2 fails. Network traffic from SW1 or SW3 toward SW2 is impacted because SW3's Gi1/0/2 port is in a blocking state. Figure 2-4 illustrates the failure scenario and events that occur to stabilize the STP topology:



Root Bridge
SW1
SW1 does not need to send a TCN as it is root bridge
2
3
SW1 sends a configuration BPDU with Topology Change flag set
Gi1/0/2
Gi1/0/3
SW1 Configuration BPDU
1
Link failure between SW1 and SW2
SW2 needs to send TCN to SW1 via its RP but cannot.
2
TCN
Gi1/0/1
Gi1/0/1
RP
RP
SW2
SW3
Gi1/0/3
Gi1/0/2
B
SW2 Configuration BPDU
SW2 advertises configuration BPDU where it is root bridge
3
4
SW3 flushes MAC address table and discards SW2's BPDUs
5
SW1 Configuration BPDU
SW2 receives SW1 BPDU and makes Gi1/0/3 RP and changes port to listening state
6
STP Max Age timer expires on SW3, and moves Gi1/0/2 to listening port. SW3 advertises SW1 BPDU to SW2

Figure 2-4 Convergence with Direct Link Failure Between SW1 and SW2

Phase 1. SW1 detects a link failure on its Gi1/0/1 interface. SW2 detects a link failure on its Gi1/0/3 interface.

Phase 2. Normally SW1 would generate a TCN flag out its root port, but it is the root bridge, so it does not. SW1 would advertise a TCN if it were not the root bridge.

SW2 removes its best BPDU received from SW1 on its Gi1/0/1 interface because it is now in a down state. At this point, SW2 would attempt to send a TCN toward the root switch to notify it of a topology change; however, its root port is down.

Phase 3. SW1 advertises a configuration BPDU with the Topology Change flag out of all its ports. This BPDU is then received and relayed to SW3. SW3 cannot relay this to SW2 as its Gi1/0/2 port is still in a blocking state.

SW2 assumes that it is now the root bridge and advertises configuration BPDUs with itself as the root bridge.

Phase 4. SW3 receives the configuration BPDU with the Topology Change flag from SW1. SW3 reduce the MAC address age timer to the forward delay timer to flush out older MAC entries. SW3 receives SW2's inferior BPDUs and discards them as it is still receiving superior BPDUs from SW1.

Phase 5. The Max Age timer on SW3 expires, and now SW3's Gi1/0/2 port transitions from blocking to listening state. SW3 must can now forward the next configuration BPDU it receives from SW1 to SW2.

Phase 6. SW2 receives SW1's configuration BPDU via SW3 and recognizes it as superior. It marks its Gi1/0/3 interface as the root port and transitions it to the listening state.

The total convergence time for SW2 is 52 seconds: 20 seconds for the Max Age timer on SW3, 2 seconds for the configuration BPDU from SW3, 15 seconds for the listening state on SW2, and 15 seconds for the learning state.

Indirect Failures

There are some failure scenarios where STP communication between switches is impaired or filtered while the network link remains up. This situation is known as an *indirect link failure*, and timers are required to detect and remediate the topology. Figure 2-5 illustrates an impediment or data corruption on the link between SW1 and SW3 along with the logic to resolve the loss of network traffic:

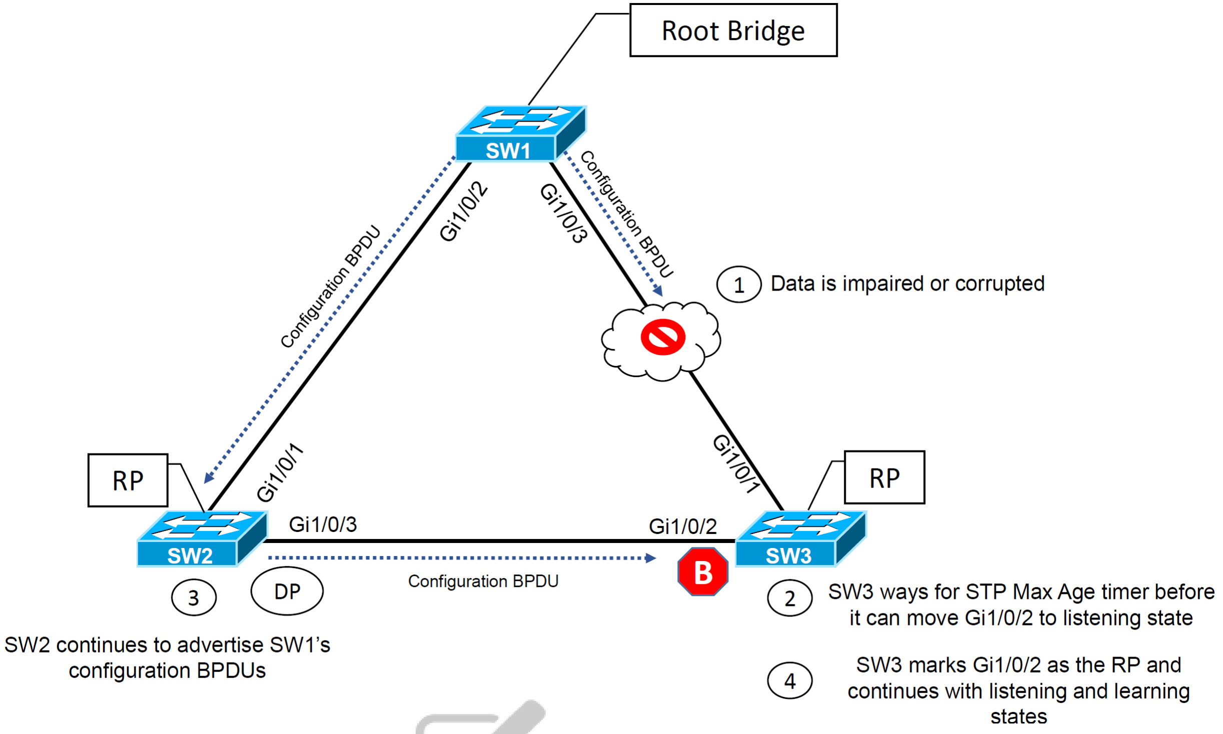

Root Bridge
SW1
Gi1/0/2
Gi1/0/3
Configuration BPDU
Configuration BPDU
1 Data is impaired or corrupted
RP
Gi1/0/1
Gi1/0/1
RP
SW2
Gi1/0/3
Gi1/0/2
SW3
B
3
DP
Configuration BPDU
SW2 continues to advertise SW1's configuration BPDUs
2 SW3 ways for STP Max Age timer before it can move Gi1/0/2 to listening state
4 SW3 marks Gi1/0/2 as the RP and continues with listening and learning states

Figure 2-5 Convergence with Indirect Link Failure

Phase 1. An event occurs that impairs or corrupts data on the link. SW1 and SW3 still report a link up condition.

Phase 2. SW3 stops receiving configuration BPDUs on its RP. It keeps a cached entry for the RP on Gi1/0/1. SW1's configuration BPDUs that are being transmitted via SW2 are discarded as its Gi1/0/2 port is in a blocking state.

Once SW3's Max Age timer expires and flushes the RP's cached entry, SW3 transitions Gi1/0/2 from blocking to listening state.

Phase 3. SW2 continues to advertise SW1's configuration BPDUs toward SW3.

Phase 4. SW3 receives SW1's configuration BPDU via SW2 on its Gi1/0/2 interface. This port is now marked as the RP and continues to transition through the listening and learning states.

The total time for reconvergence on SW3 is 52 seconds: 20 seconds for the Max Age timer on SW3, 2 seconds for the configuration BPDU advertisement on SW2, 15 seconds for the listening state on SW3, and 15 seconds the learning state on SW3.

Key Topic

RAPID SPANNING TREE PROTOCOL

802.1D did a decent job of preventing Layer 2 forwarding loops, but it used only one topology tree, which introduced scalability issues. Some larger environments with multiple VLANs need different STP topologies for traffic engineering purposes (for example, load-balancing, traffic steering). Cisco created Per-VLAN Spanning Tree (PVST) and Per-VLAN Spanning Tree Plus (PVST+) to allow more flexibility.

PVST and PVST+ were proprietary spanning protocols. The concepts in these protocols were incorporated with other enhancements to provide faster convergence into the IEEE 802.1W specification, known as Rapid Spanning Tree Protocol (RSTP).

Key Topic

RSTP (802.1W) Port States

RSTP reduces the number of port states to three:

- **Discarding:** The switch port is enabled, but the port is not forwarding any traffic to ensure that a loop is created. This state combines the traditional STP states disabled, blocking, and listening.

- **Learning:** The switch port modifies the MAC address table with any network traffic it receives. The switch still does not forward any other network traffic besides BPDUs.

- **Forwarding:** The switch port forwards all network traffic and updates the MAC address table as expected. This is the final state for a switch port to forward network traffic.

Note

A switch tries to establish an RSTP handshake with the device connected to the other end of the cable. If a handshake does not occur, the other device is assumed to be non-RSTP compatible, and the port defaults to regular 802.1D behavior. This means that host devices such as computers, printers, and so on still encounter a significant transmission delay (around 30 seconds) after the network link is established.

RSTP (802.1W) Port Roles

RSTP defines the following port roles:

- **Root port (RP):** A network port that connects to the root switch or an upstream switch in the spanning-tree topology. There should be only one root port per VLAN on a switch.

- **Designated port (DP):** A network port that receives and forwards frames to other switches. Designated ports provide connectivity to downstream devices and switches. There should be only one active designated port on a link.

- **Alternate port:** A network port that provides alternate connectivity toward the root switch through a different switch.

- **Backup port:** A network port that provides link redundancy toward the current root switch. The backup port cannot guarantee connectivity to the root bridge in the event that the upstream switch fails. A backup port exists only when multiple links connect between the same switches.

RSTP (802.1W) Port Types

RSTP defines three types of ports that are used for building the STP topology:

- **Edge port:** A port at the edge of the network where hosts connect to the Layer 2 topology with one interface and cannot form a loop. These ports directly correlate to ports that have the STP portfast feature enabled.

- **Root port:** A port that has the best path cost toward the root bridge. There can be only one root port on a switch.

- **Point-to-point port:** Any port that connects to another RSTP switch with full duplex. Full-duplex links do not permit more than two devices on a network segment, so determining whether a link is full duplex is the fastest way to check the feasibility of being connected to a switch.

Note

Multi-access Layer 2 devices such as hubs can only connect at half duplex. If a port can only connect via half duplex, it must operate under traditional 802.1D forwarding states.

Key Topic

Building the RSTP Topology

With RSTP, switches exchange handshakes with other RSTP switches to transition through the following STP states faster. When two switches first connect, they establish a bidirectional handshake across the shared link to identify the root bridge. This is straightforward for an environment with only two switches; however, large environments require greater care to avoid creating a forwarding loop. RSTP uses a synchronization process to add a switch to the RSTP topology without introducing a forwarding loop. The synchronization process starts when two switches (such as SW1 and SW2) are first connected. The process proceeds as follows:

1. As the first two switches connect to each other, they verify that they are connected with a point-to-point link by checking the full-duplex status.

2. They establish a handshake with each other to advertise a proposal (in configuration BPDUs) that their interface should be the DP for that port.

3. There can be only one DP per segment, so each switch identifies whether it is the superior or inferior switch, using the same logic as in 802.1D for the system identifier (that is, the lowest priority and then the lowest MAC address). Using the MAC addresses from Figure 2-1, SW1 (0062.ec9d.c500) is the superior switch to SW2 (0081.c4ff.8b00).

4. The inferior switch (SW2) recognizes that it is inferior and marks its local port (Gi1/0/1) as the RP. At that same time, it moves all non-edge ports to a discarding state. At this point in time, the switch has stopped all local switching for non-edge ports.

5. The inferior switch (SW2) sends an agreement (configuration BPDU) to the root bridge (SW1), which signifies to the root bridge that synchronization is occurring on that switch.

6. The inferior switch (SW2) moves its RP (Gi1/0/1) to a forwarding state. The superior switch moves its DP (Gi1/0/2) to a forwarding state, too.

7. The inferior switch (SW2) repeats the process for any downstream switches connected to it.

The RSTP convergence process can occur quickly, but if a downstream switch fails to acknowledge the proposal, the RSTP switch must default to 802.1D behaviors to prevent a forwarding loop.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 2-3 lists these key topics and the page number on which each is found.

Key Topic

Table 2-3 Key Topics for Chapter 2

Key Topic Element	Description	Page
List	802.1D port types	
Section	STP key terminology	
Section	Root bridge election	
Section	Locating root ports	
Section	Topology Changes	
Section	RSTP	
Section	RSTP (802.1W) port states	
Section	Building the RSTP topology	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

bridge protocol data unit (BPDU)

configuration BPDU

hello time

designated port (DP) forward delay

local bridge identifier

max age

root bridge

root bridge identifier

root path cost

root port

system priority

system ID extension

topology change notification (TCN)

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 2-4 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 2-4 Command Reference

Chapter 3. Advanced STP Tuning

This chapter covers the following subjects:

- **STP Topology Tuning:** This section explains some of the options for modifying the root bridge location or moving blocking ports to designated ports.

- **Additional STP Protection Mechanisms:** This section examines protection mechanisms such as root guard, BPDU guard, and STP loop guard.

This chapter reviews techniques for configuring a switch to be guaranteed as the root bridge or as a backup root bridge for a Layer 2 topology. In addition, this chapter explains features that prevent other switches from unintentionally taking over the root bridge role. The chapter also explains other common features that are used in Cisco's enterprise campus validated design guides.

“DO I KNOW THIS ALREADY?” QUIZ

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

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

Foundation Topic Section	Questions
STP Topology Tuning	1–3
Additional STP Protection Mechanisms	4–6

1. A switch’s STP priority can be configured in increments of ______.

a. 1

b. 256

c. 2048

d. 4096

2. True or false: The advertised path cost includes the advertising link's port cost as part of the configuration BPDU advertisement.

a. True

b. False

3. True or false: The switch port with the lower STP port priority is more preferred.

a. True

b. False

4. What happens to a switch port when a BPDU is received on it when BPDU guard is enabled on that port?

a. A message syslog is generated, and the BPDU is filtered.

b. A syslog message is not generated, and the BPDU is filtered.

c. A syslog message is generated, and the port is sent back to a listening state.

d. A syslog message is generated, and the port is shut down.

5. Enabling root guard on a switch port does what?

a. Upon receipt of an inferior BPDU, the port is shut down.

b. Upon receipt of a superior BPDU, the port is shut down.

c. Upon receipt of an inferior BPDU, the BPDU is filtered.

d. When the root port is shut down, only authorized designated ports can become root ports.

6. UDLD solves the problem of ______.

a. time for Layer 2 convergence

b. a cable sending traffic in only one direction

c. corrupt BPDU packets

d. flapping network links

Answers to the "Do I Know This Already?" quiz:

1. D

2. B

3. A

4. D

5. B

6. B

FOUNDATION TOPICS

STP TOPOLOGY TUNING

A properly designed network strategically places the root bridge on a specific switch and modifies which ports should be designated ports (that is, forwarding state) and which ports should be alternate ports (that is, discarding/blocking state). Design considerations factor in hardware platform, resiliency, and network topology. This chapter uses the same reference topology from Chapter 2, “Spanning Tree Protocol,” as shown in Figure 3-1.

Figure 3-1 STP Topology for Tuning

Key Topic

Root Bridge Placement

Ideally the root bridge is placed on a core switch, and a secondary root bridge is designated to minimize changes to the overall spanning tree. Root bridge placement is accomplished by lowering the system priority on the root bridge to the lowest value possible, raising the secondary root bridge to a value slightly higher than that of the root bridge, and (ideally) increasing the system priority on all other switches. This ensures consistent placement of the root bridge. The priority is set with either of the following commands:

- **spanning-tree vlan** *vlan-id* **priority** *priority***:** The priority is a value between 0 and 61,440, in increments of 4,096.

- **spanning-tree vlan** *vlan-id* **root {primary | secondary} [diameter** *diameter***]:** This command executes a script that modifies certain values. The **primary** keyword sets the priority to 24,576, and the **secondary** keyword sets the priority to 28,672.

The optional **diameter** command makes it possible to tune the Spanning Tree Protocol (STP) convergence and modifies the timers; it should reference the maximum number of Layer 2 hops between a switch and the root bridge. The timers do not need to be modified on other switches because they are carried throughout the topology through the root bridge's BPDUs.

Example 3-1 verifies the initial priority for VLAN 1 on SW1 and then checks how the change is made. Afterward, the priority is checked again to ensure that the priority is lowered.

Example 3-1 Changing the STP System Priority on SW1

```
! Verification of SW1 Priority before modifying the priority
SW1# show spanning-tree vlan 1

VLAN0001
```

```
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

! Configuring the SW1 priority as primary root for VLAN 1
SW1(config)# spanning-tree vlan 1 root primary

! Verification of SW1 Priority after modifying the priority
SW1# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    24577
             Address     0062.ec9d.c500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------
Gi1/0/2             Desg FWD 4         128.2    P2p
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/14            Desg FWD 4         128.14   P2p
```

Example 3-2 verifies the priority for VLAN 1 on SW2 before changing its priority so that it will be the backup root bridge in the event of a failure with SW1. Notice that the root bridge priority is now 24,577, and the local switch's priority is initially set to 32,769 (the default). Then the command **spanning-tree vlan 1 root secondary** is executed to modify SW2's priority, setting it to 28,673.

Example 3-2 Changing the STP System Priority on SW2

```
! Verification of SW2 Priority before modifying the priority
SW2# show spanning-tree vlan 1
! Output omitted for brevity

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    24577
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0081.c4ff.8b00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/4             Desg FWD 4         128.4    P2p
```

```
! Configuring the SW2 priority as root secondary for VLAN 1
SW2(config)# spanning-tree vlan 1 root secondary

SW2# show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    24577
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    28673  (priority 28672 sys-id-ext 1)
             Address     0081.c4ff.8b00
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ------------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/4             Desg FWD 4         128.4    P2p
```

The placement of the root bridge is an important decision and often should be chosen to minimize the number of hops to the furthest switch in the topology. The design should consider where redundant connections exist, connections that will be blocked, and the ability (performance) for the root switch to handle cross-switch traffic. Generally, root switches are at Layer 2/Layer 3 boundaries.

Key Topic

The best way to prevent erroneous devices from taking over the STP root role is to set the priority to 0 for the primary root switch and to 4096 for the secondary root switch. In addition, root guard should be used (as discussed later in this chapter).

Modifying STP Root Port and Blocked Switch Port Locations

The STP port cost is used in calculating the STP tree. When a switch generates the bridge protocol data units (BPDUs), the total path cost includes only the calculated metric to the root and does not include the cost of the port out which the BPDU is advertised. The receiving switch adds the port cost for the interface on which the BPDU was received in conjunction to the value of the total path cost in the BPDU.

In Figure 3-2, SW1 advertises its BPDUs to SW3 with a total path cost of 0. SW3 receives the BPDU and adds its STP port cost of 4 to the total path cost in the BPDU (0), resulting in a value of 4. SW3 then advertises the BPDU toward SW5 with a total path cost of 4, to which SW5 then adds its ports cost of 4. SW5 therefore reports a total path cost of 8 to reach the root bridge via SW3.

Figure 3-2 STP Path Cost Calculation

The logic is confirmed in the output of Example 3-3. Notice that there is not a total path cost in SW1’s output.

Example 3-3 Verifying the Total Path Cost

```
SW1# show spanning-tree vlan 1
! Output omitted for brevity
VLAN0001
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             This bridge is the root
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/2             Desg FWD 4         128.2    P2p
Gi1/0/3             Desg FWD 4         128.3    P2p

SW3# show spanning-tree vlan 1
! Output omitted for brevity
VLAN0001
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/2             Altn BLK 4         128.2    P2p
Gi1/0/5             Desg FWD 4         128.5    P2p

SW5# show spanning-tree vlan 1
! Output omitted for brevity
```

```
VLAN0001
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        8
             Port        3 (GigabitEthernet1/0/3)
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/3             Root FWD 4         128.3    P2p
Gi1/0/4             Altn BLK 4         128.4    P2p
Gi1/0/5             Altn BLK 4         128.5    P2p
```

Key Topic

By changing the STP port costs with the command **spanning tree** [**vlan** *vlan-id*] **cost** *cost*, you can modify the STP forwarding path. You can lower a path that is currently an alternate port while making it designated, or you can raise the cost on a port that is designated to turn it into a blocking port. The **spanning tree** command modifies the cost for all VLANs unless the optional **vlan** keyword is used to specify a VLAN.

Example 3-4 demonstrates the modification of SW3's port cost for Gi1/0/1 to a cost of 1, which impacts the port state between SW2 and SW3. SW2 receives a BPDU from SW3 with a cost of 5, and SW3 receives a BPDU from SW2 with a cost of 8. Now SW3's Gi1/0/2 is no longer an alternate port but is now a designated port. SW2's Gi1/0/3 port has changed from a designated port to an alternate port.

Example 3-4 Modifying STP Port Cost

```
SW3# conf t
SW3(config)# interface gi1/0/1
SW3(config-if)# spanning-tree cost 1

SW3# show spanning-tree vlan 1
! Output omitted for brevity
VLAN0001
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        1
             Port        1 (GigabitEthernet1/0/1)

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     189c.5d11.9980
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----------------
Gi1/0/1             Root FWD 1         128.1    P2p
Gi1/0/2             Desg FWD 4         128.2    P2p
Gi1/0/5             Desg FWD 4         128.5    P2p

SW2# show spanning-tree vlan 1
! Output omitted for brevity
VLAN0001
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        4
             Port        1 (GigabitEthernet1/0/1)
  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0081.c4ff.8b00
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root FWD 4         128.1    P2p
Gi1/0/3             Altn BLK 4         128.3    P2p
Gi1/0/4             Desg FWD 4         128.4    P2p
```

Modifying STP Port Priority

The STP port priority impacts which port is an alternate port when multiple links are used between switches. In our test topology, shutting down the link between SW3 and SW5 forces SW5 to choose one of the links connected to SW4 as a root port.

Example 3-5 verifies that this change makes SW5’s Gi1/0/4 the root port (RP) toward SW4. Remember that system ID and port cost are the same, so the next check is port priority, followed by the port number. Both the port priority and port number are controlled by the upstream switch.

Example 3-5 Viewing STP Port Priority

```
SW5# show spanning-tree vlan 1
! Output omitted for brevity
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     0062.ec9d.c500
             Cost        12
             Port        4 (GigabitEthernet1/0/4)

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     bc67.1c5c.9300
..
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/4             Root FWD 4         128.4    P2p
Gi1/0/5             Altn BLK 4         128.5    P2p
```

You can modify the port priority on SW4's Gi1/0/6 (toward R5's Gi1/0/5 interface) with the command **spanning-tree** [**vlan** *vlan-id*] **port-priority** *priority*. The optional **vlan** keyword allows you to change the priority on a VLAN-by-VLAN basis. Example 3-6 shows how to change the port priority on SW4's Gi1/0/6 port to 64.

Example 3-6 Verifying Port Priority Impact on an STP Topology

```
SW4# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW4(config)# interface gi1/0/6
SW4(config-if)# spanning-tree port-priority 64
```

Now SW4's Gi1/0/6 port has a value of 64, which is lower than the value of its Gi1/0/5 port, which is using a default value of 128. SW4's Gi1/0/6 interface is now preferred and will impact the RP on SW5, as displayed in Example 3-7.

Example 3-7 Determining the Impact of Port Priority on a Topology

```
SW4# show spanning-tree vlan 1
! Output omitted for brevity
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/2             Root FWD 4         128.2    P2p
Gi1/0/5             Desg FWD 4         128.5    P2p
Gi1/0/6             Desg FWD 4          64.6    P2p

SW5# show spanning-tree vlan 1
! Output omitted for brevity
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/4             Altn BLK 4         128.4    P2p
Gi1/0/5             Root FWD 4         128.5    P2p
```

ADDITIONAL STP PROTECTION MECHANISMS

Network packets do not decrement the time-to-live portion of the header as a packet is forwarded in a Layer 2 topology. A network forwarding loop occurs when the logical topology allows for multiple active paths between two devices. Broadcast and multicast traffic wreak havoc as they are forwarded out of every switch port and continue the forwarding loop. High CPU consumption and low free memory space are common symptoms of a Layer 2 forwarding loop. In Layer 2 forwarding loops, in addition to constantly consuming switch bandwidth, the CPU spikes. Because the packet is received on a different interface, the switch must move the media access control (MAC) address from one interface to the next. The network throughput is impacted drastically; users are likely to notice a slowdown on their network applications, and the switches might crash due to exhausted CPU and memory resources.

The following are some common scenarios for Layer 2 forwarding loops:

- STP disabled on a switch

- A misconfigured load balancer that transmits traffic out multiple ports with the same MAC address

- A misconfigured virtual switch that bridges two physical ports (Virtual switches typically do not participate in STP.)

- End users using a dumb network switch or hub

Catalyst switches detect a MAC address that is flapping between interfaces and notify via syslog with the MAC address of the host, VLAN, and ports between which the MAC address is flapping. These messages should be investigated to ensure that a forwarding loop does not exist. Example 3-8 shows a sample syslog message for a flapping MAC address where STP has been removed from the topology.

Example 3-8 Syslog Message for a Flapping MAC Address

```
12:40:30.044: %SW_MATM-4-MACFLAP_NOTIF: Host 70df.2f22.b8c7 in vl
 between port Gi1/0/3 and port Gi1/0/2
```

In this scenario, STP should be checked for all the switches hosting the VLAN mentioned in the syslog message to ensure that spanning tree is enabled and working properly.

Key Topic

Root Guard

Root guard is an STP feature that is enabled on a port-by-port basis; it prevents a configured port from becoming a root port. Root guard prevents a downstream switch (often misconfigured or rogue) from becoming a root bridge in a topology. Root guard functions by placing a port in an ErrDisabled state if a superior BPDU is received on a configured port. This prevents the configured DP with root guard from becoming an RP.

Root guard is enabled with the interface command **spanning-tree guard root**. Root guard is placed on designated ports toward other switches that should never become root bridges.

In the sample topology shown in Figure 3-1, root guard should be placed on SW2’s Gi1/0/4 port toward SW4 and on SW3’s Gi1/0/5 port toward SW5. This prevents SW4 and SW5 from ever becoming root bridges but still allows for SW2 to maintain connectivity to SW1 via SW3 if the link connecting SW1 to SW2 fails.

Key Topic

STP Portfast

The generation of TCN for hosts does not make sense as a host generally has only one connection to the network. Restricting TCN creation to only ports that connect with other switches and network devices increases the L2 network’s stability and efficiency. The STP portfast feature disables TCN generation for access ports.

Another major benefit of the STP portfast feature is that the access ports bypass the earlier 802.1D STP states (learning and listening) and forward traffic immediately. This is beneficial in environments where computers use Dynamic Host Configuration Protocol (DHCP) or Preboot Execution Environment (PXE). If a BPDU is received on a portfast-enabled port, the portfast functionality is removed from that port.

The portfast feature is enabled on a specific access port with the command **spanning-tree portfast** or globally on all access ports with the command

spanning-tree portfast default. If portfast needs to be disabled on a specific port when using the global configuration, you can use the interface configuration command **spanning-tree portfast disable** to remove portfast on that port.

Portfast can be enabled on trunk links with the command **spanning-tree portfast trunk**. However, this command should be used only with ports that are connecting to a single host (such as a server with only one NIC that is running a hypervisor with VMs on different VLANs). Running this command on interfaces connected to other switches, bridges, and so on can result in a bridging loop.

Example 3-9 shows how to enable portfast for SW1’s Gi1/0/13 port. Then the configuration is verified by examining the STP for VLAN 10 or examining the STP interface. Notice that the portfast ports are displayed with P2P Edge. The last section of output demonstrates how portfast is enabled globally for all access ports.

Example 3-9 Enabling STP Portfast on Specific Interfaces

```
SW1(config)# interface gigabitEthernet 1/0/13
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 10
SW1(config-if)# spanning-tree portfast

SW1# show spanning-tree vlan 10
! Output omitted for brevity
VLAN0010
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----------------
Gi1/0/2             Desg FWD 4         128.2    P2p
```

```
Gi1/0/3             Desg FWD 4         128.3    P2p
Gi1/0/13            Desg FWD 4         128.13   P2p Edge

SW1# show spanning-tree interface gi1/0/13 detail
 Port 13 (GigabitEthernet1/0/13) of VLAN0010 is designated forwar
   Port path cost 4, Port priority 128, Port Identifier 128.7.
   Designated root has priority 32778, address 0062.ec9d.c500
   Designated bridge has priority 32778, address 0062.ec9d.c500
   Designated port id is 128.7, designated path cost 0
   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: sent 23103, received 0
```

Example 3-10 shows how to enable portfast globally for all access ports on SW2 and then disable it for Gi1/0/8.

Example 3-10 Enabling STP Portfast Globally

```
SW2# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# spanning-tree portfast default
%Warning: this command enables portfast by default on all interfa
 should now disable portfast explicitly on switched ports leading
 switches and bridges as they may create temporary bridging loops

SW2(config)# interface gi1/0/8
SW2(config-if)# spanning-tree portfast disable
```

Key Topic

BPDU Guard

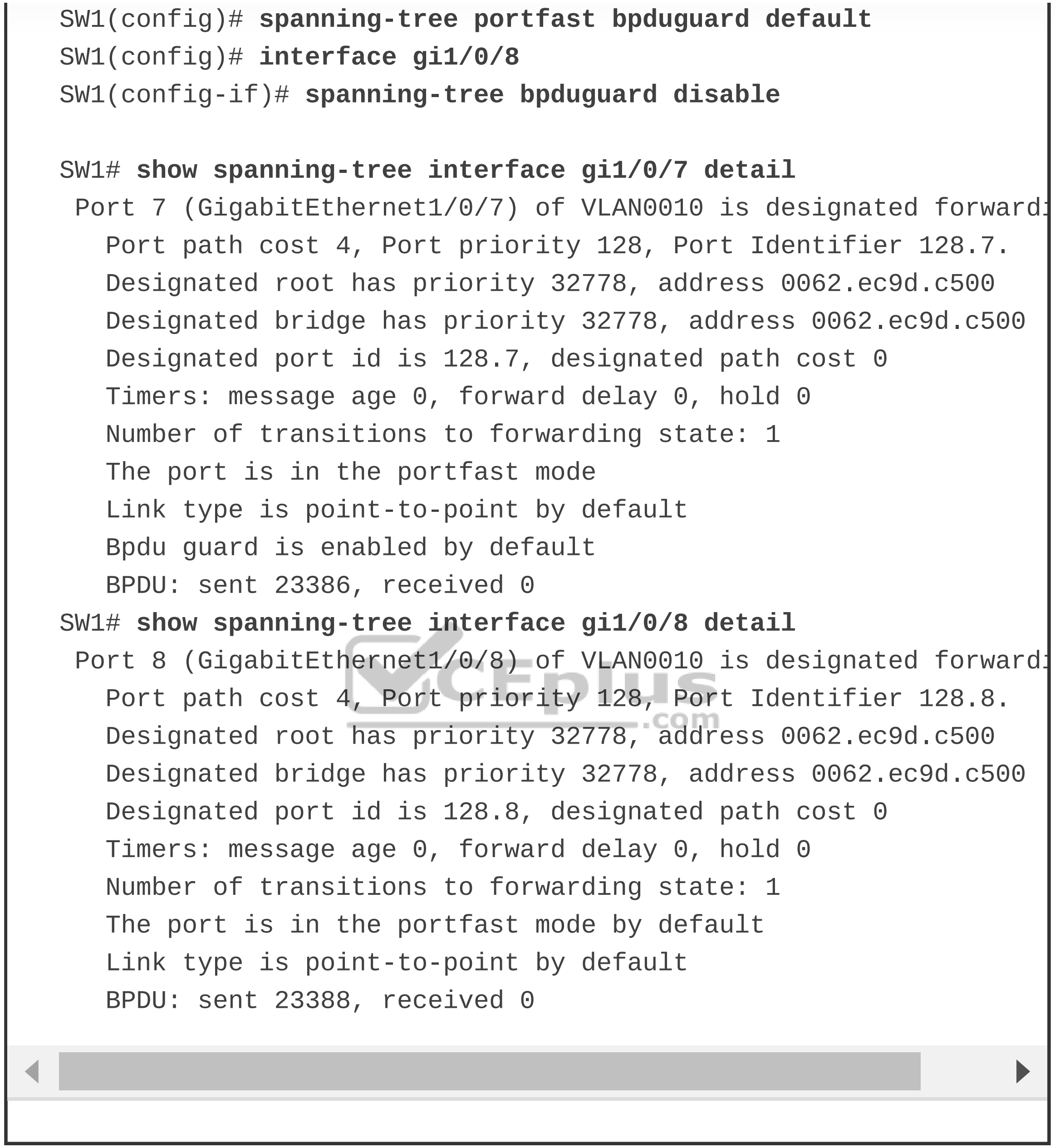
BPDU guard is a safety mechanism that shuts down ports configured with STP portfast upon receipt of a BPDU. Assuming that all access ports have portfast enabled, this ensures that a loop cannot accidentally be created if an unauthorized switch is added to a topology.

BPDU guard is enabled globally on all STP portfast ports with the command **spanning-tree portfast bpduguard default**. BPDU guard can be enabled or disabled on a specific interface with the command **spanning-tree bpduguard** {**enable** | **disable**}.

Example 3-11 shows how to configure BPDU guard globally on SW1 for all access ports but with the exception of disabling BPDU guard on Gi1/0/8. The **show spanning-tree interface** *interface-id* **detail** command displays whether BPDU guard is enabled for the specified port.

Example 3-11 Configuring BPDU Guard

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```



```
SW1(config)# spanning-tree portfast bpduguard default
SW1(config)# interface gi1/0/8
SW1(config-if)# spanning-tree bpduguard disable

SW1# show spanning-tree interface gi1/0/7 detail
 Port 7 (GigabitEthernet1/0/7) of VLAN0010 is designated forwardi
   Port path cost 4, Port priority 128, Port Identifier 128.7.
   Designated root has priority 32778, address 0062.ec9d.c500
   Designated bridge has priority 32778, address 0062.ec9d.c500
   Designated port id is 128.7, designated path cost 0
   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 by default
   BPDU: sent 23386, received 0
SW1# show spanning-tree interface gi1/0/8 detail
 Port 8 (GigabitEthernet1/0/8) of VLAN0010 is designated forwardi
   Port path cost 4, Port priority 128, Port Identifier 128.8.
   Designated root has priority 32778, address 0062.ec9d.c500
   Designated bridge has priority 32778, address 0062.ec9d.c500
   Designated port id is 128.8, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode by default
   Link type is point-to-point by default
   BPDU: sent 23388, received 0
```

Note

BPDU guard is typically configured with all host-facing ports that are enabled with portfast.

Example 3-12 shows the syslog messages that appear when a BPDU is received on a BPDU guard–enabled port. The port is then placed into an ErrDisabled state, as shown with the command **show interfaces status**.

Example 3-12 Detecting a BPDU on a BPDU Guard–Enabled Port

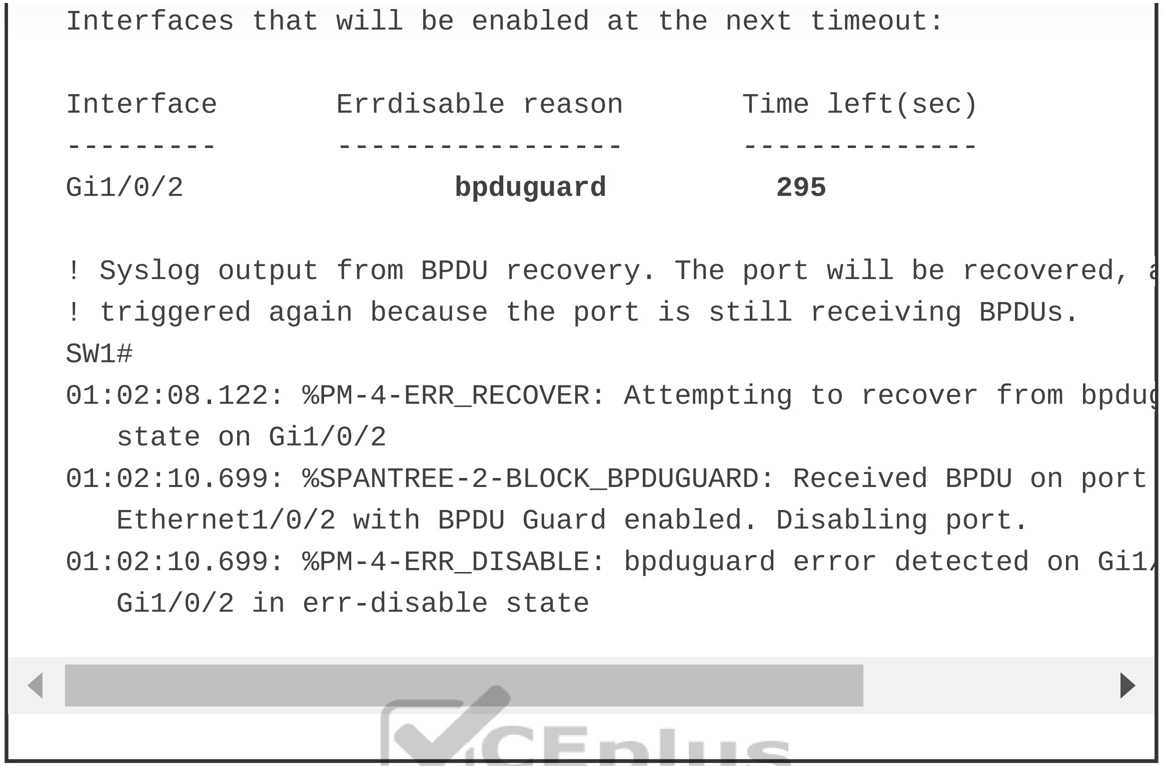
```
12:47:02.069: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port
   Ethernet1/0/2 with BPDU Guard enabled. Disabling port.
12:47:02.076: %PM-4-ERR_DISABLE: bpduguard error detected on Gi1/
   putting Gi1/0/2 in err-disable state
12:47:03.079: %LINEPROTO-5-UPDOWN: Line protocol on Interface Gig
   Ethernet1/0/2, changed state to down
12:47:04.082: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, cha
   state to down

SW1# show interfaces status

Port      Name               Status       Vlan      Duplex  Speed T
Gi1/0/1                      notconnect   1           auto   auto 1
Gi1/0/2   SW2 Gi1/0/1       err-disabled  1           auto   auto 1
Gi1/0/3   SW3 Gi1/0/1       connected    trunk      a-full a-1000 1
```

By default, ports that are put in the ErrDisabled state because of BPDU guard do not automatically restore themselves. The Error Recovery service can be used to reactivate ports that are shut down for a specific problem, thereby reducing administrative overhead. To use Error Recovery to recovers ports that were shut down from BPDU guard, use the command **errdisable recovery cause bpduguard**. The period that the Error Recovery checks for ports is configured with the command **errdisable recovery interval time-seconds**.

Example 3-13 demonstrates the configuration of the Error Recovery service for BPDU guard, verification of the Error Recovery service for BPDU guard, and the syslog messages from the process.

Example 3-13 Configuring Error Recovery Service

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# errdisable recovery cause bpduguard

SW1# show errdisable recovery
! Output omitted for brevity
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Enabled
..
Recovery command: "clear     Disabled

Timer interval: 300 seconds
```



```
Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
Gi1/0/2                bpduguard          295

! Syslog output from BPDU recovery. The port will be recovered, a
! triggered again because the port is still receiving BPDUs.
SW1#
01:02:08.122: %PM-4-ERR_RECOVER: Attempting to recover from bpdug
   state on Gi1/0/2
01:02:10.699: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port
   Ethernet1/0/2 with BPDU Guard enabled. Disabling port.
01:02:10.699: %PM-4-ERR_DISABLE: bpduguard error detected on Gi1/
   Gi1/0/2 in err-disable state
```

Note

The Error Recovery service operates every 300 seconds (5 minutes). This can be changed to 5 to 86,400 seconds with the global configuration command **errdisable recovery interval** *time*.

Key Topic

BPDU Filter

BPDU filter simply blocks BPDUs from being transmitted out a port. BPDU filter can be enabled globally or on a specific interface. The behavior changes depending on the configuration:

- The global BPDU filter configuration uses the command **spanning-tree portfast bpdufilter default**, and the port sends a series of 10 to 12 BPDUs. If the switch receives any BPDUs, it checks to identify which switch is more preferred.

- The preferred switch does not process any BPDUs that it receives, but it still transmits BPDUs to inferior downstream switches.

- A switch that is not the preferred switch processes BPDUs that are received, but it does not transmit BPDUs to the superior upstream switch.

- The interface-specific BPDU filter is enabled with the interface configuration command **spanning-tree bpdufilter enable**. The port does not send any BPDUs on an ongoing basis. If the remote port has BPDU guard on it, that generally shuts down the port as a loop prevention mechanism.

Note

Be careful with the deployment of BPDU filter as it could cause problems. Most network designs do not require BPDU filter, which adds an unnecessary level of complexity and also introduces risk.

Example 3-14 shows SW1's Gi1/0/2 statistics after BPDU is enabled on the Gi1/0/2 interface. In the first set of output, BPDU filter is enabled specifically on the Gi1/0/2 interface (thereby prohibiting any BPDUs from being sent or received). The second set of output enables BPDU filtering globally, so that BPDUs are transmitted when the port first becomes active; the filtering is verified by the number of BPDU sent changing from 56 to 58.

Example 3-14 Verifying a BPDU Filter

```
! SW1 was enabled with BPDU filter only on port Gi1/0/2
SW1# show spanning-tree interface gi1/0/2 detail | in BPDU|Bpdu|
 Port 2 (GigabitEthernet1/0/2) of VLAN0001 is designated forwardi
   Bpdu filter is enabled
   BPDU: sent 113, received 84
SW1# show spanning-tree interface gi1/0/2 detail | in BPDU|Bpdu|
 Port 2 (GigabitEthernet1/0/2) of VLAN0001 is designated forwardi
   Bpdu filter is enabled
   BPDU: sent 113, received 84

! SW1 was enabled with BPDU filter globally
```

```
SW2# show spanning-tree interface gi1/0/2 detail | in BPDU|Bpdu|
 Port 1 (GigabitEthernet1/0/2) of VLAN0001 is designated forwardi
   BPDU: sent 56, received 5
SW2# show spanning-tree interface gi1/0/2 detail | in BPDU|Bpdu|
 Port 1 (GigabitEthernet1/0/2) of VLAN0001 is designated forwardi
   BPDU: sent 58, received 5
```

Problems with Unidirectional Links

Fiber-optic cables consist of strands of glass/plastic that transmit light. A cable typically consists of one strand for sending data and another strand for receiving data on one side; the order is directly opposite at the remote site. Network devices that use fiber for connectivity can encounter unidirectional traffic flows if one strand is broken. In such scenarios, the interface still shows a line-protocol up state; however, BPDUs are not able to be transmitted, and the downstream switch eventually times out the existing root port and identifies a different port as the root port. Traffic is then received on the new root port and forwarded out the strand that is still working, thereby creating a forwarding loop.

A couple solutions can resolve this scenario:

- STP loop guard

- Unidirectional Link Detection

STP Loop Guard

STP loop guard prevents any alternative or root ports from becoming designated ports (ports toward downstream switches) due to loss of BPDUs on the root port. Loop guard places the original port in an ErrDisabled state while BPDUs are not being received. When BPDU transmission starts again on that interface, the port recovers and begins to transition through the STP states again.

Loop guard is enabled globally by using the command **spanning-tree loopguard default**, or it can be enabled on an interface basis with the interface command **spanning-tree guard loop**. It is important to note that loop guard should not be enabled on portfast-enabled ports (because it directly conflicts with the root/alternate port logic).

Example 3-15 demonstrates the configuration of loop guard on SW2’s Gi1/0/1 port.

Example 3-15 Configuring Loop Guard

```
SW2# config t
SW2(config)# interface gi1/0/1
SW2(config-if)# spanning-tree guard loop
! Placing BPDU filter on SW2’s RP (Gi1/0/1) bridge) triggers loo
SW2(config-if)# interface gi1/0/1
SW2(config-if)# spanning-tree bpdufilter enabled
01:42:35.051: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking po
   Ethernet1/0/1 on VLAN0001

SW2# show spanning-tree vlan 1 | b Interface
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/1             Root BKN*4         128.1    P2p *LOOP_Inc
```

```
Gi1/0/3             Root FWD 4          128.3    P2p
Gi1/0/4             Desg FWD 4          128.4    P2p
```

At this point, the port is considered to be in an inconsistent state and does not forward any traffic. Inconsistent ports are viewed with the command **show spanning-tree inconsistentports**, as show in Example 3-16. Notice that an entry exists for all the VLANs carried across the Gi1/0/1 port.

Example 3-16 Viewing the Inconsistent STP Ports

```
SW2# show spanning-tree inconsistentports

Name                 Interface                Inconsistency
-------------------- ------------------------ ------------------
VLAN0001             GigabitEthernet1/0/1     Loop Inconsistent
VLAN0010             GigabitEthernet1/0/1     Loop Inconsistent
VLAN0020             GigabitEthernet1/0/1     Loop Inconsistent
VLAN0099             GigabitEthernet1/0/1     Loop Inconsistent

Number of inconsistent ports (segments) in the system : 4
```

Unidirectional Link Detection

Unidirectional Link Detection (UDLD) allows for the bidirectional monitoring of fiber-optic cables. UDLD operates by transmitting UDLD packets to a neighbor device that includes the system ID and port ID of the interface transmitting the UDLD packet. The receiving device then repeats that information, including its system ID and port ID, back to the originating device. The process continues indefinitely. UDLD operates in two different modes:

- **Normal:** In normal mode, if a frame is not acknowledged, the link is considered undetermined and the port remains active.

- **Aggressive:** In aggressive mode, when a frame is not acknowledged, the switch sends another eight packets in 1-second intervals. If those packets are not acknowledged, the port is placed into an error state.

UDLD is enabled globally with the command **udld** [**aggressive**]. This enables UDLD on any small form-factor pluggable (SFP)-based port. UDLD can be disabled on a specific port with the interface configuration command **udld port disable**. UDLD recovery can be enabled with the command **udld recovery** [**interval** *time*], where the optional **interval** keyword allows for the timer to be modified from the default value of 5 minutes. UDLD can be enabled on a port-by-port basis with the interface configuration command **udld port** [**aggressive**], where the optional **aggressive** keyword places the ports in UDLD aggressive mode.

Example 3-17 shows how to enable UDLD normal mode on SW1.

Example 3-17 Configuring UDLD

```
SW1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)# udld enable
```

UDLD must be enabled on the remote switch as well. Once it is configured, the status of UDLD neighborship can be verified with the command **show udld neighbors**. More detailed information can be viewed with the command **show udld** *interface-id*.

Example 3-18 displays the verification of SW1's neighborship with SW2. The link is operating in a bidirectional state. More information is obtained with the **show udld Te1/1/3** command, which includes the current state, device IDs (that is, serial numbers), originating interface IDs, and return interface IDs.

Example 3-18 Verifying UDLD Neighbors and Switch Port Status

```
SW1# show udld neighbors
Port     Device Name   Device ID     Port ID    Neighbor State
----     -----------   ---------     -------    --------------
Te1/1/3  081C4FF8B0      1            Te1/1/3    Bidirectional

SW1# show udld Te1/1/3

Interface Te1/1/3
---
Port enable administrative configuration setting: Follows device
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detect
```

```
Message interval: 15000 ms
Time out interval: 5000 ms

Port fast-hello configuration setting: Disabled
Port fast-hello interval: 0 ms
Port fast-hello operational state: Disabled
Neighbor fast-hello configuration setting: Disabled
Neighbor fast-hello interval: Unknown


    Entry 1
    ---
    Expiration time: 41300 ms
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: 081C4FF8B0
    Port ID: Te1/1/3
    Neighbor echo 1 device: 062EC9DC50
    Neighbor echo 1 port: Te1/1/3

    TLV Message interval: 15 sec
    No TLV fast-hello interval
    TLV Time out interval: 5
    TLV CDP Device name: SW2
```

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, "Final Preparation," and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 3-2 lists these key topics and the page number on which each is found.

Key Topic

Table 3-2 Key Topics for Chapter 3

Key Topic Element	Description	Page
Section	Root bridge placement	
Paragraph	Root bridge values	
Paragraph	Spanning tree port cost	
Section	Root guard	
Section	STP portfast	
Section	BPDU guard	
Section	BPDU filter	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

BPDU filter

BPDU guard

root guard

STP portfast

STP loop guard

Unidirectional Link Detection (UDLD)

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 3-3 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 3-3 Command Reference

Chapter 4. Multiple Spanning Tree Protocol

This chapter covers the following subject:

- **Multiple Spanning Tree Protocol:** This section examines the benefits and operations of MST.

This chapter completes the section on spanning tree by explaining Multiple Spanning Tree Protocol (MST). MST is the one of three STP modes supported on Catalyst switches.

"DO I KNOW THIS ALREADY?" QUIZ

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

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

Foundation Topic Section	Questions
Multiple Spanning Tree Protocol	1–7

1. Which of the following issues does MST solve? (Choose two.)

a. Enables traffic load balancing for specific VLANs

b. Reduces the CPU and memory resources needed for environments with large numbers of VLANs

c. Overcomes MAC address table scaling limitations for environments with large numbers of devices

d. Detects issues with cabling that transmits data in one direction

e. Prevents unauthorized switches from attaching to the Layer 2 domain

2. With MST, VLANs are directly associated with ______.

a. areas

b. regions

c. instances

d. switches

3. What do CST and 802.1D have in common?

a. They support only one topology.

b. They support multiple topologies.

c. They allow for load balancing of traffic across different VLANs.

d. They provide switch authentication so that inter-switch connectivity can occur.

4. True or false: The MST root bridge advertises the VLAN-to-instance mappings to all other MST switches.

a. True

b. False

5. True or false: The MST configuration version is locally significant.

a. True

b. False

6. True or false: The MST topology can be tuned for root bridge placement, just like PVST+ and RSTP.

a. True

b. False

7. MST regions can interact with PVST+/RSTP in which of the following ways? (Choose two.)

a. The MST region is the root bridge for all VLANs.

b. The MST region is the root bridge for some VLANs.

c. The PVST+/RSTP topology is the root bridge for all VLANs.

d. The PVST+/RSTP topology is the root bridge for some VLANs.

Answers to the "Do I Know This Already?" quiz:

1. A, B

2. C

3. A

4. B

5. B

6. A

7. A, C

FOUNDATION TOPICS

Key Topic

MULTIPLE SPANNING TREE PROTOCOL

The original 802.1D standard, much like the 802.1Q standard, supported only one STP instance for an entire switch network. In this situation, referred to as *Common Spanning Tree (CST)*, all VLANs used the same topology, which meant it was not possible to load share traffic across links by blocking for specific VLANs on one link and then blocking for other VLANs on alternate links.

Figure 4-1 demonstrates shows four VLANs sharing the same topology. All network traffic from SW2 toward SW3 must traverse through SW1. If VLAN 4 contained devices only on SW2 and SW3, the topology could not be tuned with traffic going directly between the two switches.

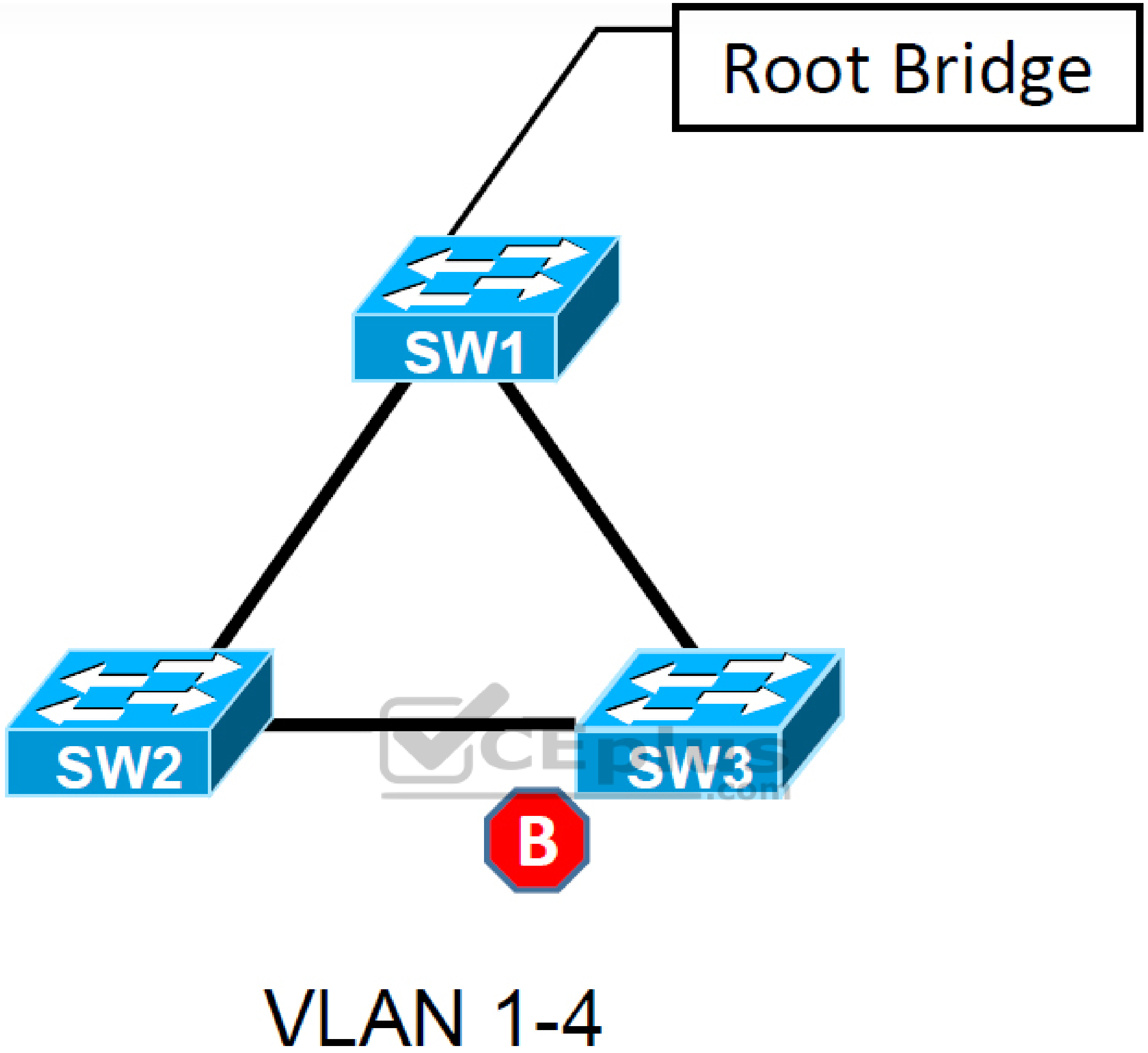


Figure 4-1 Common Spanning Tree Instance (CST) Topology

Cisco developed the Per-VLAN Spanning Tree (PVST) protocol to allow for an STP topology for each VLAN. With PVST, the root bridge can be placed on a different switch or can cost ports differently, on a VLAN-by-VLAN basis. This allows for a link to be blocked for one VLAN and forwarding for another.

Figure 4-2 demonstrates how all three switches maintain an STP topology for each of the 4 VLANs. If 10 more VLANs were added to this environment, the switches would have to maintain 14 STP topologies. With the third STP instance for VLAN 3, the blocking port moves to the SW1 ← → SW3 link due to STP tuning to address the needs of the traffic between SW2 (where servers attach) and SW3 (where clients attach). On the fourth STP instance, devices on VLAN 4 reside only on SW2 and SW3, so moving the blocking port to the SW2 ← → SW1 link allows for optimal traffic flow.

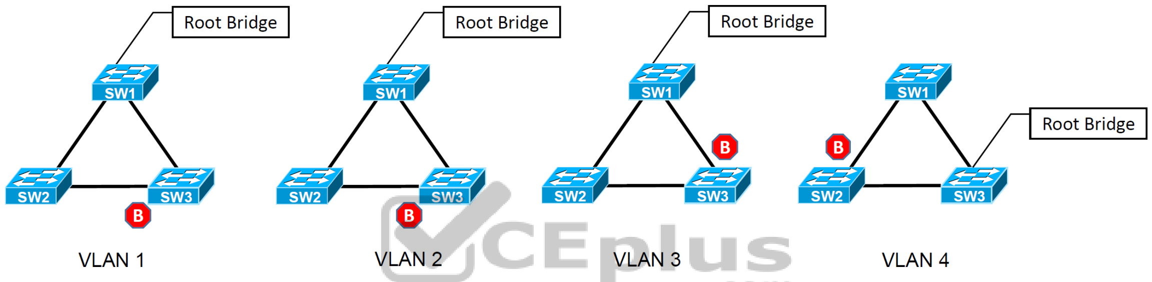


Figure 4-2 Per-VLAN Spanning Tree (PVST) Topologies

Key Topic

Now, in environments with thousands of VLANs, maintaining an STP state for all the VLANs can become a burden to the switch’s processors. The switches must process BPDUs for every VLAN, and when a major trunk link fails, they must compute multiple STP operations to converge the network. MST provides a blended approach by mapping one or multiple VLANs onto a single STP tree, called an *MST instance (MSTI)*.

Figure 4-3 shows how all three switches maintain three STP topologies for 4 VLANs. If 10 more VLANs were added to this environment, then the switches would maintain three STP topologies if they aligned to one of the three existing MSTIs. VLANs 1 and 2 correlate to one MSTI, VLAN 3 to a second MSTI, and VLAN 4 to a third MSTI.

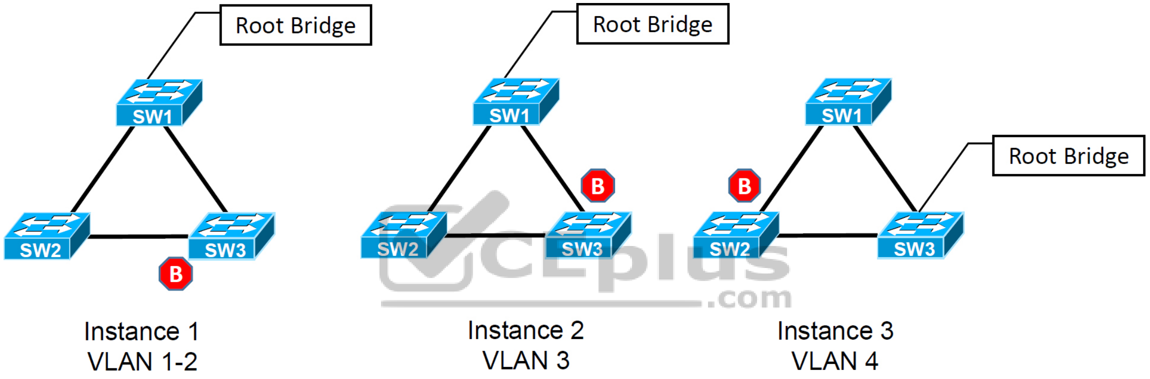


Figure 4-3 MST Topologies

Key Topic

A grouping of MST switches with the same high-level configuration is known as an *MST region*. MST incorporates mechanisms that makes an MST region appear as a single virtual switch to external switches as part of a compatibility mechanism.

Figure 4-4 demonstrates the concept further, showing the actual STP topology beside the topology perceived by devices outside the MST region. Normal STP operations would calculate SW5 blocking the port toward SW3 by using the operations explained in Chapter 2, “Spanning Tree Protocol”. But special notice should go toward SW3 blocking the port toward SW1. Normally SW3 would mark that port as an RP, but because it sees the topology from a larger collective, it is blocking that port rather than blocking the port between SW2 and SW3. In addition, SW7 is blocking the port toward the MST region. SW7 and SW5 are two physical hops away from the root bridge, but SW5 is part of the MST region virtual switch and appears to be one hop away, from SW7’s perspective. That is why SW7 places its port into a blocking state.

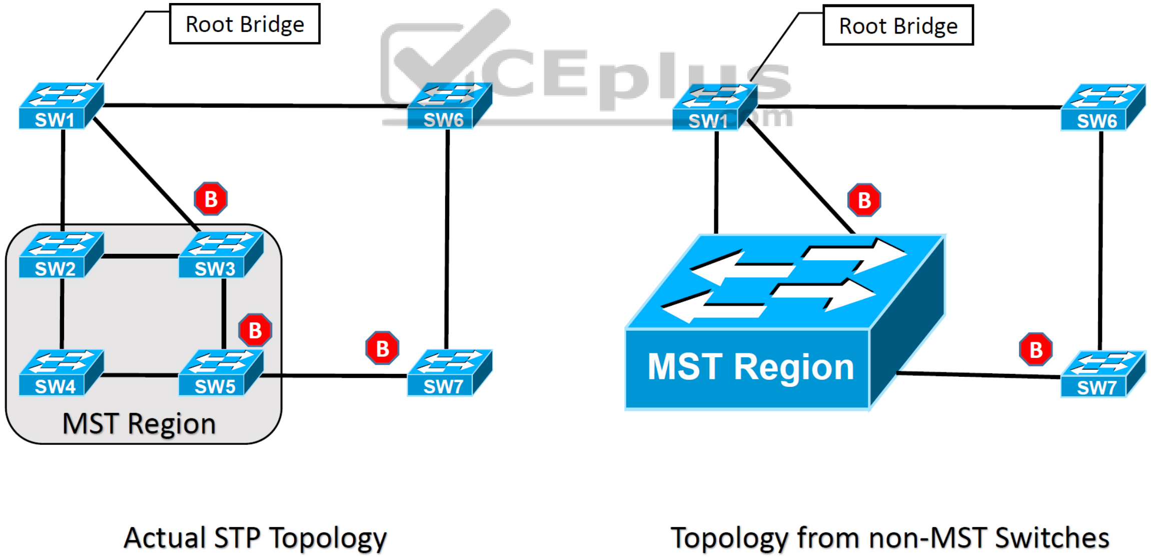


Figure 4-4 Operating Functions Within an MST Region

MST Instances (MSTIs)

MST uses a special STP instance called the *internal spanning tree (IST)*, which is always the first instance, instance 0. The IST runs on all switch port interfaces for switches in the MST region, regardless of the VLANs associated with the ports. Additional information about other MSTIs is included (nested) in the IST BPDU that is transmitted throughout the MST region. This enables the MST to advertise only one set of BPDUs, minimizing STP traffic regardless of the number of instances while providing the necessary information to calculate the STP for other MSTIs.

Note

Cisco supports up to 16 MST instances by default. The IST is always instance 0, so instances 1 to 15 can support other VLANs. There is not a special name for instances 1 to 15; they are simply known as MSTIs.

MST Configuration

MST is configured using the following process:

Step 1. Define MST as the spanning tree protocol with the command **spanning-tree mode mst**.

Step 2. (Optional) Define the MST instance priority, using one of two methods:

- **spanning-tree mst** *instance-number* **priority** *priority*

The priority is a value between 0 and 61,440, in increments of 4096.

- **spanning-tree mst** *instance-number* **root** {**primary** | **secondary**}[**diameter** *diameter*]

The **primary** keyword sets the priority to 24,576, and the **secondary** keyword sets the priority to 28,672.

Step 3. Associate VLANs to an MST instance. By default, all VLANs are associated to the MST 0 instance. The MST configuration submode must be entered with the command **spanning-tree mst configuration**. Then the VLANs are assigned to a different MST instance with the command **instance** *instance-number* **vlan** *vlan-id*.

Step 4. Specify the mst version number. The MST version number must match for all switches in the same MST region. The MST version number is configured with the submode configuration command **revision** *version*.

Step 5. (Optional) Define the MST region name. MST regions are recognized by switches that share a common name. By default, a region name is an empty string. The MST region name is set with the command **name** *mst-region-name*.

Example 4-1 demonstrates the MST configuration on SW1. MST instance 2 contains VLAN 99, MST instance 1 contains VLANs 10 and 20, and MST instance 0 contains all the other VLANs.

Example 4-1 Sample MST Configuration on SW1

```
SW1(config)# spanning-tree mode mst
SW1(config)# spanning-tree mst 0 root primary
SW1(config)# spanning-tree mst 1 root primary
SW1(config)# spanning-tree mst 2 root primary
SW1(config)# spanning-tree mst configuration
SW1(config-mst)# name ENTERPRISE_CORE
SW1(config-mst)# revision 2
SW1(config-mst)# instance 1 vlan 10,20
SW1(config-mst)# instance 2 vlan 99
```

The command **show spanning-tree mst configuration** provides a quick verification of the MST configuration on a switch. Example 4-2 shows the output. Notice that MST instance 0 contains all the VLANs except for VLANs 10, 20, and 99, regardless of whether those VLANs are configured on the switch. MST instance 1 contains VLAN 10 and 20, and MST instance 2 contains only VLAN 99.

Example 4-2 Verifying the MST Configuration

```
SW2# show spanning-tree mst configuration
Name      [ENTERPRISE_CORE]
Revision  2     Instances configured 3
```

```
Instance  Vlans mapped
--------  ---------------------------------------------------------
0         1-9,11-19,21-98,100-4094
1         10,20
2         99
```

MST Verification

The relevant spanning tree information can be obtained with the command **show spanning-tree**. However, the VLAN numbers are not shown, and the MST instance is provided instead. In addition, the priority value for a switch is the MST instance plus the switch priority. Example 4-3 shows the output of this command.

Example 4-3 Brief Review of MST Status

```
SW1# show spanning-tree
! Output omitted for brevity
! Spanning Tree information for Instance 0 (All VLANs but 10,20,
MST0
  Spanning tree enabled protocol mstp
  Root ID    Priority    24576
             Address     0062.ec9d.c500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 1

  Bridge ID  Priority    24576  (priority 0 sys-id-ext 0)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/2             Desg FWD 20000     128.2    P2p
Gi1/0/3             Desg FWD 20000     128.3    P2p

! Spanning Tree information for Instance 1 (VLANs 10 and 20)
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    24577
             Address     0062.ec9d.c500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/2             Desg FWD 20000     128.2    P2p
Gi1/0/3             Desg FWD 20000     128.3    P2p

! Spanning Tree information for Instance 0 (VLAN 30)
MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    24578
             Address     0062.ec9d.c500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15

  Bridge ID  Priority    24578  (priority 24576 sys-id-ext 2)
             Address     0062.ec9d.c500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15
```

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------
Gi1/0/2             Desg FWD 20000     128.2    P2p
Gi1/0/3             Desg FWD 20000     128.3    P2p
```

A consolidated view of the MST topology table is displayed with the command **show spanning-tree mst** [*instance-number*]. The optional *instance-number* can be included to restrict the output to a specific instance. The command is shown in Example 4-4. Notice that the VLANs are displayed next to the MST instance, which simplifies troubleshooting.

Example 4-4 Granular View of MST Topology

```
SW1# show spanning-tree mst
! Output omitted for brevity

##### MST0    vlans mapped:   1-9,11-19,21-98,100-4094
Bridge        address 0062.ec9d.c500  priority      0     (0 sys
Root          this switch for the CIST
Operational   hello time 2 , forward delay 15, max age 20, txhol
Configured    hello time 2 , forward delay 15, max age 20, max h

Interface                        Role Sts Cost      Prio.Nbr Typ
----------------                 ---- --- --------- -------- ---
Gi1/0/2                          Desg FWD 20000     128.2    P2p
Gi1/0/3                          Desg FWD 20000     128.3    P2p

##### MST1    vlans mapped:   10,20
```

```
Bridge        address 0062.ec9d.c500  priority      24577 (24576
Root          this switch for MST1

Interface                        Role Sts Cost      Prio.Nbr Typ
----------------                 ---- --- --------- -------- ---
Gi1/0/2                          Desg FWD 20000     128.2    P2p
Gi1/0/3                          Desg FWD 20000     128.3    P2p

##### MST2    vlans mapped:   99
Bridge        address 0062.ec9d.c500  priority      24578 (24576
Root          this switch for MST2

Interface                        Role Sts Cost      Prio.Nbr Typ
----------------                 ---- --- --------- -------- ---
Gi1/0/2                          Desg FWD 20000     128.2    P2p
Gi1/0/3                          Desg FWD 20000     128.3    P2p
```

The specific MST settings are viewed for a specific interface with the command **show spanning-tree mst interface** *interface-id*, as shown in Example 4-5. Notice that the output in this example includes additional information about optional STP features such as BPDU filter and BPDU guard.

Example 4-5 Viewing Interface-Specific MST Settings

```
SW2# show spanning-tree mst interface gigabitEthernet 1/0/1

GigabitEthernet1/0/1 of MST0 is root forwarding
Edge port: no             (default)        port guard : none
Link type: point-to-point (auto)           bpdu filter: disable
Boundary : internal                        bpdu guard : disable
```

```
Bpdus sent 17, received 217

Instance Role Sts Cost      Prio.Nbr Vlans mapped
-------- ---- --- --------- -------- ----------------------------
0        Root FWD 20000     128.1    1-9,11-19,21-98,100-4094
1        Root FWD 20000     128.1    10,20
2        Root FWD 20000     128.1    99
```

MST Tuning

MST supports the tuning of port cost and port priority. The interface configuration command **spanning-tree mst** *instance-number* **cost** *cost* sets the interface cost. Example 4-6 demonstrates the configuration of SW3's Gi1/0/1 port being modified to a cost of 1 and verification of the interface cost before and after the change.

Example 4-6 Changing the MST Interface Cost

```
SW3# show spanning-tree mst 0
! Output omitted for brevity
Interface                         Role Sts Cost      Prio.Nbr Type
----------------                  ---- --- --------- -------- ----
Gi1/0/1                           Root FWD 20000     128.1    P2p
Gi1/0/2                           Altn BLK 20000     128.2    P2p
Gi1/0/5                           Desg FWD 20000     128.5    P2p

SW3# configure term
Enter configuration commands, one per line. End with CNTL/Z.
SW3(config)# interface gi1/0/1
```

```
SW3(config-if)# spanning-tree mst 0 cost 1

SW3# show spanning-tree mst 0
! Output omitted for brevity
Interface                         Role Sts Cost      Prio.Nbr Type
----------------                  ---- --- --------- -------- ----
Gi1/0/1                           Root FWD 1         128.1    P2p
Gi1/0/2                           Desg FWD 20000     128.2    P2p
Gi1/0/5                           Desg FWD 20000     128.5    P2p
```

The interface configuration command **spanning-tree mst** *instance-number* **port-priority** *priority* sets the interface priority. Example 4-7 demonstrates the configuration of SW4's Gi1/0/5 port being modified to a priority of 64 and verification of the interface priority before and after the change.

Example 4-7 Changing the MST Interface Priority

```
SW4# show spanning-tree mst 0
! Output omitted for brevity
##### MST0    vlans mapped:   1-9,11-19,21-98,100-4094
Interface                         Role Sts Cost      Prio.Nbr Type
----------------                  ---- --- --------- -------- ----
Gi1/0/2                           Root FWD 20000     128.2    P2p
Gi1/0/5                           Desg FWD 20000     128.5    P2p
Gi1/0/6                           Desg FWD 20000     128.6    P2p

SW4# configure term
Enter configuration commands, one per line. End with CNTL/Z.
SW4(config)# interface gi1/0/5
SW4(config-if)# spanning-tree mst 0 port-priority 64
```

```
SW4# show spanning-tree mst 0
! Output omitted for brevity
##### MST0    vlans mapped:   1-9,11-19,21-98,100-4094
Interface                          Role Sts Cost      Prio.Nbr Type
----------------                   ---- --- --------- -------- ----
Gi1/0/2                            Root FWD 20000     128.2    P2p
Gi1/0/5                            Desg FWD 20000      64.5    P2p
Gi1/0/6                            Desg FWD 20000     128.6    P2p
```

Common MST Misconfigurations

There are two common misconfigurations within the MST region that network engineers should be aware of:

- VLAN assignment to the IST

- Trunk link pruning

These scenarios are explained in the following sections.

VLAN Assignment to the IST

Remember that the IST operates across all links in the MST region, regardless of the VLAN assigned to the actual port. The IST topology may not correlate to the access layer and might introduce a blocking port that was not intentional.

Figure 4-5 presents a sample topology in which VLAN 10 is assigned to the IST, and VLAN 20 is assigned to MSTI 1. SW1 and SW2 contain two network links between them, with VLAN 10 and VLAN20. It appears as if traffic between PC-A and PC-B would flow across the Gi1/0/2 interface, as it is an access port assigned to VLAN 10. However, all interfaces belong to the IST instance. SW1 is the root bridge, and all of its ports are designated ports (DPs), so SW2 must block either Gi1/0/1 or Gi1/0/2. SW2 blocks Gi1/0/2, based on the port identifier from SW1, which is Gi1/0/2. So now SW2 is blocking the Gi1/0/2 for the IST instance, which is the instance that VLAN 10 is mapped to.

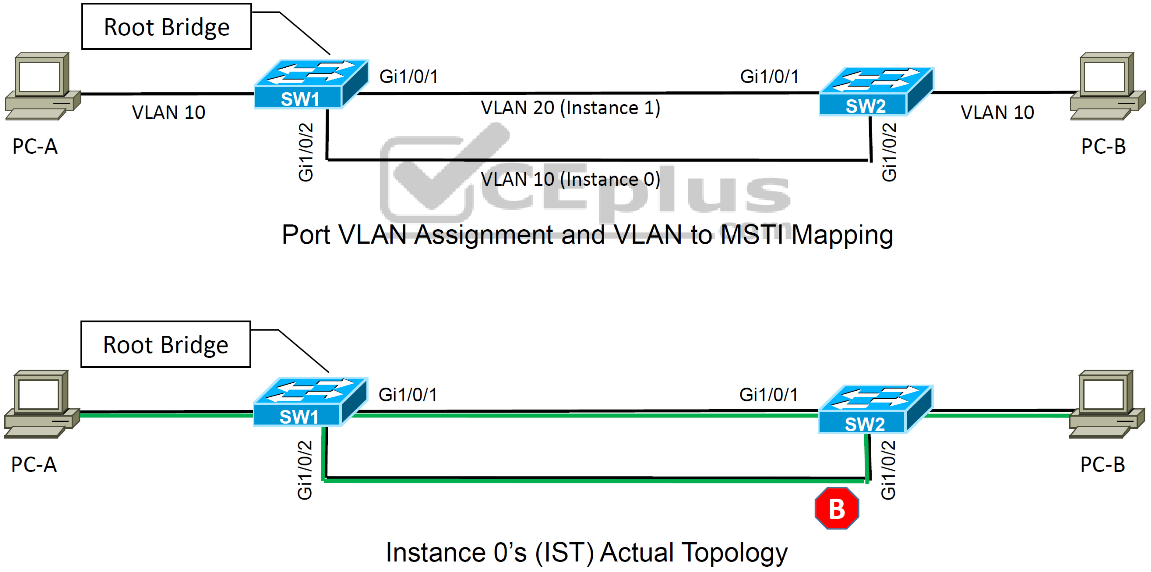


Figure 4-5 Understanding the IST Topology

There are two solutions for this scenario:

- Move VLAN 10 to an MSTI instance other than the IST. If you do this, the switches will build a topology based on the links in use by that MSTI.

- Allow the VLANs associated with the IST on all interswitch (trunk) links.

Trunk Link Pruning

Pruning of VLANs on a trunk link is a common practice for load balancing. However, it is important that pruning of VLANs does not occur for VLANs in the same MST on different network links.

Figure 4-6 presents a sample topology in which VLAN 10 and VLAN 20 are throughout the entire topology. A junior network engineer has pruned VLANs on the trunk links between SW1 to SW2 and SW1 to SW3 to help load balance traffic. Shortly after implementing the change, users attached to SW1 and SW3 cannot talk to the servers on SW2. This is because while the VLANs on the trunk links have changed, the MSTI topology has not.

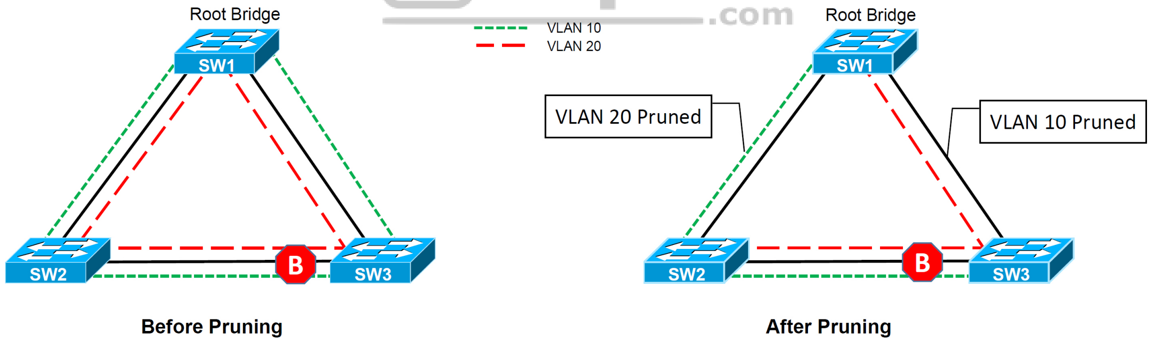


Figure 4-6 Trunk Link Pruning

A simple rule to follow is to only prune all the VLANs in the same MSTI for a trunk link.

Key Topic

MST Region Boundary

The topology for all the MST instances is contained within the IST, which operates internal to the MST region. An *MST region boundary* is any port that connects to a switch that is in a different MST region or that connects to 802.1D or 802.1W BPDUs.

MSTIs never interact outside the region. MST switches can detect PVST+ neighbors at MST region boundaries. Propagating the CST (derived from the IST) at the MST region boundary involves a feature called the *PVST simulation mechanism*.

The PVST simulation mechanism sends out PVST+ (and includes RSTP, too) BPDUs (one for each VLAN), using the information from the IST. To be very explicit, this requires a mapping of one topology (IST) to multiple VLANs (VLANs toward the PVST link). The PVST simulation mechanism is required because PVST+/RSTP topologies do not understand the IST BPDU structure.

When the MST boundary receives PVST+ BPDUs, it does not map the VLANs to the appropriate MSTIs. Instead, the MST boundary maps the PVST+ BPDU from VLAN 1 to the IST instance. The MST boundary engages the PVST simulation mechanism only when it receives a PVST BPDU on a port.

There are two design considerations when integrating an MST region with a PVST+/RSTP environment: The MST region is the root bridge or the MST region is not a root bridge for any VLAN. These scenarios are explained in the following sections.

MST Region as the Root Bridge

Making the MST region the root bridge ensures that all region boundary ports flood the same IST instance BPDU to all the VLANs in the PVST topology. Making the IST instance more preferable than any other switch in the PVST+ topology enables this design. The MST region appears as a single entity, and the PVST+ switches detect the alternate link and place it into a blocking state.

Figure 4-7 shows the IST instance as the root bridge for all VLANs. SW1 and SW2 advertise multiple superior BPDUs for each VLAN toward SW3, which is operating as a PVST+ switch. SW3 is responsible for blocking ports.

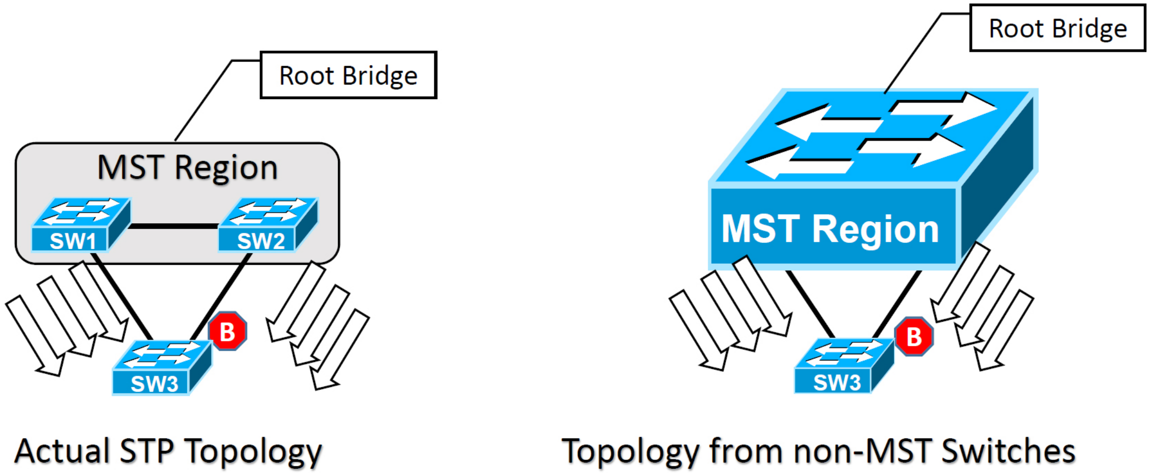


Figure 4-7 MST Region as the Root

Note

SW3 could load balance traffic between the VLANs by setting the STP port cost on a VLAN-by-VLAN basis on each uplink.

MST Region Not a Root Bridge for Any VLAN

In this scenario, the MST region boundary ports can only block or forward for all VLANs. Remember that only the VLAN 1 PVST BPDU is used for the IST and that the IST BPDU is a one-to-many translation of IST BPDUs to all PVST BPDUs There is not an option to load balance traffic because the IST instance must remain consistent.

If an MST switch detects a better BPDU for a specific VLAN on a boundary port, the switch will use BPDU guard to block this port. The port will then be placed into a root inconsistent state. While this may isolate downstream switches, it is done to ensure a loop-free topology; this is called the *PVST simulation check*.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 4-2 lists these key topics and the page number on which each is found.

Key Topic

Table 4-2 Key Topics for Chapter 4

Key Topic Element	Description	Page
Section	Multiple Spanning Tree Protocol	
Paragraph	MST instance	
Paragraph	MST region	
Paragraph	Internal spanning tree (IST)	
Section	MST region boundary	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

Common Spanning Tree (CST)

internal spanning tree (IST)

MST instance (MSTI)

MST region

MST region boundary

PVST simulation check

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 4-3 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 4-3 Command Reference

Task	Command Syntax
Configure the switch for a basic MST region that includes all VLANS and the version number 1	**spanning-tree mode mst** **spanning-tree mst configuration** **instance** 0 **vlan** 1-4094 **revision 1**
Modify a switch's MSTI priority or make it the root bridge for the MSTI	**spanning-tree mst** *instance-number* **priority** *priority* OR **spanning-tree mst** *instance-number* **root** {**primary** | **secondary**}[**diameter** *diameter*]
Specify additional VLANs to an MSTI	**spanning-tree mst configuration** **instance** *instance-number* **vlan** *vlan-id*
Change the MST version number	**spanning-tree mst configuration** **revision** *version*
Change the port cost for a specific MSTI	**spanning-tree mst** *instance-number* **cost** *cost*
Change the port priority for a specific MSTI	**spanning-tree mst** *instance-number* **port-priority** *priority*
Display the MST configuration	**show spanning-tree mst configuration**
Verify the MST switch status	**show spanning-tree mst** [*instance-number*]
View the STP topology for the MST	**show spanning-tree mst interface** *interface-id*

Chapter 5. VLAN Trunks and EtherChannel Bundles

This chapter covers the following subjects:

- **VLAN Trunking Protocol (VTP):** This section provides an overview of how switches become aware of other switches and prevent forwarding loops.

- **Dynamic Trunking Protocol (DTP):** This section examines the improvements made to STP for faster convergence.

- **EtherChannel Bundle:** This section explains how multiple physical interfaces can be combined to form a logical interface to increase throughput and provide seamless resiliency.

This chapter covers multiple features for switch-to-switch connectivity. The chapter starts off by explaining VLAN Trunking Protocol (VTP) and Dynamic Trunking Protocol (DTP) to assist with provisioning of VLANs and ensuring that switch-to-switch connectivity can carry multiple VLANs. Finally, the chapter explains using EtherChannel bundles as a method of adding bandwidth and suppressing topology changes from link failures.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 5-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 5-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topic Section	Questions
VLAN Trunking Protocol	1–4
Dynamic Trunking Protocol	5–6
EtherChannels	7–11

1. Which of the following is not a switch role for VTP?

a. Client

c. Proxy

d. Transparent

e. Off

2. True or false: The VTP summary advertisement includes the VLANs that were recently added, deleted, or modified.

a. True

b. False

3. True or false: There can be only one switch in a VTP domain that has the server role.

a. True

b. False

4. Which of the following is a common disastrous VTP problem with moving a switch from one location to another?

a. The domain certificate must be deleted and re-installed on the VTP server.

b. The moved switch sends an update to the VTP server and deletes VLANs.

c. The moved switch interrupts the VTP.

d. The moved switch causes an STP forwarding loop.

5. True or false: If two switches are connected and configured with the command **switchport mode**, the **dynamic auto** mode establishes a trunk link.

a. True

b. False

6. The command ________ prevents DTP from communicating and agreeing upon a link being a trunk port.

a. switchport dtp disable

b. switchport disable dtp

c. switchport nonegotiate

d. no switchport mode trunk handshake

e. server

7. True or false: PAgP is an industry standard dynamic link aggregation protocol.

a. True

b. False

8. An EtherChannel bundle allows for link aggregation for which types of ports? (Choose all that apply.)

a. Access

b. Trunk

c. Routed

d. Loopback

9. What are the benefits of using an EtherChannel? (Choose two.)

a. Increased bandwidth between device

b. Reduction of topology changes/convergence

c. Smaller configuration

d. Per-packet load balancing

10. One switch has EtherChannel configured as auto. What options on the other switch can be configured to establish an EtherChannel bundle?

a. Auto

b. Active

c. Desirable

d. Passive

11. True or false: LACP and PAgP allow you to set the maximum number of member links in an EtherChannel bundle.

a. True

b. False

Answers to the "Do I Know This Already?" quiz:

1. C

2. B

3. B

4. B

5. B

6. C

7. B

8. A, B ,C

9. A, B

10. C

11. B

FOUNDATION TOPICS

Key Topic

VLAN Trunking Protocol

Before APIs were available on Cisco platforms, configuring a switch was a manual process. Cisco created the proprietary protocol, VLAN Trunking Protocol (VTP), to reduce the burden of provisioning VLANs on switches. Adding a VLAN might seem like a simple task, but in an environment with 100 switches, adding a VLAN required logging in to 100 switches to provision one VLAN. Thanks to VTP, switches that participate in the same VTP domain can have a VLAN created once on a VTP server and propagated to other VTP client switches in the same VTP domain.

There are four roles in the VTP architecture:

- **Server:** The server switch is responsible for the creation, modification, and deletion of VLANs within the VTP domain.

- **Client:** The client switch receives VTP advertisements and modifies the VLANs on that switch. VLANs cannot be configured locally on a VTP client.

- **Transparent:** VTP transparent switches receive and forward VTP advertisements but do not modify the local VLAN database. VLANs are configured only locally.

- **Off:** A switch does not participate in VTP advertisements and does not forward them out of any ports either. VLANs are configured only locally.

Figure 5-1 shows a simple topology in which SW1 is the VTP server, and SW2, SW4, SW5, and SW6 are VTP clients. SW3 is in transparent mode and does not update its VLAN database as changes are propagated through the VTP domain. SW3 forwards VTP changes to SW6.

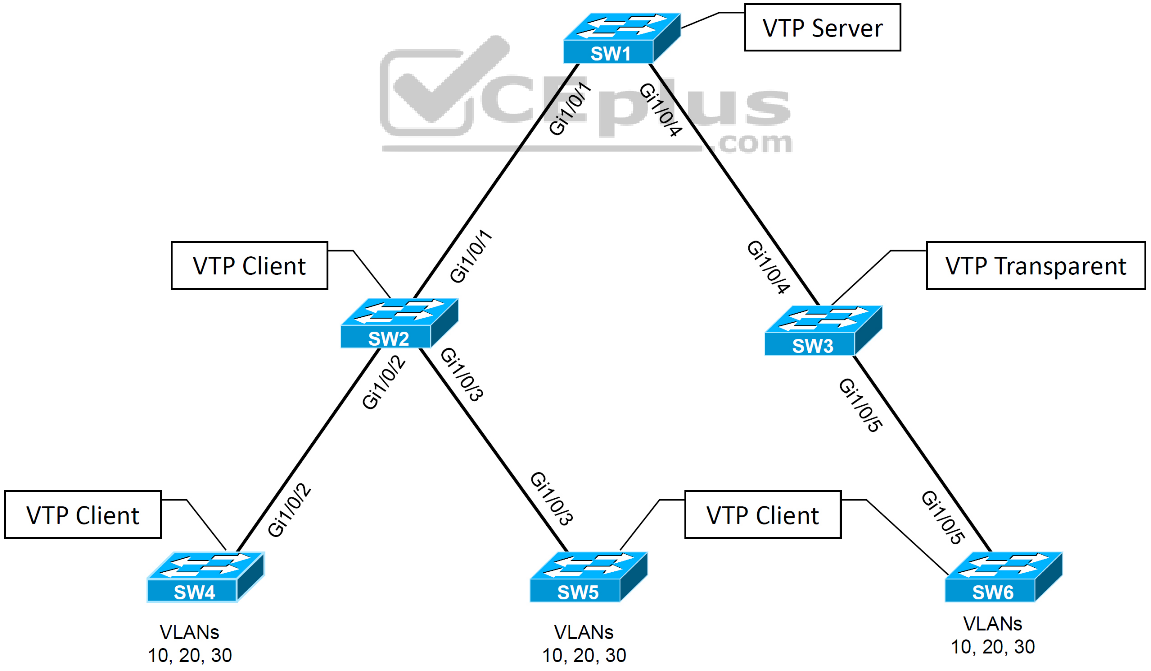


Figure 5-1 Sample Topology for VTP

There are three versions of VTP, and Version 1 is the default. At its simplest, VTP Versions 1 and 2 limited propagation to VLANs numbered 1 to 1005. VTP Version 3 allows for the full range of VLANs 1 to 4094. At the time of this writing, most switches should be capable of running VTP Version 3.

VTP supports having multiple VTP servers in a domain. These servers process updates from other VTP servers just as a client does. If a VTP domain is Version 3, the primary VTP server must be set with the executive command **vtp primary**.

VTP Communication

VTP advertises updates by using a multicast address across the trunk links for advertising updates to all the switches in the VTP domain. There are three main types of advertisements:

- **Summary:** This advertisement occurs every 300 seconds or when a VLAN is added, removed, or changed. It includes the VTP version, domain, configuration revision number, and time stamp.

- **Subset:** This advertisement occurs after a VLAN configuration change occurs. It contains all the relevant information for the switches to make changes to the VLANs on them.

- **Client requests:** This advertisement is a request by a client to receive the more detailed subset advertisement. Typically, this occurs when a switch with a lower revision number joins the VTP domain and observes a summary advertisement with a higher revision than it has stored locally.

VTP Configuration

The following are the steps for configuring VTP:

Step 1. Define the VTP version with the command **vtp version** {**1** | **2** | **3**}.

Step 2. Define the VTP domain with the command **vtp domain** *domain-name*. Changing the VTP domain resets the local switches version to 0.

Step 3. Define the VTP switch role with the command **vtp mode** { **server** | **client** | **transparent** | **none**}.

Step 4. (Optional) Secure the VTP domain with the command **vtp password** *password*. (This step is optional but recommended because it helps prevent unauthorized switches from joining the VTP domain.)

Example 5-1 demonstrates the VTP configuration on SW1, SW2, SW3, and SW6 from Figure 5-1. It shows sample configurations for three of the VTP roles: SW1 as a client, SW3 as transparent, and the other switches as VTP clients.

Example 5-1 Configuring the VTP Domain

```
SW1(config)# vtp domain CiscoPress
Changing VTP domain name from CCNP to CiscoPress
SW1(config)# vtp version 3
09:08:11.965: %SW_VLAN-6-OLD_CONFIG_FILE_READ: Old version 2 VLAN
  file detected and read OK. Version 3 files will be written in t
09:08:12.085: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name cha
SW1(config)# vtp mode server
Setting device to VTP Server mode for VLANS.
```

```
SW1(config)# vtp password PASSWORD
Setting device VTP password to PASSWORD
SW1(config)# exit
SW1# vtp primary
This system is becoming primary server for feature vlan
No conflicting VTP3 devices found.
Do you want to continue? [confirm]
09:25:02.038: %SW_VLAN-4-VTP_PRIMARY_SERVER_CHG: 0062.ec9d.c500
   primary  server for the VLAN VTP feature

SW2(config)# vtp version 3
SW2(config)# vtp domain CISCO
SW2(config)# vtp mode client
SW2(config)# vtp password PASSWORD
Setting device VTP password to PASSWORD

SW3(config)# vtp version 3
SW3(config)# vtp domain CISCO
SW3(config)# vtp mode transparent
SW3(config)# vtp password PASSWORD

SW6(config)# vtp version 3
SW6(config)# vtp domain CISCO
SW6(config)# vtp mode client
SW6(config)# vtp password PASSWORD
```

VTP Verification

The VTP status is verified with the command **show vtp status**. The most important information displayed is the VTP version, VTP domain name, VTP mode, the number of VLANs (standard and extended), and the configuration version.

Example 5-2 shows the output for SW1, SW2, SW3, and SW4. Notice the highlighted operating mode for SW2, SW3, and SW4. The last two VTP Operating Mode entries are not relevant as they are used for other functions.

Example 5-2 Verifying VTP

```
SW1# show vtp status
VTP Version capable              : 1 to 3
VTP version running              : 3
VTP Domain Name                  : CISCO
VTP Pruning Mode                 : Disabled
VTP Traps Generation             : Disabled
Device ID                        : 0062.ec9d.c500

Feature VLAN:
--------------
VTP Operating Mode                 : Server
Number of existing VLANs           : 5
Number of existing extended VLANs       : 0
Maximum VLANs supported locally    : 4096
Configuration Revision             : 1
Primary ID                         : 0062.ec9d.c500
Primary Description                : SW1
MD5 digest                         : 0x9D 0xE3 0xCD 0x04 0x22 0x7(
                                     0x96 0xDE 0x0B 0x7A 0x15 0x65
! The following information is used for other functions not cove
! Core exam and are not directly relevant and will not be explai
Feature MST:
--------------
```

```
VTP Operating Mode                : Transparent

Feature UNKNOWN:
--------------
VTP Operating Mode                : Transparent

SW2# show vtp status | i version run|Operating|VLANS|Revision
VTP version running             : 3
VTP Operating Mode                : Client
Configuration Revision            : 1
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent

SW3# show vtp status | i version run|Operating|VLANS|Revision
VTP version running             : 3
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent

SW6# show vtp status | i version run|Operating|VLANS|Revision
VTP version running             : 3
VTP Operating Mode                : Client
Configuration Revision            : 1
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent
```

Now that the VTP domain has been initialized, let’s look at how VTP works; Example 5-3 shows the creation of VLANS 10, 20, and 30 on SW1. After the VLANS are created on the VTP server, examining the VTP status provides a method to verify that the revision number has incremented (from 1 to 4 because three VLANs were added).

Example 5-3 Creating VLANs on the VTP Domain Server

```
SW1(config)# vlan 10
SW1(config-vlan)# name PCs
SW1(config-vlan)# vlan 20
SW1(config-vlan)# name VoIP
SW1(config-vlan)# vlan 30
SW1(config-vlan)# name Guest

SW1# show vtp status | i version run|Operating|VLANS|Revision
VTP version running             : 3
VTP Operating Mode                : Primary Server
Configuration Revision            : 4
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent
```

Example 5-4 confirms that SW6 has received the VTP update messages from SW3, which is operating in transparent mode. Notice that SW6 shows a configuration revision of 4, which matches the configuration revision number from SW1. The VLAN database confirms that all three VLANS were created on this switch without needing to be configured through the CLI.

Example 5-4 Verifying VTP with a Transparent Switch

```
SW6# show vtp status | i version run|Operating|VLANS|Revision
VTP version running             : 3
VTP Operating Mode                : Client
Configuration Revision            : 4
VTP Operating Mode                : Transparent
VTP Operating Mode                : Transparent

SW6# show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- ----------------
1    default                          active    Gi1/0/1, Gi1/0/2,
                                                Gi1/0/5, Gi1/0/6,
                                                Gi1/0/8, Gi1/0/9,
                                                Gi1/0/11, Gi1/0/1
                                                Gi1/0/14, Gi1/0/1
                                                Gi1/0/17, Gi1/0/1
                                                Gi1/0/20, Gi1/0/2
                                                Gi1/0/23, Gi1/0/2
10   PCs                              active
20   VoIP                             active
30   Guest                            active
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
```

Key Topic

It is very important that every switch that connects to a VTP domain has the VTP revision number reset to 0. Failing to reset the revision number on a switch could result in the switch providing an update to the VTP server. This is not an issue if

VLANs are added but is catastrophic if VLANs are removed because those VLANs will be removed throughout the domain.

When a VLAN is removed from a switch, the access port is moved to VLAN 1. It is then necessary to reassign VLANs to every port associated to the VLAN(s) that were removed.

DYNAMIC TRUNKING PROTOCOL

Chapter 1, "Packet Forwarding," describes how trunk switch ports connect a switch to another device (for example, a switch, a firewall) while carrying multiple VLANs across them. The most common format involves statically setting the switch port to a trunk port, but Cisco provides a mechanism for switch ports to dynamically form a trunk port.

Key Topic

Dynamic trunk ports are established by the switch port sending Dynamic Trunking Protocol (DTP) packets to negotiate whether the other end can be a trunk port. If both ports can successfully negotiate an agreement, the port will become a trunk switch port. DTP advertises itself every 30 seconds to neighbors so that they are kept aware of its status. DTP requires that the VTP domain match between the two switches.

There are three modes to use in setting a switch port to trunk:

- **Trunk:** This mode statically places the switch port as a trunk and advertises DTP packets to the other end to establish a dynamic trunk. Place a switch port in this mode with the command **switchport mode trunk**.

- **Dynamic desirable:** In this mode, the switch port acts as an access port, but it listens for and advertises DTP packets to the other end to establish a dynamic trunk. If it is successful in negotiation, the port becomes a trunk port. Place a switch port in this mode with the command **switchport mode dynamic desirable**.

- **Dynamic auto:** In this mode, the switch port acts as an access port, but it listens for DTP packets. It responds to DTP packets and, upon successful negation, the port becomes a trunk port. Place a switch port in this mode with the command **switchport mode dynamic auto**.

A trunk link can successfully form in almost any combination of these modes unless both ends are configured as dynamic auto. Table 5-2 shows a matrix for successfully establishing a dynamic trunk link.

Table 5-2 Matrix for Establishing a Dynamic Trunk

		Switch 2		
		Trunk	**Dynamic Desirable**	**Dynamic Auto**
Switch 1	Trunk	✓	✓	✓
	Dynamic desirable	✓	✓	✓
	Dynamic auto	✓	✓	X

Example 5-5 shows the configuration of DTP on SW1's Gi1/0/2 as a dynamic auto switch port and SW2's Gi1/0/1 as a dynamic desirable switch port.

Example 5-5 Configuring DTP on SW1 and SW2

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface gi1/0/2
SW1(config-if)# switchport mode dynamic auto

SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface gi1/0/1
SW2(config-if)# switchport mode dynamic desirable
```

The trunk port status is verified with the command **show interface** [*interface-id*] **trunk**, as shown in Example 5-6. Notice that SW1 shows the mode *auto*, and SW2 shows the mode *desirable*.

Example 5-6 Verifying Dynamic Trunk Port Status

```
SW1# show interfaces trunk
! Output omitted for brevity

Port        Mode             Encapsulation  Status        Native
Gi1/0/2     auto             802.1q         trunking      1

Port        Vlans allowed on trunk
Gi1/0/2     1-4094

SW2# show interfaces trunk
! Output omitted for brevity

Port        Mode             Encapsulation  Status        Native
Gi1/0/1     desirable        802.1q         trunking      1

Port        Vlans allowed on trunk
Gi1/0/1     1-4094
```

Note

The mode for a statically configured trunk port is on.

Key Topic

A static trunk port attempts to establish and negotiate a trunk port with a neighbor by default. However, the interface configuration command **switchport nonegotiate** prevents that port from forming a trunk port with a dynamic desirable or dynamic auto switch port. Example 5-7 demonstrates the use of this command on SW1's Gi1/0/2 interface. The setting is then verified by looking at the switch port status. Notice that Negotiation of Trunk now displays as Off.

Example 5-7 Disabling Trunk Port Negotiation

```
SW1# show run interface gi1/0/2
Building configuration...
!
interface GigabitEthernet1/0/2
 switchport mode trunk
!!!! switchport nonegotiate
end

SW1# show interfaces gi1/0/2 switchport | i Trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
```

Note

As a best practice, configure both ends of a link as a fixed port type (using **switchport mode access** or **switchport mode trunk**) to remove any uncertainty about the port's operations.

ETHERCHANNEL BUNDLE

Ethernet network speeds are based on powers of 10 (10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps). When a link between switches becomes saturated, how can more bandwidth be added to that link to prevent packet loss?

If both switches have available ports with faster throughput than the current link (for example, 10 Gbps versus 1 Gbps), then changing the link to higher-speed interfaces solves the bandwidth contingency problem. However, in most cases, this is not feasible.

Ideally, it would be nice to plug in a second cable and double the bandwidth between the switches. However, Spanning Tree Protocol (STP) will place one of the ports into a blocking state to prevent forwarding loops, as shown in Figure 5-2.

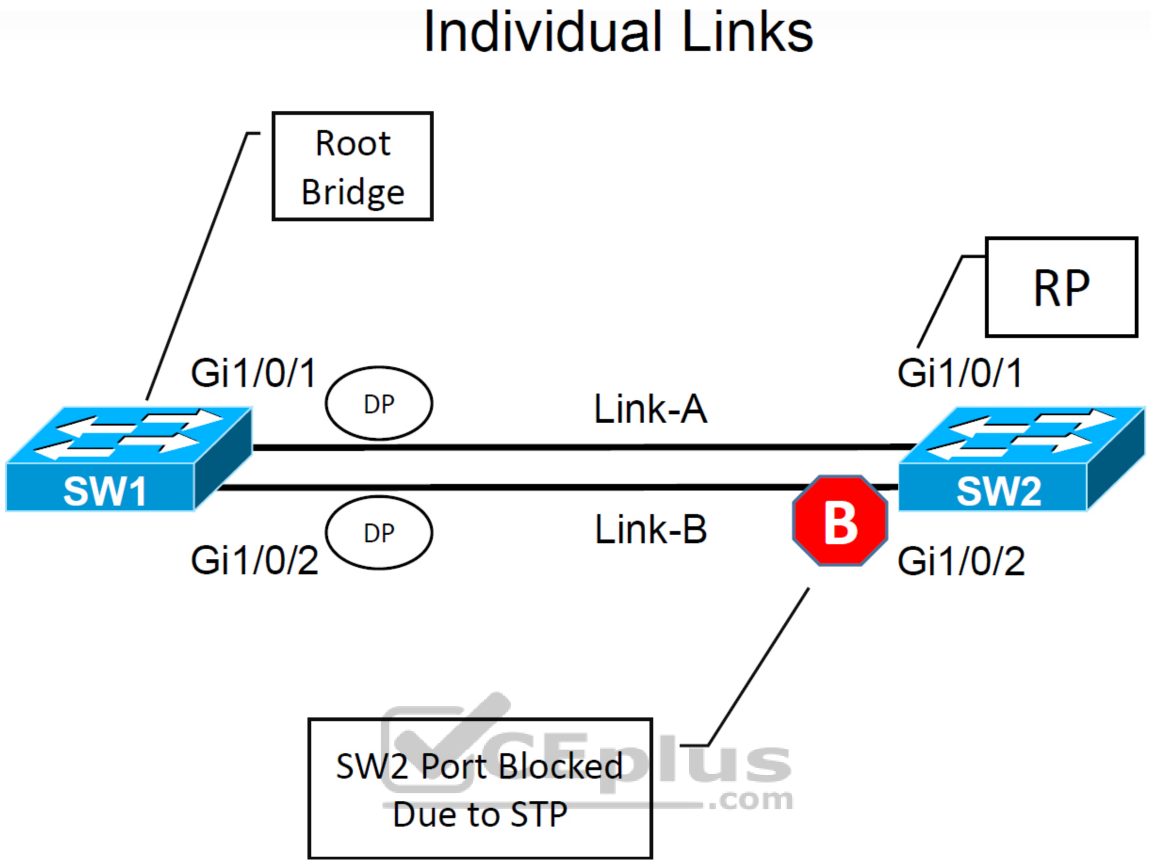


Figure 5-2 Multiple Links with STP

Fortunately, the physical links can be aggregated into a logical link called an EtherChannel bundle. The industry-based term for an EtherChannel bundle is *EtherChannel* (for short), or *port channel*, which is defined in the IEEE 802.3AD link aggregation specification. The physical interfaces that are used to assemble the logical EtherChannel are called *member interfaces*. STP operates on a logical link and not on a physical link. The logical link would then have the bandwidth of any active member interfaces, and it would be load balanced across all the links. EtherChannels can be used for either Layer 2 (access or trunk) or Layer 3 (routed) forwarding.

Note

The terms *EtherChannel*, *EtherChannel bundle*, and *port channel* are interchanged frequently on the Catalyst platform, but other Cisco platforms only use the term *port channel* exclusively.

Figure 5-3 shows some of the key components of an EtherChannel bundle between SW1 and SW2, with their Gi1/0/1 and Gi1/0/2 interfaces.

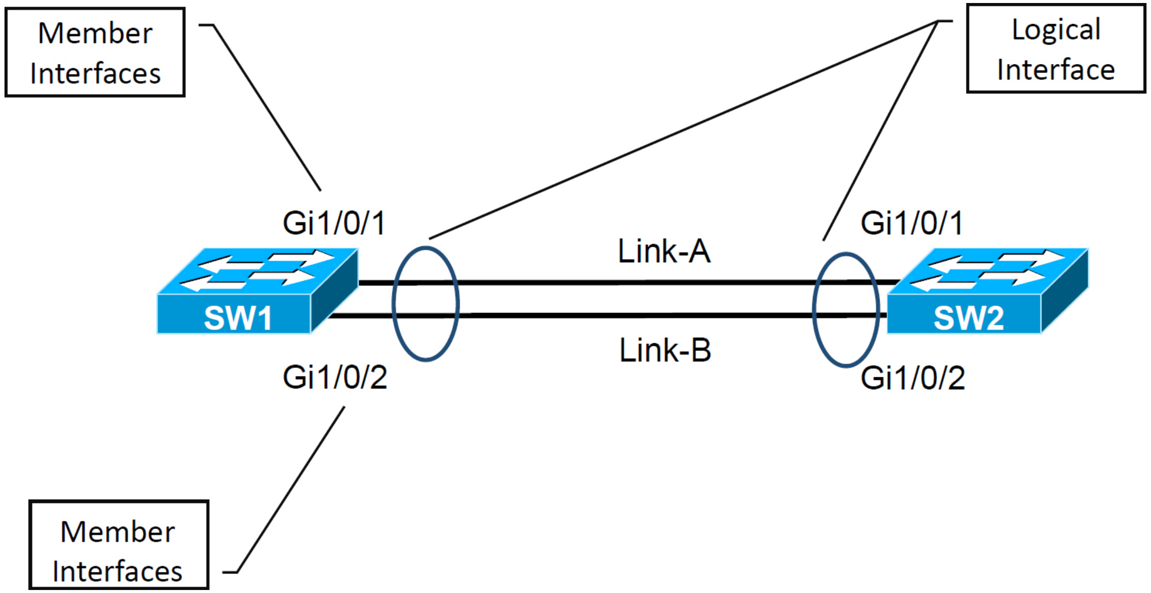


Figure 5-3 EtherChannel Components

A primary advantage of using port channels is a reduction in topology changes when a member link line protocol goes up or down. In a traditional model, a link status change may trigger a Layer 2 STP tree calculation or a Layer 3 route calculation. A member link failure in an EtherChannel does not impact those processes, as long as one active member still remains up.

A switch can successfully form an EtherChannel by statically setting them to an on state or by using a dynamic link aggregation protocol to detect connectivity between devices. Most network engineers prefer to use a dynamic method as it provides a way to ensure end-to-end connectivity between devices across all network links.

A significant downfall of statically setting an EtherChannel to an on state is that there is no health integrity check. If the physical medium degrades and keeps the line protocol in an up state, the port channel will reflect that link as viable for transferring data, which may not be the accurate and would result in sporadic packet loss.

A common scenario involves the use of intermediary devices and technologies (for example, powered network taps, IPSs, Layer 2 firewalls, DWDM) between devices. It is critical for the link state to be propagated to the other side.

Figure 5-4 illustrates a scenario in which SW1 and SW2 have combined their Gi1/0/1 and Gi1/0/2 interfaces into static EtherChannel across optical transport DWDM infrastructure. A failure on Link-A between the DWDM-1 and DWDM-

2 is not propagated to SW1 or to SW2's Gi1/0/1 interface. The switches continue to forward traffic out the Gi1/0/1 interface because those ports still maintain physical state to DWDM-1 or DWDM-2. Both SW1 and SW2 load balance traffic across the Gi1/0/1 interface, resulting in packet loss for the traffic that is sent out of the Gi1/0/1 interface.

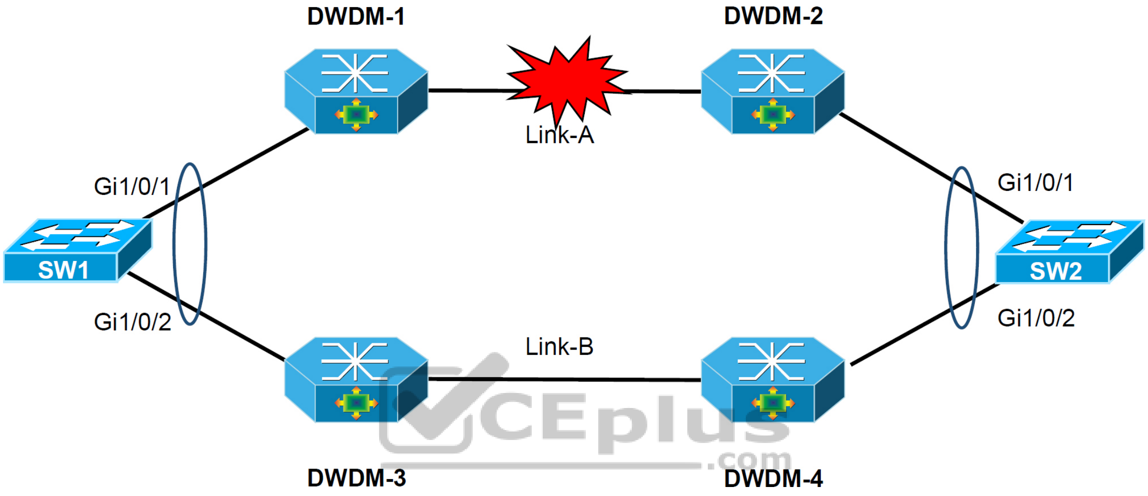


Figure 5-4 Port-Channel Link-State Propagation and Detection

There is not a health-check mechanism with the port-channel ports being statically set to on. However, if a dynamic link aggregation protocol were used between SW1 and SW2, the link failure would be detected, and the Gi1/0/1 interfaces would be made inactive for the EtherChannel.

Dynamic Link Aggregation Protocols

Two common link aggregation protocols are Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP). PAgP is Cisco proprietary and was developed first, and then LACP was created as an open industry standard.

All the member links must participate in the same protocol on the local and remote switches.

Key Topic

PAgP Port Modes

PAgP advertises messages with the multicast MAC address 0100:0CCC:CCCC and the protocol code 0x0104. PAgP can operate in two modes:

- **Auto:** In this PAgP mode, and interface does not initiate an EtherChannel to be established and does not transmit PAgP packets out of it. If an PAgP packet is received from the remote switch, this interface responds and then can establish a PAgP adjacency. If both devices are PAgP auto, a PAgP adjacency does not form.

- **Desirable:** In this PAgP mode, an interface tries to establish an EtherChannel and transmit PAgP packets out of it. Active PAgP interfaces can establish a PAgP adjacency only if the remote interface is configured to auto or desirable.

Key Topic

LACP Port Modes

LACP advertises messages with the multicast MAC address 0180:C200:0002. LACP can operate in two modes:

- **Passive:** In this LACP mode, an interface does not initiate an EtherChannel to be established and does not transmit LACP packets out of it. If an LACP packet is received from the remote switch, this interface responds and then can establish an LACP adjacency. If both devices are LACP passive, an LACP adjacency does not form.

- **Active:** In this LACP mode, an interface tries to establish an EtherChannel and transmit LACP packets out of it. Active LACP interfaces can establish an LACP adjacency only if the remote interface is configured to active or passive.

Key Topic

EtherChannel Configuration

It is possible to configure EtherChannels by going into the interface configuration mode for the member interfaces and assigning them to an EtherChannel ID and configuring the appropriate mode:

- **Static EtherChannel:** A static EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode on**.

- **LACP EtherChannel:** An LACP EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode** {**active** | **passive**}.

- **PAgP EtherChannel:** A PAgP EtherChannel is configured with the interface parameter command **channel-group** *etherchannel-id* **mode** {**auto** | **desirable**} [**non-silent**].

By default, PAgP ports operate in silent mode, which allows a port to establish an EtherChannel with a device that is not PAgP capable and rarely sends packets. Using the optional **non-silent** keyword requires a port to receive PAgP packets before adding it to the EtherChannel. The **non-silent** keyword is recommended when connecting PAgP-compliant switches together; the **non-silent** option results in a link being established more quickly than if this keyword were not used.

The following additional factors need to be considered with EtherChannel configuration:

- Configuration settings for the EtherChannel are placed in the port-channel interface.

- Member interfaces need to be in the appropriate Layer 2 or Layer 3 (that is, no switch port) before being associated with the port channel. The member interface type dictates whether the EtherChannel operates at Layer 2 or Layer 3.

Example 5-8 shows the configuration for EtherChannel 1, using the member interfaces Gi1/0/1 and Gi1/0/2. SW1 uses LACP active (which accepts and initiates a request), and SW2 uses LACP passive (which only responds to an LACP initiation). The EtherChannel will be used as a trunk port, which is configured on each switch after the EtherChannel is created.

Example 5-8 Sample Port-Channel Configuration

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# interface range gi1/0/1-2
SW1(config-if-range)# channel-group 1 mode active
Creating a port-channel interface Port-channel 1
SW1(config-if-range)# interface port-channel 1
SW1(config-if)# switchport mode trunk
13:56:20.210: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/1, changed state to down
13:56:20.216: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/2, changed state to down
13:56:32.214: %ETC-5-L3DONTBNDL2: Gi1/0/2 suspended: LACP current
  on the remote port.
13:56:32.420: %ETC-5-L3DONTBNDL2: Gi1/0/1 suspended: LACP current
  on the remote port.

SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface range gi1/0/1-2
SW2(config-if-range)# channel-group 1 mode passive
Creating a port-channel interface Port-channel 1
SW2(config-if-range)# interface port-channel 1
SW2(config-if)# switchport mode trunk
*13:57:05.434: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/1, changed state to down
*13:57:05.446: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/2, changed state to down
```

```
*13:57:12.722: %ETC-5-L3DONTBNDL2: Gi1/0/1 suspended: LACP curre
  on the remote port.
*13:57:13.072: %ETC-5-L3DONTBNDL2: Gi1/0/2 suspended: LACP curre
  on the remote port.
*13:57:24.124: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/2, changed state to up
*13:57:24.160: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  GigabitEthernet1/0/1, changed state to up
*13:57:25.103: %LINK-3-UPDOWN: Interface Port-channel1, changed s
*13:57:26.104: %LINEPROTO-5-UPDOWN: Line protocol on Interface Po
  changed state to up
```

Verifying Port-Channel Status

After a port channel has been configured, it is essential to verify that the port channel has been established. As shown in Example 5-9, the command **show etherchannel summary** provides an overview of all the configured EtherChannels, along with the status and dynamic aggregation protocol for each one. A second EtherChannel using PAgP was configured on the topology to differentiate between LACP and PAgP interfaces.

Example 5-9 Viewing EtherChannel Summary Status

```
SW1# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
        A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------------------
1      Po1(SU)         LACP      Gi1/0/1(P)  Gi1/0/2(P)
2      Po2(SU)         PAgP      Gi1/0/3(P)  Gi1/0/4(P)
```

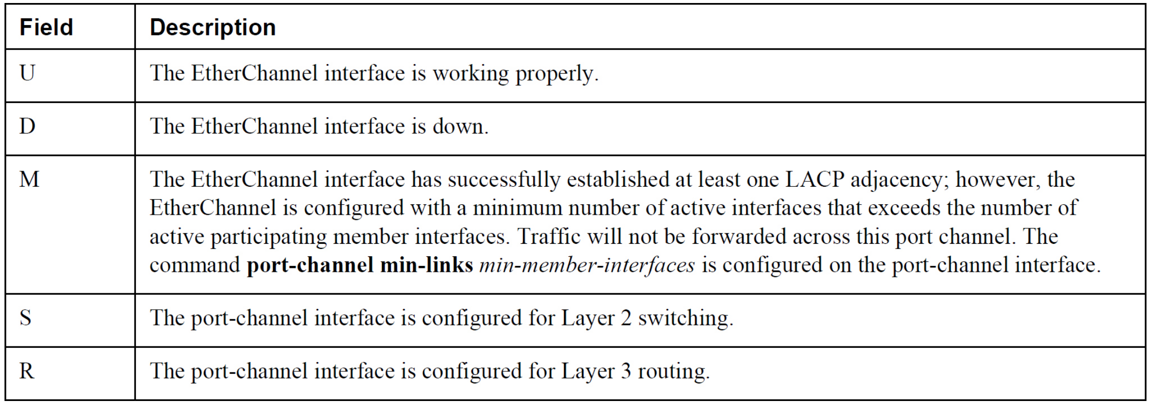
When viewing the output of the **show etherchannel summary** command, the first thing that should be checked is the EtherChannel status, which is listed in the Port-channel column. The status should be U, as highlighted in Example 5-9.

Note

The status codes are case sensitive, so please pay attention to the case of the field.

Table 5-3 provides a brief explanation of other key fields for the logical port-channel interface.

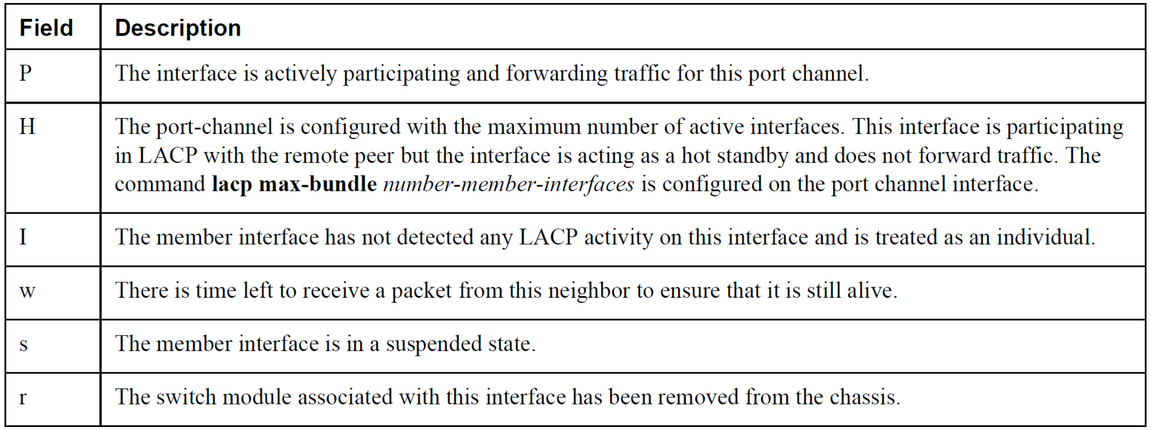
Table 5-3 Logical EtherChannel Interface Status Fields



Field	Description
U	The EtherChannel interface is working properly.
D	The EtherChannel interface is down.
M	The EtherChannel interface has successfully established at least one LACP adjacency; however, the EtherChannel is configured with a minimum number of active interfaces that exceeds the number of active participating member interfaces. Traffic will not be forwarded across this port channel. The command **port-channel min-links** *min-member-interfaces* is configured on the port-channel interface.
S	The port-channel interface is configured for Layer 2 switching.
R	The port-channel interface is configured for Layer 3 routing.

Table 5-4 provides a brief explanation of the fields that are related to the member interfaces.

Table 5-4 EtherChannel Member Interface Status Fields



Field	Description
P	The interface is actively participating and forwarding traffic for this port channel.
H	The port-channel is configured with the maximum number of active interfaces. This interface is participating in LACP with the remote peer but the interface is acting as a hot standby and does not forward traffic. The command **lacp max-bundle** *number-member-interfaces* is configured on the port channel interface.
I	The member interface has not detected any LACP activity on this interface and is treated as an individual.
w	There is time left to receive a packet from this neighbor to ensure that it is still alive.
s	The member interface is in a suspended state.
r	The switch module associated with this interface has been removed from the chassis.

The logical interface can be viewed with the command **show interface port-channel** *port-channel-id*. The output includes traditional interface statistics and lists the member interfaces and indicates that the bandwidth reflects the combined throughput of all active member interfaces. As the bandwidth changes, systems that reference the bandwidth (such as QoS policies and interface costs for routing protocols) adjust accordingly.

Example 5-10 shows the use of the **show interface port-channel** *port-channel-id* command on SW1. Notice that the bandwidth is 2 Gbps and correlates to the two 1 Gbps interfaces in the **show etherchannel summary** command.

Example 5-10 Viewing Port-Channel Interface Status

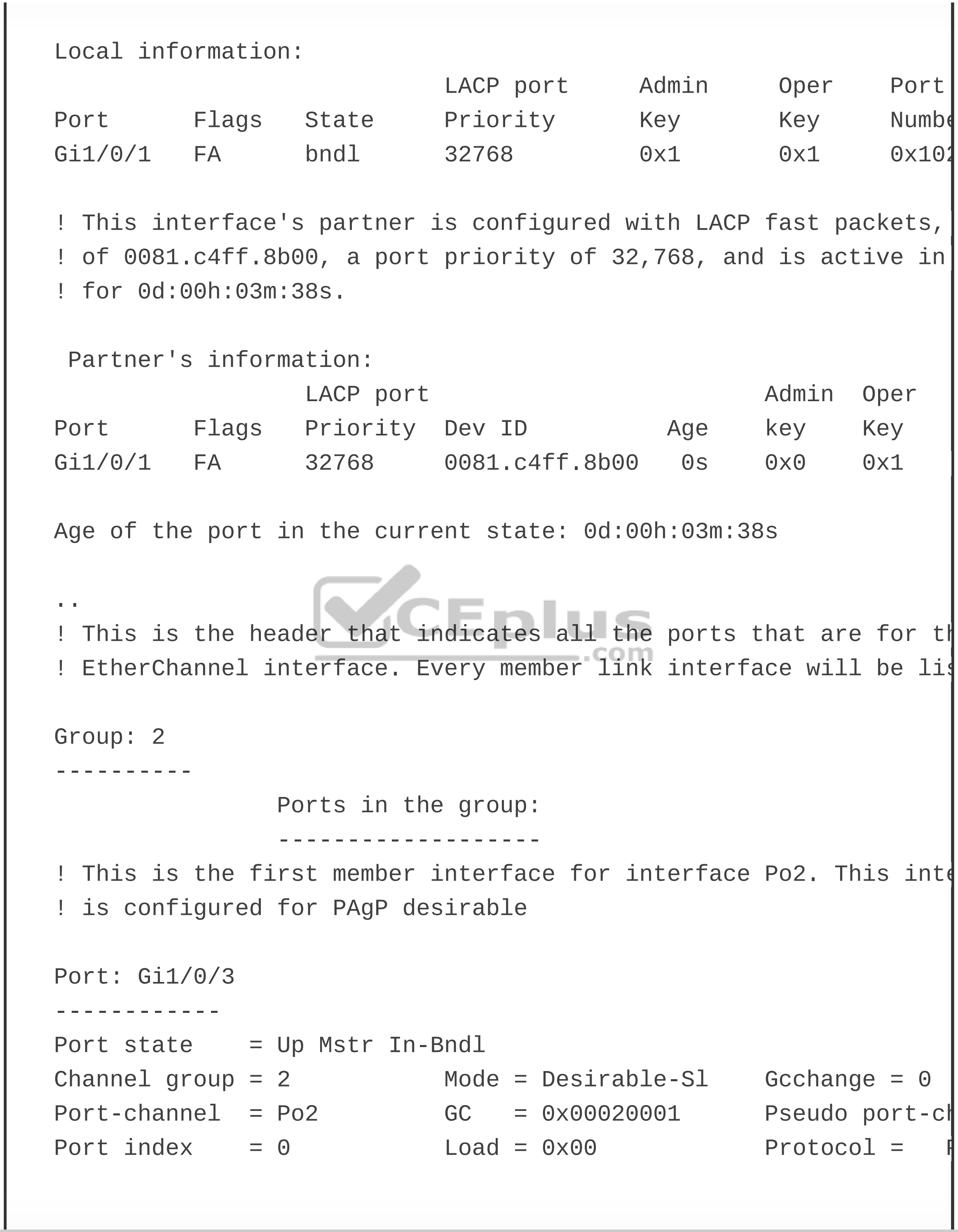
```
SW1# show interfaces port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0062.ec9d.c501 (bia 0062.e
  MTU 1500 bytes, BW 2000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, link type is auto, media type is
  input flow-control is off, output flow-control is unsupported
  Members in this channel: Gi1/0/1 Gi1/0/2
..
```

Viewing EtherChannel Neighbors

The LACP and PAgP packets include a lot of useful information that can help identify inconsistencies in configuration. The command **show etherchannel port** displays detailed instances of the local configuration and information from the packets. Example 5-11 shows the output of this command and explains key points in the output for LACP and PAgP.

Example 5-11 Viewing **show etherchannel port** Output

```
SW1# show etherchannel port
! Output omitted for brevity
                Channel-group listing:
                ----------------------
! This is the header that indicates all the ports that are for th
! EtherChannel interface. Every member link interface will be lis
Group: 1
----------
                Ports in the group:
                -------------------
! This is the first member interface for interface Po1. This inte
! is configured for LACP active
Port: Gi1/0/1
------------
Port state    = Up Mstr Assoc In-Bndl
Channel group = 1           Mode = Active           Gcchange = -
Port-channel  = Po1         GC   =   -              Pseudo port-ch
Port index    = 0           Load = 0x00             Protocol =   L

! This interface is configured with LACP fast packets, has a port
! of 32,768 and is active in the bundle.

Flags:  S - Device is sending Slow LACPDUs   F - Device is sendin
        A - Device is in active mode. P - Device is in passive mo
```



```
Local information:
                                LACP port      Admin      Oper     Port
Port       Flags    State       Priority       Key        Key      Numbe
Gi1/0/1    FA       bndl        32768          0x1        0x1      0x10

! This interface's partner is configured with LACP fast packets,
! of 0081.c4ff.8b00, a port priority of 32,768, and is active in
! for 0d:00h:03m:38s.

 Partner's information:
                    LACP port                          Admin   Oper
Port       Flags    Priority    Dev ID          Age    key     Key
Gi1/0/1    FA       32768       0081.c4ff.8b00   0s    0x0     0x1

Age of the port in the current state: 0d:00h:03m:38s

..
! This is the header that indicates all the ports that are for th
! EtherChannel interface. Every member link interface will be lis

Group: 2
----------
                Ports in the group:
                -------------------
! This is the first member interface for interface Po2. This inte
! is configured for PAgP desirable

Port: Gi1/0/3
------------
Port state    = Up Mstr In-Bndl
Channel group = 2           Mode = Desirable-Sl    Gcchange = 0
Port-channel  = Po2         GC   = 0x00020001      Pseudo port-c
Port index    = 0           Load = 0x00            Protocol =
! This interface is in a consistent state, has a neighbor with th
! 0081.c4ff.8b00 address and has been in the current state for 54

Flags:  S - Device is sending Slow hello. C - Device is in Consis
        A - Device is in Auto mode. P - Device learns on physical
        d - PAgP is down.
Timers: H - Hello timer is running. Q - Quit timer is running.
        S - Switching timer is running. I - Interface timer is ru

Local information:
                                Hello    Partner  PAgP     Learni
Port      Flags State   Timers  Interval Count   Priority   Metho
Gi1/0/3   SC    U6/S7   H       30s      1        128        Any

Partner's information:


          Partner              Partner          Partner         P
Port      Name                 Device ID        Port       Age  F
Gi1/0/3   SW2                  0081.c4ff.8b00   Gi1/0/3      1s S

Age of the port in the current state: 0d:00h:54m:45s
..
```

The output from the **show etherchannel port** command can provide too much information and slow down troubleshooting when a smaller amount of information is needed. The following sections provide some commands for each protocol that provide more succinct information.

LACP

The command **show lacp neighbor** [**detail**] displays additional information about the LACP neighbor and includes the neighbor's system ID, system priority, and whether it is using fast or slow LACP packet intervals as part of the output.

The LACP system identifier is used to verify that the member interfaces are connected to the same device and not split between devices. The local LACP system ID can be viewed by using the command **show lacp** *system-id*. Example 5-12 shows the use of this command.

Example 5-12 Viewing LACP Neighbor Information

```
SW1# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Pass

Channel group 1 neighbors

                  LACP port                        Admin  Oper
Port      Flags   Priority  Dev ID          Age    key    Key
Gi1/0/1   SA      32768     0081.c4ff.8b00   1s    0x0    0x1
Gi1/0/2   SA      32768     0081.c4ff.8b00  26s    0x0    0x1
```

PAgP

The command **show pagp neighbor** displays additional information about the PAgP neighbor and includes the neighbor's system ID, remote port number, and whether it is using fast or slow PAgP packet intervals as part of the output. Example 5-13 shows the use of this command.

Example 5-13 Viewing PAgP Neighbor Information

```
SW1# show pagp neighbor
Flags:  S - Device is sending Slow hello. C - Device is in Consis
        A - Device is in Auto mode. P - Device learns on physical

Channel group 2 neighbors
          Partner                Partner          Partner         P
Port      Name                   Device ID        Port        Age  F
Gi1/0/3   SW2                    0081.c4ff.8b00   Gi1/0/3      11s S
Gi1/0/4   SW2                    0081.c4ff.8b00   Gi1/0/4       5s S
```

Verifying EtherChannel Packets

A vital step in troubleshooting the establishment of port channels is to verify that LACP or PAgP packets are being transmitted between devices. The first troubleshooting step that can be taken is to verify the EtherChannel counters for the appropriate protocol.

LACP

The LACP counters are viewed with the command **show lacp counters**. The output includes a list of the EtherChannel interfaces, their associated member interfaces, counters for LACP packets sent/received, and any errors. An interface should see the sent and received columns increment over a time interval. The failure of the counters to increment indicates a problem. The problem could be related to a physical link, or it might have to do with an incomplete or incompatible configuration with the remote device. Check the LACP counters on the remote device to see if it is transmitting LACP packets.

Example 5-14 demonstrates the **show lacp counters** command on SW2. Notice that the received column does not increment on Gi1/0/2 for port-channel 1, but the sent column does increment. This indicates a problem that should be investigated further.

Example 5-14 Viewing LACP Packet Counters

```
SW2# show lacp counters
             LACPDUs         Marker       Marker Response    LACPD
Port       Sent   Recv     Sent   Recv     Sent   Recv      Pkts
------------------------------------------------------------------
Channel group: 1
Gi1/0/1     23     23       0      0        0      0          0
Gi1/0/2     22      0       0      0        0      0          0

SW2# show lacp counters
             LACPDUs         Marker       Marker Response    LACPD
Port       Sent   Recv     Sent   Recv     Sent   Recv      Pkts
------------------------------------------------------------------
Channel group: 1
Gi1/0/1     28     28       0      0        0      0          0
Gi1/0/2     27      0       0      0        0      0          0
```

Note

The LACP counters can be cleared with the command **clear lacp counters**.

PAgP

The output of the PAgP command **show pagp counters** includes a list of the EtherChannel interfaces, their associated member interfaces, counters for PAgP packets sent/received, and any errors. The PAgP counters can be cleared with the command **clear lacp counters**.

Example 5-15 shows the command **show pagp counters** on SW2 for the second EtherChannel interface that was created on SW1.

Example 5-15 Viewing PAgP Packet Counters

```
SW1# show pagp counters
         Information        Flush        PAgP
Port      Sent    Recv     Sent    Recv    Err Pkts
---------------------------------------------------
Channel group: 2
Gi1/0/3   31      51       0       0       0
Gi1/0/4   44      38       0       0       0
```

Advanced LACP Configuration Options

LACP provides some additional tuning that is not available with PAgP. The following sections explain some of the advanced LACP configuration options and the behavioral impact they have on member interface selection for a port channel.

LACP Fast

The original LACP standards sent out LACP packets every 30 seconds. A link is deemed unusable if an LACP packet is not received after three intervals, which results in a potential 90 seconds of packet loss for a link before that member interface is removed from a port channel.

An amendment to the standards was made so that LACP packets are advertised every 1 second. This is known as *LACP fast* because a link can be identified and removed in 3 seconds compared to the 90 seconds specified in the initial LACP standard. LACP fast is enabled on the member interfaces with the interface configuration command **lacp rate fast**.

Note

All the interfaces on both switches need to be configured the same—either using LACP fast or LACP slow—for the EtherChannel to successfully come up.

Example 5-16 shows how the current LACP state can be identified on the local and neighbor interfaces, along with how an interface can be converted to LACP fast.

Example 5-16 Configuring LACP Fast and Verifying LACP Speed State

```
SW1(config)# interface range gi1/0/1-2
SW1(config-if-range)# lacp rate fast

SW1# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Pass

Channel group 1
                                LACP port     Admin     Oper    Port
Port       Flags    State       Priority      Key       Key     Numbe
Gi1/0/1    FA       bndl        32768         0x1       0x1     0x10
Gi1/0/2    FA       bndl        32768         0x1       0x1     0x10
```

Key Topic

Minimum Number of Port-Channel Member Interfaces

An EtherChannel interface becomes active and up when only one member interface successfully forms an adjacency with a remote device. In some design scenarios using LACP, a minimum number of adjacencies is required before a port-channel interface becomes active. This option can be configured with the port-channel interface command **port-channel min-links** *min-links*.

Example 5-17 shows how to set the minimum number of port-channel interfaces to two and then shut down one of the member interfaces on SW1. This prevents the EtherChannel from meeting the required minimum links and shuts it down. Notice that the port-channel status is *not in use* in the new state.

Example 5-17 Configuring the Minimum Number of EtherChannel Member Interfaces

```
SW1(config)# interface port-channel 1
SW1(config-if)# port-channel min-links 2

SW1(config-if)# interface gi1/0/1
SW1(config-if)# shutdown
10:44:46.516: %ETC-5-MINLINKS_NOTMET: Port-channel Po1 is down bu
   doesn't meet min-links
10:44:47.506: %LINEPROTO-5-UPDOWN: Line protocol on Interface Gig
   Ethernet1/0/2, changed state to down
10:44:47.508: %LINEPROTO-5-UPDOWN: Line protocol on Interface Por
   changed state to down
10:44:48.499: %LINK-5-CHANGED: Interface GigabitEthernet1/0/1, ch
   administratively down
10:44:48.515: %LINK-3-UPDOWN: Interface Port-channel1, changed st

SW1# show etherchannel summary
```

```
! Output Ommitted for Brevity
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
..
Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------------------
1      Po1(SM)         LACP      Gi1/0/1(D)  Gi1/0/2(P)
```

Note

The minimum number of port-channel member interfaces does not need to be configured on both devices to work properly. However, configuring it on both switches is recommended to accelerate troubleshooting and assist operational staff.

Key Topic

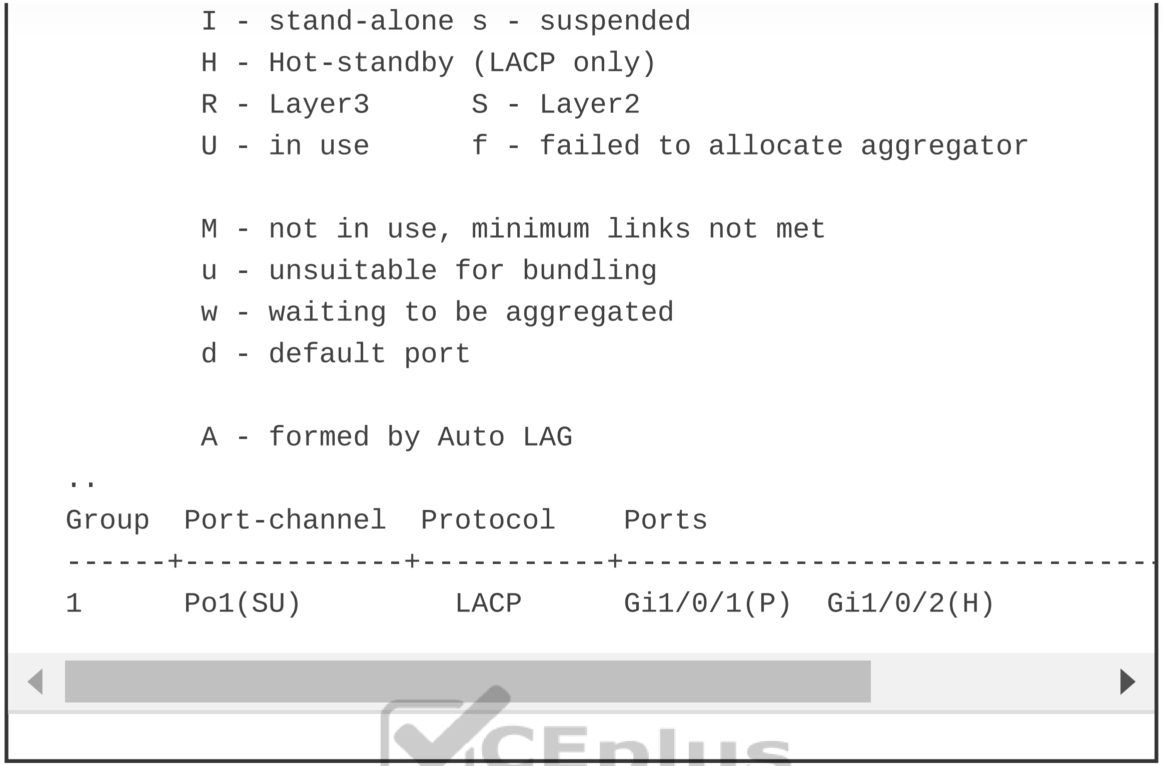
Maximum Number of Port-Channel Member Interfaces

An EtherChannel can be configured to have a specific maximum number of member interfaces in a port channel. This may be done to ensure that the active member interface count proceeds with powers of two (for example, 2, 4, 8) to accommodate load-balancing hashes. The maximum number of member interfaces in a port channel can be configured with the port-channel interface command **lacp max-bundle** *max-links*.

Example 5-18 shows the configuration of the maximum number of active member interfaces for a port channel; you can see that those interfaces now show as Hot-standby.

Example 5-18 Configuring and Verifying the Maximum Links

```
SW1(config)# interface port-channel1
SW1(config-if)# lacp max-bundle 1
11:01:11.972: %LINEPROTO-5-UPDOWN: Line protocol on Interface Gig
   Ethernet1/0/1, changed state to down
11:01:11.979: %LINEPROTO-5-UPDOWN: Line protocol on Interface Gig
   Ethernet1/0/2, changed state to down
11:01:11.982: %LINEPROTO-5-UPDOWN: Line protocol on Interface Por
   changed state to down
11:01:13.850: %LINEPROTO-5-UPDOWN: Line protocol on Interface Gig
   Ethernet1/0/1, changed state to up
11:01:13.989: %LINEPROTO-5-UPDOWN: Line protocol on Interface Por
   changed state to up

SW1# show etherchannel summary
! Output omitted for brevity
Flags:  D - down        P - bundled in port-channel
```



```
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG
..
Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------------------
1      Po1(SU)         LACP      Gi1/0/1(P)  Gi1/0/2(H)
```

The maximum number of port-channel member interfaces needs to be configured only on the master switch for that port channel; however, configuring it on both switches is recommended to accelerate troubleshooting and assist operational staff.

The port-channel master switch controls which member interfaces (and associated links) are active by examining the LACP port priority. A lower port priority is preferred. If the port priority is the same, then the lower interface number is preferred.

Key Topic

LACP System Priority

The *LACP system priority* identifies which switch is the master switch for a port channel. The master switch on a port channel is responsible for choosing which member interfaces are active in a port channel when there are more member interfaces than the maximum number of member interfaces associated with a port-channel interface. The switch with the lower system priority is preferred. The LACP system priority can be changed with the command **lacp system-priority** *priority*.

Example 5-19 shows how the LACP system priority can be viewed and changed.

Example 5-19 Viewing and Changing the LACP System Priority

```
SW1# show lacp sys-id
32768, 0062.ec9d.c500

SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# lacp system-priority 1

SW1# show lacp sys-id
1, 0062.ec9d.c500
```

Key Topic

LACP Interface Priority

LACP interface priority enables the master switch to choose which member interfaces are active in a port channel when there are more member interfaces than the maximum number of member interfaces for a port channel. A port with a lower port priority is preferred. The interface configuration command **lacp port-priority** *priority* sets the interface priority.

Example 5-20 changes the port priority on SW1 for Gi1/0/2 so that it is the most preferred interface when the LACP maximum link has been set to 1. SW1 is the master switch for port channel 1, the Gi1/0/2 interface becomes active, and port Gi1/0/1 become Hot-standby.

Example 5-20 Changing the LACP Port Priority

```
SW1# show etherchannel summary | b Group
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------------------
1      Po1(SU)         LACP      Gi1/0/1(P)  Gi1/0/2(H)

SW1(config)# interface gi1/0/2
SW1(config-if)# lacp port-priority 1
SW1# show etherchannel summary | b Group
Group  Port-channel  Protocol    Ports
```

```
------+-------------+-----------+---------------------------------
1      Po1(SU)         LACP      Gi1/0/1(H)  Gi1/0/2(P)
```

Troubleshooting EtherChannel Bundles

It is important to remember that a port channel is a logical interface, so all the member interfaces must have the same characteristics. If they do not, problems will occur.

As a general rule, when configuring port channels on a switch, place each member interface in the appropriate switch port type (Layer 2 or Layer 3) and then associate the interfaces to a port channel. All other port-channel configuration is done via the port-channel interface.

The following configuration settings must match on the member interfaces:

- **Port type:** Every port in the interface must be consistently configured to be a Layer 2 switch port or a Layer 3 routed port.

- **Port mode:** All Layer 2 port channels must be configured as either access ports or trunk ports. They cannot be mixed.

- **Native VLAN:** The member interfaces on a Layer 2 trunk port channel must be configured with the same native VLAN, using the command **switchport trunk native vlan** *vlan-id*.

- **Allowed VLAN:** The member interfaces on a Layer 2 trunk port channel must be configured to support the same VLANs, using the command **switchport trunk allowed** *vlan-ids*.

- **Speed:** All member interface must be the same speed.

- **Duplex:** The duplex must be the same for all member interfaces.

- **MTU:** All Layer 3 member interfaces must have the same MTU configured. The interface cannot be added to the port channel if the MTU does not match the MTU of the other member interfaces.

- **Load interval:** The load interval must be configured the same on all member interfaces.

- **Storm control:** The member ports must be configured with the same storm control settings on all member interfaces.

In addition to the paying attention to the configuration settings listed above, checked the following when troubleshooting the establishment of an EtherChannel bundle:

- Ensure that a member link is between only two devices.

- Ensure that the member ports are all active.

- Ensure that both end links are statically set to *on* and that either LACP is enabled with at least one side set to *active* or PAgP is enabled with at least one side set to *desirable*.

- Ensure that all member interface ports are consistently configured (except for LACP port priority).

- Verify the LACP or PAgP packet transmission and receipt on both devices.

Key Topic

Load Balancing Traffic with EtherChannel Bundles

Traffic that flows across a port-channel interface is not forwarded out member links on a round-robin basis per packet. Instead, a hash is calculated, and packets are consistently forwarded across a link based on that hash, which runs on the various packet header fields. The load-balancing hash is a systemwide configuration that uses the global command **port-channel load-balance** *hash*. The *hash* option has the following keyword choices:

- **dst-ip:** Destination IP address

- **dst-mac:** Destination MAC address

- **dst-mixed-ip-port:** Destination IP address and destination TCP/UDP port

- **dst-port:** Destination TCP/UDP port

- **src-dst-ip:** Source and destination IP addresses

- **src-dest-ip-only:** Source and destination IP addresses only

- **src-dst-mac:** Source and destination MAC addresses

- **src-dst-mixed-ip-port:** Source and destination IP addresses and source and destination TCP/UDP ports

- **src-dst-port:** Source and destination TCP/UDP ports only

- **src-ip:** Source IP address

- **src-mac:** Source MAC address

- **src-mixed-ip-port:** Source IP address and source TCP/UDP port

- **src-port:** Source TCP/UDP port

If the links are unevenly distributed, changing the hash value may provide a different distribution ratio across member links. For example, if a port channel is established with a router, using a MAC address as part of the hash could impact the traffic flow as the router's MAC address does not change (as the MAC address for the source or destination will always be the router's MAC address). A better choice would be to use the source/destination IP address or base the hash on TCP/UDP session ports.

The command **show etherchannel load-balance** displays how a switch will load balance network traffic based on its type: non-IP, IPv4, or IPv6. Example 5-21 shows the command being executed on SW1.

Example 5-21 Viewing the Port-Channel Hash Algorithm

```
SW1# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-mixed-ip-port

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address and TCP/UDP (layer-4) p
  IPv6: Source XOR Destination IP address and TCP/UDP (layer-4) p
```

Another critical point is that a hash is a binary function, so links should be in powers of two (for example, 2, 4, 8), to be consistent. A three-port EtherChannel will not load balance as effective as a two- or four-port EtherChannel. The best was to view the load of each member link is with the command **show etherchannel port**. The link utilization is displayed in hex under Load and displays the relative link utilization to the other member links of the EtherChannel.

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 5-5 lists these key topics and the page number on which each is found.

Key Topic

Table 5-5 Key Topics for Chapter 5

Key Topic Element	Description	Page
Section	VLAN Trunking Protocol (VTP)	
Paragraph	VTP revision reset	
Paragraph	Dynamic Trunking Protocol (DTP)	
Paragraph	Disabling DTP	
Section	PAgP port modes	
Section	LACP port modes	
Section	EtherChannel configuration	
Section	Minimum number of member links	
Section	Maximum number of member links	
Section	LACP system priority	
Section	LACP interface priority	
Section	Troubleshooting EtherChannel	
Section	Load balancing traffic with EtherChannel bundles	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

Dynamic Trunking Protocol (DTP)

EtherChannel bundle

member links

LACP interface priority

LACP system priority

load-balancing hash

VLAN Trunking Protocol (VTP)

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 5-6 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 5-6 Command Reference

Task	Command Syntax
Configure the VTP version	**vtp version** {**1** | **2** | **3**}
Configure the VTP domain name	**vtp domain** *domain-name*
Configure the VTP mode for a switch	**vtp mode** { **server** | **client** | **transparent** | **none**} (required for the first VTP v3 server) **vtp primary**
Configure a switch port to actively attempt to establish a trunk link	**switchport mode dynamic desirable**
Configure a switch port to respond to remote attempts to establish a trunk link	**switchport mode dynamic auto**
Configure the member ports for a static EtherChannel	**channel-group** *etherchannel-id* **mode on**
Configure the member ports for an LACP EtherChannel	**channel-group** *etherchannel-id* **mode** {**active** | **passive**}
Configure the member ports for a PAgP EtherChannel	**channel-group** *etherchannel-id* **mode** {**auto** | **desirable**} [**non-silent**]
Configure the LACP packet rate	**lacp rate** {**fast** | **slow**}
Configure the minimum number of member links for the LACP EtherChannel to become active	**port-channel min-links** *min-links*
Configure the maximum number of member links in an LACP EtherChannel	**lacp max-bundle** *max-links*
Configure a switch's LACP system priority	**lacp system-priority** *priority*
Configure a switch's LACP port priority	**lacp port-priority** *priority*
Configure the EtherChannel load-balancing hash algorithm	**port-channel load-balance** *hash*
Display the contents of all current access lists	**show access-list** [*access-list-number* | *access-list-name*}
Display the VTP system settings	**show vtp status**
Display the switch port DTP settings, native	**show interface** [*interface-id*] **trunk**

VLANs, and allowed VLANs	
Display a brief summary update on EtherChannel interfaces	**show etherchannel summary**
Display detailed information for the local EtherChannel interfaces and their remote peers	**show interface port-channel**
Display information about LACP neighbors	**show lacp neighbor** [**detail**]
Display the local LACP system identifier and priority	**show lacp** *system-id*
Display the LACP counters for configure interfaces	**show lacp counters**
Display information about PAgP neighbors	**show pagp neighbor**
Display the PAgP counters for configured interfaces	**show pagp counters**
Display the algorithm for load balancing network traffic based on the traffic type	**show etherchannel load-balance**

PREV
Chapter 4. Multiple Spanning Tree P…

NEXT
Part III: Routing

Part III: Routing

Chapter 6. IP Routing Essentials

This chapter covers the following subjects:

- **Routing Protocol Overview:** This section explains how different routing protocols advertise and identify routes.

- **Path Selection:** This section explains the logic a router uses to identify the best route and install it in the routing table.

- **Static Routing:** This section provides a brief overview of fundamental static route concepts.

- **Virtual Routing and Forwarding:** This section explains the creation of logical routers on a physical router.

This chapter revisits the fundamentals from Chapter 1, “Packet Forwarding,” as well as some of the components of the operations of a router. It reinforces the logic of the programming of the Routing Information Base (RIB), reviews differences between common routing protocols, and explains common concepts related to static routes.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 6-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 6-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topic Section	Questions
Routing Protocol Overview	1–5
Path Selection	6–8
Static Routing	9
Virtual Routing and Forwarding	10

1. Which of the following routing protocols is classified as an EGP?

a. RIP

b. EIGRP

c. OSPF

d. IS-IS

e. BGP

2. Which of the following routing protocols are classified as IGPs? (Choose all that apply.)

a. RIP

b. EIGRP

c. OSPF

d. IS-IS

e. BGP

3. A path vector routing protocol finds the best loop-free path by using ______.

a. hop count

b. bandwidth

c. delay

d. interface cost

e. path attributes

4. A distance vector routing protocol finds the best loop-free path by using ______.

a. hop count

b. bandwidth

c. delay

d. interface cost

e. path attributes

5. A link-state routing protocol finds the best loop free path by using ______.

a. hop count

b. bandwidth

c. delay

d. interface cost

e. path attributes

6. A router uses _________ as the first criterion for forwarding packets.

a. path metric

b. administrative distance

c. longest match

d. hop count

7. A router uses __________ as the second criterion for forwarding packets.

a. path metric

b. administrative distance

c. longest match

d. hop count

8. The ability to install multiple paths from the same routing protocol with the same path metric into the RIB is known as _______.

a. per-packet load balancing

b. round-robin load balancing

c. equal-cost multipathing

d. parallel link forwarding

9. Which static route should be used to avoid unintentional forwarding paths with an Ethernet link failure?

a. A directly attached static route

b. A recursive static route

c. A fully specified static route

d. A static null route

10. Virtual routing and forwarding (VRF) is useful with _____ addresses.

a. MAC

b. IPv4

c. IPv6

d. IPv4 and IPv6

Answers to the "Do I Know This Already?" quiz:

1. E

2. A, B, C, D

3. E

4. A

5. E

6. C

7. B

8. C

9. C

10. D

FOUNDATION TOPICS

As described in the previous chapters, a router is necessary to transmit packets between network segments. This chapter explains the process a router uses to insert routes into the routing table from routing protocol databases and the methodology for selecting a path. A brief overview of static routing is provided as well. By the end of this chapter, you should have a solid understanding of the routing processes on a router.

ROUTING PROTOCOL OVERVIEW

A router's primary function is to move an IP packet from one network to a different network. A router learns about nonattached networks through configuration of static routes or through dynamic IP routing protocols.

Dynamic IP routing protocols distribute network topology information between routers and provide updates without intervention when a topology change in the network occurs. Design requirements or hardware limitations may restrict IP routing to static routes, which do not accommodate topology changes very well and can burden network engineers, depending on the size of the network. With dynamic routing protocols, routers try to select the best loop-free path on which to forward a packet to its destination IP address.

A network of interconnected routers and related systems managed under a common network administration is known as an *autonomous system (AS)*, or a *routing domain*. The Internet is composed of thousands of autonomous systems spanning the globe.

The common dynamic routing protocols found on most routing platforms today are as follows:

- Routing Information Protocol Version 2 (RIPv2)
- Enhanced Interior Gateway Routing (EIGRP)
- Open Shortest Path First (OSPF)
- Intermediate System-to-Intermediate System (IS-IS)
- Border Gateway Protocol (BGP)

With the exception of BGP, the protocols in this list are designed and optimized for routing within an autonomous system and are known as Interior Gateway

Protocols (IGPs). Exterior Gateway Protocols (EGPs) route between autonomous systems. BGP is an EGP protocol but can also be used within an autonomous system. If BGP exchanges routes within an autonomous system, it is known as an *interior BGP (iBGP) session*. If it exchanges routes between different autonomous systems, it is known as an *exterior BGP (eBGP) session*.

Figure 6-1 shows an illustration of how one or many IGPs as well as iBGP can be running within an autonomous system and how eBGP sessions interconnect the various autonomous systems together.

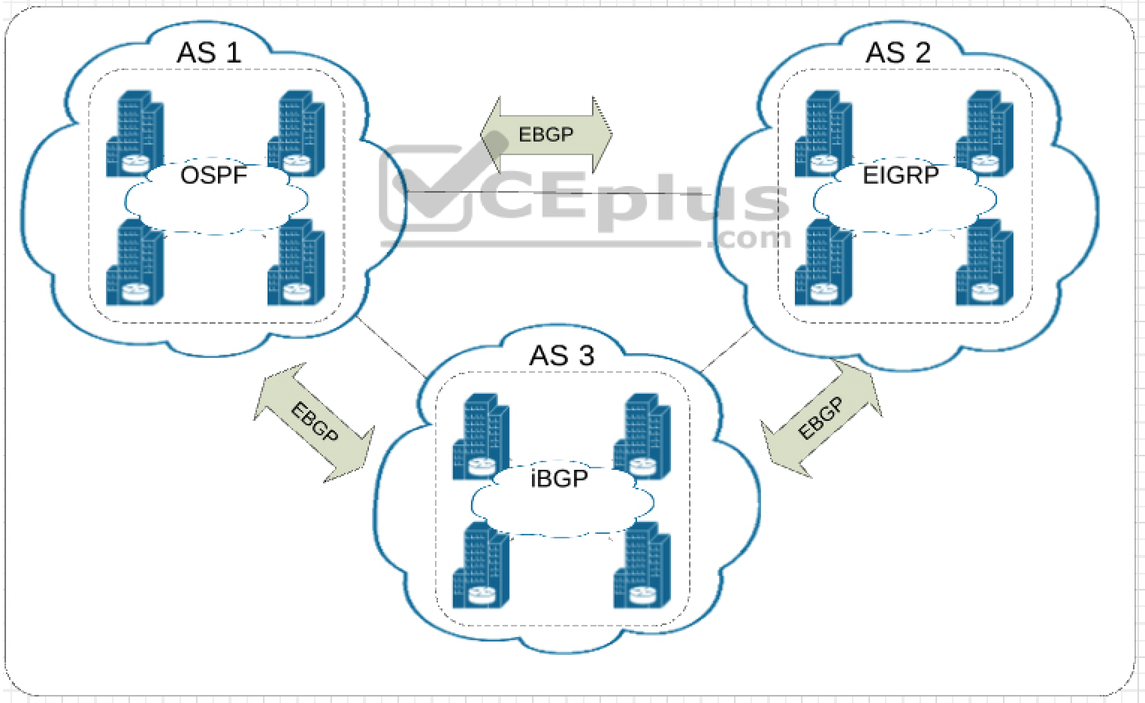


Figure 6-1 BGP Autonomous Systems and How They Interconnect

EGPs and IGPs use different algorithms for path selection and are discussed in the following sections.

Key Topic

Distance Vector Algorithms

Distance vector routing protocols, such as RIP, advertise routes as vectors, where distance is a metric (or cost) such as hop count, and vector is the next-hop router’s IP used to reach the destination:

- **Distance:** The distance is the route metric to reach the network.

- **Vector:** The vector is the interface or direction to reach the network.

When a router receives routing information from a neighbor, it stores it in a local routing database as it is received, and the distance vector algorithm (such as the Bellman-Ford and Ford-Fulkerson algorithms) is used to determine which paths are the best loop-free paths to each reachable destination. When the best paths are determined, they are installed into the routing table and are advertised to each neighbor router.

Key Topic

Routers running distance vector protocols advertise the routing information to their neighbors from their own perspective, modified from the original route received. Therefore, a distance vector protocol does not have a complete map of the whole network; instead, its database reflects that a neighbor router knows how to reach the destination network and how far the neighbor router is from the destination network. The advantage of distance vector protocols is that they require less CPU and memory and can run on low-end routers.

An analogy commonly used to describe distance vector protocols is a road sign at an intersection indicating that the destination is 2 miles to the west; drivers trust and blindly follow this information, without really knowing whether there is a shorter or better way to the destination or whether the sign is even correct. Figure 6-2 illustrates how a router using a distance vector protocol views the network and the direction that R3 needs to go to reach the 192.168.1.0/24 subnet.

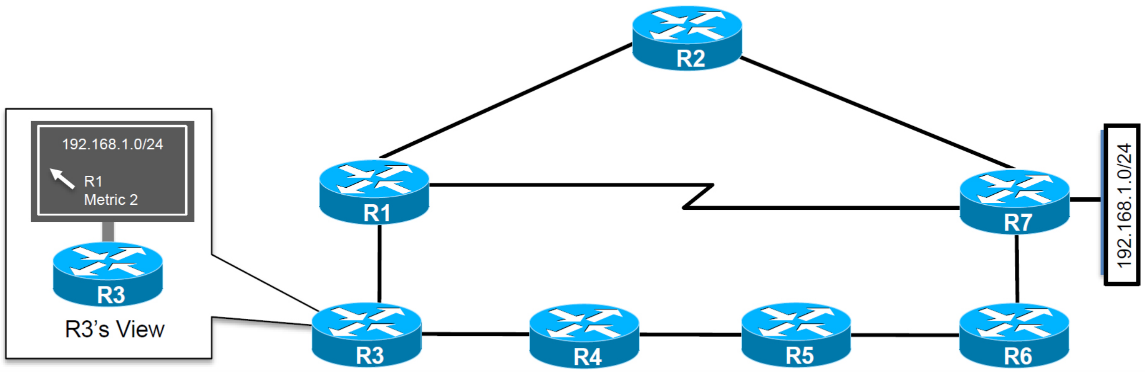


Figure 6-2 Distance Vector Protocol View of a Network

A distance vector protocol selects paths purely based on distance. It does not account for link speeds or other factors. In Figure 6-2, the link between R1 and R7 is a serial link with only 64 kbps of bandwidth, and all of the other links are 1 Gbps Ethernet links. RIP does not take this into consideration and forwards traffic across this link, which will result in packet loss when that link is oversubscribed.

Key Topic

Enhanced Distance Vector Algorithms

The diffusing update algorithm (DUAL) is an enhanced distance vector algorithm that EIGRP uses to calculate the shortest path to a destination within a network. EIGRP advertises network information to its neighbors as other distance vector protocols do, but it has some enhancements, as its name suggests. The following are some of the enhancements introduced into this algorithm compared to other distance vector algorithms:

- It offers rapid convergence time for changes in the network topology.

- It sends updates only when there is a change in the network. It does not send full routing table updates in a periodic fashion, as distance vector protocols.

- It uses hellos and forms neighbor relationships just as link-state protocols do.
- It uses bandwidth, delay, reliability, load, and maximum transmission unit (MTU) size instead of hop count for path calculations.
- It has the option to load balance traffic across equal- or unequal-cost paths.

Key Topic

EIGRP is sometimes referred to as a *hybrid routing protocol* because it has characteristics of both distance vector and link-state protocols, as shown in the preceding list. EIGRP relies on more advanced metrics other than hop count (for example, bandwidth) for its best-path calculations. By default, EIGRP advertises the total path delay and minimum bandwidth for a route. This information is advertised out every direction, as happens with a distance vector routing protocol; however, each router can calculate the best path based on the information provided by its direct neighbors.

Figure 6-3 shows the previous topology but now includes EIGRP's metric calculations for each of the links. R3 is trying to forward packets to the 192.168.1.0/24 network. If the routing domain used a distance vector routing protocol, it would take the R3→R1→R7 path, which is only two hops away, rather than the path R3→R1→R2→R7 path, which is three hops away. But the R3→R1→R7 path cannot support traffic over 64 kbps. While the

R3→R1→R2→R7 path is longer, it provides more bandwidth and does not have as much delay (because of the serialization process on lower-speed interfaces) and is the path selected by EIGRP.

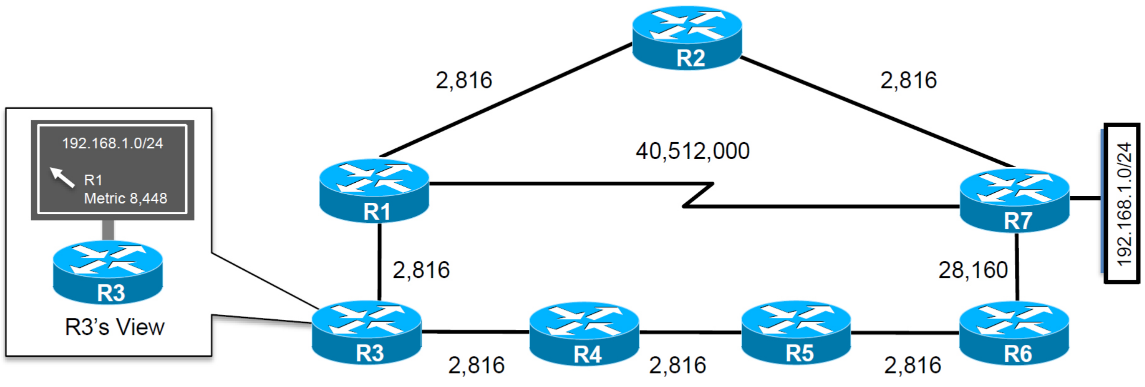


Figure 6-3 Distance Vector Protocol Versus Enhanced Distance Vector

Key Topic

Link-State Algorithms

A link-state dynamic IP routing protocol advertises the link state and link metric for each of its connected links and directly connected routers to every router in the network. OSPF and IS-IS are two link-state routing protocols commonly used in enterprise and service provider networks. OSPF advertisements are called *link-state advertisements (LSAs)*, and IS-IS uses *link-state packets (LSPs)* for its advertisements.

As a router receives an advertisement from a neighbor, it stores the information in a local database called the *link-state database (LSDB)* and advertises the link-state information on to each of its neighbor routers exactly as it was received. The link-state information is essentially flooded throughout the network, unchanged, from router to router, just as the originating router advertised it. This allows all the routers in the network to have a synchronized and identical map of the network.

Using the complete map of the network, every router in the network then runs the Dijkstra shortest path first (SPF) algorithm to calculate the best shortest loop-free paths. The link-state algorithm then populates the routing table with this information.

Due to having the complete map of the network, link-state protocols usually require more CPU and memory than distance vector protocols, but they are less prone to routing loops and make better path decisions. In addition, link-state protocols are equipped with extended capabilities such as opaque LSAs for OSPF and TLVs (type/length/value) for IS-IS that allow them to support features commonly used by service providers, such as MPLS traffic engineering.

An analogy for link-state protocols is a GPS navigation system. The GPS navigation system has a complete map and can make the best decision about which way is the shortest and best path to reach a destination. Figure 6-4 illustrates how R3 would view the network to reach the 192.168.1.0/24 subnet. R1 will use the same algorithm as R3 and take the direct link to R4.

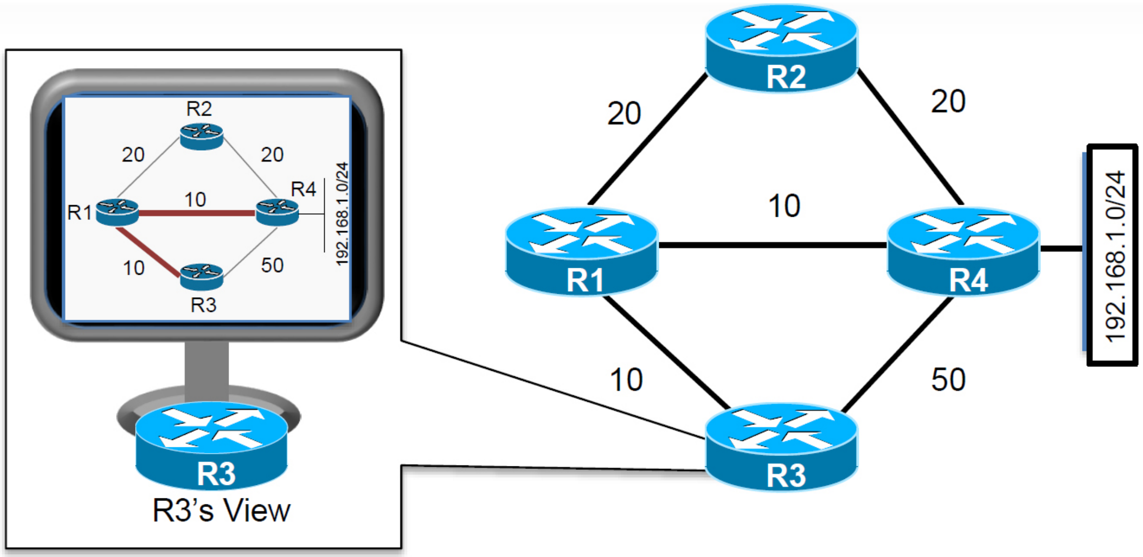


Figure 6-4 Link-State Protocol View of a Network

Key Topic

Path Vector Algorithm

A path vector protocol such as BGP is similar to a distance vector protocol; the difference is that instead of looking at the distance to determine the best loop-free path, it looks at various BGP path attributes. BGP path attributes include autonomous system path (AS_Path), multi-exit discriminator (MED), origin, next hop, local preference, atomic aggregate, and aggregator. BGP path attributes are covered in Chapter 11, “BGP,” and Chapter 12, “Advanced BGP.”

A path vector protocol guarantees loop-free paths by keeping a record of each autonomous system that the routing advertisement traverses. Any time a router receives an advertisement in which it is already part of the AS_Path, the advertisement is rejected because accepting the AS_Path would effectively result in a routing loop.

Figure 6-5 illustrates the loop prevention concept over the following steps:

1. R1 (AS1) advertises the 10.1.1.0/24 network to R2 (AS2). R1 adds the AS 1 to the AS_Path during the network advertisement to R2.

2. R2 advertises the 10.1.1.0/24 network to R4 and adds AS 2 to the AS_Path during the network advertisement to R4.

3. R4 advertises the 10.1.1.0/24 network to R3 and adds AS 4 to the AS_Path during the network advertisement to R3.

4. R3 advertises the 10.1.1.0/24 network back to R1 and R2 after adding AS 3 to the AS_Path during the network advertisement.

5. As R1 receives the 10.1.1.0/24 network advertisement from R3, it discards the route advertisement because R1 detects its AS (AS1) in the AS_Path “3 4 2 1” and considers the advertisement as a loop. R2 discards the 10.1.1.0/24 network advertisement from R3 as it detects its AS (AS 2) in the AS_Path “3 4 2 1” and considers it a loop, too.

Note

The drawing does not depict the advertisement of the 10.1.1.0/24 network toward R3 to make it easier to visualize, but the process happens in the other direction as well. R3 attempts to advertise the 10.1.1.0/24 network to R2 as well. R2 discards the route because R1 detects its AS (AS2) in the AS_Path “3 4 2 1” and considers it a loop as well—even though it did not source the original route.

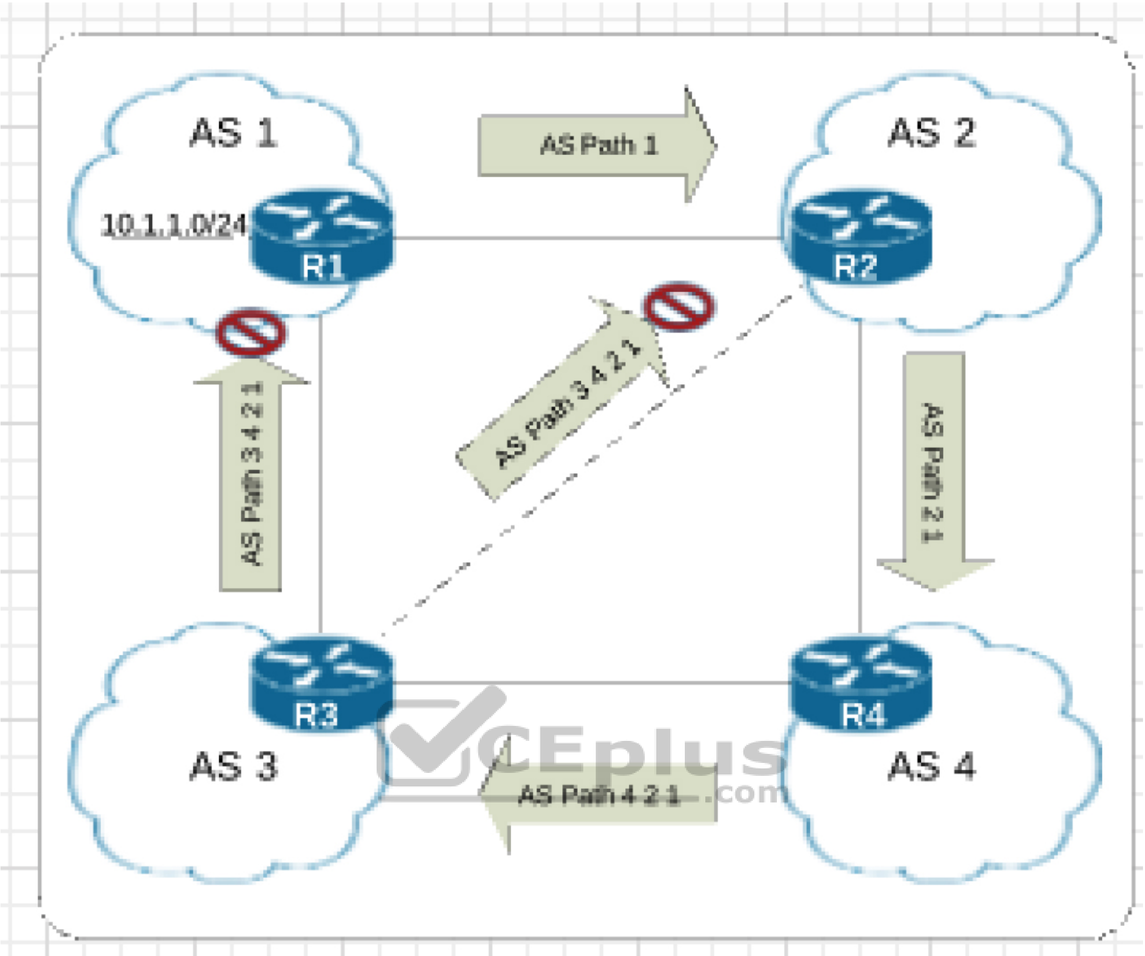


Figure 6-5 Path Vector Loop Avoidance

Key Topic

PATH SELECTION

A router identifies the path a packet should take by evaluating the prefix length that is programmed in the *Forwarding Information Base (FIB)*. The FIB is programmed through the routing table, which is also known as the *Routing Information Base (RIB)*. The RIB is composed of routes presented from the routing protocol processes. Path selection has three main components:

- **Prefix length:** The prefix length represents the number of leading binary bits in the subnet mask that are in the on position.

- **Administrative distance:** Administrative distance (AD) is a rating of the trustworthiness of a routing information source. If a router learns about a route to a destination from more than one routing protocol, and all the routes have the same prefix length, then the AD is compared.

- **Metrics:** A metric is a unit of measure used by a routing protocol in the best-path calculation. The metrics vary from one routing protocol to another.

Prefix Length

Let's look at a scenario in which a router selects a route when the packet destination is within the network range for multiple routes. Assume that a router has the following routes with various prefix lengths in the routing table:

- 10.0.3.0/28

- 10.0.3.0/26

- 10.0.3.0/24

Each of these routes, also known as *prefix routes* or simply *prefixes*, has a different prefix length (subnet mask). The routes are considered to be different destinations, and they will all be installed into the RIB, also known as the routing table. The routing table also includes the outgoing interface and the next-hop IP address (unless the prefix is a connected network). Table 6-2 shows this routing table. The applicable IP address range has been provided to help illustrate the concept.

Table 6-2 Representation of Routing Table

Prefix	IP Address Range	Next Hop	Outgoing Interface
10.0.3.0/28	10.0.3.0–10.0.3.15	10.1.1.1	Gigabit Ethernet 1/1
10.0.3.0/26	10.0.3.0–10.0.3.63	10.2.2.2	Gigabit Ethernet 2/2
10.0.3.0/24	10.0.3.0–10.0.3.255	10.3.3.3	Gigabit Ethernet 3/3

Key Topic

If a packet needs to be forwarded, the route chosen depends on the prefix length, where the longest prefix length is always preferred. For example, /28 is preferred over /26, and /26 is preferred over /24. The following is an example, using Table 6-2 as a reference:

- If a packet needs to be forwarded to 10.0.3.14, the router matches all three routes as it fits into all three IP address ranges. But the packet is forwarded to next hop 10.1.1.1 with the outgoing interface Gigabit Ethernet 1/1 because 10.0.3.0/28 has the longest prefix match.

- If a packet needs to be forwarded to 10.0.3.42, the router matches the 10.0.3.0/24 and 10.0.3.0/26 prefixes. But the packet is forwarded to 10.2.2.2 with the outgoing interface Gigabit Ethernet 2/2 because 10.0.3.0/26 has the longest prefix match.

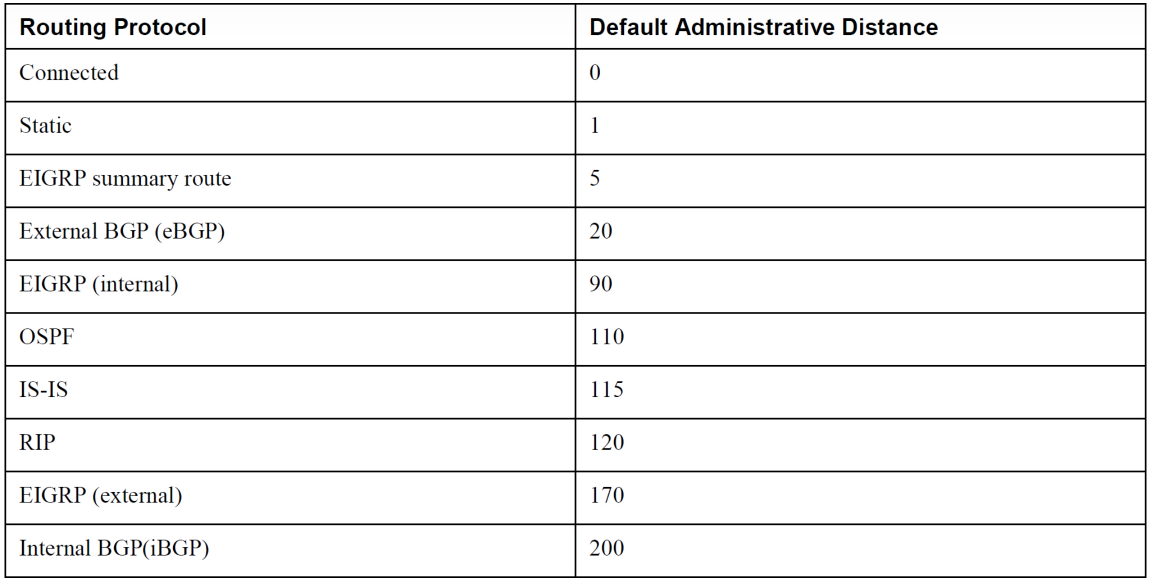
- If a packet needs to be forwarded to 10.0.3.100, the router matches only the 10.0.3.0/24 prefix. The packet is forwarded to 10.3.3.3 with the outgoing interface Gigabit Ethernet 3/3.

The forwarding decision is a function of the FIB and results from the calculations performed in the RIB. The RIB is calculated through the combination of routing protocol metrics and administrative distance.

Administrative Distance

As each routing protocol receives routing updates and other routing information, it chooses the best path to any given destination and attempts to install this path into the routing table. Table 6-3 provides the default ADs for a variety of routing protocols.

Table 6-3 Routing Protocol Default Administrative Distances



Routing Protocol	Default Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP (eBGP)	20
EIGRP (internal)	90
OSPF	110
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP(iBGP)	200

Key Topic

The RIB is programmed from the various routing protocol processes. Every routing protocol presents the same information to the RIB for insertion: the destination network, the next-hop IP address, the AD, and metric values. The RIB accepts or rejects a route based on the following logic:

• If the route does not exist in the RIB, the route is accepted.

• If the route exists in the RIB, the AD must be compared. If the AD of the route already in the RIB is lower than the process submitting the second route, the route is rejected. Then that routing process is notified.

- If the route exists in the RIB, the AD must be compared. If the AD of the route already in the RIB is higher than the routing process submitting the alternate entry, the route is accepted, and the current source protocol is notified of the removal of the entry from the RIB.

Consider another example on this topic. Say that a router has OSPF, IS-IS, and EIGRP running, and all three protocols have learned of the destination 10.3.3.0/24 network with a different best path and metric.

Each of these three protocols attempts to install the route to 10.3.3.0/24 into the routing table. Because the prefix length is the same, the next decision point is the AD, where the routing protocol with the lowest AD installs the route into the routing table.

Because the EIGRP internal route has the best AD, it is the one installed into the routing table, as demonstrated in Table 6-4.

Table 6-4 Route Selection for the RIB

Routing Protocol	AD	Network	Installs in the RIB
EIGRP	90	10.3.3.0/24	✓
OSPF	110	10.3.3.0/24	X
IS-IS	115	10.3.3.0/24	X

The routing protocol or protocols that failed to install their route into the table (in this example, OSPF and IS-IS) hang on to the route and tell the routing table process to report to them if the best path fails so that they can try to reinstall this route.

For example, if the EIGRP route 10.3.3.0/24 installed in the routing table fails for some reason, the routing table process calls OSPF and IS-IS and requests that they reinstall the route in the routing table. Out of these two protocols, the preferred route is chosen based on AD, which would be OSPF because of its lower AD.

Key Topic

Understanding the order of processing from a router is critical because in some scenarios the path with the lowest AD may not always be installed in the RIB. For example, BGP's path selection process could choose an iBGP path over an eBGP path. So BGP would present the path with an AD of 200, not 20, to the RIB, which might not preempt a route learned via OSPF that has an AD of 110. These situations are almost never seen; but remember that it is the best route from the routing protocol presented to the RIB when AD is then compared.

Note

The default AD might not always be suitable for a network; for instance, there might be a requirement to adjust it so that OSPF routes are preferred over EIGRP routes. However, changing the AD on routing protocols can have severe consequences, such as routing loops and other odd behavior, in a network. It is recommended that the AD be changed only with extreme caution and only after what needs to be accomplished has been thoroughly thought out.

Metrics

The logic for selecting the best path for a routing protocol can vary. Most IGPs prefer internally learned routes over external routes and further prioritize the path with the lowest metric.

Key Topic

Equal Cost Multipathing

If a routing protocol identifies multiple paths as a best path and supports multiple path entries, the router installs the maximum number of paths allowed per destination. This is known as *equal-cost multipathing (ECMP)* and provides load sharing across all links. RIP, EIGRP, OSPF, and IS-IS all support ECMP. ECMP provides a mechanism to increase bandwidth across multiple paths by splitting traffic equally across the links.

Figure 6-6 illustrates four routers running OSPF. All four routers belong to the same area and use the same interface metric cost. R1 has two paths with equal cost to reach R3's 10.3.3.0/24 network. R1 installs both routes in the routing table and forwards traffic across the R1–R2–R3 and R1–R4–R3 path to reach the 10.3.3.0/24 network.

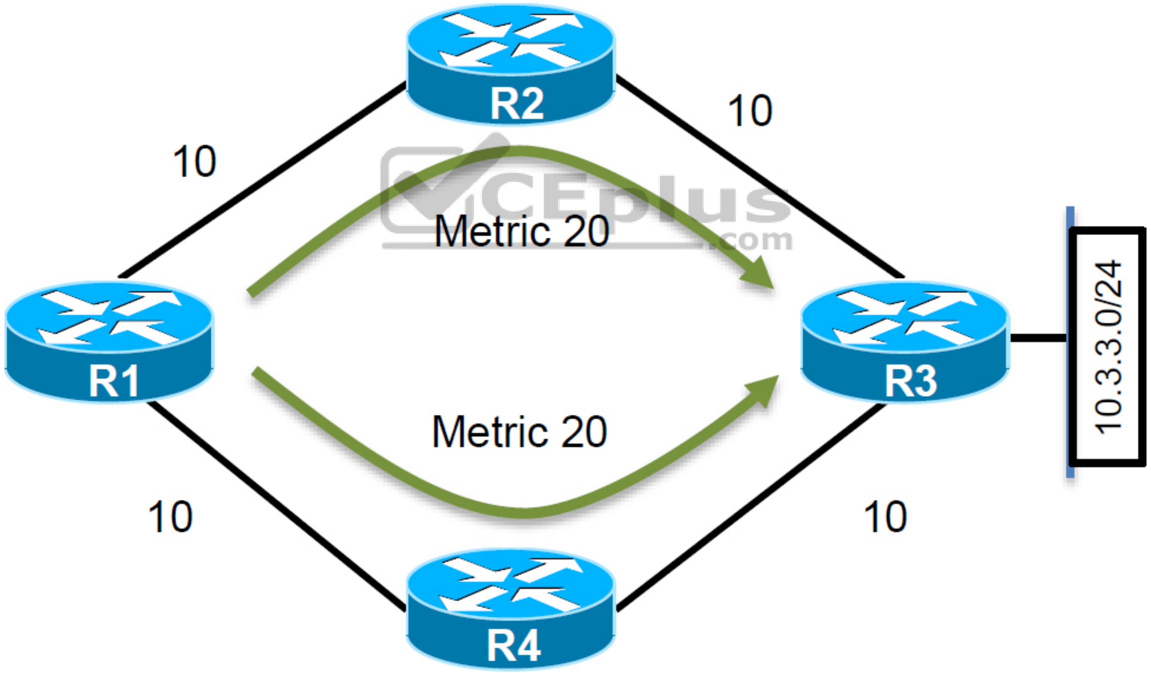


Figure 6-6 OSPF ECMP Technology

The output in Example 6-1 confirms that both paths have been installed into the RIB and, because the metrics are identical, that the router is using ECMP.

Example 6-1 R1’s Routing Table, Showing the ECMP Paths to 10.3.3.0/24

```
R1# show ip route
!  Output omitted for brevity
O     10.3.3.0/24 [110/30] via 10.12.1.2, 00:49:12, GigabitEther
                  [110/30] via 10.14.1.4, 00:49:51, GigabitEther
```

Key Topic

Unequal-Cost Load Balancing

By default, routing protocols install only routes with the lowest path metric. However, EIGRP can be configured (not enabled by default) to install multiple routes with different path metrics. This allows for unequal-cost load balancing across multiple paths. Traffic is transmitted out the router’s interfaces based on that path’s metrics in ratio to other the interface’s metrics.

Figure 6-7 shows a topology with four routers running EIGRP. The delay has been incremented on R1’s Gi0/2 interface from 1 μ to 10 μ. R1 sees the two paths with different metrics. The path from R1 to R3 via R1–R2–R3 has been assigned a path metric of 3328, and the path via R1–R4–R3 has been assigned a path metric of 5632.

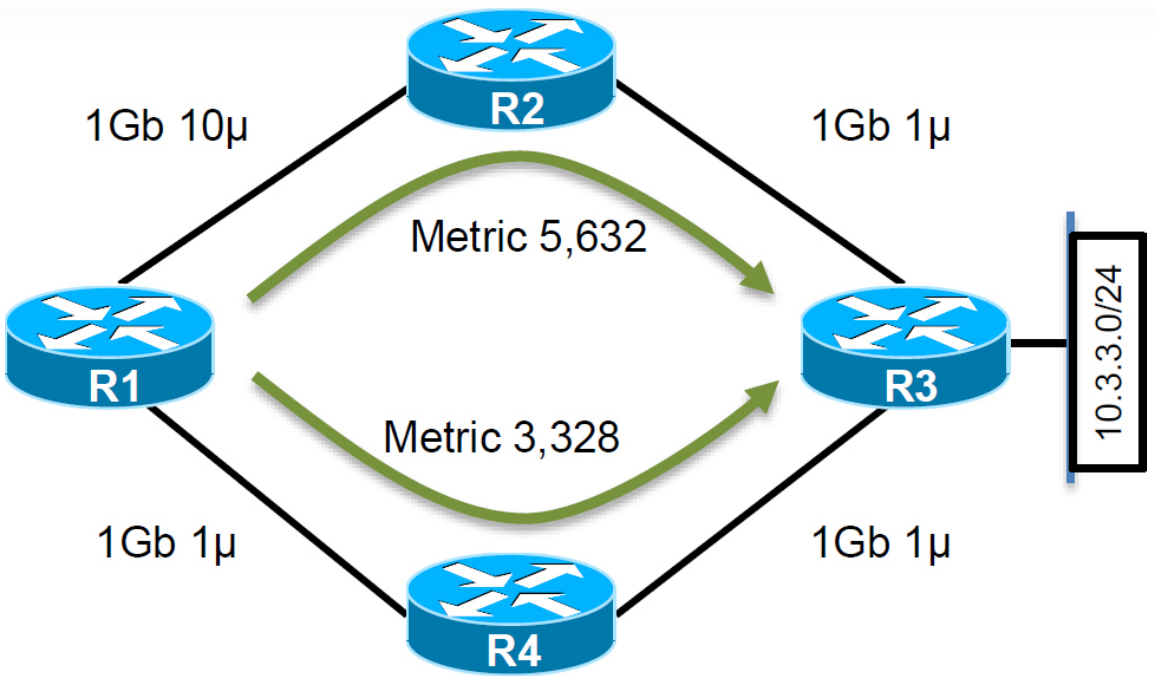


Figure 6-7 EIGRP Unequal-Cost Load Balancing

Example 6-2 shows the routing table of R1. Notice that the metrics are different for each path to the 10.3.3.0/24 network.

Example 6-2 R1's Routing Table, Showing the Unequal-Cost Load Balancing

```
R1# show ip route eigrp
!  Output omitted for brevity
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D        10.3.3.0/24 [90/3328] via 10.14.1.4, 00:00:02, GigabitE
                     [90/5632] via 10.12.1.2, 00:00:02, GigabitE
```

The explicit path must be viewed to see the traffic ratios with unequal-cost load balancing. In Example 6-3, R1 forwards 71 packets toward R2 for every 120 packets that are forwarded toward R4.

Example 6-3 Viewing the Unequal-Cost Load Balancing Ratio

```
R1# show ip route 10.3.3.0
Routing entry for 10.3.3.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
  Redistributing via eigrp 100
  Last update from 10.14.1.4 on GigabitEthernet0/4, 00:00:53 ago
  Routing Descriptor Blocks:
  * 10.14.1.4, from 10.14.1.4, 00:00:53 ago, via GigabitEthernet0
      Route metric is 3328, traffic share count is 120
      Total delay is 30 microseconds, minimum bandwidth is 100000
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
    10.12.1.2, from 10.12.1.2, 00:00:53 ago, via GigabitEthernet0
      Route metric is 5632, traffic share count is 71
      Total delay is 120 microseconds, minimum bandwidth is 100000
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

STATIC ROUTING

Static routes provide precise control over routing but may create an administrative burden as the number of routers and network segments grow.

Using static routing requires zero network bandwidth because implementing manual route entries does not require communication with other routers.

Unfortunately, because the routers are not communicating, there is no network intelligence. If a link goes down, other routers will not be aware that the network path is no longer valid. Static routes are useful when

- Dynamic routing protocols cannot be used on a router because of limited router CPU or memory

- Routes learned from dynamic routing protocols need to be superseded

Static Route Types

Static routes can be classified as one of the following:

- Directly attached static routes

- Recursive static route

- Fully specified static route

Key Topic

Directly Attached Static Routes

Point-to-point (P2P) serial interfaces do not have to worry about maintaining an adjacency table and do not use Address Resolution Protocol (ARP), so static routes can directly reference the outbound interface of a router. A static route that uses only the outbound next-hop interface is known as a *directly attached static route*, and it requires that the outbound interface be in an up state for the route to be installed into the RIB.

Directly attached static routes are configured with the command **ip route** *network subnet-mask next-hop-interface-id*.

Figure 6-8 illustrates R1 connecting to R2 using a serial connection. R1 uses a directly attached static route to the 10.22.22.0/24 network, and R2 uses a directly attached static route to the 10.11.11.0/24 network to allow connectivity between the two remote networks. Static routes are required on both routers so that return traffic will have a path back.

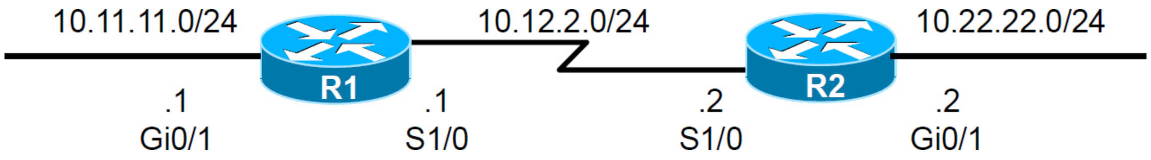


Figure 6-8 R1 and R2 Connected with a Serial Connection

Example 6-4 shows the configuration of R1 and R2 using static routes with serial 1/0 interfaces. R1 indicates that the 10.22.22.0/24 network is reachable via the S1/0 interface, and R2 indicates that the 10.11.11.0/24 network is reachable via the S1/0 interface.

Example 6-4 Configuring Directly Attached Static Routes

```
R1# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ip route 10.22.22.0 255.255.255.0 Serial 1/0

R2# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ip route 10.11.11.0 255.255.255.0 Serial 1/0
```

Example 6-5 shows the routing table with the static route configured. A directly attached static route does not display [AD/Metric] information when looking at the routing table. Notice that the static route displays *directly connected* with the outbound interface.

Example 6-5 R1 and R2 Routing Table

```
R1# show ip route
! Output omitted for brevity
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.11.11.0/24 is directly connected, GigabitEthernet0/1
C        10.12.2.0/24 is directly connected, Serial1/0
S        10.22.22.0/24 is directly connected, Serial1/0

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

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
S        10.11.11.0/24 is directly connected, Serial1/0
```

```
C        10.12.2.0/24 is directly connected, Serial1/0
C        10.22.22.0/24 is directly connected, GigabitEthernet0/1
```

Note

Configuring a directly attached static route to an interface that uses ARP (that is, Ethernet) causes problems and is not recommended. The router must repeat the ARP process for every destination that matches the static route, which consumes CPU and memory. Depending on the size of the prefix of the static route and the number of lookups, the configuration can cause system instability.

Recursive Static Routes

The forwarding engine on Cisco devices needs to know which interface an outbound packet should use. A *recursive static route* specifies the IP address of the next-hop address. The recursive lookup occurs when the router queries the

RIB to locate the route toward the next-hop IP address (connected, static, or dynamic) and then cross-references the adjacency table.

Recursive static routes are configured with the command **ip route** *network subnet-mask next-hop-ip*. Recursive static routes require the route's next-hop address to exist in the routing table to install the static route into the RIB. A recursive static route may not resolve the next-hop forwarding address using the default route (0.0.0.0/0) entry. The static route will fail next-hop reachability requirements and will not be inserted into the RIB.

Figure 6-9 shows a topology with R1 and R2 connected using the Gi0/0 port. R1 uses a recursive static route to the 10.22.22.0/24 network, and R2 uses a recursive static route to the 10.11.11.0/24 network to allow connectivity between these networks.

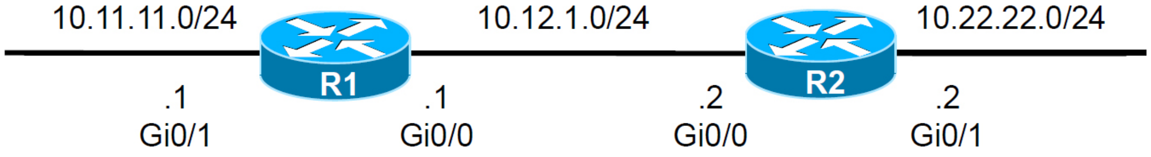


Figure 6-9 R1 and R2 Connected by Ethernet

In Example 6-6, R1's configuration states that the 10.22.22.0/24 network is reachable via the 10.12.1.2 IP address, and R2's configuration states that the 10.11.11.0/24 network is reachable via the 10.12.1.1 IP address.

Example 6-6 Configuring Recursive Static Routes

```
R1# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ip route 10.22.22.0 255.255.255.0 10.12.1.2

R2# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ip route 10.11.11.0 255.255.255.0 10.12.1.1
```

The output in Example 6-7 verifies that the static route was configured on R1 for the 10.22.22.0/24 network with the next-hop IP address 10.12.1.2. Notice that the [AD/Metric] information is present in the output and that the next-hop IP address is displayed.

Example 6-7 IP Routing Table for R1

```
R1# show ip route
! Output omitted for brevity

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.11.11.0/24 is directly connected, GigabitEthernet0/1
C        10.12.1.0/24 is directly connected, GigabitEthernet0/0
S        10.22.22.0/24 [1/0] via 10.12.1.2
```

Cisco supports the configuration of multiple recursive static routes. In Figure 6-10, R1 needs connectivity to the 10.23.1.0/24 network and to the 10.33.1.0/24 network.

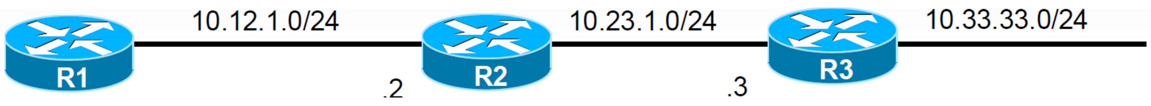


Figure 6-10 Multi-Hop Topology

R1 could configure the static route for the 10.33.33.0/24 network with a next-hop IP address as either 10.12.1.2 or 10.23.1.3. If R1 configured the static route with the 10.23.1.3 next-hop IP address, the router performs a second lookup when building the CEF entry for the 10.33.33.0/24 network.

Key Topic

Fully Specified Static Routes

Static route recursion can simplify topologies if a link fails because it may allow the static route to stay installed while it changes to a different outbound interface in the same direction as the destination. However, problems arise if the recursive lookup resolves to a different interface pointed in the opposite direction.

To correct this issue, the static route configuration should use the outbound interface and the next-hop IP address. A static route with both an interface and a next-hop IP address is known as a *fully specified static route*. If the interface listed is not in an up state, the router removes the static route from the RIB. Specifying the next-hop address along with the physical interface removes the recursive lookup and does not involve the ARP processing problems that occur when using only the outbound interface.

Fully specified static routes are configured the command **ip route** *network subnet-mask interface-id next-hop-ip*.

Revisiting Figure 6-9, R1 and R2 use fully specified static routes to connect to the 10.11.11.0/24 and 10.22.22.0/24 networks using the Gi0/0 interface. The configuration is demonstrated in Example 6-8.

Example 6-8 Configuring Fully Specified Static Routes

```
R1# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ip route 10.22.22.0 255.255.255.0 GigabitEthernet0/0

R2# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ip route 10.11.11.0 255.255.255.0 GigabitEthernet0/0
```

The output in Example 6-9 verifies that R1 can only reach the 10.22.22.0/24 network via 10.12.1.2 from the Gi0/0 interface.

Example 6-9 Verifying the Fully Specified Static Route

```
R1# show ip route
! Output omitted for brevity

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.11.11.0/24 is directly connected, GigabitEthernet0/1
C        10.12.1.0/24 is directly connected, GigabitEthernet0/0
S        10.22.22.0/24 [1/0] via 10.12.1.2, GigabitEthernet0/0
```

Key Topic

Floating Static Routing

The default AD on a static route is 1, but a static route can be configured with an AD value of 1 to 255 for a specific route. The AD is set on a static route by appending the AD as part of the command structure.

Using a floating static route is a common technique for providing backup connectivity for prefixes learned via dynamic routing protocols. A floating static route is configured with an AD higher than that of the primary route. Because the AD is higher than that of the primary route, it is installed in the RIB only when the primary route is withdrawn.

In Figure 6-11, R1 and R2 are configured with two links. The 10.12.1.0/24 transit network is preferred to the 10.12.2.0/24 network.

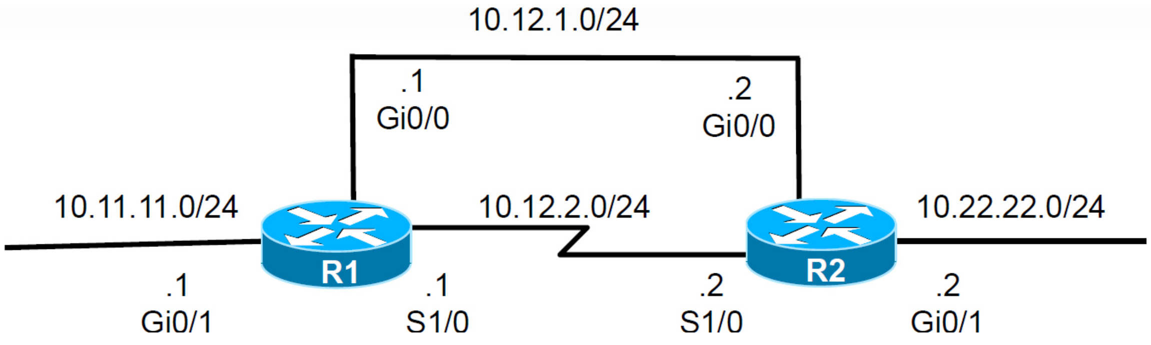


Figure 6-11 Floating Static Route Topology

Example 6-10 shows the configuration of the floating static route on R1, and R2 would be configured similarly. The static route using the Ethernet link (10.12.1.0/24) has an AD of 10, and the serial link (10.12.2.0/24) has an AD set to 210.

Example 6-10 Configuring the Floating Static Route for R1

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ip route 10.22.22.0 255.255.255.0 10.12.1.2 10
R1(config)# ip route 10.22.22.0 255.255.255.0 Serial 1/0 210
```

Example 6-11 shows the routing tables of R1. Notice that the static route across the serial link is not installed into the RIB. Only the static route for the Ethernet link (10.13.1.0/24) with an AD of 10 is installed into the RIB.

Example 6-11 Routing Table of R1 with a Floating Static Route

```
R1# show ip route
! Output omitted for brevity

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.11.11.0/24 is directly connected, GigabitEthernet0/1
C        10.12.1.0/24 is directly connected, GigabitEthernet0/0
C        10.12.2.0/24 is directly connected, Serial1/0
S        10.22.22.0/24 [10/0] via 10.12.1.2
```

Example 6-12 shows the routing table for R1 after shutting down the Gi0/0 Ethernet link to simulate a link failure. The 10.12.1.0/24 network (R1’s Gi0/0) is removed from the RIB. The floating static route through the 10.12.2.0/24 network (R1’s S1/0) is now the best path and is installed into the RIB. Notice that the AD is not shown for that static route.

Example 6-12 Routing Table After Ethernet Link Failure

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface GigabitEthernet0/0
R1(config-if)# shutdown

R1# show ip route
! Output omitted for brevity

Gateway of last resort is not set
```

```
      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.11.11.0/24 is directly connected, GigabitEthernet0/1
C        10.12.2.0/24 is directly connected, Serial1/0
S        10.22.22.0/24 is directly connected, Serial1/0
```

Even though the static route’s AD is not shown, it is still programmed in the RIB. Example 6-13 shows the explicit network entry. The output confirms that the floating static route with AD 210 is currently active in the routing table.

Example 6-13 Verifying the AD for the Floating Static Route

```
R1# show ip route 10.22.22.0
Routing entry for 10.22.22.0/24
  Known via "static", distance 210, metric 0 (connected)
  Routing Descriptor Blocks:
  * directly connected, via Serial1/0
      Route metric is 0, traffic share count is 1
```

Key Topic

Static Null Routes

The null interface is a virtual interface that is always in an up state. Null interfaces do not forward or receive network traffic and drop all traffic destined toward them without adding overhead to a router's CPU.

Configuring a static route to a null interface provides a method of dropping network traffic without requiring the configuration of an access list. Creating a static route to the Null0 interface is a common technique to prevent routing loops. The static route to the Null0 interface uses a summarized network range, and routes that are more specific point toward the actual destination.

Figure 6-12 shows a common topology in which company ABC has acquired the 172.16.0.0/20 network range from its service provider. ABC uses only a portion of the given addresses but keeps the large network block in anticipation of future growth.

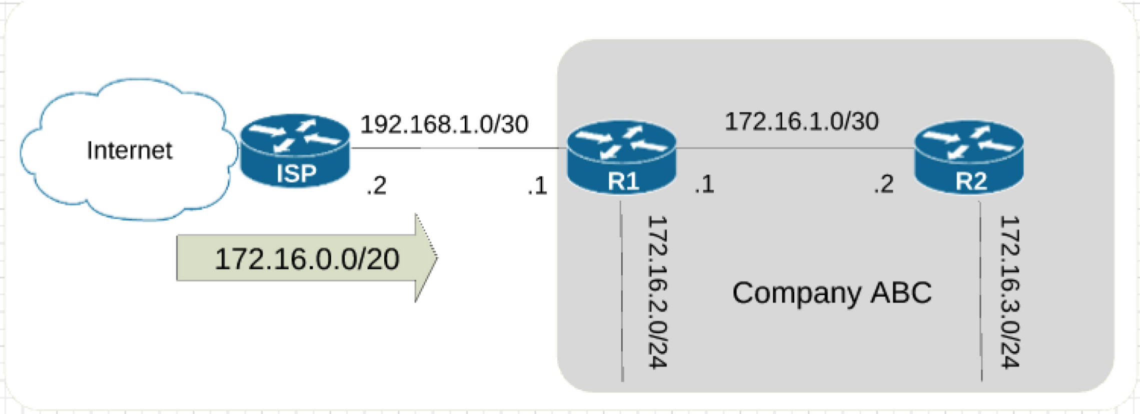


Figure 6-12 Routing Loop Topology

The service provider places a static route for the 172.16.0.0/20 network to R1's interface (192.168.1.1). R1 uses a static default route pointed toward the service provider (192.168.1.2) and a static route to the 172.16.3.0/24 network via R2 (172.16.1.2). Because R2 accesses all other networks through R1, a static default route points toward R1's interface (172.16.1.1).

If packets are sent to any address in the 172.16.0.0/20 range that is not used by company ABC, the packet gets stuck in a loop between R1 and the ISP, consuming additional bandwidth until the packet's TTL expires.

For example, a computer on the Internet sends a packet to 172.16.5.5, and the 172.16.5.0/24 network is not allocated on R1 or R2. The ISP sends the packet to R1 because of the 172.16.0.0/20 static route; R1 looks into the RIB, and the longest match for that prefix is the default route back to the ISP, so R1 sends the packet back to the ISP, creating the routing loop.

Example 6-14 shows the routing loop when packets originate from R2. Notice the IP address in the traceroute alternative between the ISP router (192.168.1.2) and R1 (192.168.1.1).

Example 6-14 Packet Traces Demonstrating the Routing Loop

```
R2# trace 172.16.5.5 source GigabitEthernet 0/2

Type escape sequence to abort.
Tracing the route to 172.16.5.5

  1 172.16.1.1 0 msec 0 msec 0 msec
  2 192.168.1.1 0 msec 0 msec 0 msec
  3 192.168.1.2 0 msec 4 msec 0 msec
  4 192.168.1.1 0 msec 0 msec 0 msec
  5 192.168.1.2 0 msec 0 msec 0 msec
! Output omitted for brevity
```

To prevent the routing loop, a static route is added for 172.16.0.0/20, pointed to the Null0 interface on R1. Any packets matching the 172.16.0.0/20 network range that do not have a longer match in R1’s RIB are dropped. Example 6-15 shows the static route configuration for R1 with the newly added null static route.

Example 6-15 R1 Static Route for 172.16.0.0/20 to Null0

```
R1
ip route 0.0.0.0 0.0.0.0 Gi0/0 192.168.1.2
ip route 172.16.3.0 255.255.255.0 Gi0/2 172.16.1.2
ip route 172.16.0.0 255.255.240.0 Null0
```

The output in Example 6-16 confirms that the null static route has removed the routing loop as intended.

Example 6-16 Packet Traces Demonstrating Loop Prevention

```
R2# trace 172.16.5.5 source GigabitEthernet 0/2
Type escape sequence to abort.
Tracing the route to 172.16.5.5

  1 172.16.1.1  *  *  *
  2 172.16.1.1  *  *  *
! Output omitted for brevity
```

Key Topic

IPv6 Static Routes

The static routing principles for IPv4 routes are exactly the same for IPv6. It is important to ensure that IPv6 routing is enabled by using the configuration command **ipv6 unicast routing**. IPv6 static routes are configured with the command **ipv6 route** *network/prefix-length* { *next-hop-interface-id* | [*next-hop-interface-id*] *next-ip-address*}.

Figure 6-13 shows R1 and R2 with IPv6 addressing to demonstrate static routing.

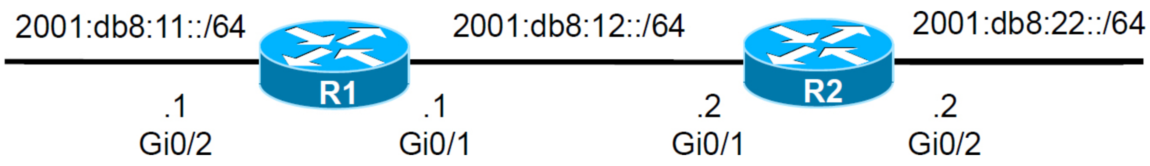


Figure 6-13 IPv6 Static Route Topology

R1 needs a static route to R2's 2001:db8:22::/64 network, and R2 needs a static route to R1's 2001:d8:11::/64 network. Example 6-17 demonstrates the IPv6 static route configuration for R1 and R2.

Example 6-17 Configuring the IPv6 Static Route

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 route 2001:db8:22::/64 2001:db8:12::2

R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ipv6 unicast-routing
R2(config)# ipv6 route 2001:db8:11::/64 2001:db8:12::1
```

Note

If the next-hop address is an IPv6 link-local address, the static route must be a fully specified static route.

The IPv6 routing table is displayed with the command **show ipv6 route**, as demonstrated in Example 6-18. The format is almost identical to that of the IPv4 routing table.

Example 6-18 Packet Traces Demonstrating the Routing Loop

```
R1# show ipv6 route
! Output omitted for brevity
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
```

```
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NE
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr -
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA e
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn
       lA - LISP away, a - Application
C   2001:DB8:11::/64 [0/0]
     via GigabitEthernet0/2, directly connected
C   2001:DB8:12::/64 [0/0]
     via GigabitEthernet0/1, directly connected
S   2001:DB8:22::/64 [1/0]
     via 2001:DB8:12::2
```

Connectivity can be verified with the **traceroute** or **ping** command. Example 6-19 shows R1 pinging R2's 2001:db8:22::2 interface IP address.

Example 6-19 Verifying IPv6 Routing

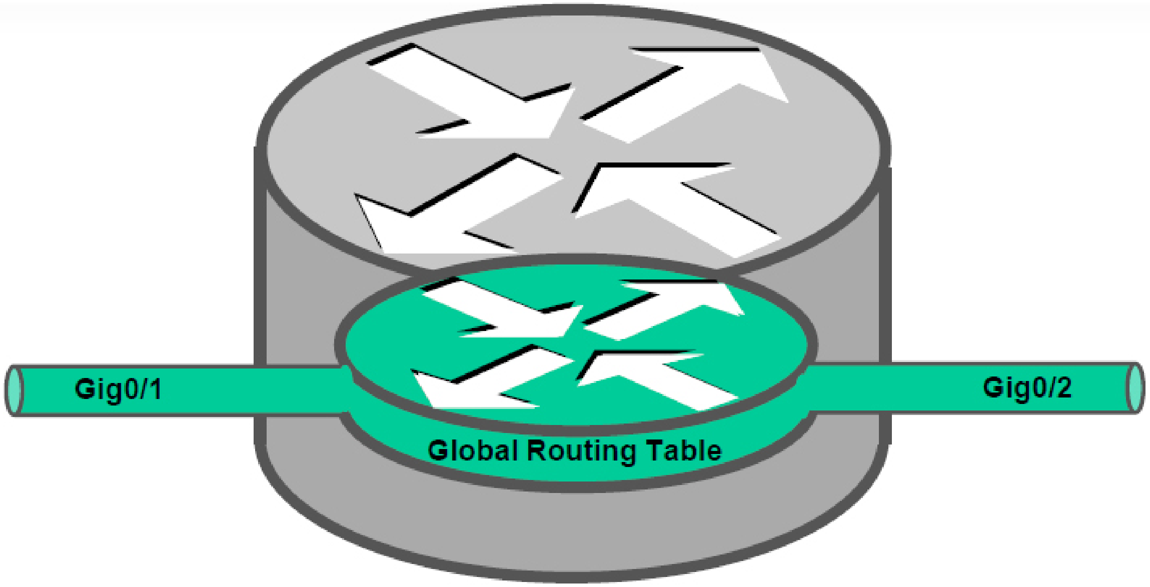
```
R1# ping 2001:db8:22::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:22::2, timeout is 2 se
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4
```

VIRTUAL ROUTING AND FORWARDING

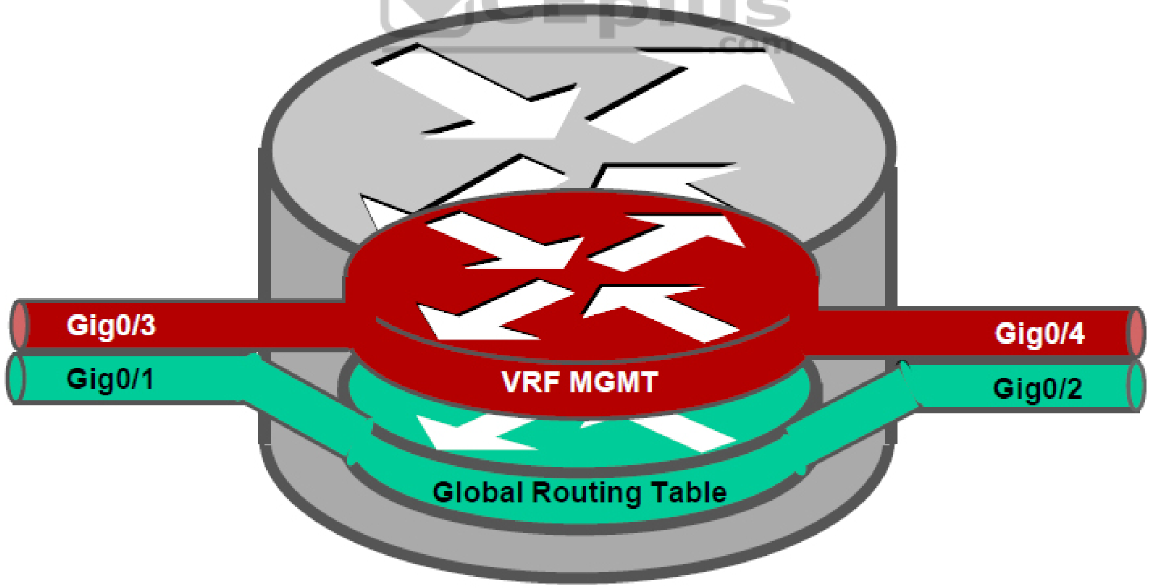
Virtual routing and forwarding (VRF) is a technology that creates separate virtual routers on a physical router. Router interfaces, routing tables, and forwarding tables are completely isolated between VRFs, preventing traffic from one VRF from forwarding into another VRF. All router interfaces belong to the global VRF until they are specifically assigned to a user-defined VRF. The global VRF is identical to the regular routing table of non-VRF routers.

Every router's VRF maintains a separate routing table; it is possible to allow for overlapping IP address ranges. VRF creates segmentation between network interfaces, network subinterfaces, IP addresses, and routing tables. Configuring VRF on a router ensures that the paths are isolated, network security is increased, and encrypting traffic on the network is not needed to maintain privacy between VRF instances.

Figure 6-14 shows two routers to help visualize the VRF routing table concept. One of the routers has no VRFs configured, and the other one has a management VRF instance named MGMT. This figure can be used as a reference for the following examples.



Without VRF Configuration



With VRF Configuration

Figure 6-14 Comparison of a Router with no VRF Instances and a Router with a VRF Instance

The creation of multiprotocol VRF instances requires the global configuration command **vrf definition** *vrf-name*. Under the VRF definition submode, the command **address-family** {**ipv4** | **ipv6**} is required to specify the appropriate address family. The VRF instance is then associated to the interface with the command **vrf forwarding** *vrf-name* under the interface configuration submode.

The following steps are required to create a VRF and assign it to an interface:

Step 1. Create a multiprotocol VRF routing table by using the command **vrf definition** *vrf-name*.

Step 2. Initialize the appropriate address family by using the command **address-family** {**ipv4** | **ipv6**}. The address family can be IPv4, IPv6, or both.

Step 3. Enter interface configuration submode and specify the interface to be associated with the VRF instance by using the command **interface** *interface-id*.

Step 4. Associate the VRF instance to the interface or subinterface by entering the command **vrf forwarding vrf-name** under interface configuration submode.

Step 5. Configure an IP address (IPv4, IPv6, or both) on the interface or subinterface by entering either or both of the following commands:

```
IPv4
ip address ip-address subnet-mask [secondary]
```

IPv6
ipv6 address *ipv6-address/prefix-length*

Table 6-5 provides a set of interfaces and IP addresses that overlap between the global routing table and the VRF instance. This information is used in the following examples.

Table 6-5 Sample Interfaces and IP Addresses

Interface	IP Address	VRF	Global
Gigabit Ethernet 0/1	10.0.3.1/24	—	✓
Gigabit Ethernet 0/2	10.0.4.1/24	—	✓
Gigabit Ethernet 0/3	10.0.3.1/24	MGMT	—
Gigabit Ethernet 0/4	10.0.4.1/24	MGMT	—

Example 6-20 shows how the IP addresses are assigned to the interfaces in the global routing table, along with the creation of the VRF instance named MGMT and two interfaces associated with it (refer to Table 6-5). The IP addresses in the MGMT VRF instance overlap with the ones configured in the global table, but there is no conflict because they are in a different routing table.

Example 6-20 IP Address Configuration in the Global Routing Table

```
R1(config)# interface GigabitEthernet0/1
R1(config-if)# ip address 10.0.3.1 255.255.255.0
R1(config)# interface GigabitEthernet0/2
```

```
R1(config-if)# ip address 10.0.4.1 255.255.255.0
R1(config)# vrf definition MGMT
R1(config-vrf)# address-family ipv4
R1(config)# interface GigabitEthernet0/3
R1(config-if)# vrf forwarding MGMT
R1(config-if)# ip address 10.0.3.1 255.255.255.0
R1(config)# interface GigabitEthernet0/4
R1(config-if)# vrf forwarding MGMT
R1(config-if)# ip address 10.0.4.1 255.255.255.0
```

Example 6-21 shows the global routing table with the command **show ip route** to highlight the IP addresses configured in Example 6-20. Notice that the interfaces in the global table do not appear with this command.

Example 6-21 Output of the Global Routing Table

```
R1# show ip route
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/1
L        10.0.3.1/32 is directly connected, GigabitEthernet0/1
C        10.0.4.0/24 is directly connected, GigabitEthernet0/2
L        10.0.4.1/32 is directly connected, GigabitEthernet0/2
```

Example 6-22 shows how the VRF IP addresses and routes configured in Example 6-20 are displayed with the command **show ip route vrf** *vrf-name*.

Example 6-22 Output of the VRF Routing Table

```
R1# show ip route vrf MGMT
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/3
L        10.0.3.1/32 is directly connected, GigabitEthernet0/3
C        10.0.4.0/24 is directly connected, GigabitEthernet0/4
L        10.0.4.1/32 is directly connected, GigabitEthernet0/4
```

VRF instances on a router can be compared to that of virtual local area networks (VLANs) on a switch. However, instead of relying on Layer 2 technologies such as spanning tree, VRF instances allow for interaction and segmenation with Layer 3 dynamic routing protocols. Using routing protocols over Layer 2 technologies has some advantages, such as improved network convergence times, dynamic traffic load sharing, and troubleshooting tools such as **ping** and **traceroute**.

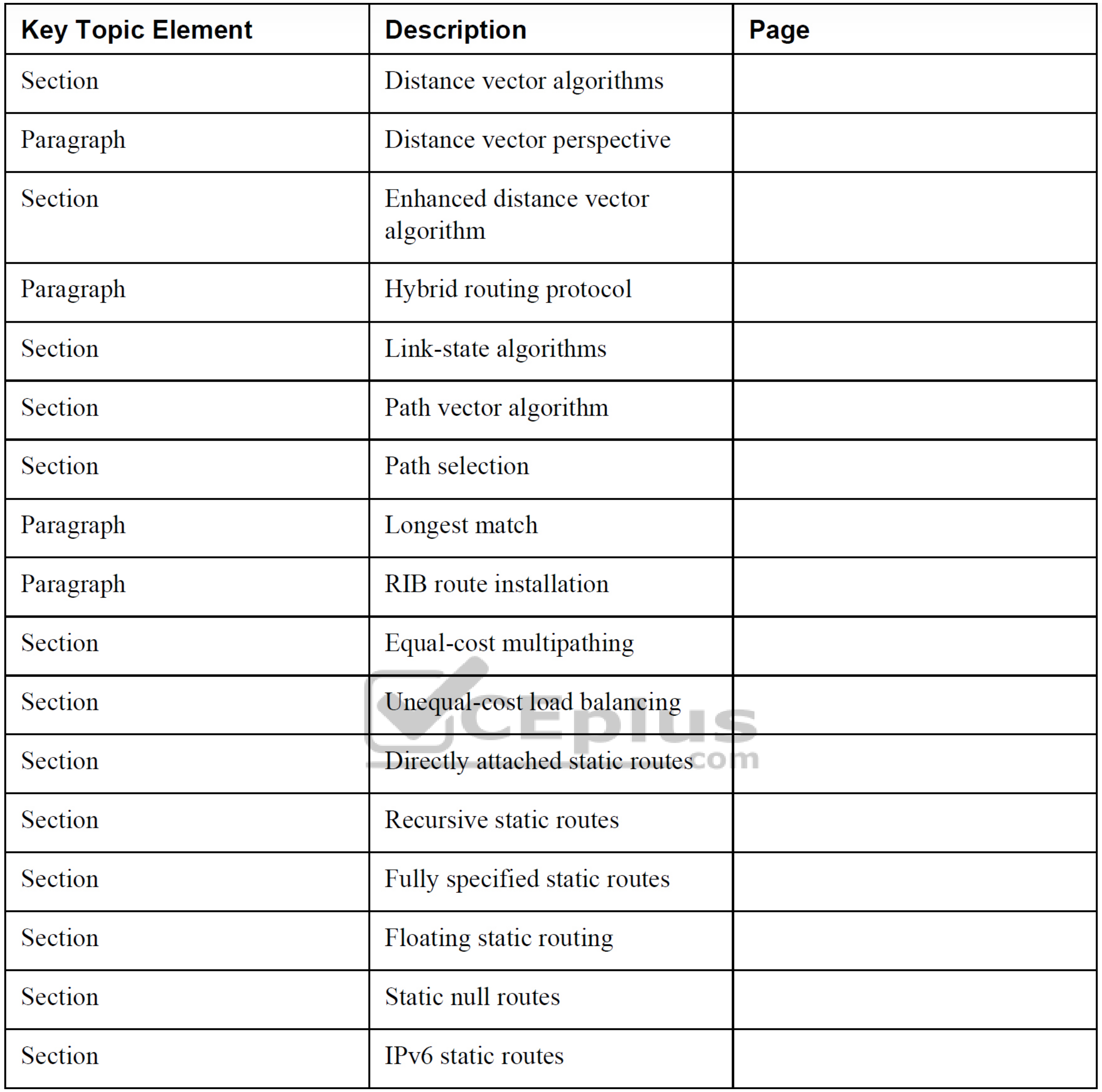
EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 6-6 lists these key topics and the page number on which each is found.

Table 6-6 Key Topics for Chapter 6



Key Topic Element	Description	Page
Section	Distance vector algorithms	
Paragraph	Distance vector perspective	
Section	Enhanced distance vector algorithm	
Paragraph	Hybrid routing protocol	
Section	Link-state algorithms	
Section	Path vector algorithm	
Section	Path selection	
Paragraph	Longest match	
Paragraph	RIB route installation	
Section	Equal-cost multipathing	
Section	Unequal-cost load balancing	
Section	Directly attached static routes	
Section	Recursive static routes	
Section	Fully specified static routes	
Section	Floating static routing	
Section	Static null routes	
Section	IPv6 static routes	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

administrative distance

directly attached static route

distance vector routing protocol

enhanced distance vector routing protocol

equal-cost multipathing

floating static route

fully specified static route

link-state routing protocol

path vector routing protocol

prefix length

recursive static route

static null route

unequal-cost load balancing

Chapter 7. EIGRP

This chapter covers the following subjects:

- **EIGRP Fundamentals:** This section explains how EIGRP establishes a neighbor adjacency with other routers and how routes are exchanged with other routers.

- **Path Metric Calculation:** This section explains how EIGRP calculates the path metric to identify the best and alternate loop-free paths.

- **Failure Detection and Timers:** This section explains how EIGRP detects the absence of a neighbor and the convergence process.

- **Route Summarization:** This section explains the logic and configuration related to summarizing routes on a router.

Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced distance vector routing protocol commonly used in enterprises networks. Initially, it was a

Cisco proprietary protocol, but it was released to the Internet Engineering Task Force (IETF) through RFC 7868, which was ratified in May 2016.

This chapter explains the underlying mechanics of the EIGRP routing protocol, the path metric calculations, and the failure detection mechanisms and techniques to optimize the operations of the routing protocol.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 7-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

Table 7-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundation Topic Section	Questions
EIGRP Fundamentals	1–5
Path Metric Calculation	6–7
Failure Detection and Timers	8–10
Summarization	11

1. EIGRP uses the protocol number _______ to identify its packets.

a. 87

b. 88

c. 89

d. 90

2. EIGRP uses ________ packet types for inter-router communication.

a. three

b. four

c. five

d. six

e. seven

3. What is an EIGRP successor?

a. The next-hop router for the path with the lowest path metric for a destination prefix

b. The path with the lowest metric for a destination prefix

c. The router selected to maintain the EIGRP adjacencies for a broadcast network

d. A route that satisfies the feasibility condition where the reported distance is less than the feasible distance

4. Which of the following attributes does the EIGRP topology table contain? (Choose all that apply.)

a. destination network prefix

b. hop count

c. total path delay

d. maximum path bandwidth

e. list of EIGRP neighbors

5. Which of the following destination addresses does EIGRP use when feasible? (Choose two.)

a. IP address 224.0.0.9

b. IP address 224.0.0.10

c. IP address 224.0.0.8

d. MAC address 01:00:5E:00:00:0A

e. MAC address 0C:15:C0:00:00:01

6. Which value can be modified on a router to manipulate the path taken by EIGRP but avoid having impacts on other routing protocols, such as OSPF?

a. interface bandwidth

b. interface MTU

c. interface delay

d. interface priority

7. EIGRP uses a reference bandwidth of ________ with the default metrics.

a. 100 Mbps

b. 1 Gbps

c. 10 Gbps

d. 40 Gbps

8. The default EIGRP hello timer for a high-speed interfaces is ______.

a. 1 second

b. 5 seconds

c. 10 seconds

d. 20 seconds

e. 30 seconds

f. 60 seconds

9. When a path has been identified using EIGRP and in a stable fashion, the route is considered ______.

a. passive

b. dead

c. active

d. alive

10. How does an EIGRP router indicate that a path computation is required for a specific route?

a. EIGRP sends out an EIGRP update packet with the topology change notification flag set.

b. EIGRP sends out an EIGRP update packet with a metric value of zero.

c. EIGRP sends out an EIGRP query with the delay set to infinity.

d. EIGRP sends a route withdrawal, notifying other neighbors to remove the route from the topology table.

11. True or false: EIGRP summarization occurs for network prefixes as it crosses all network interfaces.

a. True

b. False

Answers to the “Do I Know This Already?” quiz:

1. B

2. C

3. A

4. A, B, C, E

5. B, D

6. C

7. C

8. B

9. A

10. C

11. B

FOUNDATION TOPICS

EIGRP FUNDAMENTALS

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

Autonomous Systems

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

In Figure 7-1, EIGRP autonomous system (AS) 100 consists of R1, R2, R3, and R4, and EIGRP AS 200 consists of R3, R5, and R6. Each EIGRP process correlates to a specific autonomous system and maintains an independent EIGRP topology table. R1 does not have knowledge of routes from AS 200 because it is different from its own autonomous system, AS 100. R3 is able to participate in both autonomous systems and by default does not transfer routes learned from one autonomous system into a different autonomous system.

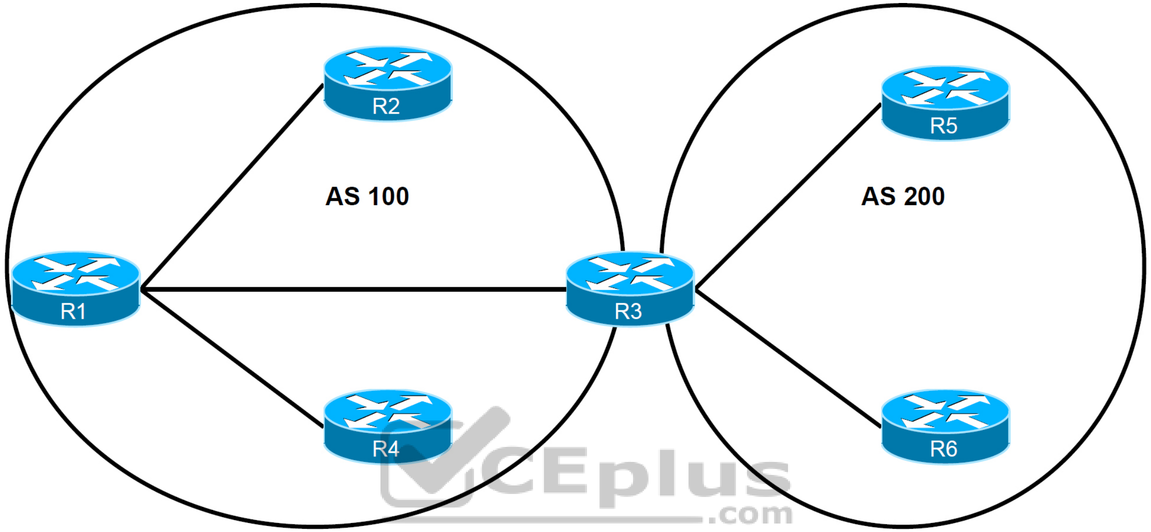


Figure 7-1 EIGRP Autonomous Systems

EIGRP Terminology

This section explains some of the core concepts of EIGRP and the path selection process in EIGRP. Figure 7-2 is the reference topology for this section; it shows R1 calculating the best path and alternative loop-free paths to the 10.4.4.0/24 network. Each value in parentheses represents a particular link's calculated metric for a segment, based on bandwidth and delay.

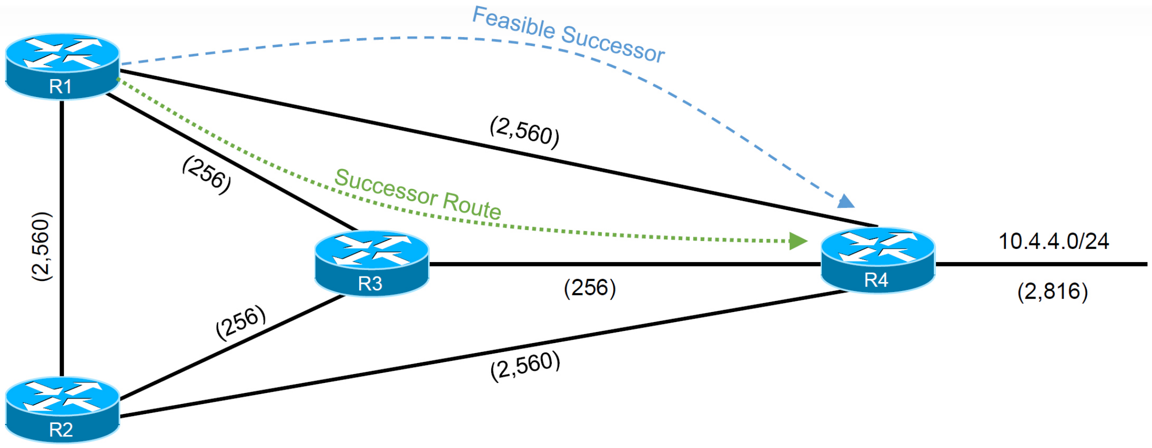
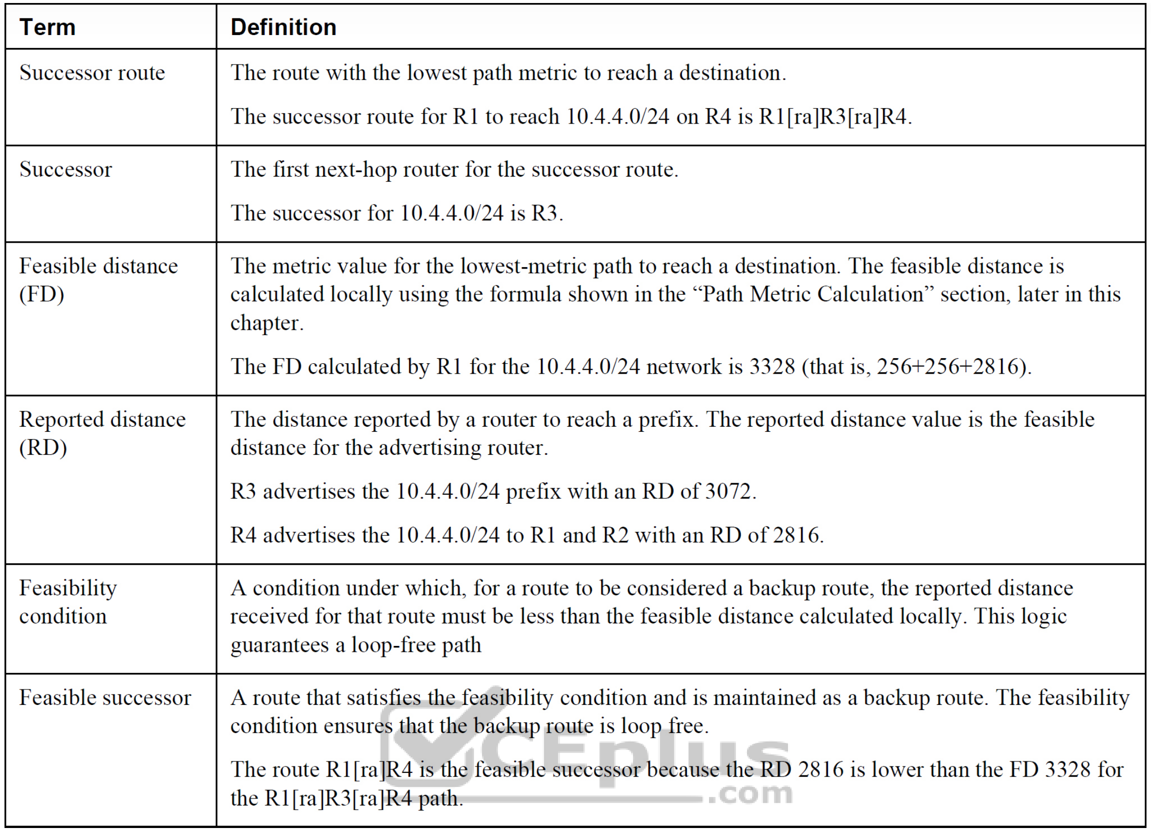


Figure 7-2 EIGRP Reference Topology

Table 7-2 lists some key terms, definitions, and their correlation to Figure 7-2.

Key Topic

Table 7-2 EIGRP Terminology



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

Key Topic

Topology Table

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

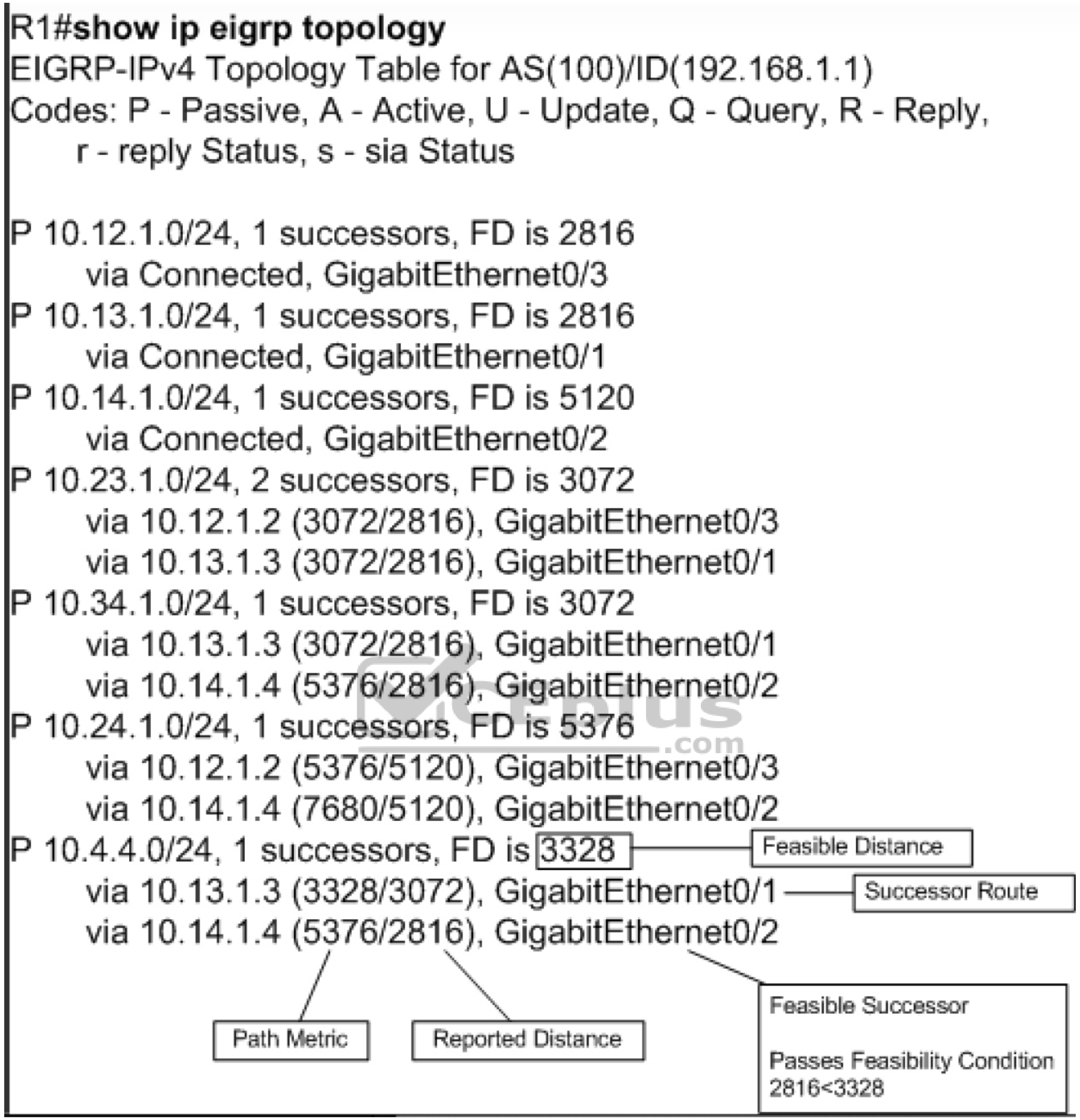
- Network prefix

- EIGRP neighbors that have advertised that prefix

- Metrics from each neighbor (for example, reported distance, hop count)

- Values used for calculating the metric (for example, load, reliability, total delay, minimum bandwidth)

Figure 7-3 shows the topology table for R1 in Figure 7-1. This section focuses on the 10.4.4.0/24 network in explaining the topology table.



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

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

Figure 7-3 EIGRP Topology Output

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

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

EIGRP Neighbors

EIGRP neighbors exchange the entire routing table when forming an adjacency, and they advertise only incremental updates as topology changes occur within a network. The neighbor adjacency table is vital for tracking neighbor status and the updates sent to each neighbor.

EIGRP uses five different packet types to communicate with other routers, as shown in Table 7-3. EIGRP uses its own IP number (88); it uses multicast packets where possible and unicast packets when necessary. Communication between routers is done with multicast, using the group address 224.0.0.10 when possible.

Key Topic

Table 7-3 EIGRP Packet Types

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

PATH METRIC CALCULATION

Metric calculation is a critical component for any routing protocol. EIGRP uses multiple factors to calculate the metric for a path. Metric calculation uses *bandwidth* and *delay* by default, but it can include interface load and reliability, too. The formula shown in Figure 7-4 illustrates the EIGRP classic metric formula.

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

Figure 7-4 EIGRP Classic Metric Formula

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

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

The EIGRP formula is based on the IGRP metric formula, except the output is multiplied by 256 to change the metric from 24 bits to 32 bits. Taking these definitions into consideration, the formula for EIGRP is shown in Figure 7-5.

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

Figure 7-5 EIGRP Classic Metric Formula with Definitions

By default, K_1 and K_3 have the value 1, and K_2, K_4, and K_5 are set to 0. Figure 7-6 places default K values into the formula and then shows a streamlined version of the formula.

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

$$\Big\Downarrow \ \text{Equals}$$

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

Figure 7-6 EIGRP Classic Metric Formula with Default K Values

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

Figure 7-7 displays the information in the EIGRP update packets for the 10.1.1.0/24 prefix propagating through the autonomous system. Notice that the hop count increments, minimum bandwidth decreases, total delay increases, and RD changes with each router in the AS.

Key Topic

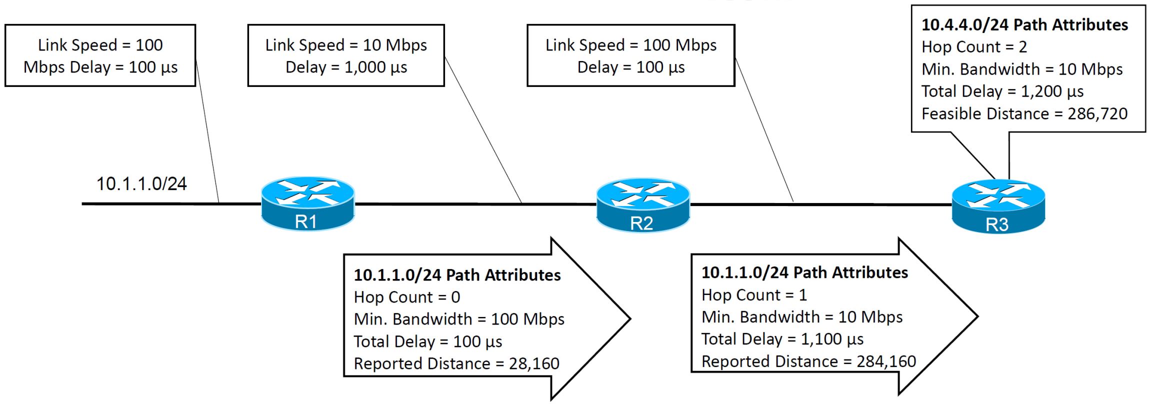


Figure 7-7 EIGRP Attribute Propagation

Table 7-4 shows some of the common network types, link speeds, delay, and EIGRP metrics, using the streamlined formula from Figure 7-6.

Table 7-4 Default EIGRP Interface Metrics for Classic Metrics

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

Using the topology from Figure 7-2, the metric from R1 and R2 for the 10.4.4.0/24 network can be calculated using the formula in Figure 7-8. The link speed for both routers is 1 Gbps, and the total delay is 30 µs (10 µs for the 10.4.4.0/24 link, 10 µs for the 10.34.1.0/24 link, and 10 µs for the 10.13.1.0/24 link).

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

Figure 7-8 EIGRP Classic Metric Formula with Default K Values

Wide Metrics

The original EIGRP specifications measured delay in 10 µs units and bandwidth in kilobytes per second, which did not scale well with higher-speed interfaces. In Table 7-4, notice that the delay is the same for the Gigabit Ethernet and 10-Gigabit Ethernet interfaces.

Example 7-1 provides some metric calculations for common LAN interface speeds. Notice that there is not a differentiation between an 11 Gbps interface and a 20 Gbps interface. The composite metric stays at 256, despite having different bandwidth rates.

Example 7-1 Calculating Metrics for Common LAN Interface Speeds

```
GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 1000000
Scaled Delay = 10 / 10
Composite Metric = 10 + 1 * 256 = 2816

10 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 10000000
Scaled Delay = 10 / 10
Composite Metric = 1 + 1 * 256 = 512

11 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 11000000
Scaled Delay = 10 / 10
Composite Metric = 0 + 1 * 256 = 256

20 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 20000000
Scaled Delay = 10 / 10
Composite Metric = 0 + 1 * 256 = 256
```

EIGRP includes support for a second set of metrics, known as *wide metrics*, that addresses the issue of scalability with higher-capacity interfaces. The original formula referenced in Figure 7-4 refers to *EIGRP classic metrics*.

Figure 7-9 shows the explicit EIGRP wide metrics formula. Notice that an additional K value (K_6) is included that adds an extended attribute to measure jitter, energy, or other future attributes.

Key Topic

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

Figure 7-9 EIGRP Wide Metrics Formula

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

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

Figure 7-10 EIGRP Wide Metrics Formula with Definitions

Metric Backward Compatibility

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

EIGRP is able to detect when peering with a router is using classic metrics, and it *unscales* a metric from the formula in Figure 7-11.

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

Figure 7-11 Formula for Calculating Unscaled EIGRP Metrics

Load Balancing

EIGRP allows multiple successor routes (using the same metric) to be installed into the RIB. Installing multiple paths into the RIB for the same prefix is called *equal-cost multipathing (ECMP)*.

Key Topic

EIGRP supports unequal-cost load balancing, which allows installation of both successor routes and feasible successors into the EIGRP RIB. EIGRP supports unequal-cost load balancing by changing EIGRP's *variance multiplier*. The

EIGRP *variance value* is the feasible distance (FD) for a route multiplied by the EIGRP variance multiplier. Any feasible successor's FD with a metric below the EIGRP variance value is installed into the RIB. EIGRP installs multiple routes where the FD for the routes is less than the EIGRP multiplier value up to the maximum number of ECMP routes, as discussed earlier.

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

Using Figure 7-2 as the example topology and output from the EIGRP topology table in Figure 7-3, the minimum EIGRP variance multiplier can be calculated so that the direct path from R1 to R4 can be installed into the RIB. The FD for the successor route is 3328, and the FD for the feasible successor is 5376. The formula provides a value of about 1.6 and is always rounded up to the nearest whole number to provide an EIGRP variance multiplier of 2. Figure 7-12 displays the calculation.

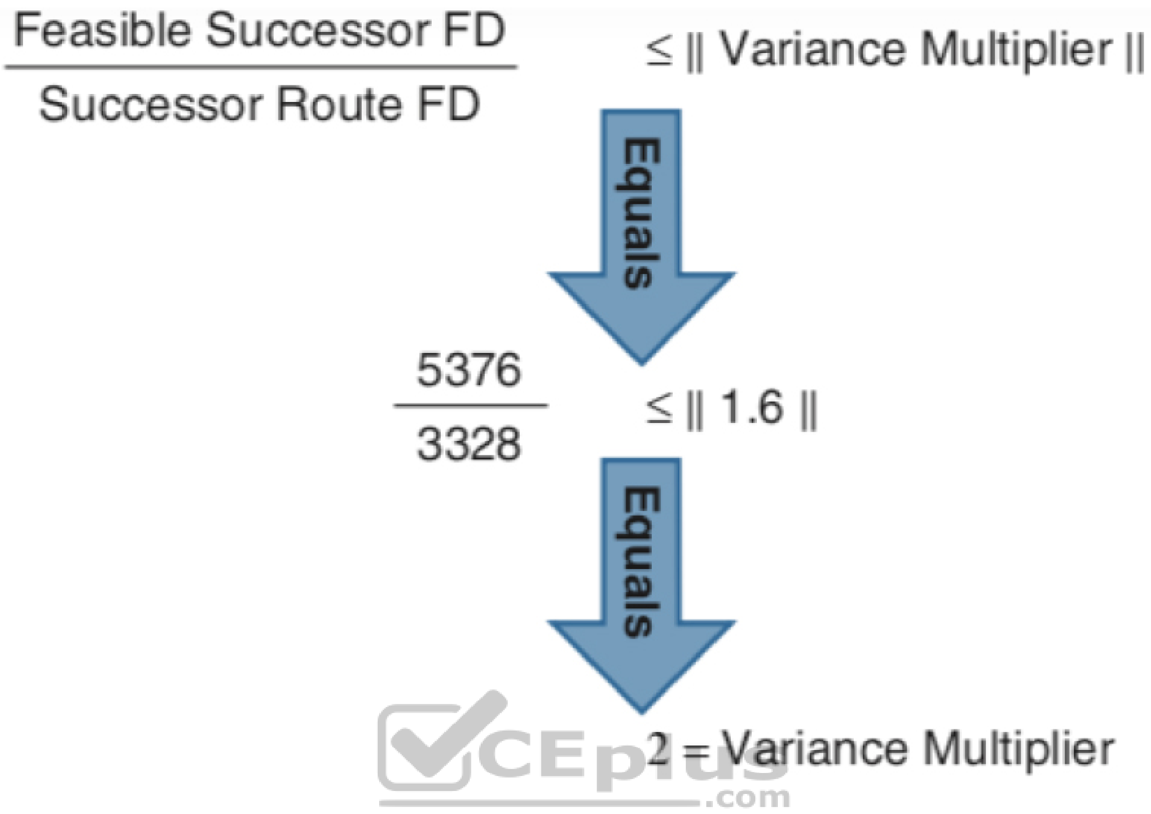


Figure 7-12 EIGRP Variance Multiplier Formula

Example 7-2 provides a brief verification that both paths have been installed into the RIB. Notice that the metrics for the paths are different. One path metric is 3328, and the other path metric is 5376. The *traffic share count* setting correlates to the ratio of traffic sent across each path.

Example 7-2 Verifying Unequal-Cost Load Balancing

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

      10.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
```

```
D        10.4.4.0/24 [90/5376] via 10.14.1.4, 00:00:03, GigabitEt
                     [90/3328] via 10.13.1.3, 00:00:03, GigabitEt

R1# show ip route 10.4.4.0
Routing entry for 10.4.4.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
  Redistributing via eigrp 100
  Last update from 10.13.1.3 on GigabitEthernet0/1, 00:00:35 ago
  Routing Descriptor Blocks:
  * 10.14.1.4, from 10.14.1.4, 00:00:35 ago, via GigabitEthernet0
      Route metric is 5376, traffic share count is 149
      Total delay is 110 microseconds, minimum bandwidth is 10000
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    10.13.1.3, from 10.13.1.3, 00:00:35 ago, via GigabitEthernet0
      Route metric is 3328, traffic share count is 240
      Total delay is 30 microseconds, minimum bandwidth is 100000
      Reliability 254/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

FAILURE DETECTION AND TIMERS

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

EIGRP uses a second timer for the *hold time*, which is the amount of time EIGRP deems the router reachable and functioning. The hold time value defaults to 3 times the hello interval. The default value is 15 seconds, and it is 180 seconds for slow-speed interfaces. The hold time decrements, and upon receipt of a hello packet, the hold time resets and restarts the countdown. If the hold time reaches 0, EIGRP declares the neighbor unreachable and notifies DUAL of a topology change.

Key Topic

Convergence

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

When EIGRP detects that it has lost its successor for a path, the feasible successor instantly becomes the successor route, providing a backup route. The router sends out an update packet for that path because of the new EIGRP path metrics. Downstream routers run their own DUAL for any impacted prefixes to account for the new EIGRP metrics. It is possible that a change of the successor route or feasible successor may occur upon receipt of new EIGRP metrics from a successor router for a prefix. Figure 7-13 demonstrates such a scenario when the link between R1 and R3 fails.

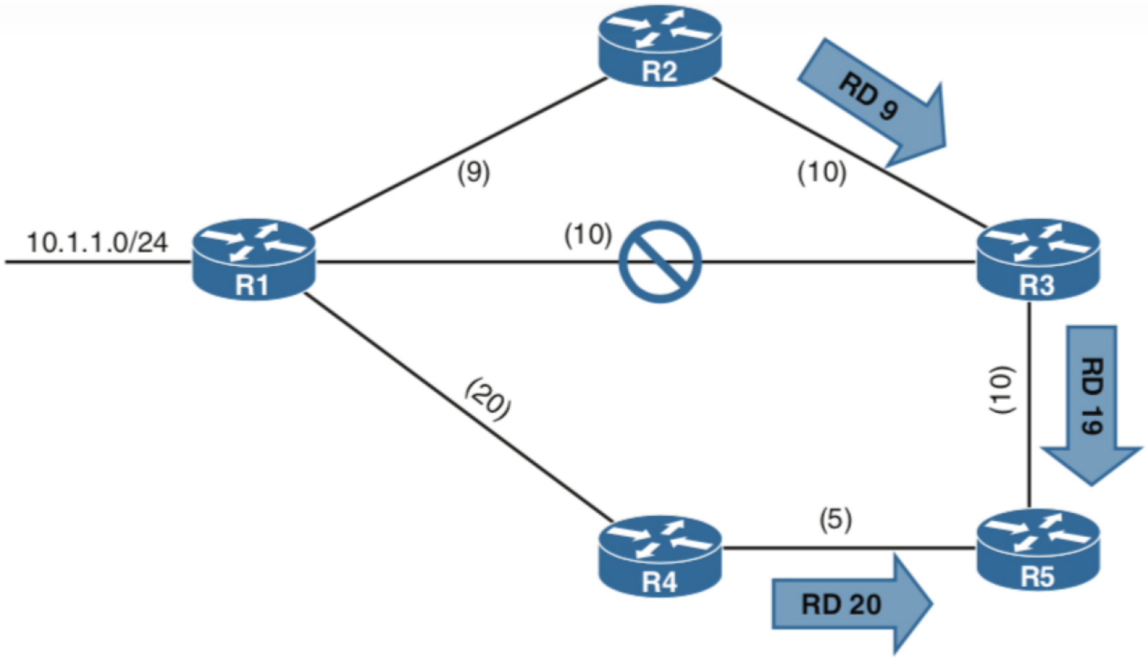


Figure 7-13 EIGRP Topology with Link Failure

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

Key Topic

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

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

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

• It might reply to the query that the router does not have a route to the prefix.

• If the query did not come from the successor for that route, it detects the delay set for infinity but ignores it because it did not come from the successor. The receiving router replies with the EIGRP attributes for that route.

• If the query came from the successor for the route, the receiving router detects the delay set for infinity, sets the prefix as active in the EIGRP topology, and sends out a query packet to all downstream EIGRP neighbors for that route.

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

• It says it does not have a route to the prefix.

- It replies with EIGRP attributes because the query did not come from the successor.

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

Figure 7-14 shows a topology where the link between R1 and R2 has failed.

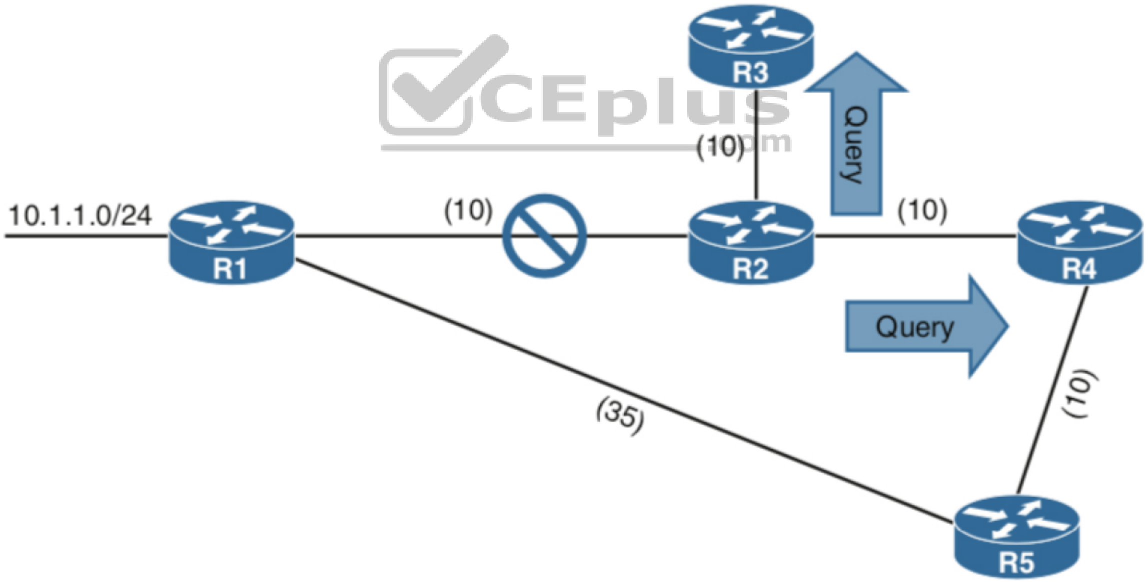


Figure 7-14 EIGRP Convergence Topology

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

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

Step 2. R3 receives the query from R2 and processes the Delay field that is set to infinity. R3 does not have any other EIGRP neighbors and sends a reply to R2 saying that a route does not exists.

R4 receives the query from R2 and processes the Delay field that is set to infinity. Because the query was received by the successor, and a feasible successor for the prefix does not exist, R4 marks the route as active and sends a query to R5.

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

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

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

ROUTE SUMMARIZATION

EIGRP works well with minimal optimizations. Scalability of an EIGRP autonomous system depends on summarization. As the size of an EIGRP autonomous system increases, convergence may take longer. Scaling an EIGRP topology requires summarizing routes in a hierarchical fashion.

EIGRP summarizes network prefixes on an interface basis. A summary aggregate is configured for the EIGRP interface. Prefixes within the summary aggregate are suppressed, and the summary aggregate prefix is advertised in lieu of the original prefixes. The summary aggregate prefix is not advertised until a prefix matches it. Interface-specific summarization can be performed in any portion of the network topology. In addition to shrinking the routing tables of all the routers, summarization creates a query boundary and shrinks the query domain when a route goes active during convergence.

Figure 7-15 illustrates the concept of EIGRP summarization. Without summarization, R2 advertises the 172.16.1.0/24, 172.16.3.0/24, 172.16.12.0/24, and 172.16.23.0/24 networks toward R4. R2 can summarize these network prefixes to the summary aggregate 172.16.0.0/16 prefix so that only one advertisement is sent to R4.

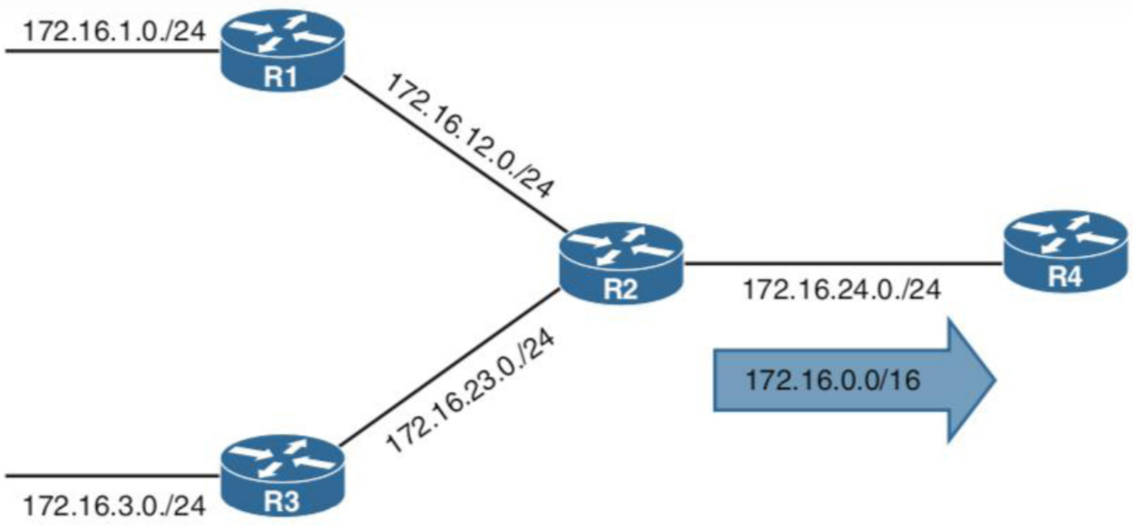


Figure 7-15 EIGRP Summarization

EXAM PREPARATION TASKS

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, "Final Preparation," and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 7-5 lists these key topics and the page number on which each is found.

Key Topic

Table 7-5 Key Topics for Chapter 7

Key Topic Element	Description	Page
Table 7-2	EIGRP Terminology	
Section	Topology table	
Table 7-3	EIGRP Packet Types	
Figure 7-7	EIGRP Attribute Propagation	
Figure 7-9	EIGRP Wide Metric Formula	
Paragraph	EIGRP unequal-cost load balancing	
Section	Convergence	
Paragraph	Active route state	

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the companion website, includes completed tables and lists you can use to check your work.

DEFINE KEY TERMS

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

autonomous system

feasible distance

feasibility condition

feasibility successor

hello packets

hello timer

K values

reported distance

successor

successor route

summarization

topology table

variance value

wide metric

REFERENCES IN THIS CHAPTER

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

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

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

Chapter 8. OSPF

This chapter covers the following subjects:

- **OSPF Fundamentals:** This section provides an overview of communication between OSPF routers.

- **OSPF Configuration:** This section describes the OSPF configuration techniques and commands that can be executed to verify the exchange of routes.

- **Default Route Advertisement:** This section explains how default routes are advertised in OSPF.

- **Common OSPF Optimizations:** This section reviews common OSPF settings for optimizing the operation of the protocol.

The Open Shortest Path First (OSPF) protocol is the first link-state routing protocol covered in this book. OSPF is a nonproprietary Interior Gateway Protocol (IGP) that overcomes the deficiencies of other distance vector routing protocols and distributes routing information within a single OSPF routing domain. OSPF introduced the concept of variable-length subnet masking (VLSM), which supports classless routing, summarization, authentication, and external route tagging. There are two main versions of OSPF in production networks today:

- **OSPF Version 2 (OSPFv2):** Defined in RFC 2328 and supports IPv4

- **OSPF Version 3 (OSPFv3):** Defined in RFC 5340 and modifies the original structure to support IPv6

This chapter explains the core concepts of OSPF and the basics of establishing neighborships and exchanging routes with other OSPF routers. Two other chapters in this book also cover OSPF-related topics. Here is an overview of them:

- **Chapter 9, "Advanced OSPF":** Explains the function of segmenting the OSPF domain into smaller areas to support larger topologies.

- **Chapter 10, "OSPFv3":** Explains how OSPF can be used for routing IPv6 packets.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 8-1 lists the major headings in this chapter and the "Do I Know

This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 8-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topic Section	Questions
OSPF Fundamentals	1–3
OSPF Configuration	4–5
Default Route Advertisement	6
Common OSPF Optimizations	7–10

1. OSPF uses the protocol number ______ for its inter-router communication.

a. 87

b. 88

c. 89

d. 90

2. OSPF uses _______ packet types for inter-router communication.

a. three

b. four

c. five

d. six

e. seven

3. What destination addresses does OSPF use, when feasible? (Choose two.)

a. IP address 224.0.0.5

b. IP address 224.0.0.10

c. IP address 224.0.0.8

d. MAC address 01:00:5E:00:00:05

e. MAC address 01:00:5E:00:00:0A

4. True or false: OSPF is only enabled on a router interface by using the command **network** *ip-address wildcard-mask* **area** *area-id* under the OSPF router process.

a. True

b. False

5. True or false: The OSPF process ID must match for routers to establish a neighbor adjacency.

a. True

b. False

6. True or false: A default route advertised with the command **default information-originate** in OSPF will always appear as an OSPF inter-area route.

a. True

b. False

7. True or false: The router with the highest IP address is the designated router when using a serial point-to-point link.

a. True

b. False

8. OSPF automatically assigns a link cost to an interface based on a reference bandwidth of ______.

a. 100 Mbps

b. 1 Gbps

c. 10 Gbps

d. 40 Gbps

9. What command is configured to prevent a router from becoming the designated router for a network segment?

a. The interface **command ip ospf priority 0**

b. The interface **command ip ospf priority 255**

c. The command **dr-disable** *interface-id* under the OSPF process

d. The command **passive interface** *interface-id* under the OSPF process

e. The command **dr-priority** *interface-id* **255** under the OSPF process

10. What is the advertised network for the loopback interface with IP address 10.123.4.1/30?

a. 10.123.4.1/24

b. 10.123.4.0/30

c. 10.123.4.1/32

d. 10.123.4.0/24

Answers to the “Do I Know This Already?” quiz:

1. C

2. C

3. A, D

4. B

5. B

6. B

7. B

8. A

9. A

10. C

FOUNDATION TOPICS

OSPF FUNDAMENTALS

OSPF sends to neighboring routers link-state advertisements (LSAs) that contain the link state and link metric. The received LSAs are stored in a local database called the link-state database (LSDB), and they are flooded throughout the OSPF routing domain, just as the advertising router advertised them. All OSPF routers maintain a synchronized identical copy of the LSDB for the same area.

The LSDB provides the topology of the network, in essence providing for the router a complete map of the network. All OSPF routers run the Dijkstra shortest path first (SPF) algorithm to construct a loop-free topology of shortest paths. OSPF dynamically detects topology changes within the network and calculates loop-free paths in a short amount of time with minimal routing protocol traffic.

Each router sees itself as the root or top of the SPF tree (SPT), and the SPT contains all network destinations within the OSPF domain. The SPT differs for each OSPF router, but the LSDB used to calculate the SPT is identical for all OSPF routers.

Figure 8-1 shows a simple OSPF topology and the SPT from R1’s and R4’s perspective. Notice that the local router’s perspective will always be the root (top of the tree). There is a difference in connectivity to the 10.3.3.0/24 network from R1’s SPT and R4’s SPT. From R1’s perspective, the serial link between R3 and R4 is missing; from R4’s perspective, the Ethernet link between R1 and R3 is missing.

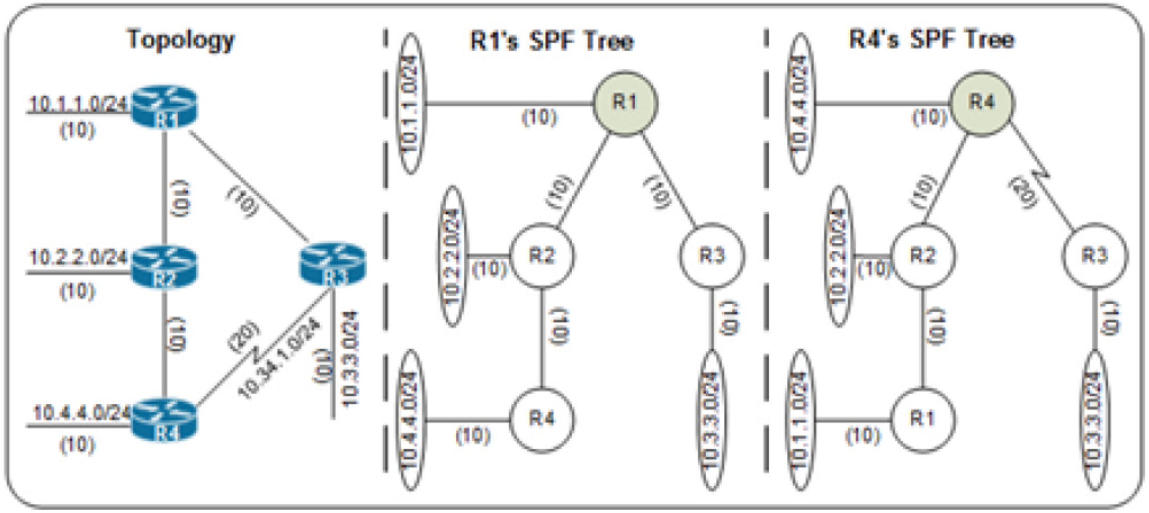


Figure 8-1 OSPF Shortest Path First (SPF) Tree

The SPTs give the illusion that no redundancy exists to the networks, but remember that the SPT shows the shortest path to reach a network and is built from the LSDB, which contains all the links for an area. During a topology change, the SPT is rebuilt and may change.

Key Topic

OSPF provides scalability for the routing table by using multiple OSPF areas within the routing domain. Each OSPF area provides a collection of connected networks and hosts that are grouped together. OSPF uses a two-tier hierarchical architecture, where Area 0 is a special area known as the backbone, to which all other areas must connect. In other words, Area 0 provides transit connectivity between nonbackbone areas. Nonbackbone areas advertise routes into the backbone, and the backbone then advertises routes into other nonbackbone areas.

Figure 8-2 shows route advertisement into other areas. Area 12 routes are advertised to Area 0 and then into Area 34. Area 34 routes are advertised to Area 0 and then into Area 12. Area 0 routes are advertised into all other OSPF areas.

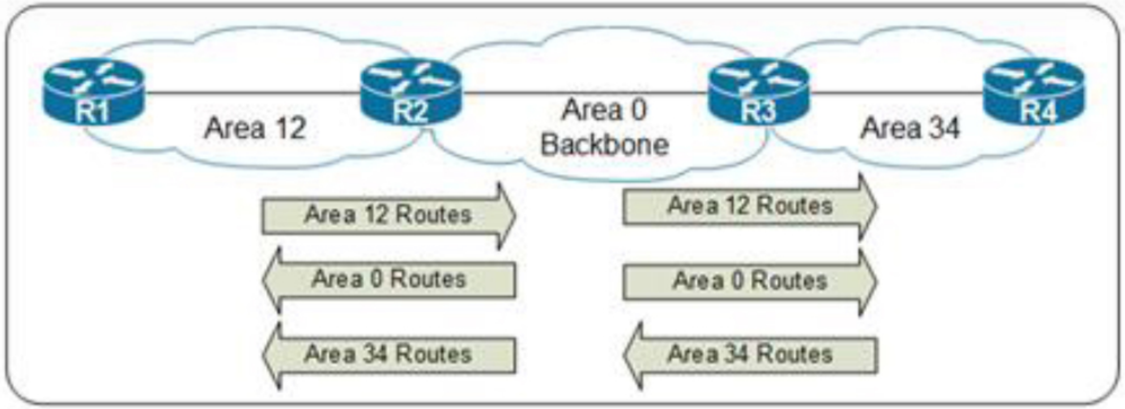


Figure 8-2 Two-Tier Hierarchical Area Structure

The exact topology of the area is invisible from outside the area while still providing connectivity to routers outside the area. This means that routers outside the area do not have a complete topological map for that area, which reduces OSPF traffic in that area. When you segment an OSPF routing domain into multiple areas, it is no longer true that all OSPF routers will have identical LSDBs; however, all routers within the same area will have identical area LSDBs.

The reduction in routing traffic uses less router memory and resources and therefore provides scalability. Chapter 9 explains areas in greater depth; this chapter focuses on the core OSPF concepts. For the remainder of this chapter, OSPF Area 0 is used as a reference area.

A router can run multiple OSPF processes. Each process maintains its own unique database, and routes learned in one OSPF process are not available to a different OSPF process without redistribution of routes between processes. The

OSPF process numbers are locally significant and do not have to match among routers. Running OSPF process number 1 on one router and running OSPF process number 1234 will still allow the two routers to become neighbors.

Key Topic

Inter-Router Communication

OSPF runs directly over IPv4, using its own protocol 89, which is reserved for OSPF by the Internet Assigned Numbers Authority (IANA). OSPF uses multicast where possible to reduce unnecessary traffic. The two OSPF multicast addresses are as follows:

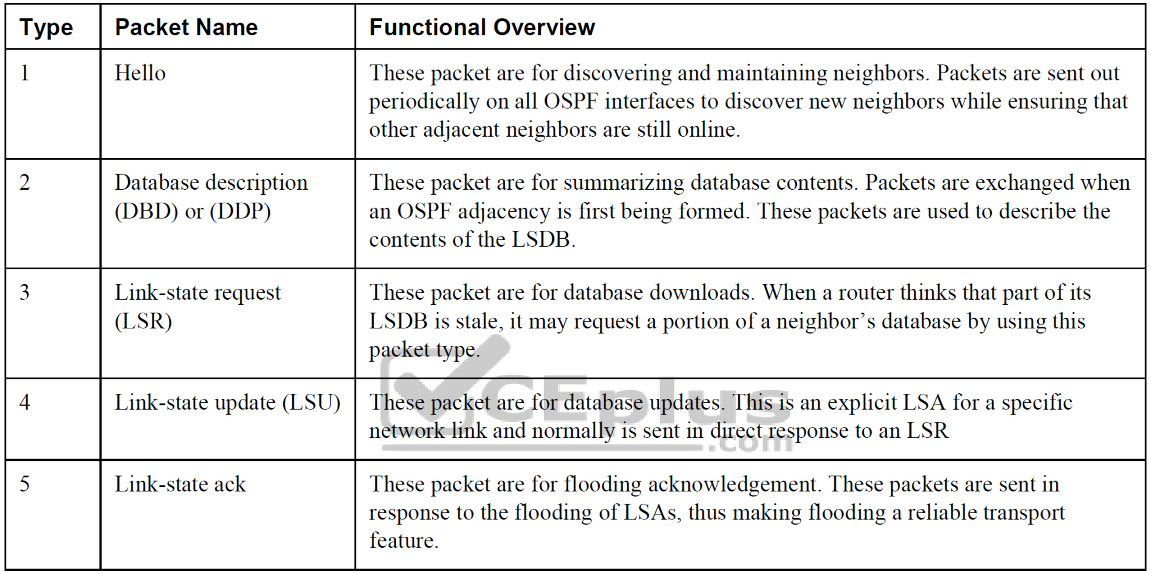
- **AllSPFRouters:** IPv4 address 224.0.0.5 or MAC address 01:00:5E:00:00:05. All routers running OSPF should be able to receive these packets.

- **AllDRouters:** IPv4 address 224.0.0.6 or MAC address 01:00:5E:00:00:06. Communication with designated routers (DRs) uses this address.

Within the OSPF protocol, five types of packets are communicated. Table 8-2 briefly describes these OSPF packet types.

Key Topic

Table 8-2 OSPF Packet Types

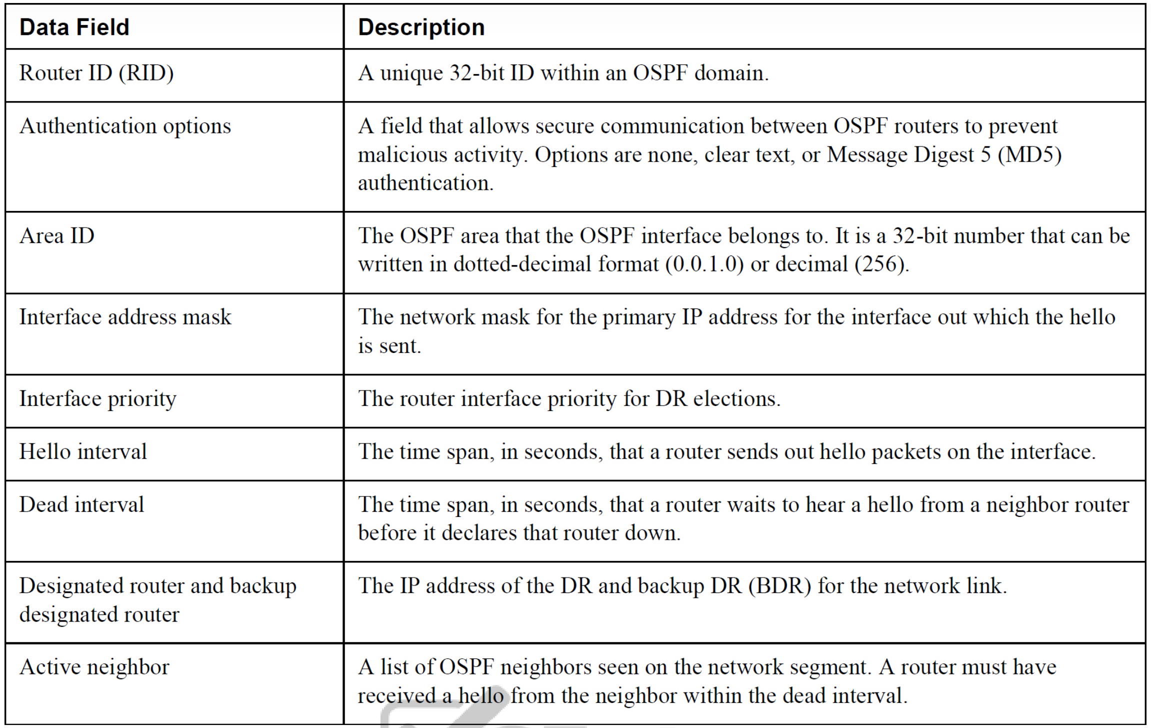


Type	Packet Name	Functional Overview
1	Hello	These packet are for discovering and maintaining neighbors. Packets are sent out periodically on all OSPF interfaces to discover new neighbors while ensuring that other adjacent neighbors are still online.
2	Database description (DBD) or (DDP)	These packet are for summarizing database contents. Packets are exchanged when an OSPF adjacency is first being formed. These packets are used to describe the contents of the LSDB.
3	Link-state request (LSR)	These packet are for database downloads. When a router thinks that part of its LSDB is stale, it may request a portion of a neighbor's database by using this packet type.
4	Link-state update (LSU)	These packet are for database updates. This is an explicit LSA for a specific network link and normally is sent in direct response to an LSR
5	Link-state ack	These packet are for flooding acknowledgement. These packets are sent in response to the flooding of LSAs, thus making flooding a reliable transport feature.

OSPF Hello Packets

OSPF hello packets are responsible for discovering and maintaining neighbors. In most instances, a router sends hello packets to the AllSPFRouters address (224.0.0.5). Table 8-3 lists some of the data contained within an OSPF hello packet.

Table 8-3 OSPF Hello Packet Fields



Data Field	Description
Router ID (RID)	A unique 32-bit ID within an OSPF domain.
Authentication options	A field that allows secure communication between OSPF routers to prevent malicious activity. Options are none, clear text, or Message Digest 5 (MD5) authentication.
Area ID	The OSPF area that the OSPF interface belongs to. It is a 32-bit number that can be written in dotted-decimal format (0.0.1.0) or decimal (256).
Interface address mask	The network mask for the primary IP address for the interface out which the hello is sent.
Interface priority	The router interface priority for DR elections.
Hello interval	The time span, in seconds, that a router sends out hello packets on the interface.
Dead interval	The time span, in seconds, that a router waits to hear a hello from a neighbor router before it declares that router down.
Designated router and backup designated router	The IP address of the DR and backup DR (BDR) for the network link.
Active neighbor	A list of OSPF neighbors seen on the network segment. A router must have received a hello from the neighbor within the dead interval.

Router ID

The OSPF router ID (RID) is a 32-bit number that uniquely identifies an OSPF router. In some OSPF output commands, *neighbor ID* refers to the RID; the terms are synonymous. The RID must be unique for each OSPF process in an OSPF domain and must be unique between OSPF processes on a router.

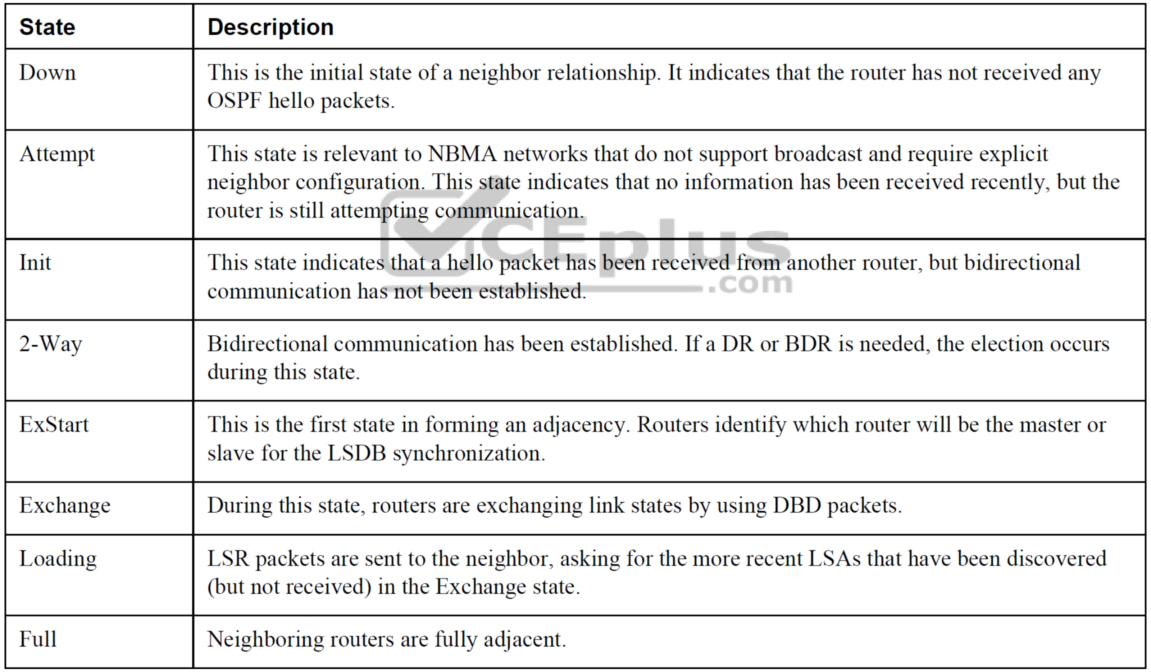
Neighbors

An OSPF neighbor is a router that shares a common OSPF-enabled network link. OSPF routers discover other neighbors via the OSPF hello packets. An adjacent OSPF neighbor is an OSPF neighbor that shares a synchronized OSPF database between the two neighbors.

Each OSPF process maintains a table for adjacent OSPF neighbors and the state of each router. Table 8-4 briefly describes the OSPF neighbor states.

Key Topic

Table 8-4 OSPF Neighbor States



State	Description
Down	This is the initial state of a neighbor relationship. It indicates that the router has not received any OSPF hello packets.
Attempt	This state is relevant to NBMA networks that do not support broadcast and require explicit neighbor configuration. This state indicates that no information has been received recently, but the router is still attempting communication.
Init	This state indicates that a hello packet has been received from another router, but bidirectional communication has not been established.
2-Way	Bidirectional communication has been established. If a DR or BDR is needed, the election occurs during this state.
ExStart	This is the first state in forming an adjacency. Routers identify which router will be the master or slave for the LSDB synchronization.
Exchange	During this state, routers are exchanging link states by using DBD packets.
Loading	LSR packets are sent to the neighbor, asking for the more recent LSAs that have been discovered (but not received) in the Exchange state.
Full	Neighboring routers are fully adjacent.

Designated Router and Backup Designated Router

Multi-access networks such as Ethernet (LANs) and Frame Relay allow more than two routers to exist on a network segment. Such a setup could cause scalability problems with OSPF as the number of routers on a segment increases.

Additional routers flood more LSAs on the segment, and OSPF traffic becomes excessive as OSPF neighbor adjacencies increase. If four routers share the same multi-access network, six OSPF adjacencies form, along with six occurrences of database flooding on a network. Figure 8-3 shows a simple four-router physical topology and the adjacencies established.

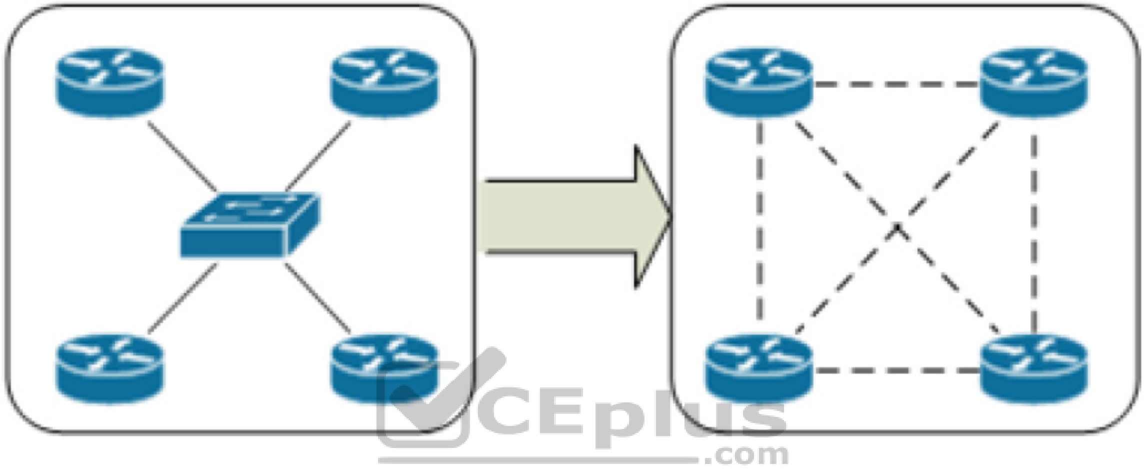


Figure 8-3 Multi-Access Physical Topology Versus Logical Topology

The number of edges formula, $n(n - 1) / 2$, where n represents the number of routers, is used to identify the number of sessions in a full mesh topology. If 5 routers were present on a segment, $5(5 - 1) / 2 = 10$, then 10 OSPF adjacencies would exist for that segment. Continuing the logic, adding 1 additional router would makes 15 OSPF adjacencies on a network segment. Having so many adjacencies per segment consumes more bandwidth, more CPU processing, and more memory to maintain each of the neighbor states.

Figure 8-4 illustrates the exponential rate of OSPF adjacencies needed as routers on a network segment increase.

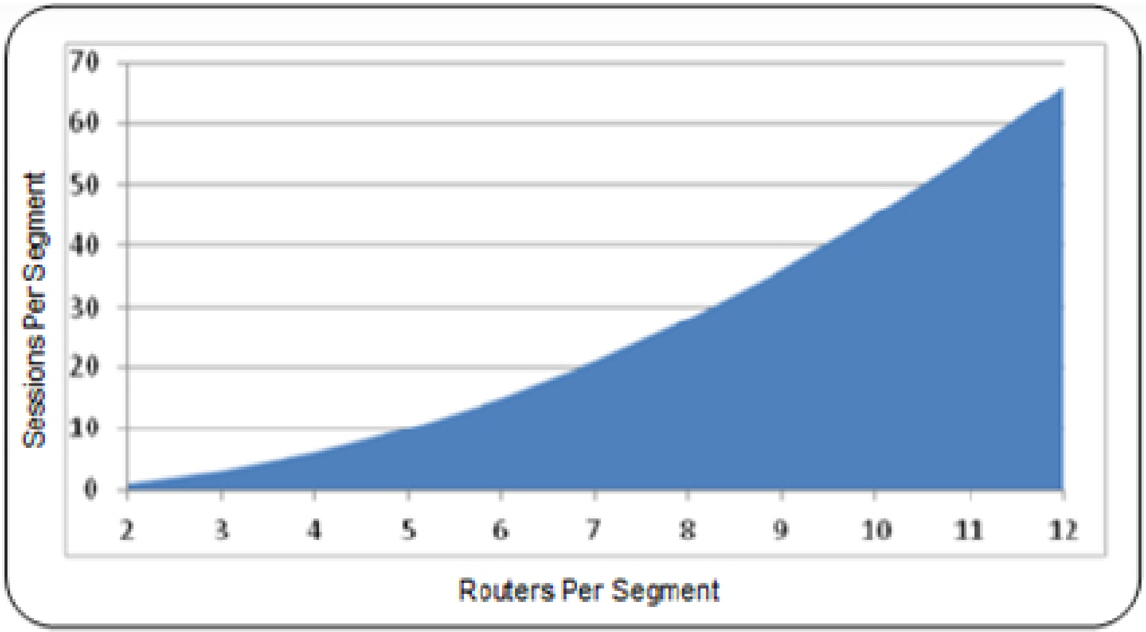


Figure 8-4 Exponential LSA Sessions for Routers on the Same Segment

Key Topic

OSPF overcomes this inefficiency by creating a pseudonode (virtual router) to manage the adjacency state with all the other routers on that broadcast network segment. A router on the broadcast segment, known as the *designated router (DR)*, assumes the role of the pseudonode. The DR reduces the number of OSPF adjacencies on a multi-access network segment because routers only form a full OSPF adjacency with the DR and not each other. The DR is responsible for flooding updates to all OSPF routers on that segment as the updates occur. Figure

8-5 demonstrates how using a DR simplifies a four-router topology with only three neighbor adjacencies.

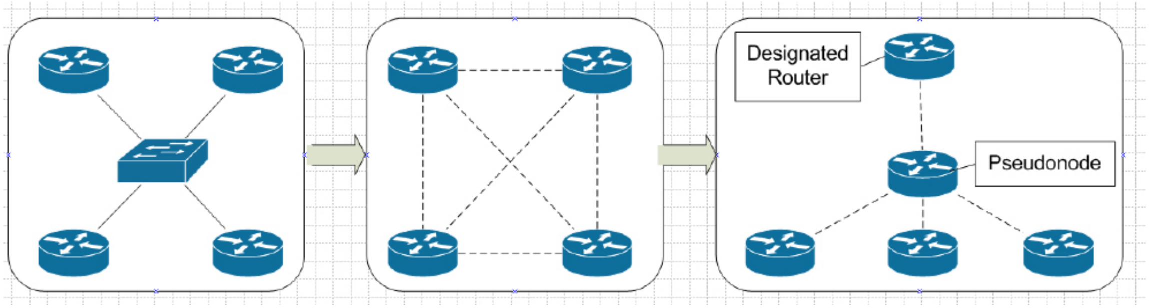


Figure 8-5 OSPF DR Concept

If the DR were to fail, OSPF would need to form new adjacencies, invoking all new LSAs, and could potentially cause a temporary loss of routes. In the event of DR failure, a *backup designated router (BDR)* becomes the new DR; then an election occurs to replace the BDR. To minimize transition time, the BDR also forms full OSPF adjacencies with all OSPF routers on that segment.

The DR/BDR process distributes LSAs in the following manner:

1. All OSPF routers (DR, BDR, and DROTHER) on a segment form full OSPF adjacencies with the DR and BDR.

2. As an OSPF router learns of a new route, it sends the updated LSA to the AllDRouters (224.0.0.6) address, which only the DR and BDR receive and process, as illustrated in step 1 of Figure 8-6.

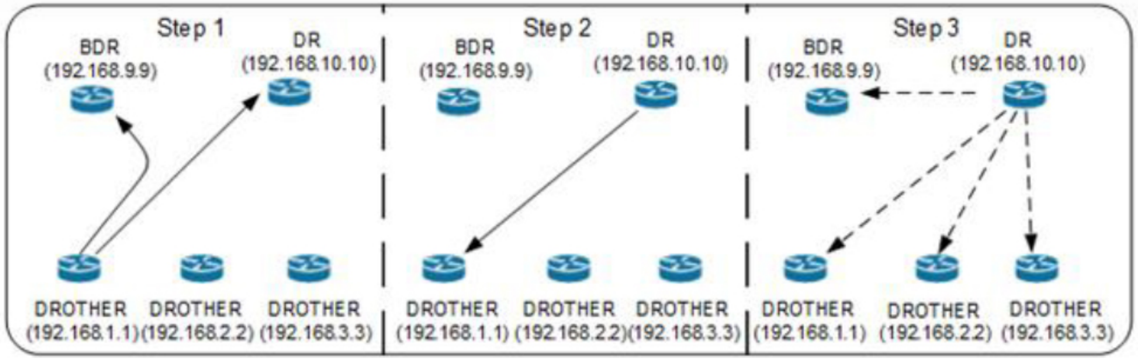


Figure 8-6 Network Prefix Advertisement with DR Segments

3. The DR sends a unicast acknowledgement to the router that sent the initial LSA update, as illustrated in step 2 of Figure 8-6.

4. The DR floods the LSA to all the routers on the segment via the AllSPFRouters (224.0.0.5) address, as shown in step 3 of Figure 8-6.

OSPF CONFIGURATION

The configuration process for OSPF resides mostly under the OSPF process, but some OSPF options go directly on the interface configuration submode. The command **router ospf** *process-id* defines and initializes the OSPF process. The OSPF process ID is locally significant but is generally kept the same for operational consistency. OSPF is enabled on an interface using two methods:

- An OSPF network statement

- Interface-specific configuration

The following section describes these techniques.

OSPF Network Statement

The OSPF network statement identifies the interfaces that the OSPF process will use and the area that those interfaces participate in. The network statements match against the primary IPv4 address and netmask associated with an interface.

Key Topic

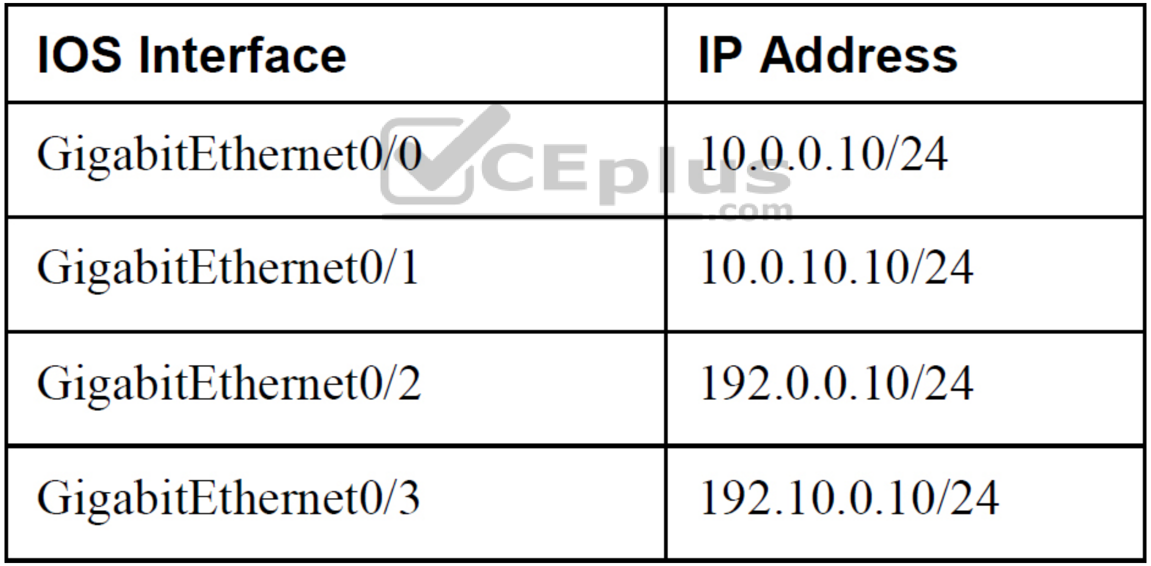
A common misconception is that the network statement advertises the networks into OSPF; in reality, though, the network statement is selecting and enabling OSPF on the interface. The interface is then advertised in OSPF through the LSA. The network statement uses a wildcard mask, which allows the configuration to be as specific or vague as necessary. The selection of interfaces within the OSPF process is accomplished by using the command **network** *ip-address wildcard-mask* **area** *area-id*.

The concept is similar to the configuration of Enhanced Interior Gateway Routing Protocol (EIGRP), except that the OSPF area is specified. If the IP address for an interface matches two network statements with different areas, the most explicit network statement (that is, the longest match) preempts the other network statements for area allocation.

The connected network for the OSPF-enabled interface is added to the OSPF LSDB under the corresponding OSPF area in which the interface participates. Secondary connected networks are added to the LSDB only if the secondary IP address matches a network statement associated with the same area.

To help illustrate the concept, the following scenarios explain potential use cases of the network statement for a router with four interfaces. Table 8-5 provides IP addresses and interfaces.

Table 8-5 Table of Sample Interfaces and IP Addresses



IOS Interface	IP Address
GigabitEthernet0/0	10.0.0.10/24
GigabitEthernet0/1	10.0.10.10/24
GigabitEthernet0/2	192.0.0.10/24
GigabitEthernet0/3	192.10.0.10/24

The configuration in Example 8-1 enables OSPF for Area 0 only on the interfaces that explicitly match the IP addresses in Table 8-4.

Example 8-1 Configuring OSPF with Explicit IP Addresses

```
router ospf  1
    network 10.0.0.10 0.0.0.0 area 0
    network 10.0.10.10 0.0.0.0 area 0
    network 192.0.0.10 0.0.0.0 area 0
    network 192.10.0.10 0.0.0.0 area 0
```

Example 8-2 displays the OSPF configuration for Area 0, using network statements that match the subnets used in Table 8-4. If you set the last octet of the IP address to 0 and change the wildcard mask to 255, the network statements match all IP addresses within the /24 network.

Example 8-2 Configuring OSPF with Explicit Subnet

```
router ospf  1
    network 10.0.0.0 0.0.0.255 area 0
    network 10.0.10.0 0.0.0.255 area 0
    network 192.0.0.0 0.0.0.255 area 0
    network 192.10.0.0 0.0.0.255 area 0
```

Example 8-3 displays the OSPF configuration for Area 0, using network statements for interfaces that are within the 10.0.0.0/8 or 192.0.0.0/8 network ranges, and will result in OSPF being enabled on all four interfaces, as in the previous two examples.

Example 8-3 Configuring OSPF with Large Subnet Ranges

```
router ospf  1
    network 10.0.0.0 0.255.255.255 area 0
    network 192.0.0.0 0.255.255.255 area 0
```

Example 8-4 displays the OSPF configuration for Area 0 to enable OSPF on all interfaces.

Example 8-4 Configuring OSPF for All Interfaces

```
router ospf  1
    network 0.0.0.0 255.255.255.255 area 0
```

Note

For simplicity, this chapter focuses on OSPF operation from a single area, Area 0. Chapter 9 explains multi-area OSPF behavior in detail.

Key Topic

Interface-Specific Configuration

The second method for enabling OSPF on an interface for IOS is to configure it specifically on an interface with the command **ip ospf** *process-id* **area** *area-id* [**secondaries none**]. This method also adds secondary connected networks to the LSDB unless the **secondaries none** option is used.

This method provides explicit control for enabling OSPF; however, the configuration is not centralized and increases in complexity as the number of interfaces on the routers increases. If a hybrid configuration exists on a router, interface-specific settings take precedence over the network statement with the assignment of the areas.

Example 8-5 provides a sample interface-specific configuration.

Example 8-5 Configuring OSPF on IOS for a Specific Interface

```
interface GigabitEthernet 0/0
    ip address 10.0.0.1 255.255.255.0
    ip ospf 1 area 0
```

Statically Setting the Router ID

By default, the RID is dynamically allocated using the highest IP address of any *up* loopback interfaces. If there are no *up* loopback interfaces, the highest IP address of any active *up* physical interfaces becomes the RID when the OSPF process initializes.

The OSPF process selects the RID when the OSPF process initializes, and it does not change until the process restarts. Interface changes (such as addition/removal of IP addresses) on a router are detected when the OSPF process restarts, and the RID changes accordingly.

The OSPF topology is built on the RID. Setting a static RID helps with troubleshooting and reduces LSAs when a RID changes in an OSPF environment. The RID is four octets in length but generally represents an IPv4 address that resides on the router for operational simplicity; however, this is not a requirement. The command **router-id** *router-id* statically assigns the OSPF RID under the OSPF process.

The command **clear ip ospf process** restarts the OSPF process on a router so that OSPF can use the new RID.

Key Topic

Passive Interfaces

Enabling an interface with OSPF is the quickest way to advertise a network segment to other OSPF routers. However, it might be easy for someone to plug in an unauthorized OSPF router on an OSPF-enabled network segment and introduce false routes, thus causing havoc in the network. Making the network interface passive still adds the network segment into the LSDB but prohibits the interface from forming OSPF adjacencies. A *passive interface* does not send out OSPF hellos and does not process any received OSPF packets.

The command **passive** *interface-id* under the OSPF process makes the interface passive, and the command **passive interface default** makes all interfaces passive. To allow for an interface to process OSPF packets, the command **no passive** *interface-id* is used.

Key Topic

Requirements for Neighbor Adjacency

The following list of requirements must be met for an OSPF neighborship to be formed:

- RIDs must be unique between the two devices. They should be unique for the entire OSPF routing domain to prevent errors.

- The interfaces must share a common subnet. OSPF uses the interface's primary IP address when sending out OSPF hellos. The network mask (netmask) in the hello packet is used to extract the network ID of the hello packet.

- The MTUs (maximum transmission units) on the interfaces must match. The OSPF protocol does not support fragmentation, so the MTUs on the interfaces should match.

- The area ID must match for the segment.
- The DR enablement must match for the segment.
- OSPF hello and dead timers must match for the segment.
- Authentication type and credentials (if any) must match for the segment.
- Area type flags must match for the segment (for example, Stub, NSSA). (These are not discussed in this book.)

Sample Topology and Configuration

Figure 8-7 shows a topology example of a basic OSPF configuration. All four routers have loopback IP addresses that match their RIDs (R1 equals 192.168.1.1, R2 equals 192.168.2.2, and so on).

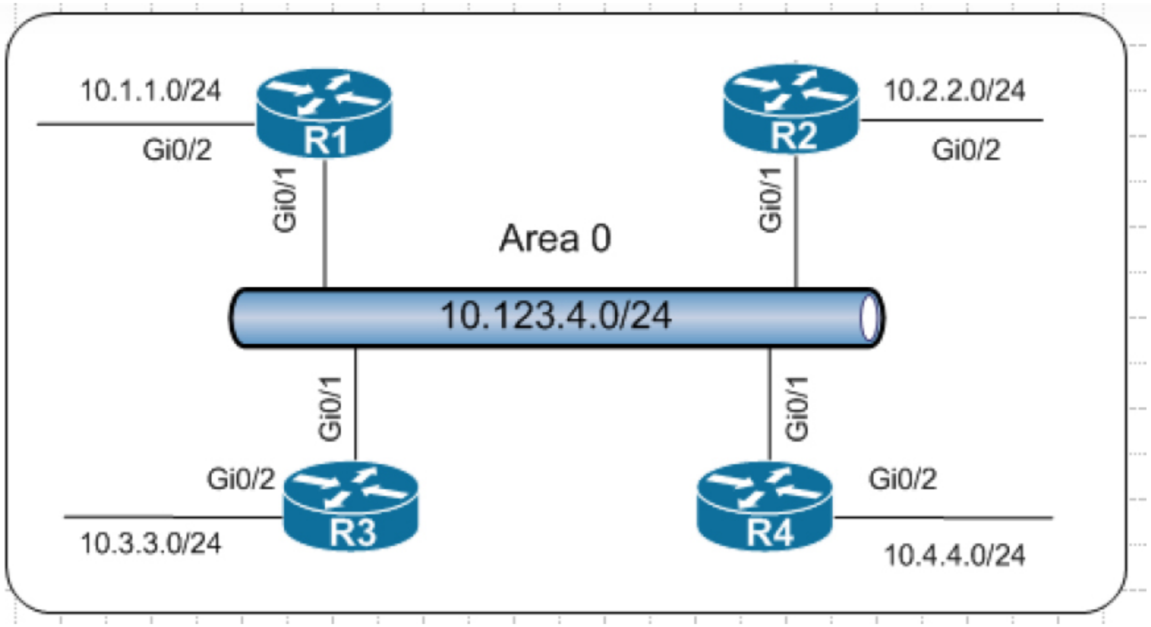


Figure 8-7 Sample OSPF Topology

On R1 and R2, OSPF is enabled on all interfaces with one command, R3 uses specific network-based statements, and R4 uses interface-specific commands. R1 and R2 set the Gi0/2 interface as passive, and R3 and R4 make all interfaces passive by default but make Gi0/1 active.

Example 8-6 provides a sample configuration for all four routers.

Example 8-6 Configuring OSPF for the Topology Example

```
! OSPF is enabled with a single command, and the passive interfac
! set individually
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface Loopback0
```

```
R1(config-if)# ip address 192.168.1.1 255.255.255.255
R1(config-if)# interface GigabitEthernet0/1
R1(config-if)# ip address 10.123.4.1 255.255.255.0
R1(config-if)# interface GigabitEthernet0/2
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)#
R1(config-if)# router ospf 1
R1(config-router)# router-id 192.168.1.1
R1(config-router)# passive-interface GigabitEthernet0/2
R1(config-router)# network 0.0.0.0 255.255.255.255 area 0

! OSPF is enabled with a single command, and the passive interfac
! set individually
R2(config)# interface Loopback0
R2(config-if)# ip address 192.168.2.2 255.255.255.255
R2(config-if)# interface GigabitEthernet0/1
R2(config-if)# ip address 10.123.4.2 255.255.255.0
R2(config-if)# interface GigabitEthernet0/2
R2(config-if)# ip address 10.2.2.2 255.255.255.0
R2(config-if)#
R2(config-if)# router ospf 1
R2(config-router)# router-id 192.168.2.2
R2(config-router)# passive-interface GigabitEthernet0/2
R2(config-router)# network 0.0.0.0 255.255.255.255 area 0

! OSPF is enabled with a network command per interface, and the p
! is enabled globally while the Gi0/1 interface is reset to activ
R3(config)# interface Loopback0
R3(config-if)# ip address 192.168.3.3 255.255.255.255
R3(config-if)# interface GigabitEthernet0/1
R3(config-if)# ip address 10.123.4.3 255.255.255.0
R3(config-if)# interface GigabitEthernet0/2
R3(config-if)# ip address 10.3.3.3 255.255.255.0
R3(config-if)#
R3(config-if)# router ospf 1
```

```
R3(config-router)# router-id 192.168.3.3
R3(config-router)# passive-interface default
R3(config-router)# no passive-interface GigabitEthernet0/1
R3(config-router)# network 10.3.3.3 0.0.0.0 area 0
R3(config-router)# network 10.123.4.3 0.0.0.0 area 0
R3(config-router)# network 192.168.3.3 0.0.0.0 area 0

! OSPF is enabled with a single command under each interface, and
! passive interface is enabled globally while the Gi0/1 interface
R4(config-router)# interface Loopback0
R4(config-if)# ip address 192.168.4.4 255.255.255.255
R4(config-if)# ip ospf 1 area 0
R4(config-if)# interface GigabitEthernet0/1
R4(config-if)# ip address 10.123.4.4 255.255.255.0
R4(config-if)# ip ospf 1 area 0
R4(config-if)# interface GigabitEthernet0/2
R4(config-if)# ip address 10.4.4.4 255.255.255.0
R4(config-if)# ip ospf 1 area 0
R4(config-if)#
R4(config-if)# router ospf 1
R4(config-router)# router-id 192.168.4.4
R4(config-router)# passive-interface default
R4(config-router)# no passive-interface GigabitEthernet0/1
```

Confirmation of Interfaces

It is a good practice to verify that the correct interfaces are running OSPF after making changes to the OSPF configuration. The command **show ip ospf interface** [**brief** | *interface-id*] displays the OSPF-enabled interfaces.

Example 8-7 displays a snippet of the output from R1. The output lists all the OSPF-enabled interfaces, the IP address associated with each interface, the RID for the DR and BDR (and their associated interface IP addresses for that segment), and the OSPF timers for that interface.

Example 8-7 OSPF Interface Output in Detailed Format

```
R1# show ip ospf interface
! Output omitted for brevity
Loopback0 is up, line protocol is up
  Internet Address 192.168.1.1/32, Area 0, Attached via Network S
  Process ID 1, Router ID 192.168.1.1, Network Type LOOPBACK, Cos
  Topology-MTID    Cost    Disabled    Shutdown      Topology Nar
        0           1         no          no             Base
  Loopback interface is treated as a stub Host
GigabitEthernet0/1 is up, line protocol is up
  Internet Address 10.123.4.1/24, Area 0, Attached via Network St
  Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Co
  Topology-MTID    Cost    Disabled    Shutdown      Topology Nar
        0           1         no          no             Base
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 192.168.4.4, Interface address 10.123.4
  Backup Designated router (ID) 192.168.3.3, Interface address 10
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retrans
..
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 192.168.3.3  (Backup Designated Router
    Adjacent with neighbor 192.168.4.4  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Example 8-8 shows the **show ip ospf interface** command with the **brief** keyword.

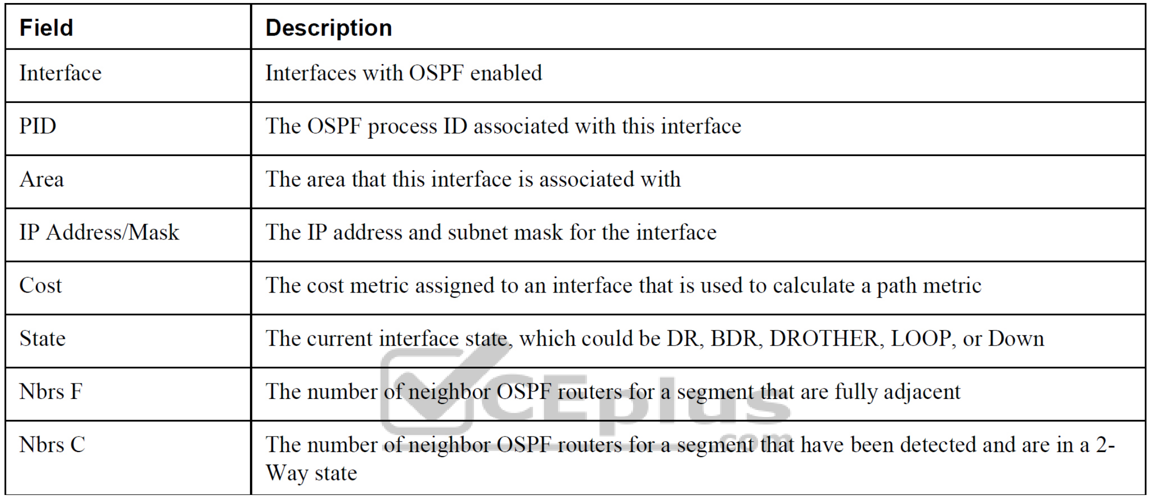
Example 8-8 OSPF Interface Output in Brief Format

```
R1# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Lo0           1     0               192.168.1.1/32     1     LOOP
Gi0/2         1     0               10.1.1.1/24        1     DR
Gi0/1         1     0               10.123.4.1/24      1     DROTH

R2# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Lo0           1     0               192.168.2.2/32     1     LOOP
Gi0/2         1     0               10.2.2.2/24        1     DR
Gi0/1         1     0               10.123.4.2/24      1     DROTH

R3# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Lo0           1     0               192.168.3.3/32     1     LOOP
Gi0/1         1     0               10.123.4.3/24      1     BDR
Gi0/2         1     0               10.3.3.3/24        1     DR

R4# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Lo0           1     0               192.168.4.4/32     1     LOOP
Gi0/1         1     0               10.123.4.4/24      1     DR
Gi0/2         1     0               10.4.4.4/24        1     DR
```

Table 8-6 provides an overview of the fields in the output in Example 8-8.

Key Topic

Table 8-6 OSPF Interface Columns



Field	Description
Interface	Interfaces with OSPF enabled
PID	The OSPF process ID associated with this interface
Area	The area that this interface is associated with
IP Address/Mask	The IP address and subnet mask for the interface
Cost	The cost metric assigned to an interface that is used to calculate a path metric
State	The current interface state, which could be DR, BDR, DROTHER, LOOP, or Down
Nbrs F	The number of neighbor OSPF routers for a segment that are fully adjacent
Nbrs C	The number of neighbor OSPF routers for a segment that have been detected and are in a 2-Way state

Note

The DROTHER is a router on the DR-enabled segment that is not the DR or the BDR; it is simply the other router. DROTHERs do not establish full adjacency with other DROTHERs.

Verification of OSPF Neighbor Adjacencies

The command **show ip ospf neighbor** [**detail**] provides the OSPF neighbor table. Example 8-9 shows sample output on R1, R2, R3, and R4.

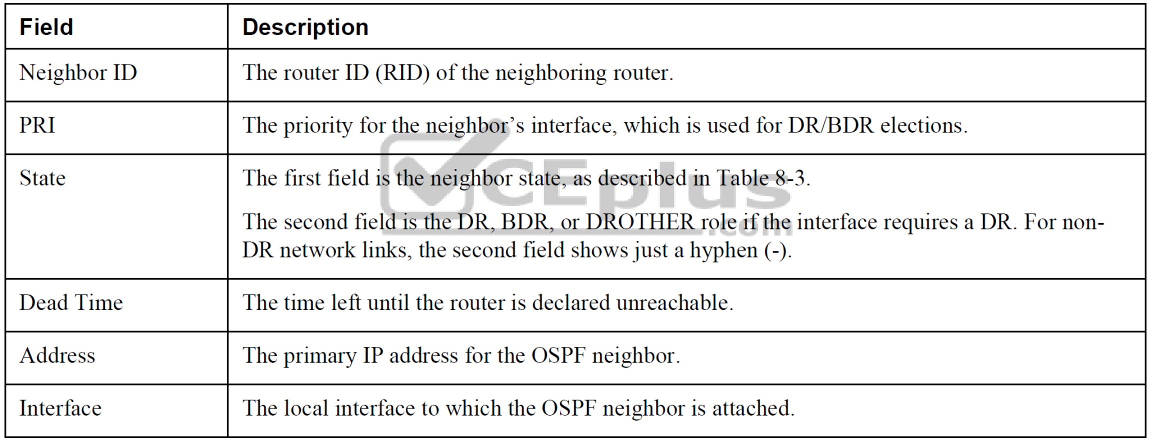
Example 8-9 OSPF Neighbor Output

```
R1# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
192.168.2.2       1   2WAY/DROTHER    00:00:37    10.123.4.2
192.168.3.3       1   FULL/BDR        00:00:35    10.123.4.3
192.168.4.4       1   FULL/DR         00:00:33    10.123.4.4

R2# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
192.168.1.1       1   2WAY/DROTHER    00:00:30    10.123.4.1
192.168.3.3       1   FULL/BDR        00:00:32    10.123.4.3
192.168.4.4       1   FULL/DR         00:00:31    10.123.4.4

R3# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
192.168.1.1       1   FULL/DROTHER    00:00:35    10.123.4.1
192.168.2.2       1   FULL/DROTHER    00:00:34    10.123.4.2
192.168.4.4       1   FULL/DR         00:00:31    10.123.4.4

R4# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address
192.168.1.1       1   FULL/DROTHER    00:00:36    10.123.4.1
192.168.2.2       1   FULL/DROTHER    00:00:34    10.123.4.2
192.168.3.3       1   FULL/BDR        00:00:35    10.123.4.3
```

Table 8-7 provides a brief overview of the fields shown in Example 8-9. The neighbor state on R1 identify R3 as the BDR and R4 as the BDR. R3 and R4 identify R1 and R2 as DROTHER in the output.

Key Topic

Table 8-7 OSPF Neighbor State Fields



Field	Description
Neighbor ID	The router ID (RID) of the neighboring router.
PRI	The priority for the neighbor's interface, which is used for DR/BDR elections.
State	The first field is the neighbor state, as described in Table 8-3. The second field is the DR, BDR, or DROTHER role if the interface requires a DR. For non-DR network links, the second field shows just a hyphen (-).
Dead Time	The time left until the router is declared unreachable.
Address	The primary IP address for the OSPF neighbor.
Interface	The local interface to which the OSPF neighbor is attached.

Verification of OSPF Routes

The next step is to verify the OSPF routes installed in the IP routing table. OSPF routes that install into the Routing Information Base (RIB) are shown with the command **show ip route ospf**.

Example 8-10 provides sample output of the OSPF routing table for R1. In the output, where two sets of numbers are in the brackets (for example, [110/2]/0, the first number is the administrative distance (AD), which is 110 by default for OSPF, and the second number is the metric of the path used for that network. The output for R2, R3 and R4 would be similar to the output in Example 8-10.

Example 8-10 OSPF Routes Installed in the RIB

```
R1# show ip route ospf
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external ty
       E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O        10.2.2.0/24 [110/2] via 10.123.4.2, 00:35:03, GigabitEth
O        10.3.3.0/24 [110/2] via 10.123.4.3, 00:35:03, GigabitEth
O        10.4.4.0/24 [110/2] via 10.123.4.4, 00:35:03, GigabitEth
      192.168.2.0/32 is subnetted, 1 subnets
O        192.168.2.2 [110/2] via 10.123.4.2, 00:35:03, GigabitEth
      192.168.3.0/32 is subnetted, 1 subnets
O        192.168.3.3 [110/2] via 10.123.4.3, 00:35:03, GigabitEth
      192.168.4.0/32 is subnetted, 1 subnets
O        192.168.4.4 [110/2] via 10.123.4.4, 00:35:03, GigabitEth
```

Note

The terms *path cost* and *path metric* are synonymous from OSPF's perspective.

Key Topic

DEFAULT ROUTE ADVERTISEMENT

OSPF supports advertising the default route into the OSPF domain. The default route is advertised by using the command **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] underneath the OSPF process.

If a default route does not exist in a routing table, the **always** optional keyword advertises a default route even if a default route does not exist in the RIB. In addition, the route metric can be changed with the **metric** *metric-value* option, and the metric type can be changed with the **metric-type** *type-value* option.

Figure 8-8 illustrates a common scenario, where R1 has a static default route to a firewall that is connected to the Internet. To provide connectivity to other parts of the network (for example, R2 and R3), R1 advertises a default route into OSPF.

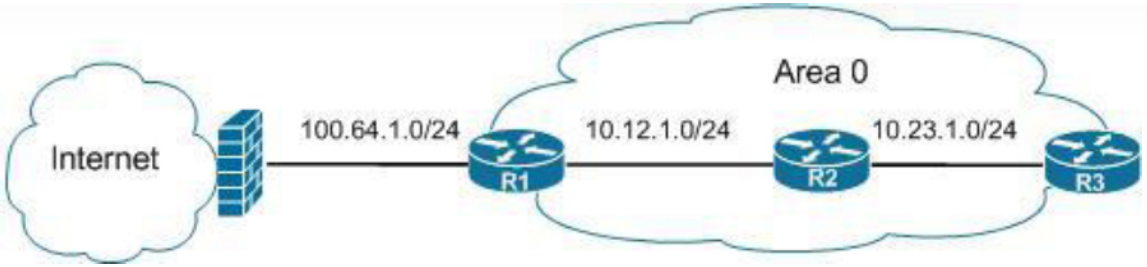


Figure 8-8 Default Route Topology

Example 8-11 provides the relevant configuration on R1. Notice that R1 has a static default route to the firewall (100.64.1.2) to satisfy the requirement of having the default route in the RIB.

Example 8-11 OSPF Default Information Origination Configuration

```
R1
ip route 0.0.0.0 0.0.0.0 100.64.1.2
!
router ospf 1
 network 10.0.0.0 0.255.255.255 area 0
 default-information originate
```

Example 8-12 provides the routing tables of R2 and R3. Notice that OSPF advertises the default route as an external OSPF route.

Example 8-12 Routing Tables for R2 and R3

```
R2# show ip route | begin Gateway
Gateway of last resort is 10.12.1.1 to network 0.0.0.0
```

```
O*E2  0.0.0.0/0 [110/1] via 10.12.1.1, 00:02:56, GigabitEthernet(
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.12.1.0/24 is directly connected, GigabitEthernet0/1
C        10.23.1.0/24 is directly connected, GigabitEthernet0/2

R3# show ip route | begin Gateway
Gateway of last resort is 10.23.1.2 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 10.23.1.2, 00:01:47, GigabitEthernet(
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O        10.12.1.0/24 [110/2] via 10.23.1.2, 00:05:20, GigabitEth
C        10.23.1.0/24 is directly connected, GigabitEthernet0/1
```

COMMON OSPF OPTIMIZATIONS

Almost every network requires tuning based on the equipment, technical requirements, or a variety of other factors. The following sections explain common concepts involved with the tuning of an OSPF network.

Key Topic

Link Costs

Interface cost is an essential component of Dijkstra’s SPF calculation because the shortest path metric is based on the cumulative interface cost (that is, metric) from the router to the destination. OSPF assigns the OSPF link cost (that is, metric) for an interface by using the formula in Figure 8-9.

$$\text{Cost} = \frac{\text{Reference Bandwidth}}{\text{Interface Bandwidth}}$$

Figure 8-9 OSPF Interface Cost Formula

The default reference bandwidth is 100 Mbps. Table 8-8 provides the OSPF cost for common network interface types using the default reference bandwidth.

Table 8-8 OSPF Interface Costs Using Default Settings

Interface Type	OSPF Cost
T1	64
Ethernet	10
FastEthernet	1
GigabitEthernet	1
10 GigabitEthernet	1

Notice in Table 8-8 that there is no differentiation in the link cost associated with a FastEthernet interface and a 10 GigabitEthernet interface. Changing the reference bandwidth to a higher value allows for a differentiation of cost between higher-speed interfaces. Making the value too high could cause issues because low-bandwidth interfaces would not be distinguishable. The OSPF LSA metric field is 16-bits, and the interface cost cannot exceed 65,535.

Under the OSPF process, the command **auto-cost reference-bandwidth** *bandwidth-in-mbps* changes the reference bandwidth for all OSPF interfaces associated with that process. If the reference bandwidth is changed on one router, the reference bandwidth should be changed on all OSPF routers to ensure that SPF uses the same logic to prevent routing loops. It is a best practice to set the same reference bandwidth for all OSPF routers.

The OSPF cost can be set manually with the command **ip ospf cost** *1-65535* underneath the interface. While the interface cost is limited to 65,535 because of LSA field limitations, the path metric can exceed a 16-bit value (65,535) because all the link metrics are calculated locally.

Key Topic

Failure Detection

A secondary function of the OSPF hello packets is to ensure that adjacent OSPF neighbors are still healthy and available. OSPF sends hello packets at set intervals, based on the hello timer. OSPF uses a second timer called the *OSPF dead interval timer*, which defaults to four times the hello timer. Upon receipt of a hello packet from a neighboring router, the OSPF dead timer resets to the initial value and then starts to decrement again.

If a router does not receive a hello before the OSPF dead interval timer reaches 0, the neighbor state is changed to down. The OSPF router immediately sends out the appropriate LSA, reflecting the topology change, and the SPF algorithm processes on all routers within the area.

Hello Timer

The default OSPF hello timer interval varies based on the OSPF network type. OSPF allows modification to the hello timer interval with values between 1 and 65,535 seconds. Changing the hello timer interval modifies the default dead interval, too. The OSPF hello timer is modified with the interface configuration submode command **ip ospf hello-interval** *1-65535*.

Dead Interval Timer

The dead interval timer can be changed to a value between 1 and 65,535 seconds. The OSPF dead interval timer can be change with the command **ip ospf dead-interval** *1-65535* under the interface configuration sub mode.

Note

Always make sure that the dead interval timer setting is greater than the hello timer setting to ensure that the dead interval timer does not reach 0 in between hello packets.

Verifying OSPF Timers

The timers for an OSPF interfaces are shown with the command **show ip ospf interface**, as demonstrated in Example 8-13. Notice the highlighted hello and dead timers.

Example 8-13 OSPF Interface Timers

```
R1# show ip ospf interface | i Timer|line
Loopback0 is up, line protocol is up
GigabitEthernet0/2 is up, line protocol is up
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retrans
GigabitEthernet0/1 is up, line protocol is up
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retrans
```

Note

Hello and dead interval timers must match for OSPF neighbors to become adjacent.

DR Placement

The DR and BDR roles for a broadcast network consume CPU and memory on the host routers in order to maintain states with all the other routers for that segment. Placing the DR and BDR roles on routers with adequate resources is recommended.

The following sections explain the DR election process and how the DR role can be assigned to specific hardware.

Key Topic

Designated Router Elections

The DR/BDR election occurs during OSPF neighborship—specifically during the last phase of 2-Way neighbor state and just before the ExStart state. When a router enters the 2-Way state, it has already received a hello from the neighbor. If the hello packet includes a RID other than 0.0.0.0 for the DR or BDR, the new router assumes that the current routers are the actual DR and BDR.

Any router with OSPF priority of 1 to 255 on its OSPF interface attempts to become the DR. By default, all OSPF interfaces use a priority of 1. The routers place their RID and OSPF priorities in their OSPF hellos for that segment.

Routers then receive and examine OSPF hellos from neighboring routers. If a router identifies itself as being a more favorable router than the OSPF hellos it receives, it continues to send out hellos with its RID and priority listed. If the hello received is more favorable, the router updates its OSPF hello packet to use the more preferable RID in the DR field. OSPF deems a router more preferable if the priority for the interface is the highest for that segment. If the OSPF priority is the same, the higher RID is more favorable.

Once all the routers have agreed on the same DR, all routers for that segment become adjacent with the DR. Then the election for the BDR takes place. The election follows the same logic for the DR election, except that the DR does not add its RID to the BDR field of the hello packet.

The OSPF DR and BDR roles cannot be preempted after the DR/BDR election. Only upon the failure (or process restart of the DR or BDR) does the election start to replace the role that is missing.

Note

To ensure that all routers on a segment have fully initialized, OSPF initiates a wait timer when OSPF hello packets do not contain a DR/BDR router for a segment. The default value for the wait timer is the dead interval timer. Once the wait timer has expired, a router participates in the DR election. The wait timer starts when OSPF first starts on an interface; so that a router can still elect itself as the DR for a segment without other OSPF routers, it waits until the wait timer expires.

The easiest way to determine the interface role is by viewing the OSPF interface with the command **show ip ospf interface brief**. Example 8-14 shows this command executed on R1 and R3 of the sample topology . Notice that R1’s Gi0/2 interface is the DR for the 10.1.1.0/24 network (as no other router is present), and R1’s Gi0/1 interface is DROTHER for the 10.123.4.0/24 segment. R3’s Gi0/1 interface is the BDR for the 10.123.4.0/24 network segment.

Example 8-14 OSPF Interface State

```
R1# show ip ospf interface brief
Interface    PID   Area           IP Address/Mask    Cost  State
Lo0          1     0              192.168.1.1/32     1     LOOP
Gi0/2        1     0              10.1.1.1/24        1     DR
Gi0/1        1     0              10.123.4.1/24      1     DROTH
```

```
R3# show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State
Lo0          1     0               192.168.3.3/32     1     LOOP
Gi0/1        1     0               10.123.4.3/24      1     BDR
Gi0/2        1     0               10.3.3.3/24        1     DR
```

The neighbor’s full adjacency field reflects the number of routers that have become adjacent on that network segment; the neighbors count field is the number of other OSPF routers on that segment. You might assume that all routers will become adjacent with each other, but that would defeat the purpose of using a DR. Only the DR and BDR become adjacent with routers on a network segment.

DR and BDR Placement

In Example 8-14, R4 won the DR election, and R3 won the BDR election because all the OSPF routers had the same OSPF priority, so the next decision point was the higher RID. The RIDs matched the Loopback 0 interface IP addresses, and R4’s loopback address is the highest on that segment; R3’s is the second highest.

Modifying a router’s RID for DR placement is a bad design strategy. A better technique involves modifying the interface priority to a higher value than the existing DR has. In our current topology, the DR role for the segment (10.123.4.0/24) requires that the priority change to a higher value than 1 (the existing DR’s priority) on the desired node. Remember that OSPF does not preempt the DR or BDR roles, and the OSPF process might need to be restarted on the current DR/BDR for the changes to take effect.

The priority can be set manually under the interface configuration with the command **ip ospf priority** *0-255* for IOS nodes. Setting an interface priority to 0 removes that interface from the DR/BDR election immediately. Raising the priority above the default value (1) makes that interface more favorable compared to interfaces with the default value.

Figure 8-10 provides a topology example to illustrate modification of DR/BDR placement in a network segment. R4 should never become the DR/BDR for the 10.123.4.0/24 segment, and R1 should always become the DR for the 10.123.4.0/24 segment.

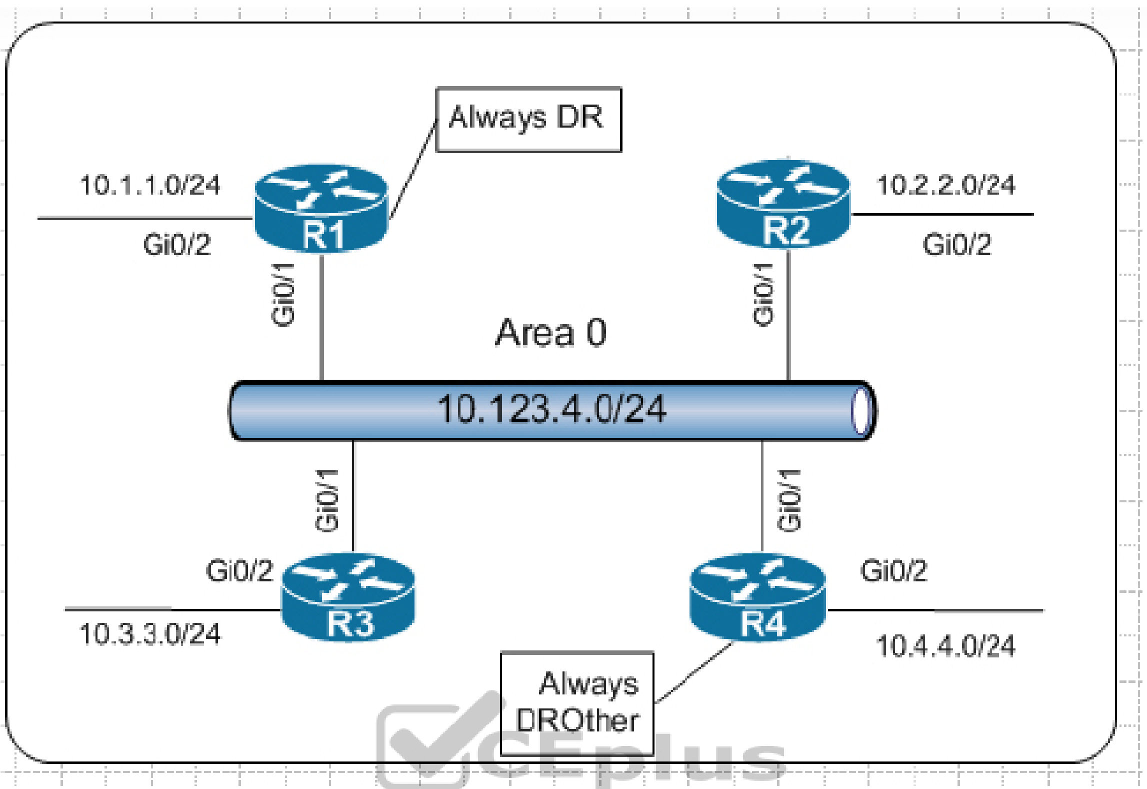


Figure 8-10 OSPF Topology for DR/BDR Placement

To prevent R4 from entering into the DR/BDR election, the OSPF priority changes to 0. R1's interface priority will change to a value higher than 1 to ensure that it always wins the DR election.

Example 8-15 provides the relevant configuration for R1 and R4. No configuration changes have occurred on R2 and R3.

Example 8-15 Configuring OSPF with DR Manipulation

```
R1# configuration terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface GigabitEthernet 0/1
R1(config-if)# ip ospf priority 100

R4# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)# interface gigabitEthernet 0/1
R4(config-if)# ip ospf priority 0
21:52:54.479: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Gigab
FULL, Loading Done
```

Notice that upon configuring the interface priority to 0 on R4, the neighbor state with R1 changed. When the interface DR priority changed to zero, R4 removed itself as DR, R3 was promoted from the BDR to the DR, and then R1 was elected to the BDR. Because R1 is now a BDR, any half-open neighborships were allowed to progress with establishing a complete neighborship with other routers.

Example 8-16 checks the status of the topology. R1 shows a priority of 100, and R4 shows a priority of 0. However, R1 is in the BDR position and not the DR role, as intended.

Example 8-16 Verifying DR Manipulation

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         In
```

```
192.168.1.1     100   FULL/BDR        00:00:31    10.123.4.1    Gi
192.168.3.3       1   FULL/DR         00:00:33    10.123.4.3    Gi
192.168.4.4       0   2WAY/DROTHER    00:00:31    10.123.4.4    Gi
```

This example shows normal operation because the DR/BDR role does not support preemption. If all routers started as the same type, R1 would be the DR because of the wait timer in the initial OSPF DR election process. To complete the migration of the DR to R1, the OSPF process must be restarted on R3, as demonstrated in Example 8-17. After the process is restarted, the OSPF neighborship is checked again, and now R1 is the DR for the 10.123.4.0/24 network segment.

Example 8-17 Clearing the DR OSPF Process

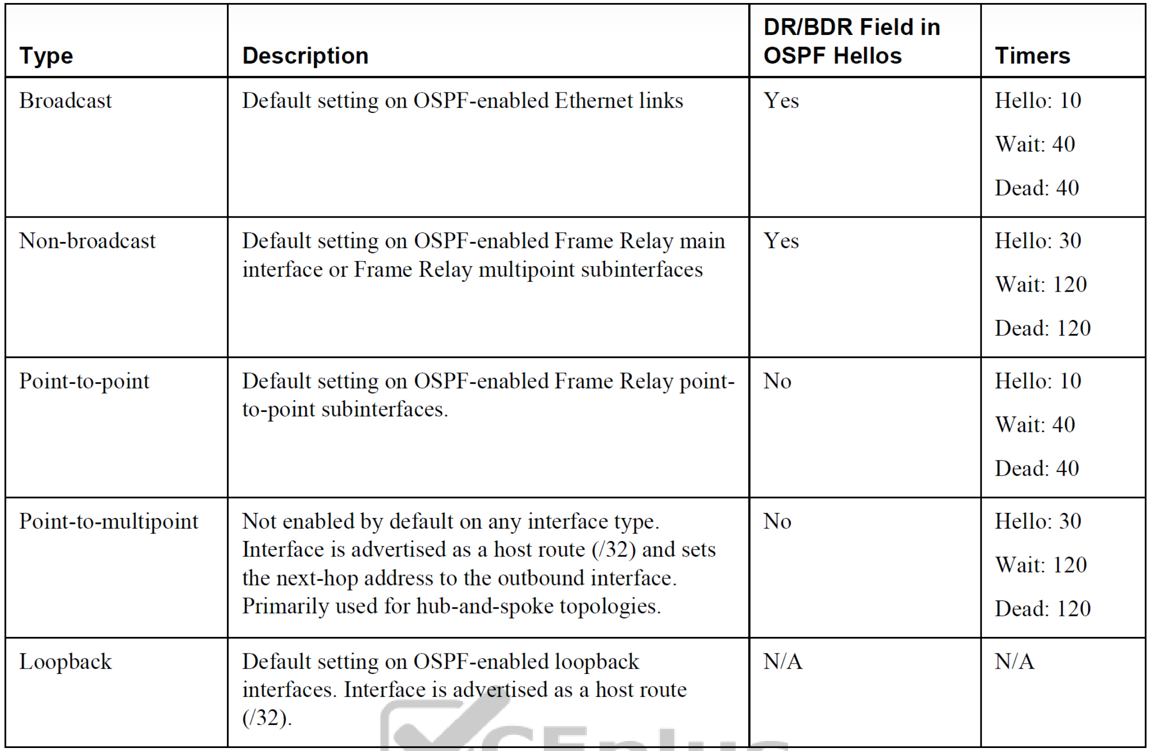
```
R3# clear ip ospf process
Reset ALL OSPF processes? [no]: y
21:55:09.054: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on Gigab
  from FULL to DOWN, Neighbor Down: Interface down or detached
21:55:09.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.2 on Gigab
  from FULL to DOWN, Neighbor Down: Interface down or detached
21:55:09.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.4.4 on Gigab
  from FULL to DOWN, Neighbor Down: Interface down or detached

R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address       In
192.168.1.1     100   FULL/DR         00:00:37    10.123.4.1    Gi
192.168.2.2       1   FULL/DROTHER    00:00:34    10.123.4.2    Gi
192.168.4.4       0   FULL/DROTHER    00:00:35    10.123.4.4    Gi
```

OSPF Network Types

Different media can provide different characteristics or might limit the number of nodes allowed on a segment. Frame Relay and Ethernet are a common multi-access media, and because they support more than two nodes on a network segment, the need for a DR exists. Other network circuits, such as serial links (with HDLC or PPP encapsulation), do not require a DR, and having one would just waste router CPU cycles.

The default OSPF network type is set based on the media used for the connection and can be changed independently of the actual media type used. Cisco's implementation of OSPF considers the various media and provides five OSPF network types, as listed in Table 8-9.

Table 8-9 OSPF Network Types



Type	Description	DR/BDR Field in OSPF Hellos	Timers
Broadcast	Default setting on OSPF-enabled Ethernet links	Yes	Hello: 10 Wait: 40 Dead: 40
Non-broadcast	Default setting on OSPF-enabled Frame Relay main interface or Frame Relay multipoint subinterfaces	Yes	Hello: 30 Wait: 120 Dead: 120
Point-to-point	Default setting on OSPF-enabled Frame Relay point-to-point subinterfaces.	No	Hello: 10 Wait: 40 Dead: 40
Point-to-multipoint	Not enabled by default on any interface type. Interface is advertised as a host route (/32) and sets the next-hop address to the outbound interface. Primarily used for hub-and-spoke topologies.	No	Hello: 30 Wait: 120 Dead: 120
Loopback	Default setting on OSPF-enabled loopback interfaces. Interface is advertised as a host route (/32).	N/A	N/A

The non-broadcast or point-to-multipoint network types are beyond the scope of the Enterprise Core exam, but the other OSPF network types are explained in the following sections.

Broadcast

Broadcast media such as Ethernet are better defined as broadcast multi-access to distinguish them from non-broadcast multi-access (NBMA) networks. Broadcast networks are multi-access in that they are capable of connecting more than two devices, and broadcasts sent out one interface are capable of reaching all interfaces attached to that segment.

The OSPF network type is set to broadcast by default for Ethernet interfaces. A DR is required for this OSPF network type because of the possibility that multiple nodes can exist on a segment, and LSA flooding needs to be controlled. The hello timer defaults to 10 seconds, as defined in RFC 2328.

The interface parameter command **ip ospf network broadcast** overrides the automatically configured setting and statically sets an interface as an OSPF broadcast network type.

Point-to-Point Networks

A network circuit that allows only two devices to communicate is considered a point-to-point (P2P) network. Because of the nature of the medium, point-to-point networks do not use Address Resolution Protocol (ARP), and broadcast traffic does not become the limiting factor.

The OSPF network type is set to point-to-point by default for serial interfaces (HDLC or PPP encapsulation), generic routing encapsulation (GRE) tunnels, and point-to-point Frame Relay subinterfaces. Only two nodes can exist on this type of network medium, so OSPF does not waste CPU cycles on DR functionality. The hello timer is set to 10 seconds on OSPF point-to-point network types.

Figure 8-11 shows a serial connection between R1 and R2.

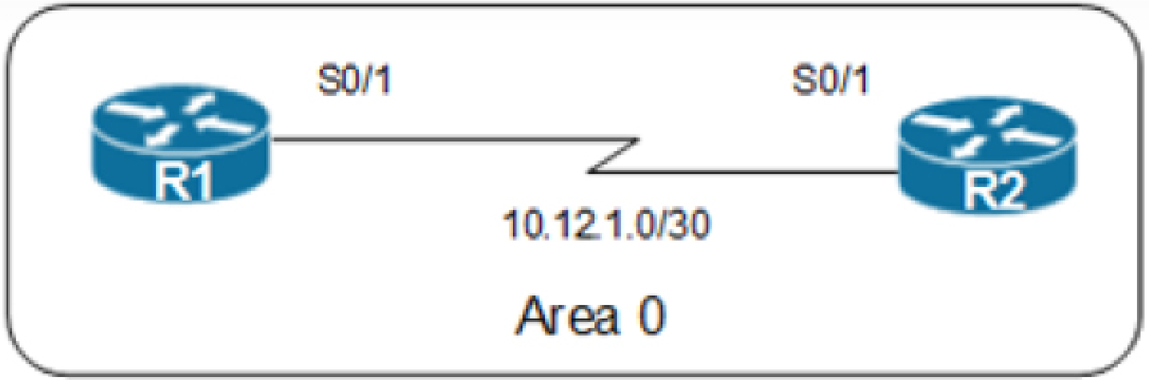


Figure 8-11 OSPF Topology with Serial Interfaces

Example 8-18 shows the relevant serial interface and OSPF configuration for R1 and R2. Notice that there are not any special commands placed in the configuration.

Example 8-18 Configuring R1 and R2 Serial and OSPF

```
R1
interface serial 0/1
  ip address 10.12.1.1 255.255.255.252
!
router ospf 1
   router-id 192.168.1.1
   network 0.0.0.0 255.255.255.255 area 0

R2
interface serial 0/1
  ip address 10.12.1.2 255.255.255.252
!
router ospf 1
```

```
   router-id 192.168.2.2
   network 0.0.0.0 255.255.255.255 area 0
```

Example 8-19 verifies that the OSPF network type is set to POINT_TO_POINT, indicating the OSPF point-to-point network type.

Example 8-19 Verifying the OSPF P2P Interfaces

```
R1# show ip ospf interface s0/1 | include Type
  Process ID 1, Router ID 192.168.1.1, Network Type POINT_TO_POIN

R2# show ip ospf interface s0/1 | include Type
  Process ID 1, Router ID 192.168.2.2, Network Type POINT_TO_POIN
```

Example 8-20 shows that point-to-point OSPF network types do not use a DR. Notice the hyphen (-) in the State field.

Example 8-20 Verifying OSPF Neighbors on P2P Interfaces

```
R1# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address
192.168.2.2       0   FULL/  -        00:00:36    10.12.1.2
```

Interfaces using an OSPF P2P network type form an OSPF adjacency more quickly because the DR election is bypassed, and there is no wait timer. Ethernet interfaces that are directly connected with only two OSPF speakers in the subnet could be changed to the OSPF point-to-point network type to form adjacencies more quickly and to simplify the SPF computation. The interface parameter command **ip ospf network point-to-point** sets an interface as an OSPF point-to-point network type.

Loopback Networks

The OSPF network type loopback is enabled by default for loopback interfaces and can be used only on loopback interfaces. The OSPF loopback network type states that the IP address is always advertised with a /32 prefix length, even if the IP address configured on the loopback interface does not have a /32 prefix length. It is possible to demonstrate this behavior by reusing Figure 8-11 and advertising a Loopback 0 interface. Example 8-21 provides the updated configuration. Notice that the network type for R2's loopback interface is set to the OSPF point-to-point network type.

Example 8-21 OSPF Loopback Network Type

```
R1
interface Loopback0
    ip address 192.168.1.1 255.255.255.0
interface Serial 0/1
    ip address 10.12.1.1 255.255.255.252
!
router ospf 1
   router-id 192.168.1.1
   network 0.0.0.0 255.255.255.255 area 0

R2
interface Loopback0
    ip address 192.168.2.2 255.255.255.0
    ip ospf network point-to-point
interface Serial 0/0
    ip address 10.12.1.2 255.255.255.252
!
router ospf 1
   router-id 192.168.2.2
   network 0.0.0.0 255.255.255.255 area 0
```

The network types for the R1 and R2 loopback interfaces are checked to verify that they changed and are different, as demonstrated in Example 8-22.

Example 8-22 Displaying OSPF Network Type for Loopback Interfaces

```
R1# show ip ospf interface Loopback 0 | include Type
  Process ID 1, Router ID 192.168.1.1, Network Type LOOPBACK, Co

R2# show ip ospf interface Loopback 0 | include Type
Process ID 1, Router ID 192.168.2.2, Network Type POINT_TO_POINT,
```

Example 8-23 shows the R1 and R2 routing tables. Notice that R1's loopback address is a /32 network, and R2's loopback is a /24 network. Both loopbacks were configured with a /24 network; however, because R1’s Lo0 is an OSPF network type of loopback, it is advertised as a /32 network.

Example 8-23 OSPF Route Table for OSPF Loopback Network Types

```
R1# show ip route ospf
! Output omitted for brevity
Gateway of last resort is not set

O        192.168.2.0/24 [110/65] via 10.12.1.2, 00:02:49, Serial0

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

      192.168.1.0/32 is subnetted, 1 subnets
O        192.168.1.1 [110/65] via 10.12.1.1, 00:37:15, Serial0/0
```

EXAM PREPARATION TASKS

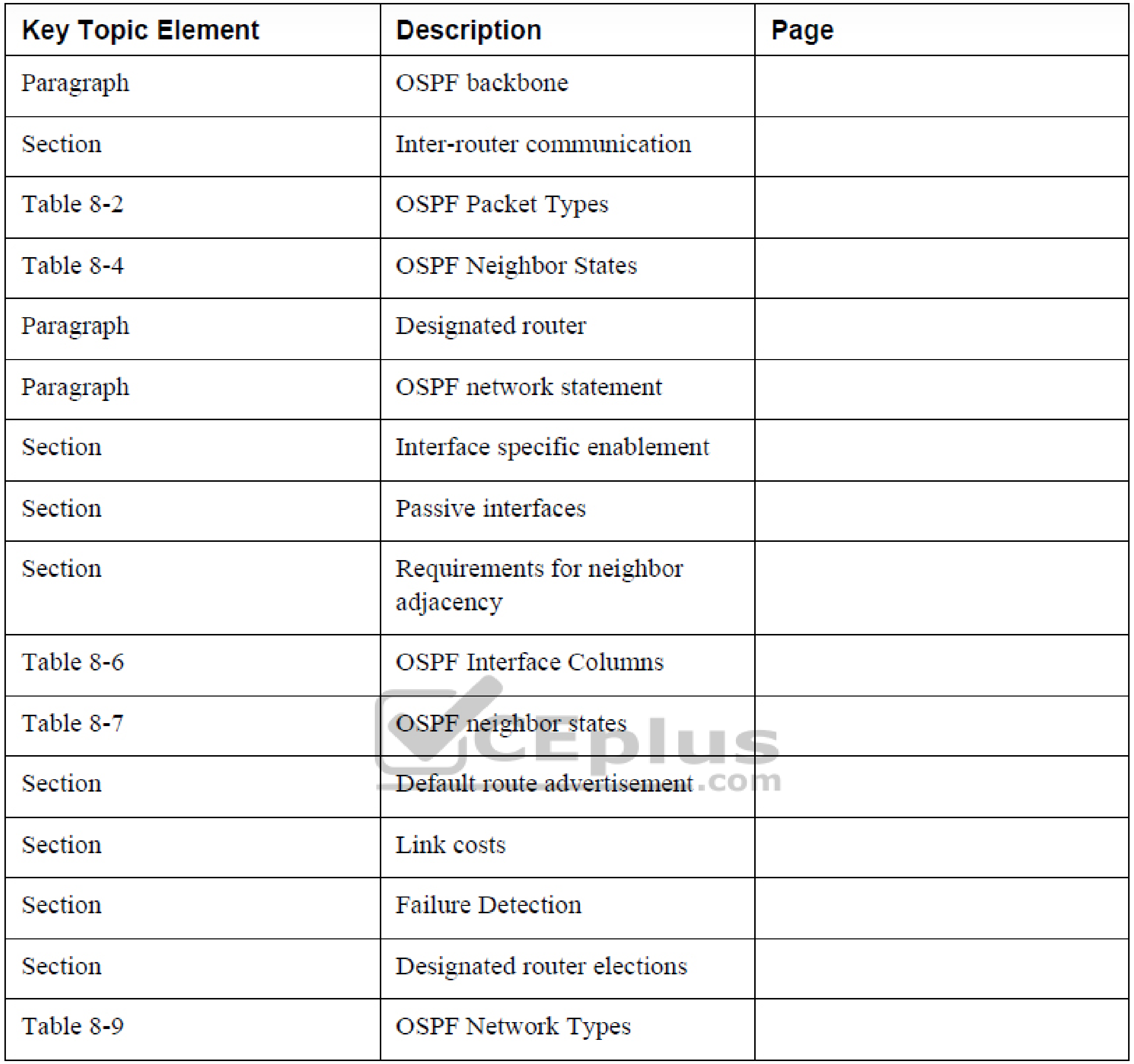
As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 8-10 lists these key topics and the page number on which each is found.

Key Topic

Table 8-10 Key Topics for Chapter 8



Key Topic Element	Description	Page
Paragraph	OSPF backbone	
Section	Inter-router communication	
Table 8-2	OSPF Packet Types	
Table 8-4	OSPF Neighbor States	
Paragraph	Designated router	
Paragraph	OSPF network statement	
Section	Interface specific enablement	
Section	Passive interfaces	
Section	Requirements for neighbor adjacency	
Table 8-6	OSPF Interface Columns	
Table 8-7	OSPF neighbor states	
Section	Default route advertisement	
Section	Link costs	
Section	Failure Detection	
Section	Designated router elections	
Table 8-9	OSPF Network Types	

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix C, “Memory Tables” (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix C, “Memory Tables Answer Key,” also on the companion website, includes completed tables and lists you can use to check your work.

DEFINE KEY TERMS

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

backup designated router (BDR)

dead interval

designated router (DR)

hello interval

hello packets

interface priority

passive interface

router ID (RID)

shortest path tree (SPT)

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 8-11 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 8-11 Command Reference

Task	Command Syntax
Initialize the OSPF process	**router ospf** *process-id*
Enable OSPF on network interfaces matching a specified network range for a specific OSPF area	**network** *ip-address wildcard-mask* **area** *area-id*
Enable OSPF on an explicit specific network interface for a specific OSPF area	**ip ospf** *process-id* **area** *area-id*
Configure a specific interface as passive	**passive** *interface-id*
Configure all interfaces as passive	**passive interface default**
Advertise a default route into OSPF	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*]
Modify the OSPF reference bandwidth for dynamic interface metric costing	**auto-cost reference-bandwidth** *bandwidth-in-mbps*
Statically set the OSPF metric for an interface	**ip ospf cost** *1-65535*
Configure the OSPF priority for a DR/BDR election	**ip ospf priority** *0-255*
Statically configure an interface as a broadcast OSPF network type	**ip ospf network broadcast**
Statically configure an interface as a point-to-point OSPF network type	**ip ospf network point-to-point**
Restart the OSPF process	**clear ip ospf process**
Display the OSPF interfaces on a router	**show ip ospf interface** [**brief** | *interface-id*]
Display the OSPF neighbors and their current states	**show ip ospf neighbor** [**detail**]
Display the OSPF routes that are installed in the RIB	**show ip route ospf**

REFERENCES IN THIS CHAPTER

Chapter 9. Advanced OSPF

This chapter covers the following subjects:

- **Areas:** This section describes the benefits and functions of areas within an OSPF routing domain.

- **Link-State Announcements:** This section explains how OSPF stores, communicates, and builds a topology from the link-state announcements (LSAs).

- **Discontiguous Networks:** This section demonstrates a discontiguous network and explains why such a network cannot distribute routes to all areas properly.

- **OSPF Path Selection:** This section explains how OSPF makes path selection choices for routes learned within the OSPF routing domain.

- **Summarization of Routes:** This section explains how network summarization works with OSPF.

- **Route Filtering:** This section explains how OSPF routes can be filtered on a router.

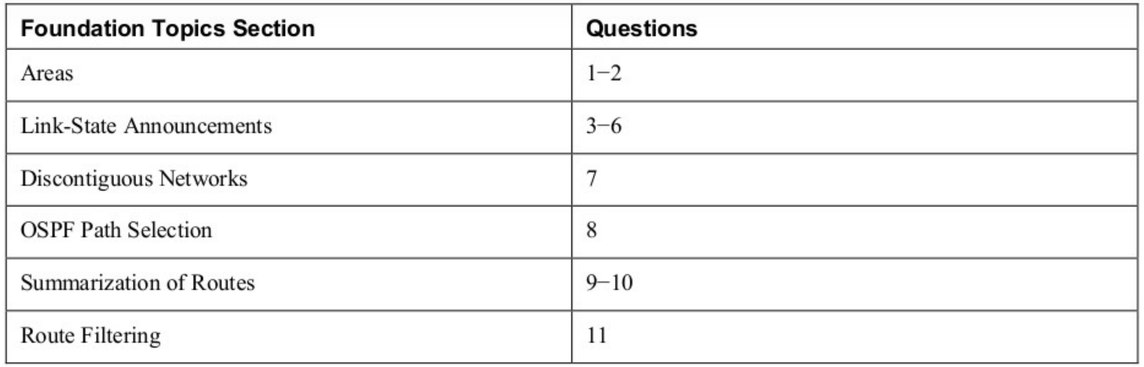
The Open Shortest Path First (OSPF) protocol scales well with proper network planning. IP addressing schemes, area segmentation, address summarization, and hardware capabilities for each area should all be taken into consideration for a network design.

This chapter expands on Chapter 8, “OSPF,” and explains the functions and features found in larger enterprise networks. By the end of this chapter, you should have a solid understanding of the route advertisement within a multi-area OSPF domain, path selection, and techniques to optimize an OSPF environment.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the “Exam Preparation Tasks” section. Table 9-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

Table 9-1 Do I Know This Already?” Foundation Topics Section-to-Question Mapping



Foundation Topics Section	Questions
Areas	1–2
Link-State Announcements	3–6
Discontiguous Networks	7
OSPF Path Selection	8
Summarization of Routes	9–10
Route Filtering	11

1. True or false: A router with an interface associated with Area 1 and Area 2 will be able to inject routes learned from one area into another area.

a. True

b. False

2. True or false: A member router contains a complete copy of the LSDBs for every area in the routing domain.

a. True

b. False

3. How many OSPF link-state announcement (LSA) types are used for routing traditional IPv4 packets?

a. Two

b. Three

c. Five

d. Six

e. Seven

4. What is the LSA age field in the LSDB used for?

a. For version control—to ensure that the most recent LSA is present

b. To age out old LSAs by removing an LSA when its age reaches zero

c. For troubleshooting—to identify exactly when the LSA was advertised

d. To age out old LSAs by removing an LSA when it reaches 3600 seconds

5. Which LSA type exists in all OSPF areas?

a. Network

b. Summary

c. Router

d. AS external

6. True or false: When an ABR receives a network LSA, the ABR forwards the network LSA to the other connected areas.

a. True

b. False

7. When a type 3 LSA is received in a nonbackbone area, what does the ABR do?

a. Discards the type 3 LSA and does not process it

b. Installs the type 3 LSA for only the area where it was received

c. Advertises the type 3 LSA to the backbone area and displays an error

d. Advertises the type 3 LSA to the backbone area

8. True or false: OSPF uses the shortest total path metric to identify the best path for every internal OSPF route (intra-area and interarea).

a. True

b. False

9. True or false: Breaking a large OSPF topology into smaller OSPF areas can be considered a form of summarization.

a. True

b. False

10. How is the process of summarizing routes on an OSPF router accomplished?

a. By using the interface configuration command **summary-address** *network prefix-length*

b. By using the OSPF process configuration command **summary-address** *network prefix-length*

c. By using the OSPF process configuration command **area** *area-id* **range** *network subnet-mask*

d. By using the interface configuration command **area** *area-id* **summary-address** *network subnet-mask*

11. OSPF supports filtering of routes using which of the following techniques? (Choose two.)

a. Summarization, using the no-advertise option

b. LSA filtering, which prevents type 1 LSAs from being advertised through a member router

c. Area filtering, which prevents type 1 LSAs from being generated into a type 3 LSA

d. Injection of an OSPF discard route on the router that filtering should apply

Answers to the "Do I Know This Already?" quiz:

1. B

2. B

3. D

4. D

5. C

6. B

7. B

8. B

9. A

10. C

11. A, C

FOUNDATION TOPICS

AREAS

An OSPF area is a logical grouping of routers or, more specifically, a logical grouping of router interfaces. Area membership is set at the interface level, and the area ID is included in the OSPF hello packet. An interface can belong to only one area. All routers within the same OSPF area maintain an identical copy of the link-state database (LSDB).

An OSPF area grows in size as network links and the number of routers increase in the area. While using a single area simplifies the topology, there are trade-offs:

• Full shortest path first (SPF) tree calculation runs when a link flaps within the area.

• The LSDB increases in size and becomes unmanageable.

• The LSDB for the area grows, consuming more memory, and taking longer during the SPF computation process.

• No summarization of route information occurs.

Proper design addresses each of these issues by segmenting the routers into multiple OSPF areas, thereby keeping the LSDB to a manageable size. Sizing and design of OSPF networks should account for the hardware constraints of the smallest router in that area.

If a router has interfaces in multiple areas, the router has multiple LSDBs (one for each area). The internal topology of one area is invisible from outside that area. If a topology change occurs (such as a link flap or an additional network being added) within an area, all routers in the same OSPF area calculate the SPF tree again. Routers outside that area do not calculate the full SPF tree again but perform a partial SPF calculation if the metrics have changed or a prefix is removed.

In essence, an OSPF area hides the topology from another area but enables the networks to be visible in other areas within the OSPF domain. Segmenting the OSPF domain into multiple areas reduces the size of the LSDB for each area, making SPF tree calculations faster, and decreasing LSDB flooding between routers when a link flaps.

Just because a router connects to multiple OSPF areas does not mean the routes from one area will be injected into another area. Figure 9-1 shows router R1 connected to Area 1 and Area 2. Routes from Area 1 will not advertise into Area 2 and vice versa.

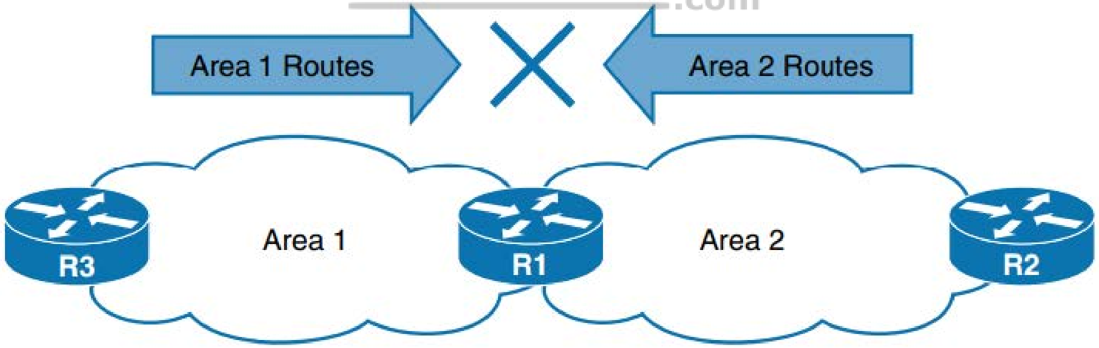


Figure 9-1 Failed Route Advertisement Between Areas

Key Topic

Area 0 is a special area called *the backbone*. By design, all areas must connect to Area 0 because OSPF expects all areas to inject routing information into the backbone, and Area 0 advertises the routes into other areas. The backbone design is crucial to preventing routing loops.

Key Topic

Area border routers (ABRs) are OSPF routers connected to Area 0 and another OSPF area, per Cisco definition and according to RFC 3509. ABRs are responsible for advertising routes from one area and injecting them into a different OSPF area. Every ABR needs to participate in Area 0; otherwise, routes will not advertise into another area. ABRs compute an SPF tree for every area that they participate in.

Figure 9-2 shows that R1 is connected to Area 0, Area 1, and Area 2. R1 is a proper ABR because it now participates in Area 0. The following occurs on R1:

- Routes from Area 1 advertise into Area 0.

- Routes from Area 2 advertise into Area 0.

- Routes from Area 0 advertise into Area 1 and 2. This includes the local Area 0 routes, in addition to the routes that were advertised into Area 0 from Area 1 and Area 2.

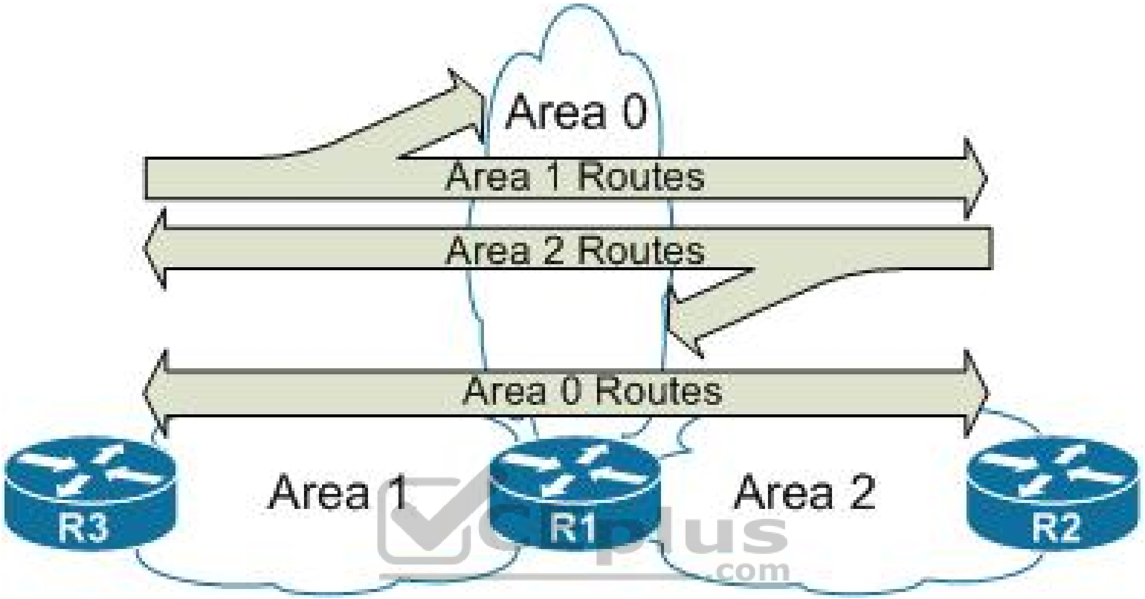


Figure 9-2 Successful Route Advertisement Between Areas

Figure 9-3 shows a larger-scale OSPF multi-area topology that is used throughout this chapter to describe various OSPF concepts.

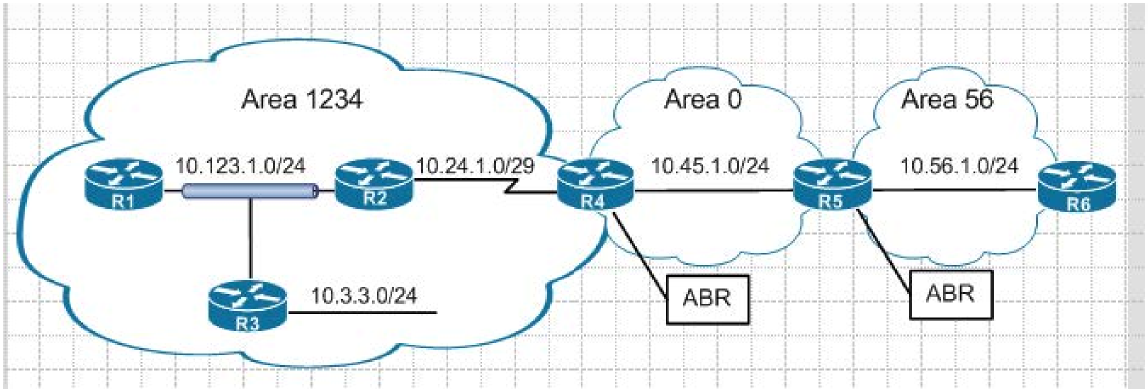


Figure 9-3 OSPF Multi-Area Topology

In the topology:

- R1, R2, R3, and R4 belong to Area 1234.

- R4 and R5 belong to Area 0.

- R5 and R6 belong to Area 56.

- R4 and R5 are ABRs.

- Area 1234 connects to Area 0, and Area 56 connects to Area 0.

- Routers in Area 1234 can see routes from routers in Area 0 and Area 56 and vice versa.

Example 9-1 shows the OSPF configuration for the ABRs R4 and R5. Notice that multiple areas in the configuration have Area 0 as one of the areas.

Example 9-1 Sample Multi-Area OSPF Configuration

```
R4
router ospf 1
 router-id 192.168.4.4
 network 10.24.1.0 0.0.0.255 area 1234
 network 10.45.1.0 0.0.0.255 area 0

R5
router ospf 1
 router-id 192.168.5.5
 network 10.45.1.0 0.0.0.255 area 0
 network 10.56.1.0 0.0.0.255 area 56
```

Example 9-2 verifies that interfaces on R4 belong to Area 1234 and Area 0 and that interfaces on R5 belong to Area 0 and Area 56.

Example 9-2 Verifying Interfaces for ABRs

```
R4# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Gi0/0         1     0               10.45.1.4/24       1     DR
Se1/0         1     1234            10.24.1.4/29       64    P2P

R5# show ip ospf interface brief
Interface     PID   Area            IP Address/Mask    Cost  State
Gi0/0         1     0               10.45.1.5/24       1     DR
Gi0/1         1     56              10.56.1.5/24       1     BDR
```

Key Topic

Area ID

The area ID is a 32-bit field and can be formatted in simple decimal (0 through 4,294,967,295) or dotted decimal (0.0.0.0 through 255.255.255.255). During router configuration, the area can use decimal format on one router and dotted-decimal format on a different router, and the routers can still form an adjacency. OSPF advertises the area ID in dotted-decimal format in the OSPF hello packet.

OSPF Route Types

Network routes that are learned from other OSPF routers within the same area are known as *intra-area routes*. In Figure 9-3, the network link between R2 and R4 (10.24.1.0/29) is an intra-area route to R1. The IP routing table displays OSPF intra-area routes with an *O*.

Network routes that are learned from other OSPF routers from a different area using an ABR are known as *interarea routes*. In Figure 9-3, the network link between R4 and R5 (10.45.1.0/24) is an interarea route to R1. The IP routing table displays OSPF interarea routers with *O IA*.

Example 9-3 provides the routing table for R1 from Figure 9-3. Notice that R1's OSPF routing table shows routes from within Area 1234 as intra-area (*O* routes) and routes from Area 0 and Area 56 as interarea (*O IA* routes).

Example 9-3 OSPF Routing Tables for Sample Multi-Area OSPF Topology

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

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
! The following two routes are OSPF intra-area routes as they al
! Area 1234
O        10.3.3.0/24 [110/20] via 10.123.1.3, 00:12:07, GigabitEt
O        10.24.1.0/29 [110/74] via 10.123.1.2, 00:12:07, GigabitE
! The following two routes are OSPF interarea routes as they all
! outside of Area 1234
O IA     10.45.1.0/24 [110/84] via 10.123.1.2, 00:12:07, GigabitE
O IA     10.56.1.0/24 [110/94] via 10.123.1.2, 00:12:07, GigabitE
C        10.123.1.0/24 is directly connected, GigabitEthernet0/0
```

Example 9-4 provides the routing table for R4 from Figure 9-3. Notice that R4's routing table shows the routes from within Area 1234 and Area 0 as intra-area and routes from Area 56 as interarea because R4 does not connect to Area 56.

Notice that the metric for the 10.123.1.0/24 and 10.3.3.0/24 networks has drastically increased compared to the metric for the 10.56.1.0/24 network. This is because it must cross the slow serial link, which has an interface cost of 64.

Example 9-4 OSPF Routing Table for ABR R4

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

      10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
O        10.3.3.0/24 [110/66] via 10.24.1.2, 00:03:45, Serial1/0
C        10.24.1.0/29 is directly connected, Serial1/0
C        10.45.1.0/24 is directly connected, GigabitEthernet0/0
O IA     10.56.1.0/24 [110/2] via 10.45.1.5, 00:04:56, GigabitEth
O        10.123.1.0/24 [110/65] via 10.24.1.2, 00:13:19, Serial1/
```

Example 9-5 provides the routing tables with filtering for OSPF for R5 and R6 from Figure 9-3. R5 and R6 only contain interarea routes in the OSPF routing table because intra-area routes are directly connected.

Example 9-5 OSPF Routing Tables for R5 and R6

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

      10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
O IA     10.3.3.0/24 [110/67] via 10.45.1.4, 00:04:13, GigabitEth
O IA     10.24.1.0/29 [110/65] via 10.45.1.4, 00:04:13, GigabitEt
O IA     10.123.1.0/24 [110/66] via 10.45.1.4, 00:04:13, GigabitE

R6# show ip route ospf | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O IA     10.3.3.0/24 [110/68] via 10.56.1.5, 00:07:04, GigabitEth
```

```
O IA     10.24.1.0/24 [110/66] via 10.56.1.5, 00:08:19, GigabitEt
O IA     10.45.1.0/24 [110/2] via 10.56.1.5, 00:08:18, GigabitEth
O IA     10.123.1.0/24 [110/67] via 10.56.1.5, 00:08:19, GigabitE
```

External routes are routes learned from outside the OSPF domain but injected into an OSPF domain through redistribution. External OSPF routes can come from a different OSPF domain or from a different routing protocol. External OSPF routes are beyond the scope of the CCNP and CCIE Enterprise Core ENCOR 300-401 exam and are not covered in this book.

Key Topic

LINK-STATE ANNOUNCEMENTS

When OSPF neighbors become adjacent, the LSDBs synchronize between the OSPF routers. As an OSPF router adds or removes a directly connected network link to or from its database, the router floods the link-state advertisement (LSA) out all active OSPF interfaces. The OSPF LSA contains a complete list of networks advertised from that router.

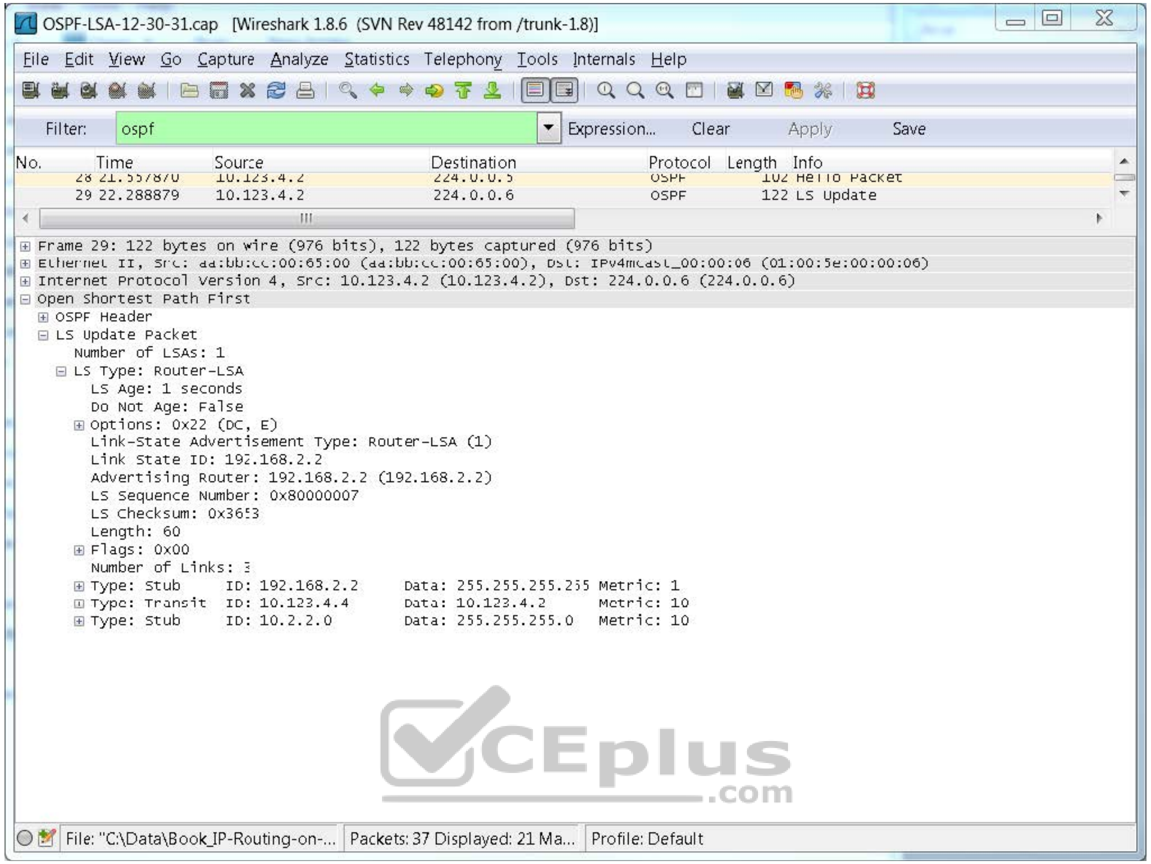
OSPF uses six LSA types for IPv4 routing:

- **Type 1, router LSA:** Advertises the LSAs that originate within an area

- **Type 2, network LSA:** Advertises a multi-access network segment attached to a DR
- **Type 3, summary LSA:** Advertises network prefixes that originated from a different area
- **Type 4, ASBR summary LSA:** Advertises a summary LSA for a specific ASBR
- **Type 5, AS external LSA:** Advertises LSAs for routes that have been redistributed
- **Type 7, NSSA external LSA:** Advertises redistributed routes in NSSAs

LSA types 1, 2, and 3, which are used for building the SPF tree for intra-area and interarea routes, are explained in this section.

Figure 9-4 shows a packet capture of an OSPF update LSA and outlines the important components of the LSA: the LSA type, LSA age, sequence number, and advertising router. Because this is a type 1 LSA, the link IDs add relevance as they list the attached networks and the associated OSPF cost for each interface.


OSPF-LSA-12-30-31.cap [Wireshark 1.8.6 (SVN Rev 48142 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: ospf
Expression... Clear Apply Save

```
No.   Time        Source         Destination   Protocol Length Info
   28 21.557870   10.123.4.2     224.0.0.5     OSPF        102 Hello Packet
   29 22.288879   10.123.4.2     224.0.0.6     OSPF        122 LS Update

⊞ Frame 29: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
⊞ Ethernet II, Src: aa:bb:cc:00:65:00 (aa:bb:cc:00:65:00), Dst: IPv4mcast_00:00:06 (01:00:5e:00:00:06)
⊞ Internet Protocol Version 4, Src: 10.123.4.2 (10.123.4.2), Dst: 224.0.0.6 (224.0.0.6)
⊟ Open Shortest Path First
  ⊞ OSPF Header
  ⊟ LS Update Packet
      Number of LSAs: 1
    ⊟ LS Type: Router-LSA
        LS Age: 1 seconds
        Do Not Age: False
      ⊞ Options: 0x22 (DC, E)
        Link-State Advertisement Type: Router-LSA (1)
        Link State ID: 192.168.2.2
        Advertising Router: 192.168.2.2 (192.168.2.2)
        LS Sequence Number: 0x80000007
        LS Checksum: 0x3653
        Length: 60
      ⊞ Flags: 0x00
        Number of Links: 3
      ⊞ Type: Stub      ID: 192.168.2.2     Data: 255.255.255.255 Metric: 1
      ⊞ Type: Transit   ID: 10.123.4.4      Data: 10.123.4.2      Metric: 10
      ⊞ Type: Stub      ID: 10.2.2.0        Data: 255.255.255.0   Metric: 10
```

File: "C:\Data\Book_IP-Routing-on-...
Packets: 37 Displayed: 21 Ma...
Profile: Default

Figure 9-4 Packet Capture of an LSA Update for the Second Interface

LSA Sequences

OSPF uses the sequence number to overcome problems caused by delays in LSA propagation in a network. The LSA sequence number is a 32-bit number for controlling versioning. When the originating router sends out LSAs, the LSA sequence number is incremented. If a router receives an LSA sequence that is greater than the one in the LSDB, it processes the LSA. If the LSA sequence number is lower than the one in the LSDB, the router deems the LSA old and discards the LSA.

LSA Age and Flooding

Every OSPF LSA includes an age that is entered into the local LSDB and that will increment by 1 every second. When a router's OSPF LSA age exceeds 1800 seconds (30 minutes) for its networks, the originating router advertises a new LSA with the LSA age set to 0. As each router forwards the LSA, the LSA age is incremented with a calculated (minimal) delay that reflects the link. If the LSA age reaches 3600, the LSA is deemed invalid and is purged from the LSDB. The repetitive flooding of LSAs is a secondary safety mechanism to ensure that all routers maintain a consistent LSDB within an area.

LSA Types

All routers within an OSPF area have an identical set of LSAs for that area. The ABRs maintain a separate set of LSAs for each OSPF area. Most LSAs in one area will be different from the LSAs in another area. Generic router LSA output is shown with the command **show ip ospf database**.

LSA Type 1: Router Link

Every OSPF router advertises a type 1 LSA. Type 1 LSAs are the essential building blocks within the LSDB. A type 1 LSA entry exists for each OSPF-enabled link (that is, every interface and its attached networks). Figure 9-5 shows that in this example, the type 1 LSAs are not advertised outside Area 1234, which means the underlying topology in an area is invisible to other areas.

Note

Type 1 LSAs for an area are shown with the command **show ip ospf database router**.

Key Topic

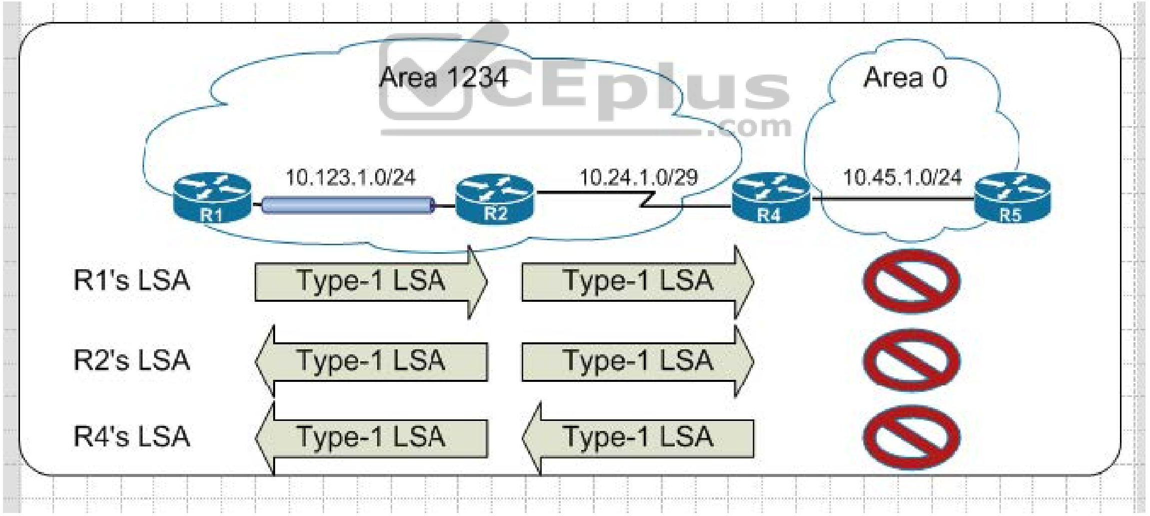


Figure 9-5 Type 1 LSA Flooding in an Area

Figure 9-6 is a reference subsection of Area 1234 taken from the original Figure 9-3.

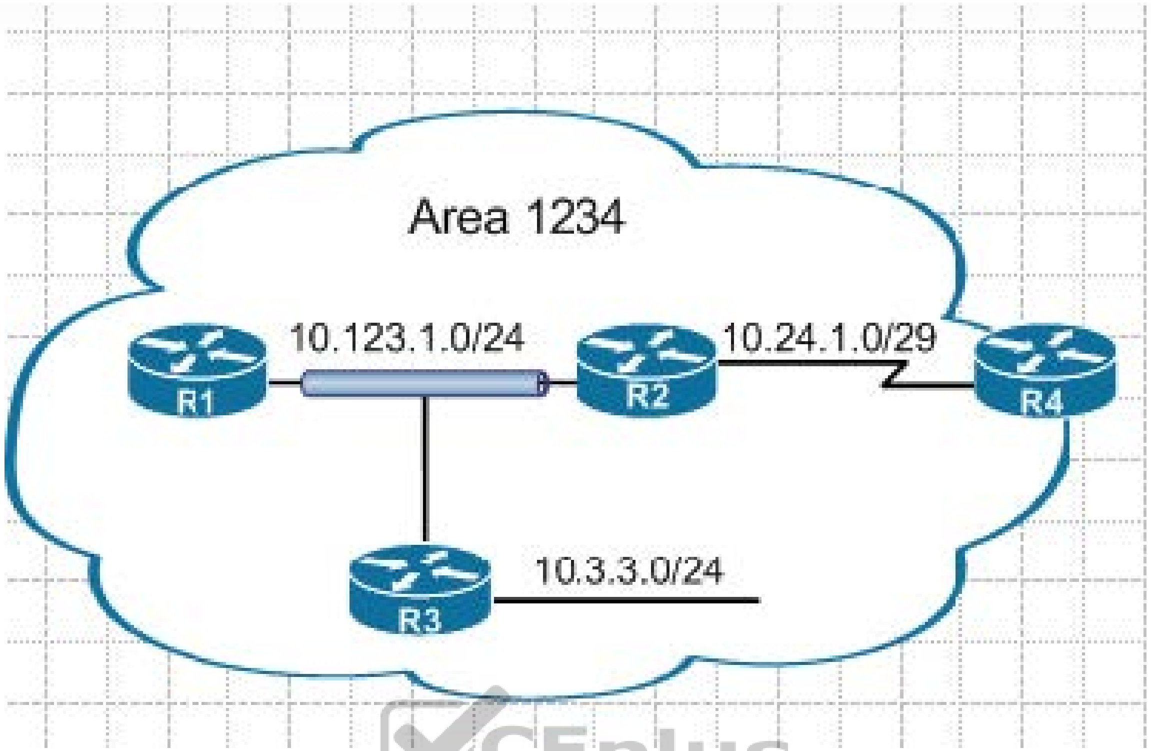


Figure 9-6 Type 1 LSA Flooding Within an Area

The initial fields of each type 1 LSA indicate the RID for the LSA's advertising router, age, sequence, link count, and link ID. Every OSPF-enabled interface is listed under the number of links for each router. Each network link on a router contains the link type, correlating information for neighbor router identification, and interface metric.

The correlating information for neighbor router identification is often the neighbor RID, with the exception of multi-access network segments that contain designated routers (DRs). In those scenarios, the interface address of the DR identifies the neighbor router.

If we correlate just type 1 LSAs from the sample topology of Figure 9-6, then Figure 9-7 demonstrates the topology built by all routers in Area 1234 using the LSA attributes for Area 1234 from all four routers. Using only type 1 LSAs, a connection is made between R2 and R4 because they point to each other's RID in the point-to-point LSA. Notice that the three networks on R1, R2, and R3 (10.123.1.0) have not been directly connected yet.

Key Topic

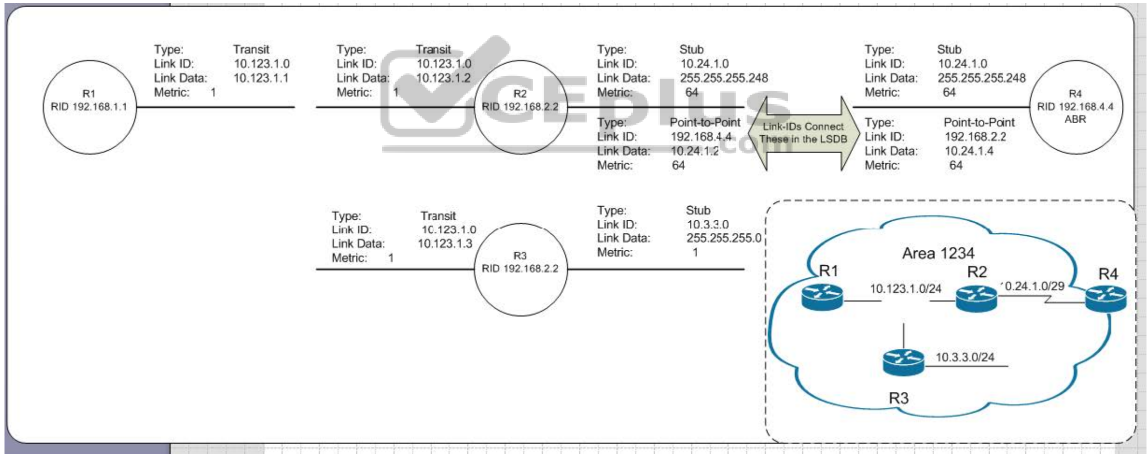


Figure 9-7 Visualization of Type 1 LSAs

Key Topic

LSA Type 2: Network Link

A type 2 LSA represents a multi-access network segment that uses a DR. The DR always advertises the type 2 LSA and identifies all the routers attached to that network segment. If a DR has not been elected, a type 2 LSA is not present in the LSDB because the corresponding type 1 transit link type LSA is a stub. Like type 1 LSAs, Type 2 LSAs are not flooded outside the originating OSPF area.

Area 1234 has only one DR segment that connects R1, R2, and R3 because R3 has not formed an OSPF adjacency on the 10.3.3.0/24 network segment. On the 10.123.1.0/24 network segment, R3 is elected as the DR, and R2 is elected as the BDR because of the order of the RIDs.

Note

Detailed type 2 LSA information is shown with the command **show ip ospf database network**.

Now that we have the type 2 LSA for Area 1234, all the network links are connected. Figure 9-8 provides a visualization of the type 1 and type 2 LSAs, which correspond with Area 1234 perfectly.

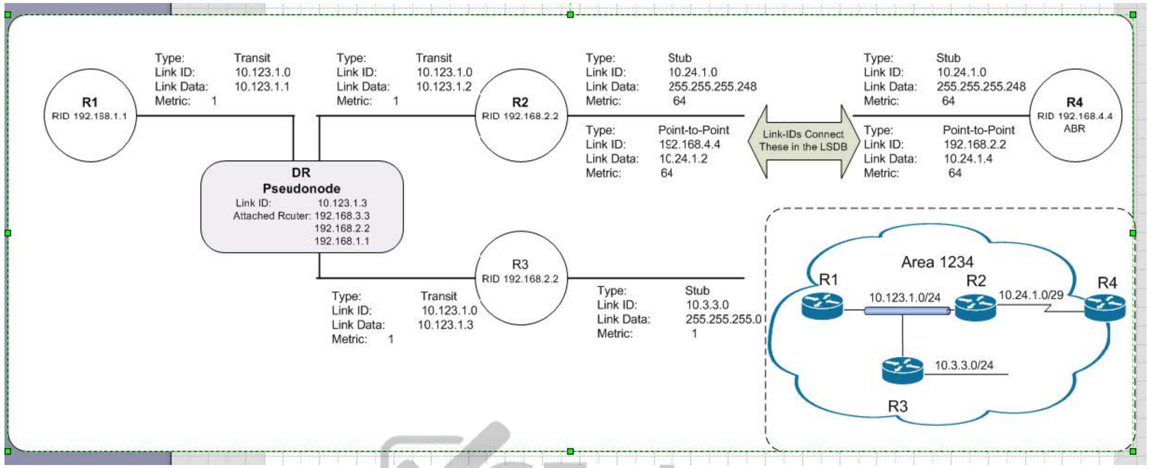


Figure 9-8 Visualization of Area 1234 with Type 1 and Type 2 LSAs

Note

When the DR changes for a network segment, a new type 2 LSA is created, causing SPF to run again within the OSPF area.

Key Topic

LSA Type 3: Summary Link

Type 3 LSAs represent networks from other areas. The role of the ABRs is to participate in multiple OSPF areas and ensure that the networks associated with type 1 LSAs are reachable in the non-originating OSPF areas.

As explained earlier, ABRs do not forward type 1 or type 2 LSAs into other areas. When an ABR receives a type 1 LSA, it creates a type 3 LSA referencing the network in the original type 1 LSA; the type 2 LSA is used to determine the network mask of the multi-access network. The ABR then advertises the type 3 LSA into other areas. If an ABR receives a type 3 LSA from Area 0 (the backbone), it regenerates a new type 3 LSA for the nonbackbone area and lists itself as the advertising router, with the additional cost metric.

Figure 9-9 demonstrates the concept of a type 3 LSA interaction with type 1 LSAs.

Key Topic

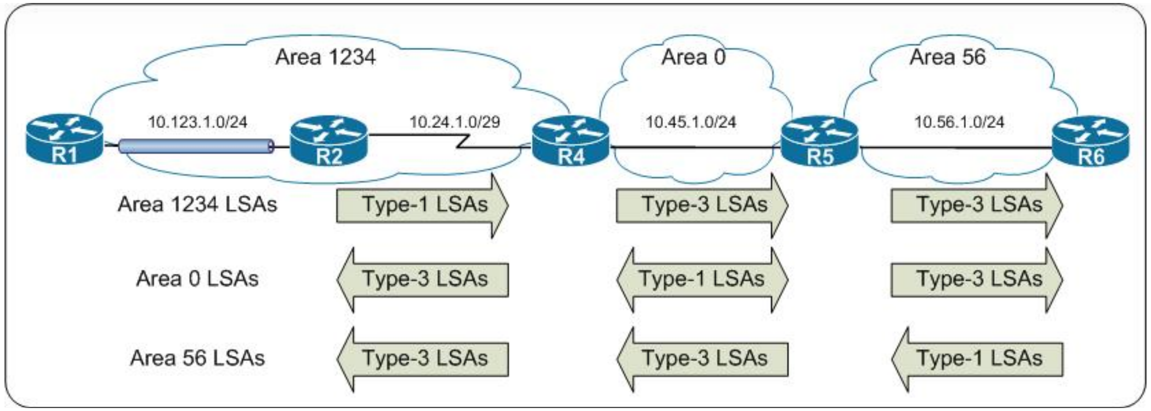


Figure 9-9 Type 3 LSA Conceptual Overview

The type 3 LSAs show up under the appropriate areas where they exist in the OSPF domain. For example, the 10.56.1.0 type 3 LSA is in Area 0 and Area 1234 on R4; however, on R5 the type 3 LSA exists only in Area 0 because the 10.56.1.0 network is a type 1 LSA in Area 56.

Detailed type 3 LSA information is shown with the command **show ip ospf database summary**. The output can be restricted to a specific LSA by appending the network prefix to the end of the command.

The advertising router for type 3 LSAs is the last ABR that advertises the prefix. The metric within the type 3 LSA uses the following logic:

- If the type 3 LSA is created from a type 1 LSA, it is the total path metric to reach the originating router in the type 1 LSA.

- If the type 3 LSA is created from a type 3 LSA from Area 0, it is the total path metric to the ABR plus the metric in the original type 3 LSA.

For example, from Figure 9-9, as R2 advertises the 10.123.1.0/24 network, the following happens:

- R4 receives R2’s type 1 LSA and creates a new type 3 LSA by using the metrics 65. The cost of 1 for R2’s LAN interface and 64 for the serial link between R2 and R4.

- R4 advertises the type 3 LSA with the metric 65 into Area 0.

- R5 receives the type 3 LSA and creates a new type 3 LSA for Area 56, using the metric 66. The cost of 1 for the link between R4 and R5 plus the original type 3 LSA metric 65.

- R6 receives the type 3 LSA. Part of R6’s calculation is the metric to reach the ABR (R5), which is 1 plus the metric in the type 3 LSA (66). R6 therefore calculates the metric 67 to reach the 10.123.1.0/24.

The type 3 LSA contains the link-state ID (network number), the subnet mask, the IP address of the advertising ABR, and the metric for the network prefix.

Figure 9-10 provides R4’s perspective of the type 3 LSA created by ABR (R5) for the 10.56.1.0/24 network. R4 does not know if the 10.56.1.0/24 network is directly attached to the ABR (R5) or multiple hops away. R4 knows that its metric to the ABR (R5) is 1 and that the type 3 LSA already has a metric of 1, so its total path metric reach the 10.56.1.0/24 network is 2.

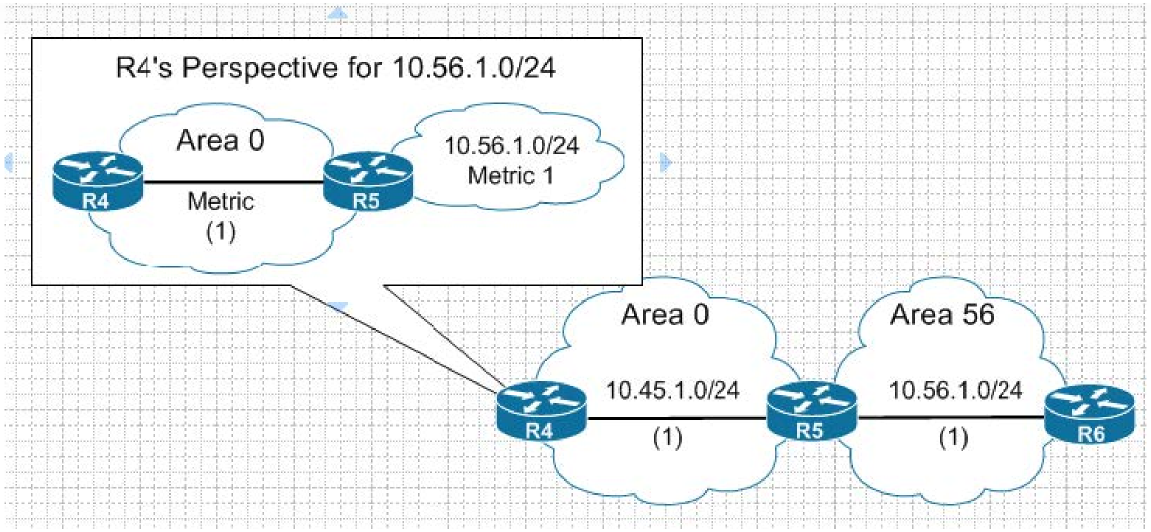


Figure 9-10 Visualization of the 10.56.1.0/24 Type 3 LSA from Area 0

Figure 9-11 provides R3’s perspective of the type 3 LSA created by ABR (R4) for the 10.56.1.0/24 network. R3 does not know if the 10.56.1.0/24 network is directly attached to the ABR (R4) or multiple hops away. R3 knows that its metric to the ABR (R4) is 65 and that the type 3 LSA already has a metric of 2, so its total path metric to reach the 10.56.1.0/24 network is 67.

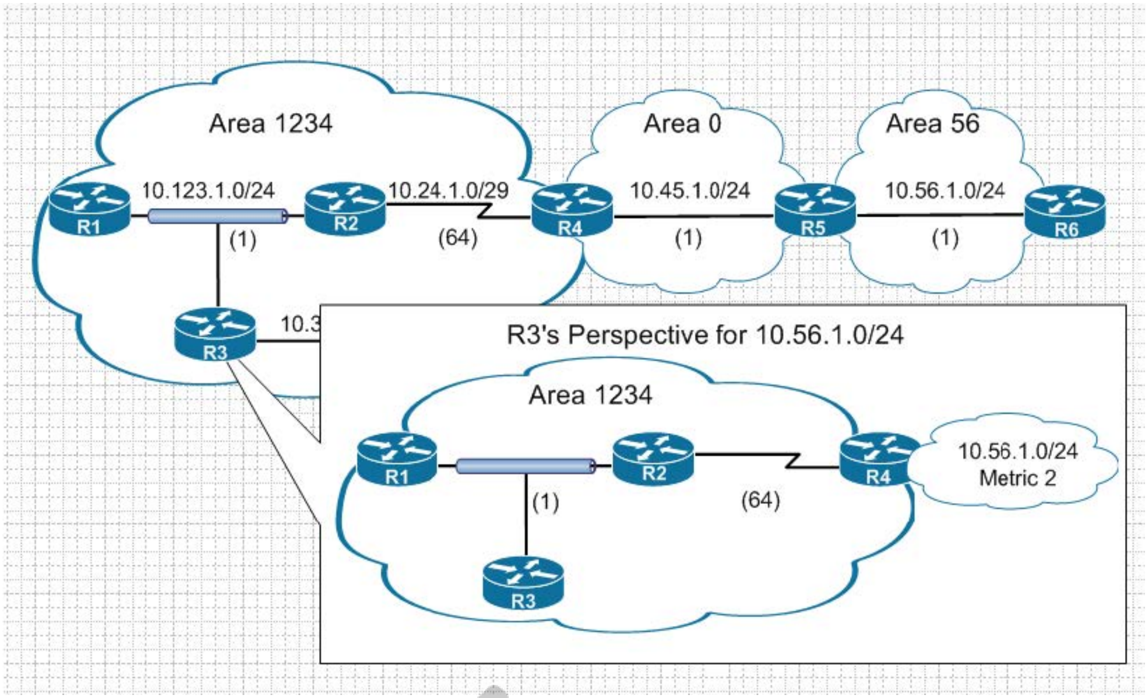


Figure 9-11 Visualization of 10.56.1.0/24 Type 3 LSA from Area 1234

Note

An ABR advertises only one type 3 LSA for a prefix, even if it is aware of multiple paths from within its area (type 1 LSAs) or from outside its area (type 3 LSAs). The metric for the best path will be used when the LSA is advertised into a different area.

DISCONTIGUOUS NETWORKS

Network engineers who do not fully understand OSPF design may create a topology such as the one illustrated in Figure 9-12. While R2 and R3 have OSPF interfaces in Area 0, traffic from Area 12 must cross Area 23 to reach Area 34. An OSPF network with this design is discontiguous because interarea traffic is trying to cross a nonbackbone area.

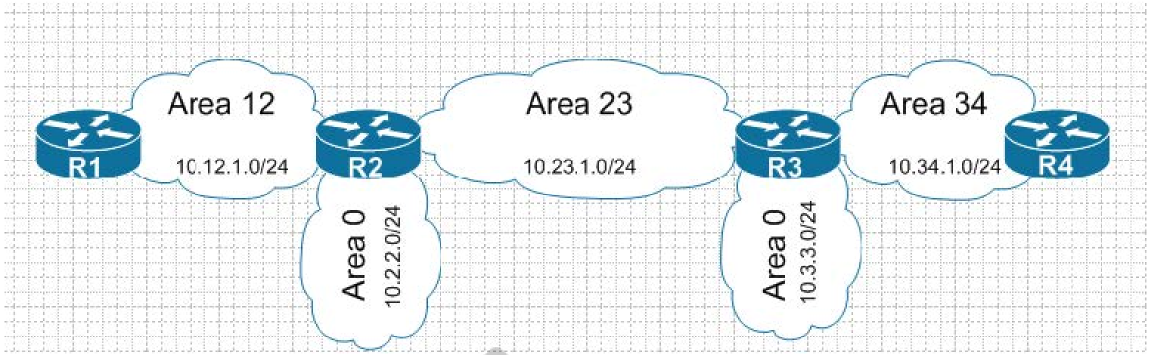
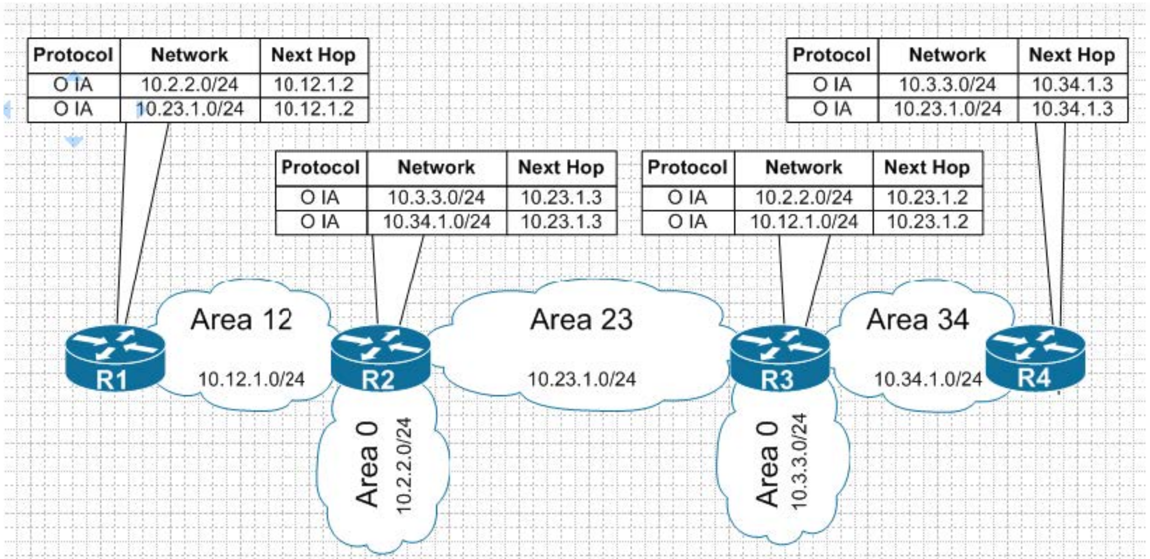


Figure 9-12 Discontiguous Network

At first glance, it looks like routes in the routing tables on R2 and R3 in Figure 9-13 are being advertised across area 23. The 10.34.1.0/24 network was advertised into OSPF by R3 and R4 as a type 1 LSA. R3 is an ABR and converts Area 34’s 10.34.1.0/24 type 1 LSA into a type 3 LSA in Area 0. R3 uses the type 3 LSA from Area 0 to generate the type 3 LSA for Area 23. R2 is able to install the type 3 LSA from Area 23 into its routing table.



Protocol	Network	Next Hop
O IA	10.2.2.0/24	10.12.1.2
O IA	10.23.1.0/24	10.12.1.2

Protocol	Network	Next Hop
O IA	10.3.3.0/24	10.34.1.3
O IA	10.23.1.0/24	10.34.1.3

Protocol	Network	Next Hop
O IA	10.3.3.0/24	10.23.1.3
O IA	10.34.1.0/24	10.23.1.3

Protocol	Network	Next Hop
O IA	10.2.2.0/24	10.23.1.2
O IA	10.12.1.0/24	10.23.1.2

Figure 9-13 OSPF Routes for Discontiguous Network

Key Topic

Most people would assume that the 10.34.1.0/24 route learned by Area 23 would then advertise into R2’s Area 0 and then propagate to Area 12. However, they would be wrong. There are three fundamental rules ABRs use the for creating type 3 LSAs:

• Type 1 LSAs received from an area create type 3 LSAs into the backbone area and nonbackbone areas.

• Type 3 LSAs received from Area 0 are created for the nonbackbone area.

- Type 3 LSAs received from a nonbackbone area only insert into the LSDB for the source area. ABRs do not create a type 3 LSA for the other areas (including a segmented Area 0).

The simplest fix for a discontiguous network is to ensure that Area 0 is contiguous. There are other functions, like virtual link or usage of GRE tunnels; however, they are beyond the scope of this book and complicate the operational environment.

Note

Real-life scenarios of discontiguous networks involve Area 0 becoming partitioned due to hardware failures. Ensuring that multiple paths exist to keep the backbone contiguous is an important factor in network design.

Key Topic

OSPF PATH SELECTION

OSPF executes Dijkstra's shortest path first (SPF) algorithm to create a loop-free topology of shortest paths. All routers use the same logic to calculate the shortest path for each network. Path selection prioritizes paths by using the following logic:

1. Intra-area

2. Interarea

3. External routes (which involves additional logic not covered in this book)

Intra-Area Routes

Routes advertised via a type 1 LSA for an area are always preferred over type 3 LSAs. If multiple intra-area routes exist, the path with the lowest total path metric is installed in the OSPF Routing Information Base (RIB), which is then presented to the router's global RIB. If there is a tie in metric, both routes install into the OSPF RIB.

In Figure 9-14, R1 is computing the route to 10.4.4.0/24. Instead of taking the faster Ethernet connection (R1–R2–R4), R1 takes the path across the slower serial link (R1–R3–R4) to R4 because that is the intra-area path.

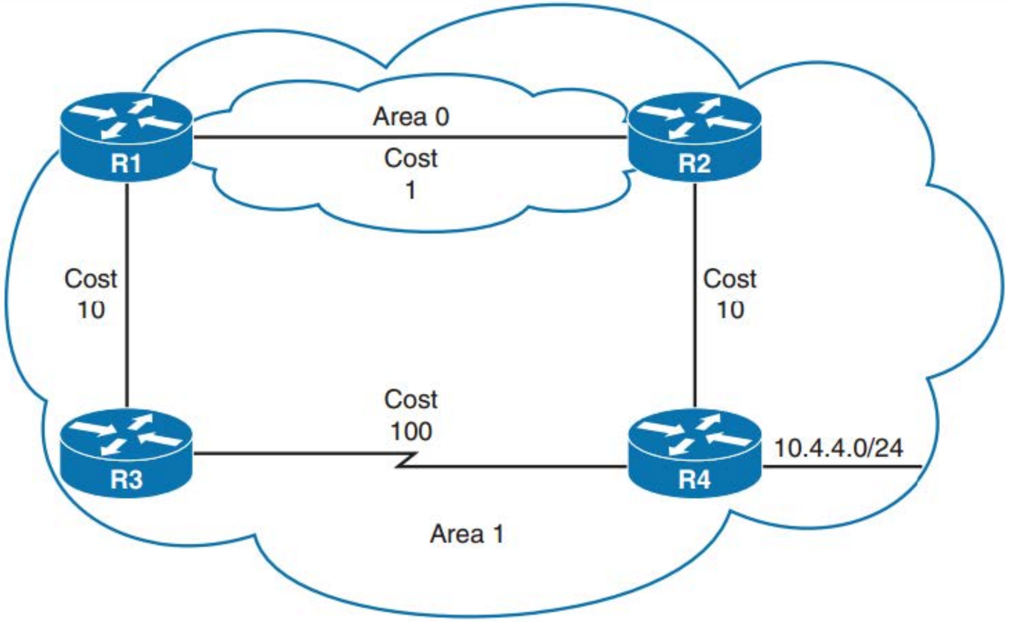


Figure 9-14 Intra-Area Routes over Interarea Routes

Example 9-6 shows R1’s routing table entry for the 10.4.4.0/24 network. Notice that the metric is 111 and that the intra-area path was selected over the interarea path with the lower total path metric.

Example 9-6 R1’s Routing Table for the 10.4.4.0/24 Network

```
R1# show ip route 10.4.4.0
Routing entry for 10.4.4.0/24
  Known via "ospf 1", distance 110, metric 111, type intra area
  Last update from 10.13.1.3 on GigabitEthernet0/1, 00:00:42 ago
  Routing Descriptor Blocks:
  * 10.13.1.3, from 10.34.1.4, 00:00:42 ago, via GigabitEthernet
      Route metric is 111, traffic share count is 1
```

Interarea Routes

The next priority for selecting a path to a network is selection of the path with the lowest total path metric to the destination. If there is a tie in metric, both routes install into the OSPF RIB. All interarea paths for a route must go through Area 0 to be considered.

In Figure 9-15, R1 is computing the path to R6. R1 uses the path R1–R3–R5–R6 because its total path metric is 35 versus the R1–R2–R4–R6 path, with a metric of 40.

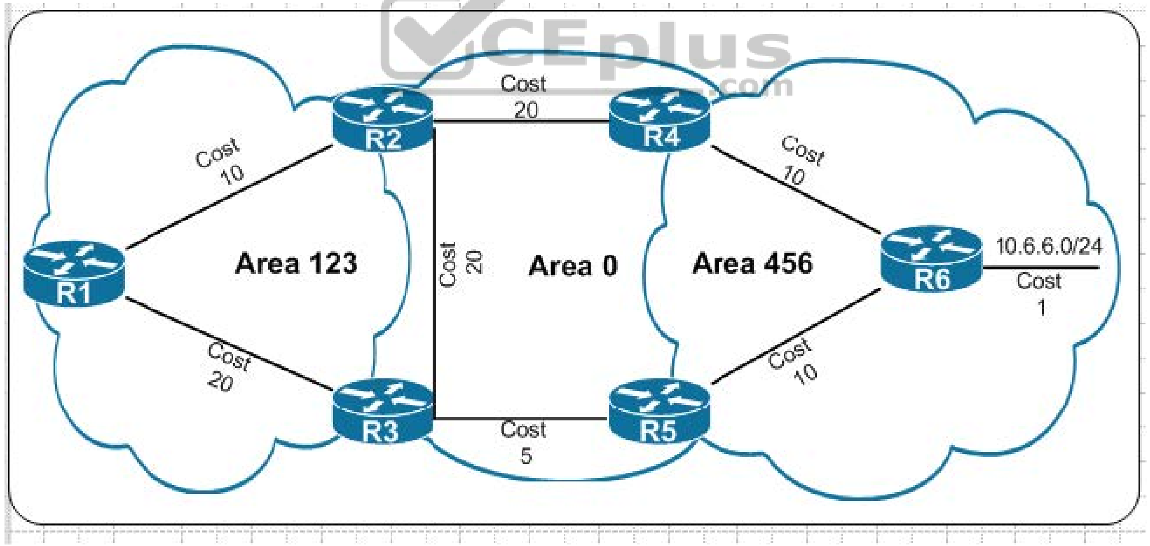


Figure 9-15 Interarea Route Selection

Equal-Cost Multipathing

If OSPF identifies multiple paths in the path selection algorithms, those routes are installed into the routing table as equal-cost multipathing (ECMP) routes. The default maximum number of ECMP paths is four paths. The default ECMP setting can be overwritten with the command **maximum-paths** *maximum-paths* under the OSPF process to modify the default setting.

Key Topic

SUMMARIZATION OF ROUTES

Route scalability is a large factor for the IGP routing protocols used by service providers because there can be thousands of routers running in a network. Splitting up an OSPF routing domain into multiple areas reduces the size of the LSDB for each area. While the number of routers and networks remains the same within the OSPF routing domain, the detailed type 1 and type 2 LSAs are exchanged for simpler type 3 LSAs.

For example, referencing our topology for LSAs, in Figure 9-16 for Area 1234, there are three type 1 LSAs and one type 2 LSA for the 10.123.1.0/24 network. Those four LSAs become one type 3 LSA outside Area 1234. Figure 9-16 illustrates the reduction of LSAs through area segmentation for the 10.123.1.0/24 network.

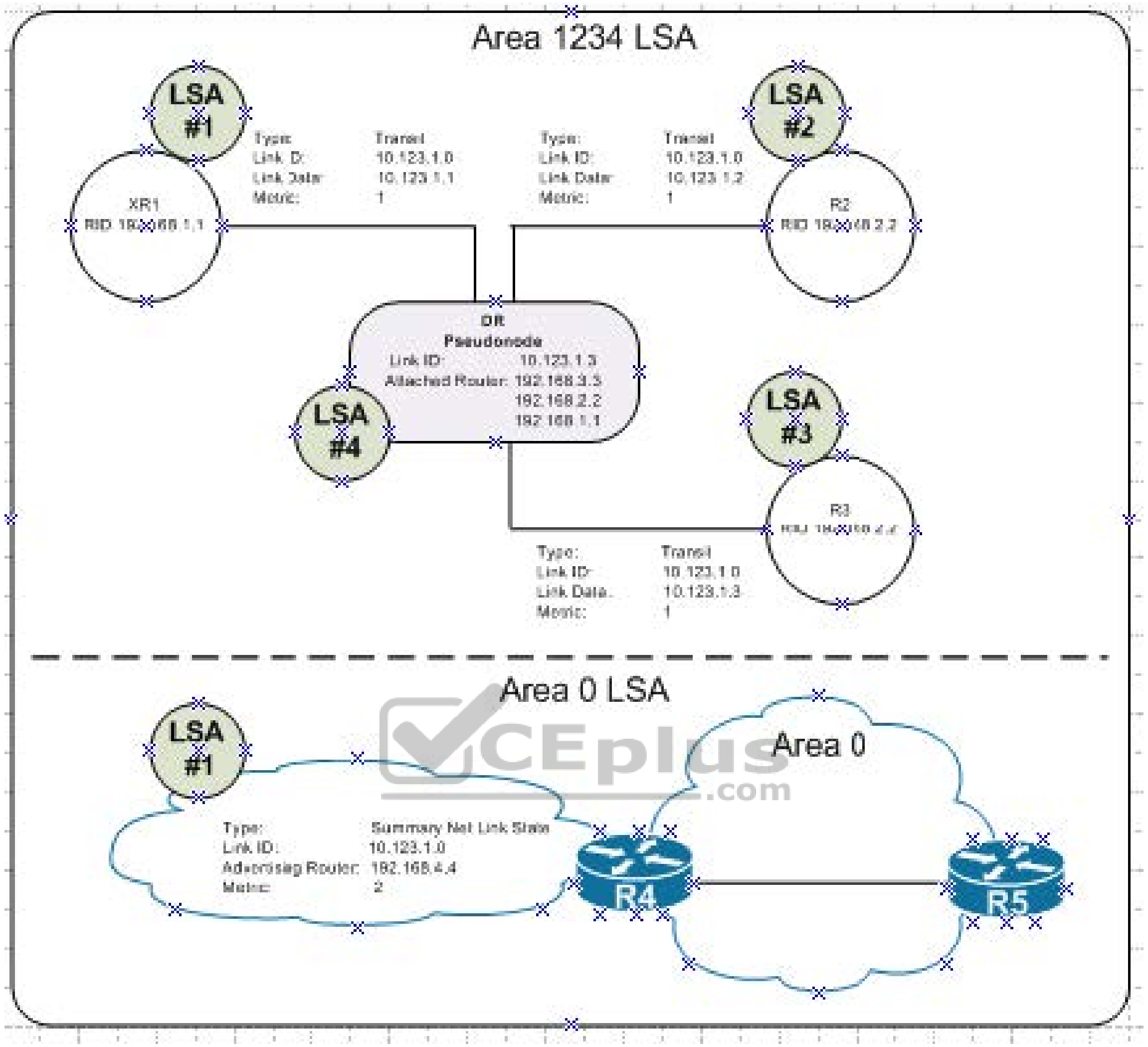


Figure 9-16 LSA Reduction Through Area Segmentation

Summarization Fundamentals

Another method of shrinking the LSDB involves summarizing network prefixes. Newer routers have more memory and faster processors than those in the past, but because all routers have an identical copy of the LSDB, an OSPF area needs to accommodate the smallest and slowest router in that area.

Summarization of routes also helps SPF calculations run faster. A router that has 10,000 network entries will take longer to run the SPF calculation than a router with 500 network entries. Because all routers within an area must maintain an identical copy of the LSDB, summarization occurs between areas on the ABRs.

Summarization can eliminate the SPF calculation outside the area for the summarized prefixes because the smaller prefixes are hidden. Figure 9-17 provides a simple network topology where the serial link between R3 and R4 adds to the path metric, and all traffic uses the other path to reach the 172.16.46.0/24 network. If the 10.1.12.0/24 link fails, all routers in Area 1 have to run SPF calculations. R4 identifies that the 10.1.13.0/24 and 10.1.34.0/24 networks will change their next hop through the serial link. Both of the type 3 LSAs for these networks need to be updated with new path metrics and advertised into Area 0. The routers in Area 0 run an SPF calculation only on those two prefixes.

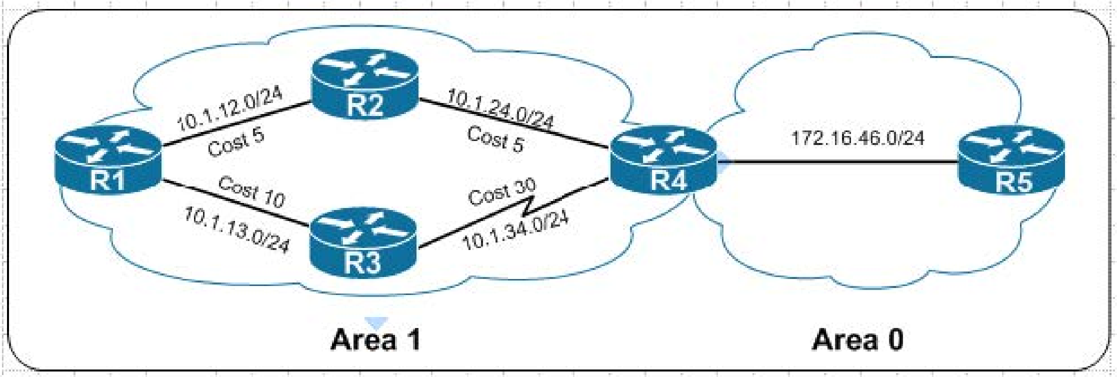


Figure 9-17 The Impact of Summarization on SPF Topology Calculation

Figure 9-18 shows the networks in Area 1 being summarized at the ABR into the aggregate 10.1.0.0/18 prefix. If the 10.1.12.0/24 link fails, all the routers in Area 1 still run the SPF calculation, but routers in Area 0 are not impacted because the 10.1.13.0/24 and 10.1.34.0/24 networks are not known outside Area 1.

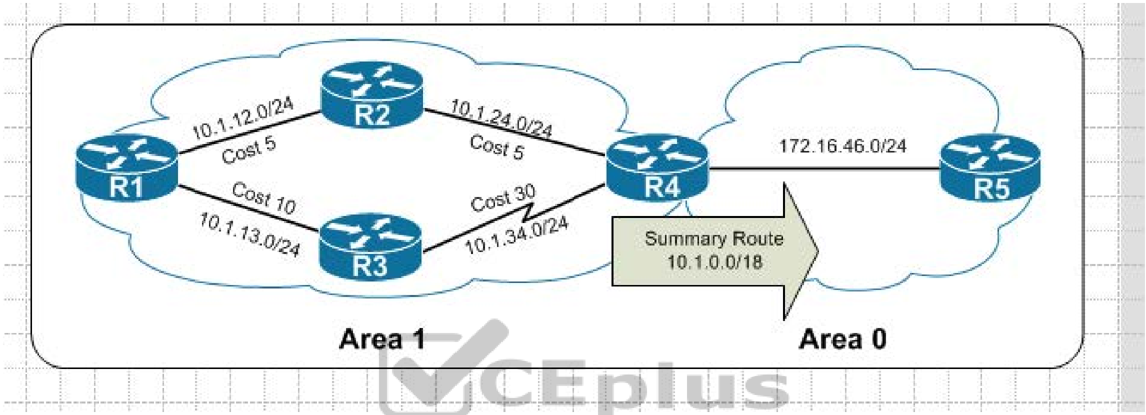


Figure 9-18 Topology Example with Summarization

This concept applies to networks of various sizes but is beneficial for networks with a carefully developed IP addressing scheme and proper summarization. The following sections explain summarization in more detail.

Key Topic

Interarea Summarization

Interarea summarization reduces the number of type 3 LSAs that an ABR advertises into an area when it receives type 1 LSAs. The network summarization range is associated with a specific source area for type 1 LSAs.

When a type 1 LSA within the summarization range reaches the ABR from the source area, the ABR creates a type 3 LSA for the summarized network range. The ABR suppresses the more specific type 3 LSAs, thereby preventing the generation of the subordinate route's type 3 LSAs. Interarea summarization does not impact the type 1 LSAs in the source area.

Figure 9-19 shows five type 1 LSAs (172.16.1.0/24 through 172.16.15.0/24) being summarized into one type 3 LSA (the 172.16.0.0/20 network).

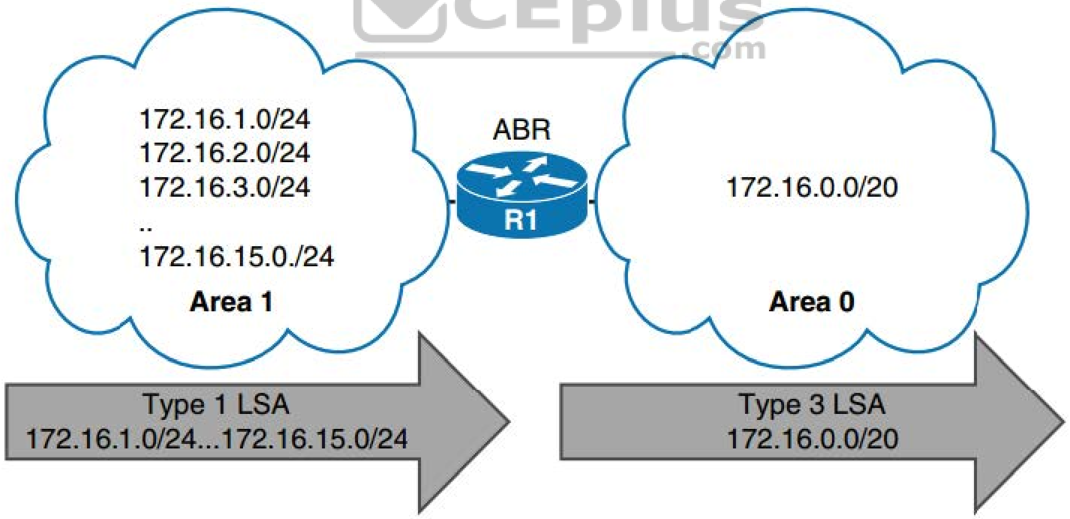


Figure 9-19 OSPF Interarea Summarization

Summarization works only on type 1 LSAs and is normally configured (or designed) so that summarization occurs as routes enter the backbone from nonbackbone areas.

Summarization Metrics

The default metric for the summary LSA is the smallest metric associated with an LSA; however, it can be set as part of the configuration. In Figure 9-20, R1 summarizes three prefixes with various path costs. The 172.16.3.0/24 prefix has the lowest metric, so that metric is used for the summarized route.

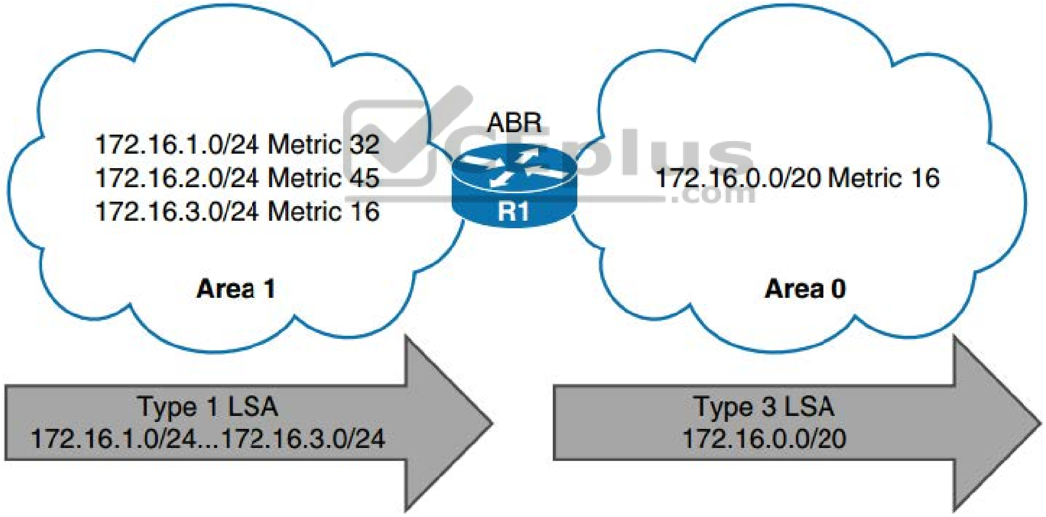


Figure 9-20 Interarea Summarization Metric

OSPF behaves identically to Enhanced Interior Gateway Routing Protocol (EIGRP) and checks every prefix within the summarization range when a matching type 1 LSA is added or removed. If a lower metric is available, the summary LSA is advertised with the newer metric; if the lowest metric is removed, a newer and higher metric is identified, and a new summary LSA is advertised with the higher metric.

Key Topic

Configuration of Interarea Summarization

To define the summarization range and associated area, use the command **area** *area-id* **range** *network subnet-mask* [**advertise** | **not-advertise**] [**cost** *metric*] under the OSPF process on the ABR. The default behavior is to advertise the summary prefix, so the keyword **advertise** is not necessary. Appending the **cost** *metric* keyword to the command statically sets the metric on the summary route.

Figure 9-21 provides a topology example in which R1 is advertising the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 networks.

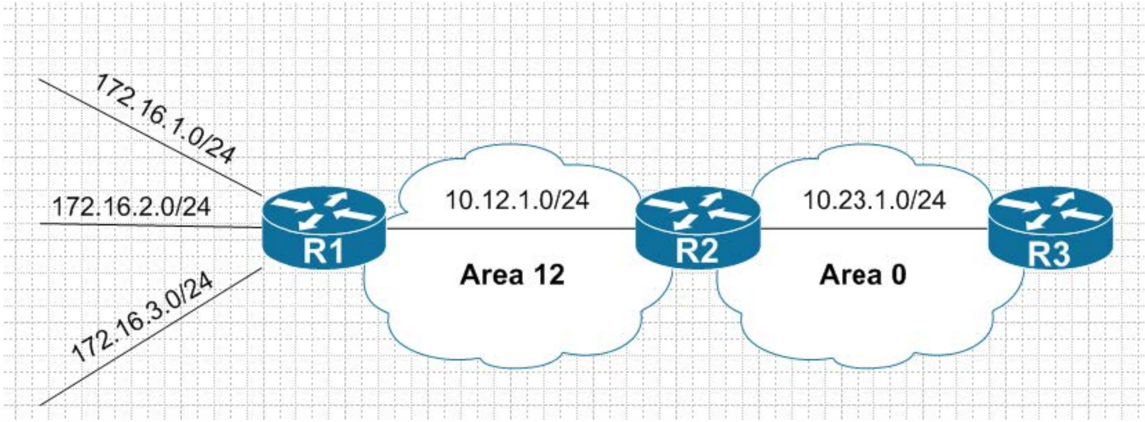


Figure 9-21 OSPF Interarea Summarization Example

Example 9-7 displays the routing table on R3 before summarization. Notice that the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 networks are all present.

Example 9-7 Routing Table Before OSPF Interarea Route Summarization

```
R3# show ip route ospf | b Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O IA     10.12.1.0/24 [110/20] via 10.23.1.2, 00:02:22, GigabitEt
      172.16.0.0/24 is subnetted, 3 subnets
O IA     172.16.1.0 [110/3] via 10.23.1.2, 00:02:12, GigabitEther
O IA     172.16.2.0 [110/3] via 10.23.1.2, 00:02:12, GigabitEther
O IA     172.16.3.0 [110/3] via 10.23.1.2, 00:02:12, GigabitEther
```

R2 summarizes the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 networks into a single summary network, 172.16.0.0/16, as those networks are advertised into Area 0. Example 9-8 provides R2's configuration for interarea summarization into an aggregate route of 172.16.0.0/16. A static cost of 45 is added to the summary route to reduce CPU load if any of the three networks flap.

Example 9-8 R2's Interarea Route Summarization Configuration

```
router ospf 1
 router-id 192.168.2.2
 area 12 range 172.16.0.0 255.255.0.0 cost 45
 network 10.12.0.0 0.0.255.255 area 12
 network 10.23.0.0 0.0.255.255 area 0
```

Example 9-9 displays R3’s routing table for verification that the smaller routes were suppressed while the summary route was aggregated. Notice that the path metric is 46, whereas previously the metric for the 172.16.1.0/24 network was 3.

Example 9-9 Routing Table After OSPF Interarea Route Summarization

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

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA    10.12.1.0/24 [110/2] via 10.23.1.2, 00:02:04, GigabitEthe
O IA  172.16.0.0/16 [110/46] via 10.23.1.2, 00:00:22, GigabitEthe
```

The ABR performing interarea summarization installs a discard route—that is, a route to the Null0 interface that match the summarized network range. Discard routes prevent routing loops where portions of the summarized network range do not have a more specific route in the RIB. The AD for the OSPF summary discard route for internal networks is 110, and it is 254 for external networks.

Example 9-10 shows the discard route on R2 for the 172.16.0.0/16 prefix.

Example 9-10 Discarding a Route for Loop Prevention

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

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
O        172.16.0.0/16 is a summary, 00:03:11, Null0
O        172.16.1.0/24 [110/2] via 10.12.1.1, 00:01:26, GigabitEt
O        172.16.2.0/24 [110/2] via 10.12.1.1, 00:01:26, GigabitEt
O        172.16.3.0/24 [110/2] via 10.12.1.1, 00:01:26, GigabitEt
```

ROUTE FILTERING

Route filtering is a method for selectively identifying routes that are advertised or received from neighbor routers. Route filtering may be used to manipulate traffic flows, reduce memory utilization, or improve security.

Filtering of routes with vector-based routing protocols is straightforward as the routes are filtered as routing updates are advertised to downstream neighbors. However, with link-state routing protocols such as OSPF, every router in an area shares a complete copy of the link-state database. Therefore, filtering of routes generally occurs as routes enter the area on the ABR.

The following sections describe three techniques for filtering routes with OSPF.

Filtering with Summarization

One of the easiest methodologies for filtering routes is to use the **not-advertise** keyword during prefix summarization. Using this keyword prevents creation of any type 3 LSAs for any networks in that range, thus making the subordinate routes visible only within the area where the route originates.

The full command structure is **area** *area-id* **range** *network subnet-mask* **not-advertise** under the OSPF process.

If we revisit Figure 9-21, where R1 is advertising the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 networks, we see that R2 can filter out any of the type 1 LSAs that are generated in Area 12 from being advertised into Area 0. The configuration is displayed in Example 9-11.

Example 9-11 R2's Configuration for Filtering via Summarization

```
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# router ospf 1
R2(config-router)# area 12 range 172.16.2.0 255.255.255.0 not-adv
```

Example 9-12 shows R3's routing table after the area filtering configuration has been placed on R2. The 172.16.2.0/24 network has been removed from Area 0. If a larger network range were configured, then more of the subordinate routes would be filtered.

Example 9-12 Verifying Removal of 172.16.2.0 from Area 0

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

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA     10.12.1.0/24 [110/3] via 10.34.1.3, 00:02:24, GigabitEth
      172.16.0.0/24 is subnetted, 2 subnets
O IA     172.16.1.0 [110/4] via 10.34.1.3, 00:00:17, GigabitEthe
O IA     172.16.3.0 [110/4] via 10.34.1.3, 00:00:17, GigabitEthe
```

Area Filtering

Although filtering via summarization is very easy, it is limited in its ability. For example, in Figure 9-22, if the 172.16.1.0/24 network needs to be present in Area 0 but removed in Area 34, it is not possible to filter the route using summarization.

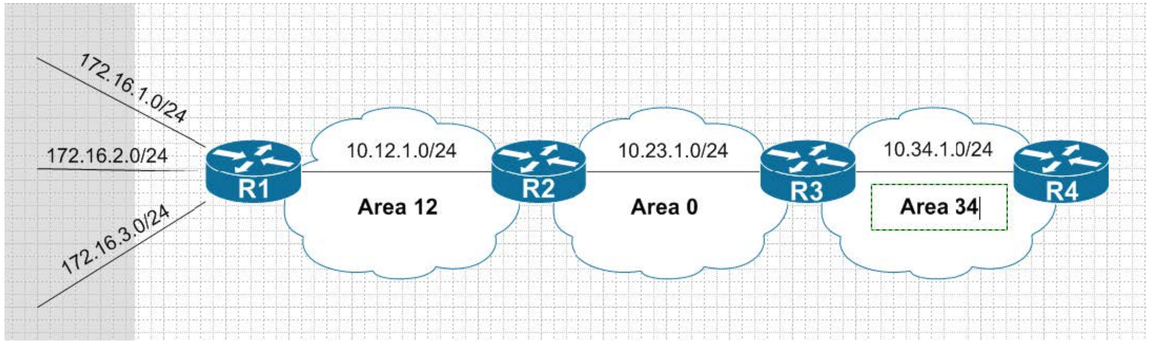


Figure 9-22 Expanded Topology for Filtering Routes

Other network designs require filtering of OSPF routes based on other criteria. OSPF supports filtering when type 3 LSA generation occurs. This allows for the original route to be installed in the LSDB for the source area so that the route can be installed in the RIB of the ABR. Filtering can occur in either direction on the ABR. Figure 9-23 demonstrates the concept.

Key Topic

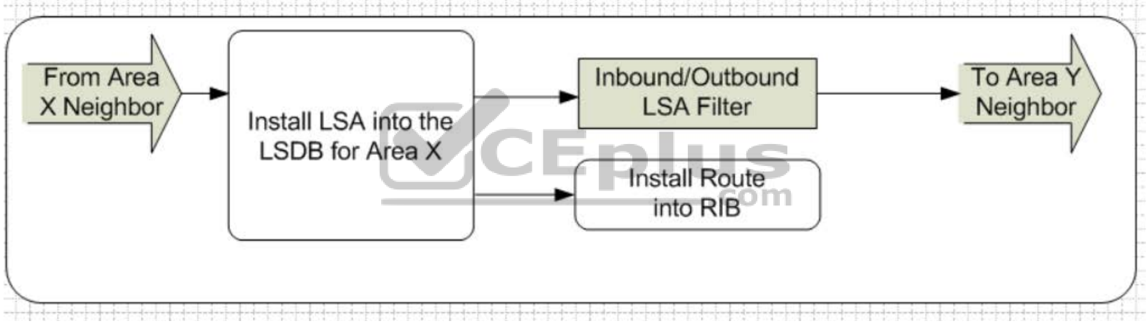

From Area X Neighbor
Install LSA into the LSDB for Area X
Inbound/Outbound LSA Filter
Install Route into RIB
To Area Y Neighbor

Figure 9-23 OSPF Area Filtering

Figure 9-24 expands on the sample topology and demonstrates that the ABR can filter routes as they advertise out of an area or into an area. R2 is able to filter routes (LSAs) outbound as they leave Area 12 or inbound as they enter Area 0. In addition, R3 can filter routes as they leave Area 0 or enter Area 34. The same logic applies with routes advertised in the opposition direction.

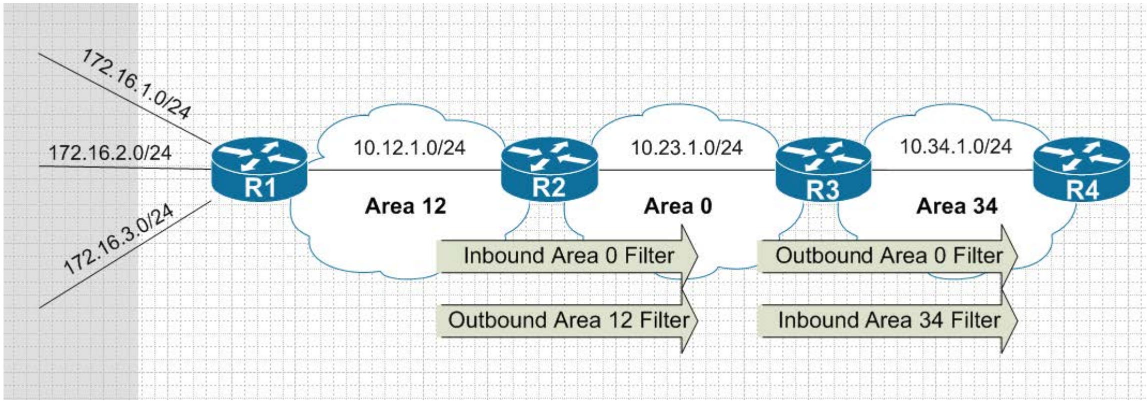


Figure 9-24 OSPF Area Filtering Topology

OSPF area filtering is accomplished by using the command **area** *area-id* **filter-list prefix** *prefix-list-name* {**in** | **out**} on the ABR. Say that R1 is advertising the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 network prefixes. R2 is configured to filter the 172.16.1.0/24 prefix as it enters Area 0, and R3 is configured to filter the 172.16.2.0/24 prefix as it leaves Area 0. Example 9-13 provides the necessary configuration for R2 and R3.

Example 9-13 Configuring OSPF Area Filtering

```
R2
ip prefix-list PREFIX-FILTER seq 5 deny 172.16.1.0/24
ip prefix-list PREFIX-FILTER seq 10 permit 0.0.0.0/0 le 32
!
router ospf 1
 router-id 192.168.2.2
 network 10.12.1.0 0.0.0.255 area 12
 network 10.23.1.0 0.0.0.255 area 0
 area 0 filter-list prefix PREFIX-FILTER in
```

```
R3
ip prefix-list PREFIX-FILTER seq 5 deny 172.16.2.0/24
ip prefix-list PREFIX-FILTER seq 10 permit 0.0.0.0/0 le 32
!
router ospf 1
 router-id 192.168.3.3
 network 10.23.1.0 0.0.0.255 area 0
 network 10.34.1.0 0.0.0.255 area 34
 area 0 filter-list prefix PREFIX-FILTER out
```

Example 9-14 shows the routing table on R3 where the 172.16.1.0/24 network has been filtered from all the routers in Area 0. The 172.16.2.0/24 network has been filtered from all the routers in Area 34. This verifies that the area filtering was successful for routes entering the backbone and leaving the backbone.

Example 9-14 Verifying OSPF Area Filtering

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

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O IA     10.12.1.0/24 [110/2] via 10.23.1.2, 00:17:39, GigabitEth
      172.16.0.0/24 is subnetted, 2 subnets
O IA     172.16.2.0 [110/3] via 10.23.1.2, 00:16:30, GigabitEther
O IA     172.16.3.0 [110/3] via 10.23.1.2, 00:16:30, GigabitEther

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

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O IA     10.12.1.0/24 [110/3] via 10.34.1.3, 00:19:41, GigabitEth
```

```
O IA     10.23.1.0/24 [110/2] via 10.34.1.3, 00:19:41, GigabitEt
      172.16.0.0/24 is subnetted, 1 subnets
O IA     172.16.3.0 [110/4] via 10.34.1.3, 00:17:07, GigabitEthe
```

Local OSPF Filtering

In some scenarios, routes need to be removed only on specific routers in an area. OSPF is a link-state protocol that requires all routers in the same area to maintain an identical copy of the LSDB for that area. A route can exist in the OSPF LSDB, but it could be prevented from being installed in the local RIB. This is accomplished by using a distribute lists. Figure 9-25 illustrates this concept.

Key Topic

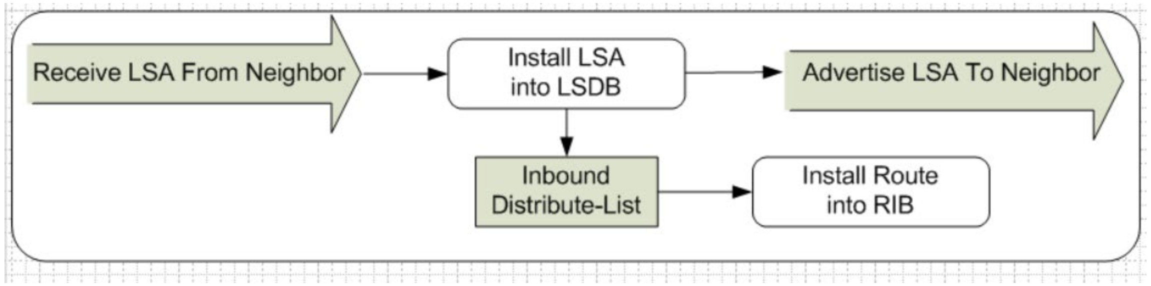


Figure 9-25 OSPF Distribute List Filtering Logic

A distribute list on an ABR does not prevent type 1 LSAs from becoming type 3 LSAs in a different area because the type 3 LSA generation occurs before the distribute list is processed.

However, a distribute list on an ABR prevents type 3 LSAs coming from the backbone from being regenerated into nonbackbone areas because this regeneration process happens after the distribute list is processed. A distribute list should not be used for filtering of prefixes between areas; the following section identifies more preferred techniques.

A distribute list is configured under the OSPF process with the command **distribute-list** {*acl-number* | *acl-name* | **prefix** *prefix-list-name* | **route-map** *route-map-name*} **in**. To demonstrate this concept, the topology from Figure 9-24 is used again. Say that R1 is advertising the 172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24 network prefixes. R2 filters the 172.16.3.0/24 network from entering its RIB. The configuration is provided in Example 9-15.

Example 9-15 Configuring the OSPF Distribute List

```
R2
ip access-list standard ACL-OSPF-FILTER
 deny   172.16.3.0
 permit any
!
router ospf 1
 router-id 192.168.2.2
 network 10.12.1.0 0.0.0.255 area 12
 network 10.23.1.0 0.0.0.255 area 0
 distribute-list ACL-OSPF-FILTER in
```

Example 9-16 shows the routing tables for R2 and R3. The 172.16.3.0/24 network is removed from R2's RIB but is present on R3's RIB.

Example 9-16 Verifying the OSPF Distribute List

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

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O IA     10.34.1.0/24 [110/2] via 10.23.1.3, 00:02:21, GigabitEth
      172.16.0.0/24 is subnetted, 2 subnets
O        172.16.1.0 [110/2] via 10.12.1.1, 00:02:21, GigabitEthe
O        172.16.2.0 [110/2] via 10.12.1.1, 00:02:21, GigabitEthe

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

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O IA     10.12.1.0/24 [110/2] via 10.23.1.2, 00:24:11, GigabitEth
      172.16.0.0/24 is subnetted, 3 subnets
O IA     172.16.1.0 [110/3] via 10.23.1.2, 00:01:54, GigabitEthe
O IA     172.16.2.0 [110/3] via 10.23.1.2, 00:23:02, GigabitEthe
O IA     172.16.3.0 [110/3] via 10.23.1.2, 00:23:02, GigabitEthe
```

EXAM PREPARATION TASKS

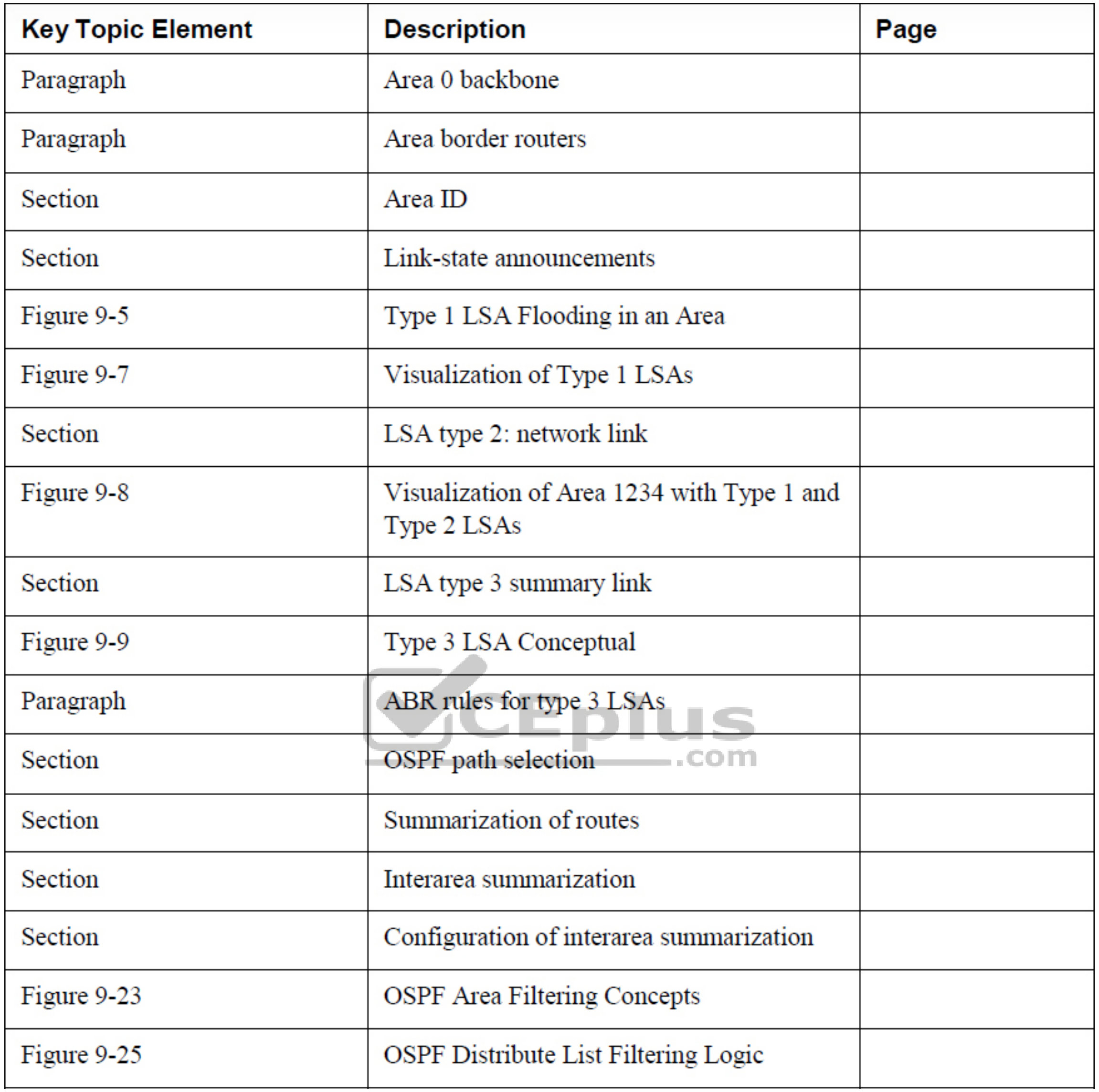
As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 9-2 lists these key topics and the page number on which each is found.

Key Topic

Table 9-2 Key Topics for Chapter 9



Key Topic Element	Description	Page
Paragraph	Area 0 backbone	
Paragraph	Area border routers	
Section	Area ID	
Section	Link-state announcements	
Figure 9-5	Type 1 LSA Flooding in an Area	
Figure 9-7	Visualization of Type 1 LSAs	
Section	LSA type 2: network link	
Figure 9-8	Visualization of Area 1234 with Type 1 and Type 2 LSAs	
Section	LSA type 3 summary link	
Figure 9-9	Type 3 LSA Conceptual	
Paragraph	ABR rules for type 3 LSAs	
Section	OSPF path selection	
Section	Summarization of routes	
Section	Interarea summarization	
Section	Configuration of interarea summarization	
Figure 9-23	OSPF Area Filtering Concepts	
Figure 9-25	OSPF Distribute List Filtering Logic	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

area border router (ABR)

backbone area

discontiguous network

intra-area route

interarea route

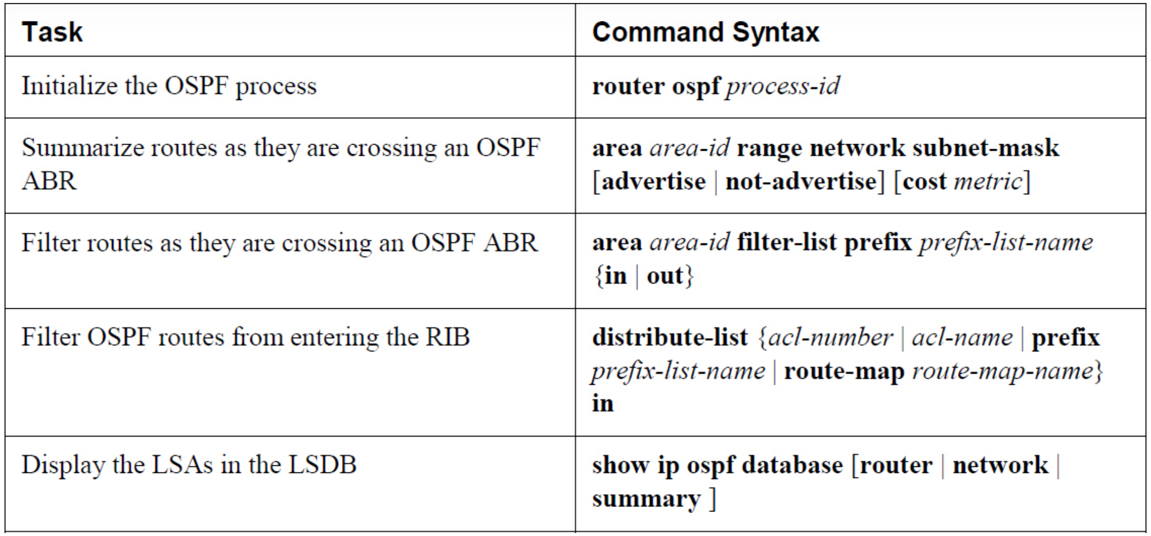
router LSA

summary LSA

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 9-3 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 9-3 Command Reference



Task	Command Syntax
Initialize the OSPF process	**router ospf** *process-id*
Summarize routes as they are crossing an OSPF ABR	**area** *area-id* **range network subnet-mask** [**advertise** | **not-advertise**] [**cost** *metric*]
Filter routes as they are crossing an OSPF ABR	**area** *area-id* **filter-list prefix** *prefix-list-name* {**in** | **out**}
Filter OSPF routes from entering the RIB	**distribute-list** {*acl-number* | *acl-name* | **prefix** *prefix-list-name* | **route-map** *route-map-name*} **in**
Display the LSAs in the LSDB	**show ip ospf database** [**router** | **network** | **summary**]

REFERENCES IN THIS CHAPTER

RFC 2328, *OSPF Version 2*, by John Moy, http://www.ietf.org/rfc/rfc2328.txt (http://www.ietf.org/rfc/rfc2328.txt), April 1998.

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*, by Alex Zinin, Acee Lindem, and Derek Yeung, https://tools.ietf.org/html/rfc3509, April 2003.

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

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

Chapter 10. OSPFv3

This chapter covers the following subjects:

- **OSPFv3 Fundamentals:** This section provides an overview of the OSPFv3 routing protocol and the similarities to OSPFv2.

- **OSPFv3 Configurations:** This section demonstrates the configuration and verification of an OSPFv3 environment.

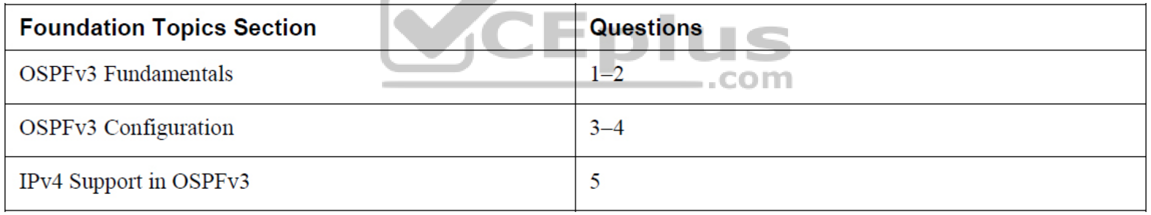
- **IPv4 Support in OSPFv3:** This section explains and demonstrates how OPSFv3 can be used for exchanging IPv4 routes.

OSPF Version 3 (OSPFv3), which is the latest version of the OSPF protocol, includes support for both the IPv4 and IPv6 address families. The OSPFv3 protocol is not backward compatible with OSPFv2, but the protocol mechanisms described in Chapters 8, “OSPF,” and 9, “Advanced OSPF,” are essentially the same for OSPFv3. This chapter expands on Chapter 9 and discusses OSPFv3 and its support of IPv6.

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the “Exam Preparation Tasks” section. Table 10-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

Table 10-1 Do I Know This Already?” Foundation Topics Section-to-Question Mapping



Foundation Topics Section	Questions
OSPFv3 Fundamentals	1–2
OSPFv3 Configuration	3–4
IPv4 Support in OSPFv3	5

1. OSPFv3 uses _______ packet types for inter-router communication.

a. three

b. four

c. five

d. six

e. seven

2. The OSPFv3 hello packet uses the ______ for the destination address.

a. MAC address 00:C1:00:5C:00:FF

b. MAC address E0:00:00:06:00:AA

c. IP address 224.0.0.8

d. IP address 224.0.0.10

e. IPv6 address FF02::A

f. IPv6 address FF02::5

3. How do you enable OSPFv3 on an interface?

a. Use the command **network** *prefix/prefix-length* under the OSPF process.

b. Use the command **network** *interface-id* under the OSPF process.

c. Use the command **ospfv3** *process-id* **ipv6 area** *area-id* under the interface.

d. Nothing. OSPFv6 is enabled on all IPv6 interfaces upon initialization of the OSPF process.

4. True or false: On a brand-new router installation, OSPFv3 requires only that an IPv6 link-local address be configured and that OSPFv3 be enabled on that interface to form an OSPFv3 neighborship with another router.

a. True

b. False

5. True or false: OSPFv3 support for IPv4 networks only requires that an IPv4 address be assigned to the interface and that the OSPFv3 process be initialized for IPv4.

a. True

b. False

Answers to the "Do I Know This Already?" quiz:

1. C

2. F

3. C

4. B

5. B

FOUNDATION TOPICS

Key Topic

OSPFV3 FUNDAMENTALS

OSPFv3 is different from OSPFv2 in the following ways:

- **Support for multiple address families:** OSPFv3 supports IPv4 and IPv6 address families.

- **New LSA types:** New LSA types have been created to carry IPv6 prefixes.

- **Removal of addressing semantics:** The IP prefix information is no longer present in the OSPF packet headers. Instead, it is carried as LSA payload information, making the protocol essentially address family independent, much like IS-IS. OSPFv3 uses the term *link* instead of *network* because the SPT calculations are per link instead of per subnet.

- **LSA flooding:** OSPFv3 includes a new link-state type field that is used to determine the flooding scope of LSA, as well as the handling of unknown LSA types.

- **Packet format:** OSPFv3 runs directly over IPv6, and the number of fields in the packet header has been reduced.

- **Router ID:** The router ID is used to identify neighbors, regardless of the network type in OSPFv3. When configuring OSPFv3 on IOS routers, the ID must always be manually assigned in the routing process.

- **Authentication:** Neighbor authentication has been removed from the OSPF protocol and is now performed through IPsec extension headers in the IPv6 packet.

- **Neighbor adjacencies:** OSPFv3 inter-router communication is handled by IPv6 link-local addressing. Neighbors are not automatically detected over non-broadcast multiple access (NBMA) interfaces. A neighbor must be manually specified using the link-local address. IPv6 allows for multiple subnets to be assigned to a single interface, and OSPFv3 allows for neighbor adjacency to form even if the two routers do not share a common subnet.

- **Multiple instances:** OSPFv3 packets include an instance ID field that may be used to manipulate which routers on a network segment are allowed to form adjacencies.

> **Note**
>
> RFC 5340 provides in-depth coverage of all the differences between OSPFv2 and OSPFv3.

OSPFv3 Link-State Advertisement

OSPFv3 packets use protocol ID 89, and routers communicate with each other using the local interface's IPv6 link-local address. The OSPF link-state database information is organized and advertised differently in Version 3 than in Version 2. OSPFv3 modifies the structure of the router LSA (type 1), renames the network summary LSA to the interarea prefix LSA, and renames the ASBR summary LSA to the interarea router LSA. The principal difference is that the router LSA is only responsible for announcing interface parameters such as the interface type (point-to-point, broadcast, NBMA, point-to-multipoint, and virtual links) and metric (cost).

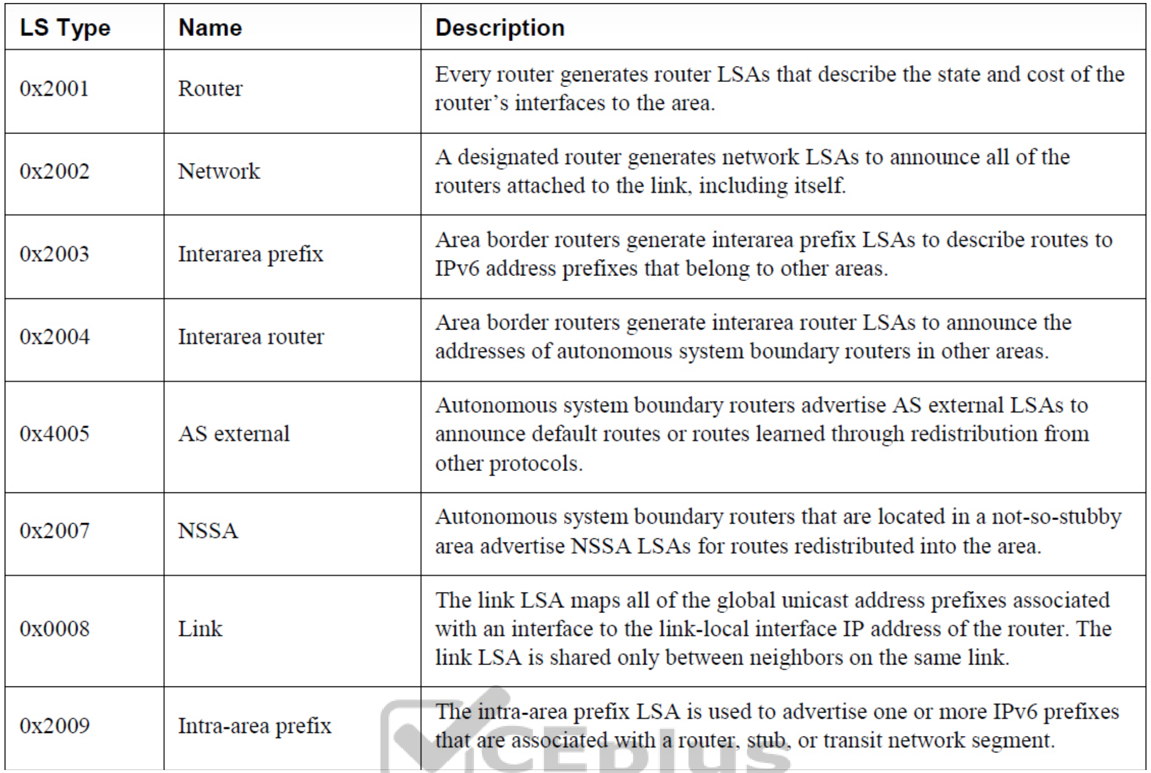
IP address information is advertised independently by two new LSA types:

- Intra-area prefix LSA
- Link-local LSA

The OSPF Dijkstra calculation used to determine the shortest path tree (SPT) only examines the router and network LSAs. Advertising the IP address information using new LSA types eliminates the need for OSPF to perform full shortest path first (SPF) tree calculations every time a new address prefix is added or changed on an interface. The OSPFv3 link-state database (LSDB) creates a shortest path topology tree based on links instead of networks.

Table 10-2 provides a brief description of each OSPFv3 LSA type.

Table 10-2 OSPFv3 LSA Types



LS Type	Name	Description
0x2001	Router	Every router generates router LSAs that describe the state and cost of the router's interfaces to the area.
0x2002	Network	A designated router generates network LSAs to announce all of the routers attached to the link, including itself.
0x2003	Interarea prefix	Area border routers generate interarea prefix LSAs to describe routes to IPv6 address prefixes that belong to other areas.
0x2004	Interarea router	Area border routers generate interarea router LSAs to announce the addresses of autonomous system boundary routers in other areas.
0x4005	AS external	Autonomous system boundary routers advertise AS external LSAs to announce default routes or routes learned through redistribution from other protocols.
0x2007	NSSA	Autonomous system boundary routers that are located in a not-so-stubby area advertise NSSA LSAs for routes redistributed into the area.
0x0008	Link	The link LSA maps all of the global unicast address prefixes associated with an interface to the link-local interface IP address of the router. The link LSA is shared only between neighbors on the same link.
0x2009	Intra-area prefix	The intra-area prefix LSA is used to advertise one or more IPv6 prefixes that are associated with a router, stub, or transit network segment.

OSPFv3 Communication

OSPFv3 packets use protocol ID 89, and routers communicate with each other using the local interface's IPv6 link-local address as the source. Depending on the packet type, the destination address is either a unicast link-local address or a multicast link-local scoped address:

- **FF02::05:** OSPFv3 AllSPFRouters

- **FF02::06:** OSPFv3 AllDRouters designated router (DR)

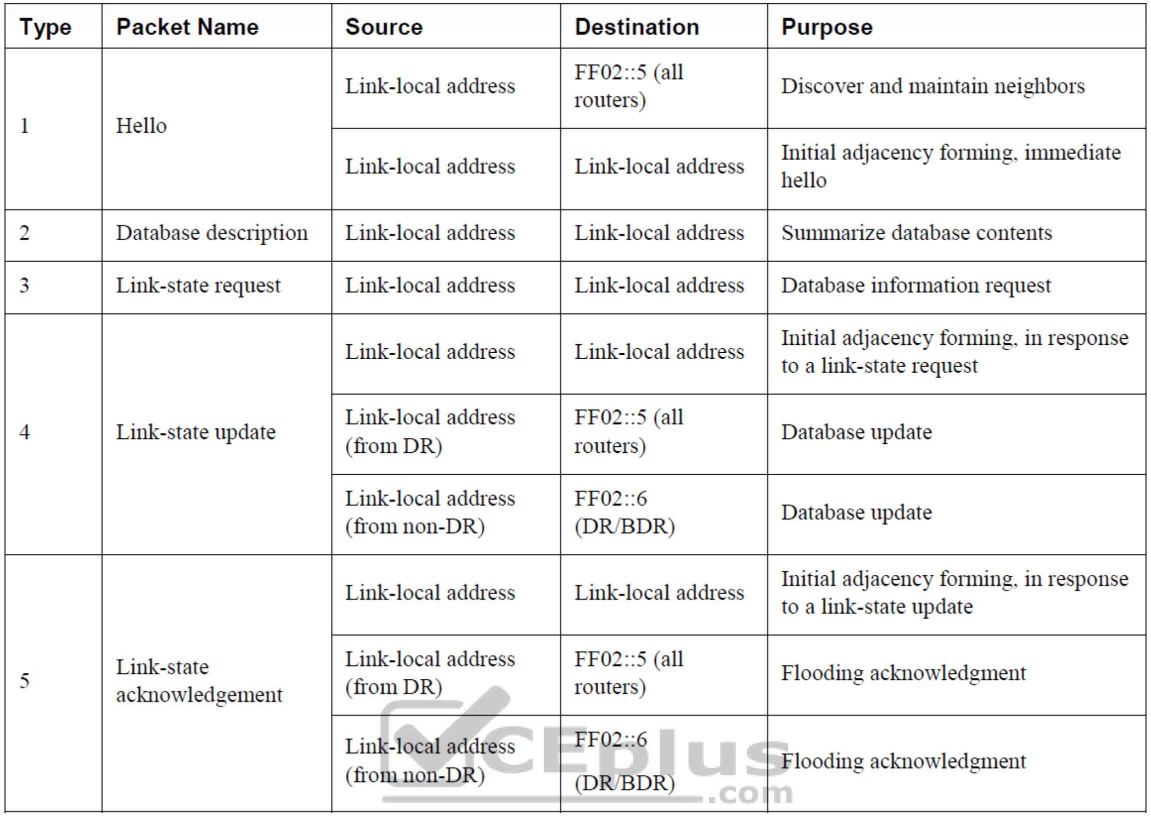
Every router uses the AllSPFRouters multicast address FF02::5 to send OSPF hello messages to routers on the same link. The hello messages are used for neighbor discovery and detecting whether a neighbor relationship is down. The DR and BDR routers also use this address to send link-state update and flooding acknowledgement messages to all routers.

Non-DR/BDR routers send an update or link-state acknowledgement message to the DR and BDR by using the AllDRouters address FF02::6.

OSPFv3 uses the same five packet types and logic as OSPFv2. Table 10-3 shows the name, address, and purpose of each of the five packets types.

Key Topic

Table 10-3 OSPFv3 Packet Types



Type	Packet Name	Source	Destination	Purpose
1	Hello	Link-local address	FF02::5 (all routers)	Discover and maintain neighbors
		Link-local address	Link-local address	Initial adjacency forming, immediate hello
2	Database description	Link-local address	Link-local address	Summarize database contents
3	Link-state request	Link-local address	Link-local address	Database information request
4	Link-state update	Link-local address	Link-local address	Initial adjacency forming, in response to a link-state request
		Link-local address (from DR)	FF02::5 (all routers)	Database update
		Link-local address (from non-DR)	FF02::6 (DR/BDR)	Database update
5	Link-state acknowledgement	Link-local address	Link-local address	Initial adjacency forming, in response to a link-state update
		Link-local address (from DR)	FF02::5 (all routers)	Flooding acknowledgment
		Link-local address (from non-DR)	FF02::6 (DR/BDR)	Flooding acknowledgment

OSPFV3 CONFIGURATION

The process of configuring OSPFv3 involves the following steps:

Step 1. Initialize the routing process. As a prerequisite, **ipv6 unicast-routing** must be enabled on the router. Afterward, the OSPFv3 process is configured with the command **router ospfv3** [*process-id*].

Step 2. Define the router ID. The command **router-id** *router-id* assigns a router ID to the OSPF process. The router ID is a 32-bit value that does not need to match an IPv4 address. It may be any number, as long as the value is unique within the OSPF domain.

OSPFv3 uses the same algorithm as OSPFv2 for dynamically locating the RID. If there are not any IPv4 interfaces available, the RID is set to 0.0.0.0 and does not allow for adjacencies to form.

Step 3. (Optional) Initialize the address family. The address family is initialized within the routing process with the command **address-family** {**ipv6** | **ipv4**} **unicast**. The appropriate address family is enabled automatically when OSPFv3 is enabled on an interface.

Step 4. Enable OSPFv3 on an interface. The interface command **ospfv3** *process-id* **ipv6 area** *area-id* enables the protocol and assigns the interface to an area.

Figure 10-1 displays a simple four-router topology to demonstrate OPSFv3 configuration. Area 0 consists of R1, R2, and R3, and Area 34 contains R3 and R4. R3 is the ABR.

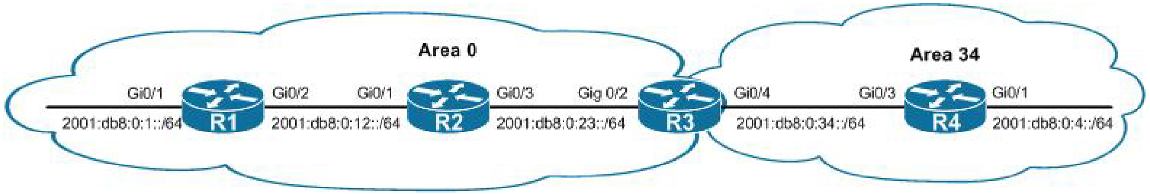


Figure 10-1 OSPFv3 Topology

Example 10-1 provides the OSPFv3 and IPv6 address configurations for R1, R2, R3, and R4. IPv6 link-local addressing has been configured so that all router interfaces reflect their local numbers (for example, R1's interfaces are set to FE80::1) in addition to traditional IPv6 addressing. The link-local addressing is statically configured to assist with any diagnostic output in this chapter. The OSPFv3 configuration has been highlighted in this example.

Example 10-1 IPv6 Addressing and OSPFv3 Configuration

```
R1
interface Loopback0
 ipv6 address 2001:DB8::1/128
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/1
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:0:1::1/64
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/2
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:0:12::1/64
 ospfv3 1 ipv6 area 0
!
router ospfv3 1
 router-id 192.168.1.1

R2
interface Loopback0
 ipv6 address 2001:DB8::2/128
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/1
 ipv6 address FE80::2 link-local
 ipv6 address 2001:DB8:0:12::2/64
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/3
 ipv6 address FE80::2 link-local
 ospfv3 1 ipv6 area 0
!
router ospfv3 1
 router-id 192.168.2.2
```

R3

```
interface Loopback0
 ipv6 address 2001:DB8::3/128
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/2
 ipv6 address FE80::3 link-local
 ipv6 address 2001:DB8:0:23::3/64
 ospfv3 1 ipv6 area 0
!
interface GigabitEthernet0/4
 ipv6 address FE80::3 link-local
 ipv6 address 2001:DB8:0:34::3/64
 ospfv3 1 ipv6 area 34
!
router ospfv3 1
 router-id 192.168.3.3
```

R4

```
interface Loopback0
 ipv6 address 2001:DB8::4/128
 ospfv3 1 ipv6 area 34
!
interface GigabitEthernet0/1
 ipv6 address FE80::4 link-local
 ipv6 address 2001:DB8:0:4::4/64
 ospfv3 1 ipv6 area 34
!
interface GigabitEthernet0/3
 ipv6 address FE80::4 link-local
 ipv6 address 2001:DB8:0:34::4/64
 ospfv3 1 ipv6 area 34
!
router ospfv3 1
 router-id 192.168.4.4
```

Note

Earlier versions of IOS used the commands **ipv6 router ospf** for initialization of the OSPF process and **ipv6 ospf** *process-id* **area** *area-id* for identification of the interface. These commands are considered legacy and should be migrated to the ones used in this book.

Key Topic

OSPFv3 Verification

The commands for viewing OSPFv3 settings and statuses are very similar to those used in OSPFv2; they essentially replace **ip ospf** with **ospfv3 ipv6**. Supporting OSPFv3 requires verifying the OSPFv3 interfaces, neighborship, and the routing table.

For example, to view the neighbor adjacency for OSPFv2, the command **show ip ospf neighbor** is executed, and for OSPFv3, the command **show ospfv3 ipv6 neighbor** is used. Example 10-2 shows this the command executed on R3.

Example 10-2 Identifying R3’s OSPFv3 Neighbors

```
R3# show ospfv3 ipv6 neighbor

          OSPFv3 1 address-family ipv6 (router-id 192.168.3.3)

Neighbor ID   Pri   State         Dead Time    Interface ID    Inter
192.168.2.2    1   FULL/DR       00:00:32     5               Gigab
192.168.4.4    1   FULL/BDR      00:00:33     5               Gigab
```

Example 10-3 shows R1’s GigabitEthernet0/2 OSPFv3-enabled interface status with the command **show ospfv3 interface** [*interface-id*]. Notice that address semantics have been removed compared to OSPFv2. The interface maps to the interface ID value 3 rather than an IP address value, as in OSPFv2. In addition, some helpful topology information describes the link. The local router is the DR (192.168.1.1), and the adjacent neighbor router is the BDR (192.168.2.2).

Example 10-3 Viewing the OSPFv3 Interface Configuration

```
R1# show ospfv3 interface GigabitEthernet0/2
GigabitEthernet0/2 is up, line protocol is up
  Link Local Address FE80::1, Interface ID 3
  Area 0, Process ID 1, Instance ID 0, Router ID 192.168.1.1
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.1, local address FE80::1
  Backup Designated router (ID) 192.168.2.2, local address FE80:
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retrans
    Hello due in 00:00:01
  Graceful restart helper support enabled
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.2.2  (Backup Designated Router
  Suppress hello for 0 neighbor(s)
```

A brief version of the OSPFv3 interface settings can be viewed with the command **show ospfv3 interface brief**. The associated process ID, area, address family (IPv4 or IPv6), interface state, and neighbor count are provided in the output.

Example 10-4 demonstrates this command being executed on the ABR, R3. Notice that some interfaces reside in Area 0, and others reside in Area 34.

Example 10-4 Viewing a Brief Version of OSPFv3 Interfaces

```
R3# show ospfv3 interface brief
Interface    PID   Area            AF         Cost  State Nbrs F/
Lo0          1     0               ipv6       1     LOOP  0/0
Gi0/2        1     0               ipv6       1     BDR   1/1
Gi0/4        1     34              ipv6       1     DR    1/1
```

The OSPFv3 IPv6 routing table is viewed with the command **show ipv6 route ospf**. Intra-area routes are indicated with *O*, and interarea routes are indicated with *OI*.

Example 10-5 shows this command being executed on R1. The forwarding address for the routes is the link-local address of the neighboring router.

Example 10-5 Viewing the OSPFv3 Routes in the IPv6 Routing Table

```
R1# show ipv6 route ospf
! Output omitted for brevity
IPv6 Routing Table - default - 11 entries
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA e
..
O   2001:DB8::2/128 [110/1]
     via FE80::2, GigabitEthernet0/2
O   2001:DB8::3/128 [110/2]
     via FE80::2, GigabitEthernet0/2
OI  2001:DB8::4/128 [110/3]
```

```
     via FE80::2, GigabitEthernet0/2
OI  2001:DB8:0:4::/64 [110/4]
     via FE80::2, GigabitEthernet0/2
O   2001:DB8:0:23::/64 [110/2]
     via FE80::2, GigabitEthernet0/2
OI  2001:DB8:0:34::/64 [110/3]
     via FE80::2, GigabitEthernet0/2
```

Passive Interface

OSPFv3 supports the ability to mark an interface as passive. The command is placed under the OSPFv3 process or under the specific address family. Placing the command under the global process cascades the setting to both address families. An interface is marked as being passive with the command **passive-interface** *interface-id* or globally with **passive-interface default**, and then the interface is marked as active with the command **no passive-interface** *interface-id*.

Example 10-6 shows how to make the LAN interface on R1 explicitly passive and how to make all interfaces passive on R4 while marking the Gi0/3 interface as active.

Example 10-6 Configuring OSPFv3 Passive Interfaces

```
R1(config)# router ospfv3 1
R1(config-router)# passive-interface GigabitEthernet0/1
```

```
R4(config)# router ospfv3 1
R4(config-router)# passive-interface default
22:10:46.838: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 192.168.3.3
R4(config-router)# no passive-interface GigabitEthernet 0/3
```

The active/passive state of an interface is verified by examining the OSPFv3 interface status using the command **show ospfv3 interface** [*interface-id*] and searching for the *Passive* keyword. In Example 10-7, R1 confirms that the Gi0/3 interface is passive.

Example 10-7 Viewing an OSPFv3 Interface State

```
R1# show ospfv3 interface gigabitEthernet 0/1 | include Passive
    No Hellos (Passive interface)
```

Summarization

The ability to summarize IPv6 networks is as important as summarizing routes in IPv4 (and it may even be more important, due to hardware scale limitations). Example 10-8 shows the IPv6 routing table on R4 before summarization is applied on R3.

Example 10-8 R4’s IPv6 Routing Table Before Summarization

```
R4# show ipv6 route ospf | begin Application
       lA - LISP away, a - Application
OI  2001:DB8::1/128 [110/3]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8::2/128 [110/2]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8::3/128 [110/1]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:1::/64 [110/4]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:12::/64 [110/3]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:23::/64 [110/2]
     via FE80::3, GigabitEthernet0/3
```

Summarizing the Area 0 router’s loopback interfaces (2001:db8:0::1/128, 2001:db8:0::2/128, and 2001:db8:0::3/128) removes three routes from the routing table.

Note

A common mistake with summarization of IPv6 addresses is to confuse hex with decimal. We typically perform summarization logic in decimal, and the first and third digits in an octet should not be confused as decimal values. For example, the IPv6 address 2001::1/128 is not 20 and 1 in decimal format. The number 2001::1/128 is 32 and 1.

Key Topic

Summarization of internal OSPFv3 routes follows the same rules as in OSPFv2 and must occur on ABRs. In our topology, R3 summarizes the three loopback addresses into the 2001:db8:0:0::/65 network. Summarization involves the command **area** *area-id* **range** *prefix/prefix-length*, which resides under the address family in the OSPFv3 process.

Example 10-9 shows R3's configuration for summarizing these prefixes.

Example 10-9 IPv6 Summarization

```
R3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# router ospfv3 1
R3(config-router)# address-family ipv6 unicast
R3(config-router-af)# area 0 range 2001:db8:0:0::/65
```

Example 10-10 shows R4's IPv6 routing table after configuring R3 to summarize the Area 0 loopback interfaces. The summary route is highlighted in this example.

Example 10-10 R4's IPv6 Routing Table After Summarization

```
R4# show ipv6 route ospf | begin Application
       lA - LISP away, a - Application
OI  2001:DB8::/65 [110/4]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:1::/64 [110/4]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:12::/64 [110/3]
     via FE80::3, GigabitEthernet0/3
OI  2001:DB8:0:23::/64 [110/2]
     via FE80::3, GigabitEthernet0/3
```

Network Type

OSPFv3 supports the same OSPF network types as OSPFv2. Example 10-11 shows that R2's Gi0/3 interface is set as a broadcast OSPF network type and is confirmed as being in a DR state.

Example 10-11 Viewing the Dynamic Configured OSPFv3 Network Type

```
R2# show ospfv3 interface GigabitEthernet 0/3 | include Network
  Network Type BROADCAST, Cost: 1

R2# show ospfv3 interface brief
Interface     PID   Area            AF         Cost  State Nbrs F
Lo0           1     0               ipv6       1     LOOP  0/0
Gi0/3         1     0               ipv6       1     DR    1/1
Gi0/1         1     0               ipv6       1     BDR   1/1
```

The OSPFv3 network type is changed with the interface parameter command **ospfv3 network** {**point-to-point** | **broadcast**}. Example 10-12 shows the interfaces associated with the 2001:DB8:0:23::/64 network being changed to point-to-point.

Example 10-12 Changing the OSPFv3 Network Type

```
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# interface GigabitEthernet 0/3
R2(config-if)# ospfv3 network point-to-point

R3(config)# interface GigabitEthernet 0/2
R3(config-if)# ospfv3 network point-to-point
```

After typing in the changes, the new settings are verified in Example 10-13. The network is now a point-to-point link, and the interface state shows as P2P for confirmation.

Example 10-13 Viewing the Statically Configured OSPFv3 Network Type

```
R2# show ospfv3 interface GigabitEthernet 0/3 | include Network
  Network Type POINT_TO_POINT, Cost: 1

R2# show ospfv3 interface brief
Interface    PID   Area            AF         Cost  State Nbrs F
Lo0          1     0               ipv6       1     LOOP  0/0
Gi0/3        1     0               ipv6       1     P2P   1/1
Gi0/1        1     0               ipv6       1     BDR   1/1
```

IPV4 SUPPORT IN OSPFV3

OSPFv3 supports multiple address families by setting the instance ID value from the IPv6 reserved range to the IPv4 reserved range (64 to 95) in the link LSAs.

Key Topic

Enabling IPv4 support for OSPFv3 is straightforward:

Step 1. Ensure that the IPv4 interface has an IPv6 address (global or link local) configured. Remember that configuring a global address also places a global address; alternatively, a link-local address can statically be configured.

Step 2. Enable the OSPFv3 process for IPv4 on the interface with the command **ospfv3** *process-id* **ipv4 area** *area-id*.

Using the topology shown in Figure 10-1, IPv4 addressing has been placed onto R1, R2, R3, and R4 using the conventions outlined earlier. Example 10-14 demonstrates the deployment of IPv4 using the existing OSPFv3 deployment.

Example 10-14 Configuration Changes for IPv4 Support

```
R1(config)# interface Loopback 0
R1(config-if)# ospfv3 1 ipv4 area 0
R1(config-if)# interface GigabitEthernet0/1
R1(config-if)# ospfv3 1 ipv4 area 0
R1(config-if)# interface GigabitEthernet0/2
R1(config-if)# ospfv3 1 ipv4 area 0

R2(config)# interface Loopback 0
R2(config-if)# ospfv3 1 ipv4 area 0
R2(config-if)# interface GigabitEthernet0/1
R2(config-if)# ospfv3 1 ipv4 area 0
R2(config-if)# interface GigabitEthernet0/3
R2(config-if)# ospfv3 1 ipv4 area 0

R3(config)# interface Loopback 0
R3(config-if)# ospfv3 1 ipv4 area 0
R3(config-if)# interface GigabitEthernet0/2
R3(config-if)# ospfv3 1 ipv4 area 0
R3(config-if)# interface GigabitEthernet0/4
R3(config-if)# ospfv3 1 ipv4 area 34

R4(config)# interface Loopback 0
R4(config-if)# ospfv3 1 ipv4 area 34
R4(config-if)# interface GigabitEthernet0/1
R4(config-if)# ospfv3 1 ipv4 area 34
R4(config-if)# interface GigabitEthernet0/3
R4(config-if)# ospfv3 1 ipv4 area 34
```

Example 10-15 verifies that the routes were exchanged and installed into the IPv4 RIB.

Example 10-15 Verifying IPv4 Route Exchange with OSPFv3

```
R4# show ip route ospfv3 | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O IA     10.1.1.0/24 [110/4] via 10.34.1.3, 00:00:39, GigabitEthe
O IA     10.12.1.0/24 [110/3] via 10.34.1.3, 00:00:39, GigabitEth
O IA     10.23.1.0/24 [110/2] via 10.34.1.3, 00:00:39, GigabitEth
      192.168.1.0/32 is subnetted, 1 subnets
O IA     192.168.1.1 [110/3] via 10.34.1.3, 00:00:39, GigabitEthe
      192.168.2.0/32 is subnetted, 1 subnets
O IA     192.168.2.2 [110/2] via 10.34.1.3, 00:00:39, GigabitEthe
      192.168.3.0/32 is subnetted, 1 subnets
O IA     192.168.3.3 [110/1] via 10.34.1.3, 00:00:39, GigabitEthe
```

The command **show ospfv3 interface** [**brief**] displays the address families enabled on an interface. When IPv4 and IPv6 are both configured on an interface, an entry appears for each address family. Example 10-16 lists the interfaces and associated address families.

Example 10-16 Listing of OSPFv3 Interfaces and Their Address Families

```
R4# show ospfv3 interface brief
Interface    PID   Area            AF         Cost  State Nbrs F/
Lo0          1     34              ipv4       1     LOOP  0/0
Gi0/1        1     34              ipv4       1     DR    1/1
Gi0/3        1     34              ipv4       1     DR    1/1
Lo0          1     34              ipv6       1     LOOP  0/0
Gi0/1        1     34              ipv6       1     DR    0/0
Gi0/3        1     34              ipv6       1     BDR   1/1
```

Example 10-17 shows how to view the OSPFv3 neighbors to display the neighbors enabled for IPv4 and IPv6 as separate entities.

Example 10-17 Verifying OSPFv3 IPv4 Neighbors

```
R4# show ospfv3 neighbor

          OSPFv3 1 address-family ipv4 (router-id 192.168.4.4)

Neighbor ID     Pri   State           Dead Time   Interface ID
192.168.3.3       1   FULL/BDR        00:00:30    6

          OSPFv3 1 address-family ipv6 (router-id 192.168.4.4)

Neighbor ID     Pri   State           Dead Time   Interface ID
192.168.3.3       1   FULL/DR         00:00:31    6
```

EXAM PREPARATION TASKS

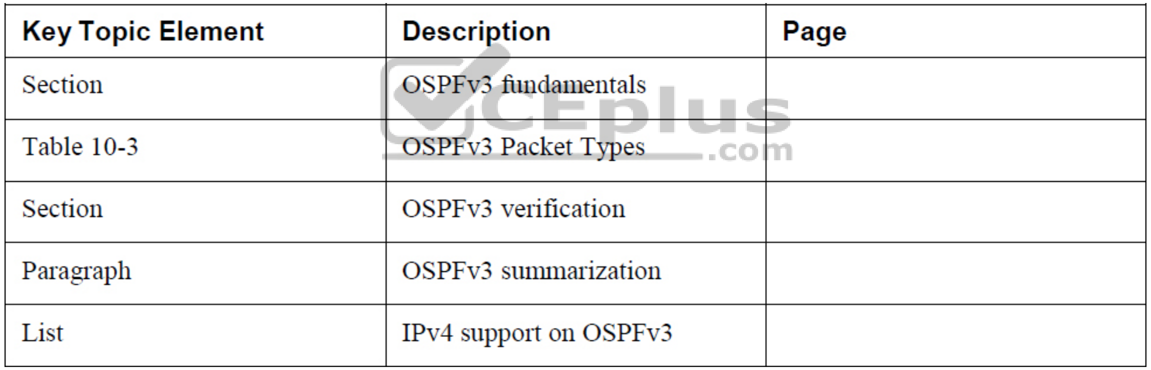
As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 10-4 lists these key topics and the page number on which each is found.

Key Topic

Table 10-4 Key Topics for Chapter 10



Key Topic Element	Description	Page
Section	OSPFv3 fundamentals	
Table 10-3	OSPFv3 Packet Types	
Section	OSPFv3 verification	
Paragraph	OSPFv3 summarization	
List	IPv4 support on OSPFv3	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

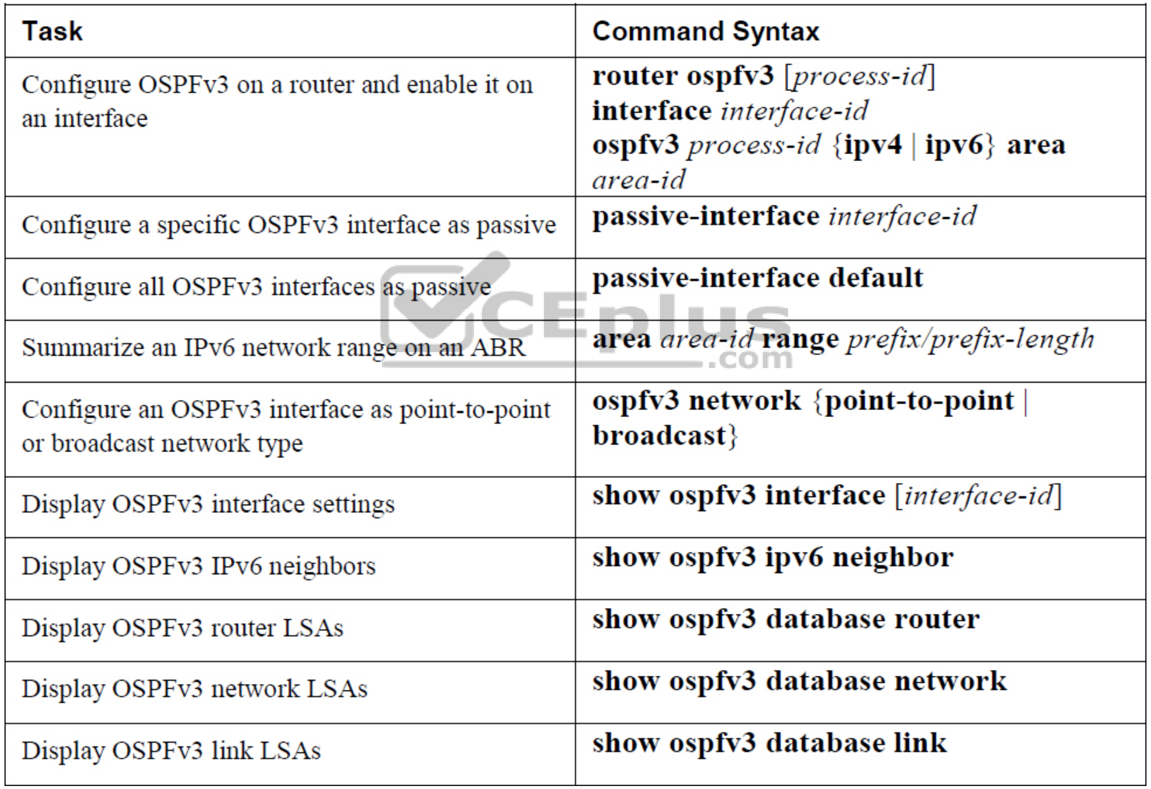
DEFINE KEY TERMS

There are no key terms in this chapter.

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 10-5 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 10-5 Command Reference



Task	Command Syntax
Configure OSPFv3 on a router and enable it on an interface	**router ospfv3** [*process-id*] **interface** *interface-id* **ospfv3** *process-id* {**ipv4** | **ipv6**} **area** *area-id*
Configure a specific OSPFv3 interface as passive	**passive-interface** *interface-id*
Configure all OSPFv3 interfaces as passive	**passive-interface default**
Summarize an IPv6 network range on an ABR	**area** *area-id* **range** *prefix/prefix-length*
Configure an OSPFv3 interface as point-to-point or broadcast network type	**ospfv3 network** {**point-to-point** | **broadcast**}
Display OSPFv3 interface settings	**show ospfv3 interface** [*interface-id*]
Display OSPFv3 IPv6 neighbors	**show ospfv3 ipv6 neighbor**
Display OSPFv3 router LSAs	**show ospfv3 database router**
Display OSPFv3 network LSAs	**show ospfv3 database network**
Display OSPFv3 link LSAs	**show ospfv3 database link**

REFERENCES IN THIS CHAPTER

Chapter 11. BGP

This chapter covers the following subjects:

- **BGP Fundamentals:** This section provides an overview of the fundamentals of the BGP routing protocol.

- **Basic BGP Configuration:** This section walks through the process of configuring BGP to establish a neighbor session and how routes are exchanged between peers.

- **Route Summarization:** This section provides an overview of the how route summarization works with BGP and some of the design considerations with summarization.

- **Multiprotocol BGP for IPv6:** This section explains how BGP provides support for IPv6 routing and configuration.

RFC 1654 defines *Border Gateway Protocol (BGP)* as an EGP standardized path vector routing protocol that provides scalability, flexibility, and network stability.

When BGP was created, the primary design consideration was for IPv4 inter-organization connectivity on public networks like the Internet and on private dedicated networks. BGP is the only protocol used to exchange networks on the Internet, which has more than 780,000 IPv4 routes and continues to grow. Due to the large size of the BGP tables, BGP does not advertise incremental updates or refresh network advertisements as OSPF and IS-IS do. BGP prefers stability within the network, as a link flap could result in route computation for thousands of routes.

This chapter covers the fundamentals of BGP (path attributes, address families, and inter-router communication), BGP configuration, route summarization, and support for IPv6. Chapter 12, "Advanced BGP," explains common scenarios in enterprise environments for BGP, route filtering and manipulation, BGP communities, and the logic BGP uses for identifying a route as the best path.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 11-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 11-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
BGP Fundamentals	1–4
Basic BGP Configuration	5–8
Route Summarization	9
Multiprotocol BGP for IPv6	10

1. Which of the following autonomous systems are private? (Choose two.)

a. 64,512–65,535

b. 65,000–65,535

c. 4,200,000,000–4,294,967,294

d. 4,265,000–4,265,535,016

2. Which BGP attribute must be recognized by all BGP implementations and advertised to other autonomous systems?

a. Well-known mandatory

b. Well-known discretionary

c. Optional transitive

d. Optional non-transitive

3. True or false: BGP supports dynamic neighbor discovery by both routers.

a. True

b. False

4. True or false: A BGP session is always one hop away from a neighbor.

a. True

b. False

5. True or false: The IPv4 address family must be initialized to establish a BGP session with a peer using IPv4 addressing.

a. True

b. False

6. Which command is used to view the BGP neighbors and their hello intervals?

a. show bgp neighbors

b. show bgp *afi safi* **neighbors**

c. show bgp *afi safi* **summary**

d. show *afi* **bgp interface brief**

7. How many tables does BGP use for storing prefixes?

a. One

b. Two

c. Three

d. Four

8. True or false: BGP advertises all its paths for every prefix so that every neighbor can build its own topology table.

a. True

b. False

9. Which BGP command advertises a summary route to prevent link-flap processing by downstream BGP routers?

a. aggregate-address *network subnet-mask* **as-set**

b. aggregate-address *network subnet-mask* **summary-only**

c. summary-address *network subnet-mask*

d. summary-address *network* **mask** *subnet-mask*

10. True or false: The IPv6 address family must be initialized to establish a BGP session with a peer using IPv6 addressing.

a. True

b. False

Answers to the “Do I Know This Already?” quiz:

1. A, D

2. A

3. B

4. B

5. B

6. B

7. C

8. B

9. B

10. A

FOUNDATION TOPICS

BGP FUNDAMENTALS

From the perspective of BGP, an *autonomous system (AS)* is a collection of routers under a single organization's control, using one or more IGPs and common metrics to route packets within the AS. If multiple IGPs or metrics are used within an AS, the AS must appear consistent to external ASs in routing policy. An IGP is not required within an AS; an AS could use BGP as the only routing protocol.

Key Topic

Autonomous System Numbers

An organization requiring connectivity to the Internet must obtain an autonomous system number (ASN). ASNs were originally 2 bytes (16-bit range), which made 65,535 ASNs possible. Due to exhaustion, RFC 4893 expanded the ASN field to accommodate 4 bytes (32-bit range). This allows for 4,294,967,295 unique ASNs, providing quite an increase from the original 65,535 ASNs.

Two blocks of private ASNs are available for any organization to use, as long as they are never exchanged publicly on the Internet. ASNs 64,512–65,535 are private ASNs in the 16-bit ASN range, and 4,200,000,000–4,294,967,294 are private ASNs within the extended 32-bit range.

The *Internet Assigned Numbers Authority (IANA)* is responsible for assigning all public ASNs to ensure that they are globally unique. IANA requires the following items when requesting a public ASN:

- Proof of a publicly allocated network range

- Proof that Internet connectivity is provided through multiple connections

- Need for a unique routing policy from providers

In the event that an organization cannot provide this information, it should use the ASN provided by its service provider.

Note

It is imperative to use only the ASN assigned by IANA, the ASN assigned by your service provider, or a private ASNs. Using another organization’s ASN without permission could result in traffic loss and cause havoc on the Internet.

Key Topic

Path Attributes

BGP uses path attributes (PAs) associated with each network path. The PAs provide BGP with granularity and control of routing policies within BGP. The BGP prefix PAs are classified as follows:

- Well-known mandatory

- Well-known discretionary

- Optional transitive

- Optional non-transitive

Per RFC 4271, well-known attributes must be recognized by all BGP implementations. Well-known mandatory attributes must be included with every prefix advertisement; well-known discretionary attributes may or may not be included with a prefix advertisement.

Optional attributes do not have to be recognized by all BGP implementations. Optional attributes can be set so that they are transitive and stay with the route advertisement from AS to AS. Other PAs are *non-transitive* and cannot be shared from AS to AS. In BGP, the *Network Layer Reachability Information (NLRI)* is a routing update that consists of the network prefix, prefix length, and any BGP PAs for the specific route.

Loop Prevention

BGP is a path vector routing protocol and does not contain a complete topology of the network, as link-state routing protocols do. BGP behaves like distance vector protocols, ensuring that a path is loop free.

Key Topic

The BGP attribute AS_Path is a well-known mandatory attribute and includes a complete list of all the ASNs that the prefix advertisement has traversed from its source AS. AS_Path is used as a loop-prevention mechanism in BGP. If a BGP router receives a prefix advertisement with its AS listed in the AS_Path attribute, it discards the prefix because the router thinks the advertisement forms a loop.

Figure 11-1 shows the loop-prevention mechanism:

- AS 100 advertises the 172.16.1.0/24 prefix to AS 200.

- AS 200 advertises the prefix to AS 400, which then advertises the prefix to AS 300.

- AS 300 advertises the prefix back to AS 100 with an AS_Path of 300 400 200 100. AS 100 see itself in the AS_Path variable and discards the prefix.

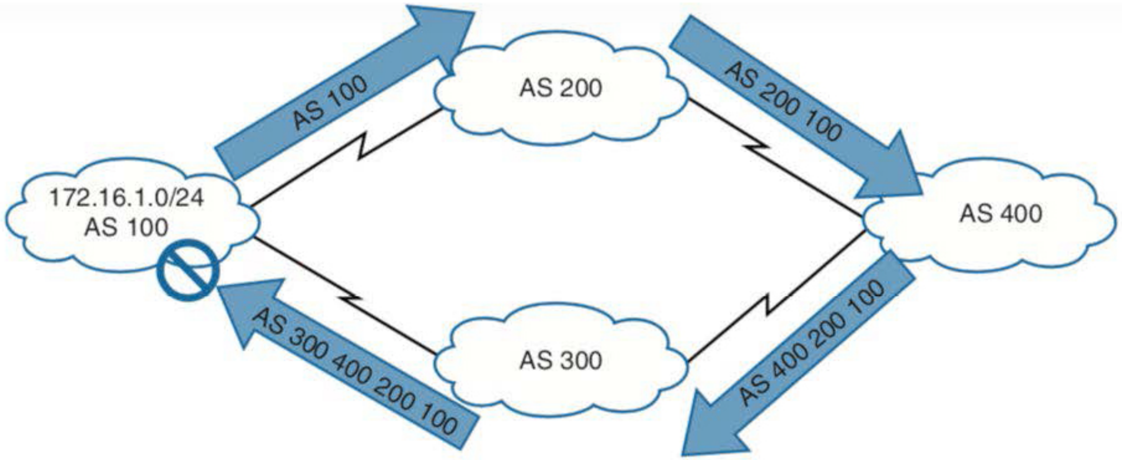


Figure 11-1 Path Vector Loop Prevention

Address Families

Originally, BGP was intended for routing of IPv4 prefixes between organizations, but RFC 2858 added Multi-Protocol BGP (MP-BGP) capability by adding an extension called the address family identifier (AFI). An address family correlates to a specific network protocol, such as IPv4 or IPv6, and additional granularity is provided through a subsequent address-family identifier (SAFI) such as unicast or multicast. MBGP achieves this separation by using the BGP path attributes (PAs) MP_REACH_NLRI and MP_UNREACH_NLRI. These attributes are carried inside BGP update messages and are used to carry network reachability information for different address families.

Note

Some network engineers refer to Multiprotocol BGP as MP-BGP, and other network engineers use the term MBGP. Both terms refer to the same thing.

Key Topic

Every address family maintains a separate database and configuration for each protocol (address family + sub-address family) in BGP. This allows for a routing policy in one address family to be different from a routing policy in a different address family, even though the router uses the same BGP session with the other router. BGP includes an AFI and SAFI with every route advertisement to differentiate between the AFI and SAFI databases.

Key Topic

Inter-Router Communication

BGP does not use hello packets to discover neighbors, as do IGP protocols, and it cannot discover neighbors dynamically. BGP was designed as an inter-autonomous routing protocol, implying that neighbor adjacencies should not change frequently and are coordinated. BGP neighbors are defined by IP address.

BGP uses TCP port 179 to communicate with other routers. TCP allows for handling of fragmentation, sequencing, and reliability (acknowledgement and retransmission) of communication packets. Most recent implementations of BGP set the do-not-fragment (DF) bit to prevent fragmentation and rely on path MTU discovery.

IGPs follow the physical topology because the sessions are formed with hellos that cannot cross network boundaries (that is, single hop only). BGP uses TCP, which is capable of crossing network boundaries (that is, multi-hop capable). While BGP can form neighbor adjacencies that are directly connected, it can also form adjacencies that are multiple hops away.

A BGP session refers to the established adjacency between two BGP routers. Multi-hop sessions require that the router use an underlying route installed in the RIB (static or from any routing protocol) to establish the TCP session with the remote endpoint.

In Figure 11-2, R1 is able to establish a direct BGP session with R2. In addition, R2 is able to establish a BGP session with R4, even though it passes through R3. R1 and R2 use a directly connected route to locate each other. R2 uses a static route to reach the 10.34.1.0/24 network, and R4 has a static route to reach the

10.23.1.0/24 network. R3 is unaware that R2 and R4 have established a BGP session even though the packets flow through R3.

Key Topic

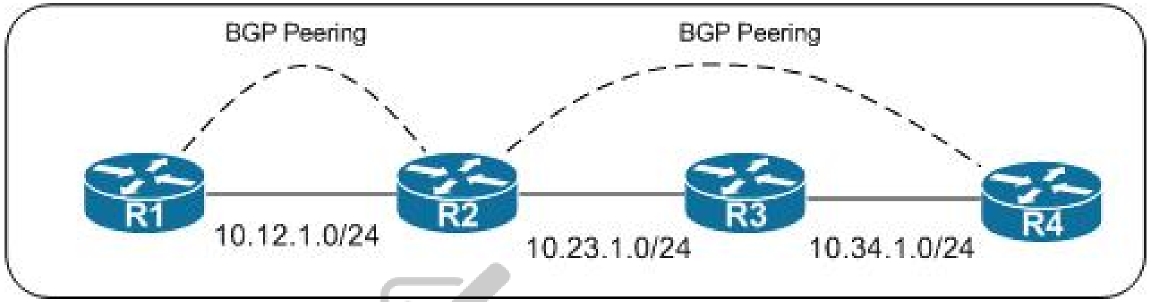


Figure 11-2 BGP Single- and Multi-Hop Sessions

Note

BGP neighbors connected to the same network use the ARP table to locate the IP address of the peer. Multi-hop BGP sessions require routing table information for finding the IP address of the peer. It is common to have a static route or an IGP running between iBGP neighbors for providing the topology path information to establish the BGP TCP session. A default route is not sufficient to establish a multi-hop BGP session.

BGP can be thought of as a control plane routing protocol or as an application because it allows for the exchange of routes with a peer that is multiple hops away. BGP routers do not have to be in the data plane (path) to exchange prefixes, but all routers in the data path need to know all the routes that will be forwarded through them.

Key Topic

BGP Session Types

BGP sessions are categorized into two types:

- **Internal BGP (iBGP):** Sessions established with an iBGP router that are in the same AS or that participate in the same BGP confederation. iBGP prefixes are assigned an administrative distance (AD) of 200 upon installation in the router's RIB.

- **External BGP (eBGP):** Sessions established with a BGP router that are in a different AS. eBGP prefixes are assigned an AD of 20 upon installation in the router's RIB.

The following sections review these two types of BGP sessions.

iBGP

The need for BGP within an AS typically occurs when multiple routing policies are required or when transit connectivity is provided between autonomous systems. In Figure 11-3, AS 65200 provides transit connectivity to AS 65100 and AS 65300. AS 65100 connects at R2, and AS 65300 connects at R4.

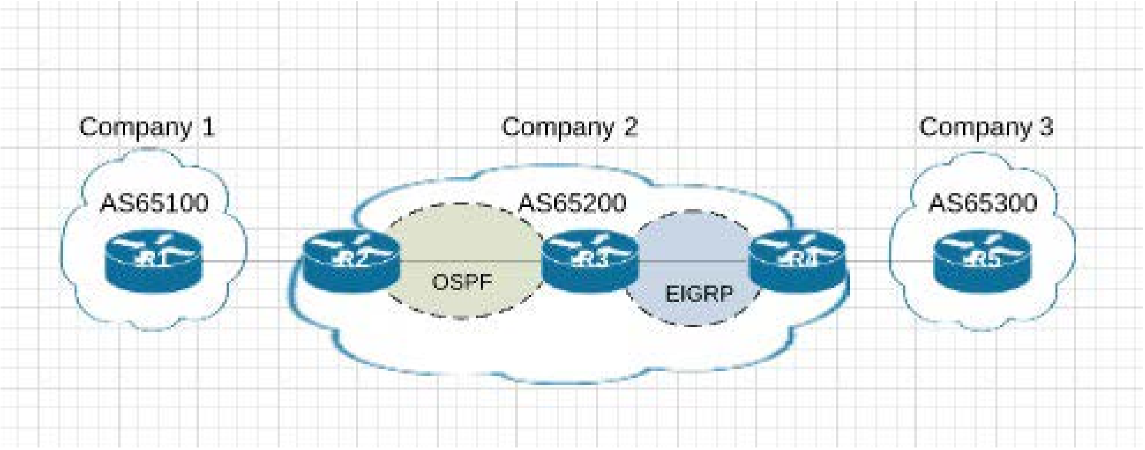
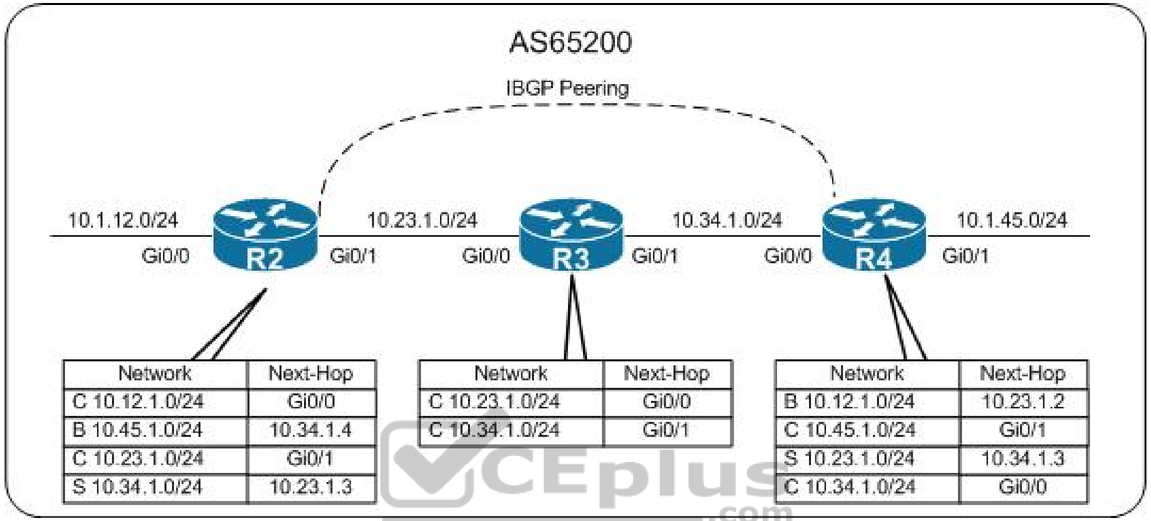


Figure 11-3 AS 65200 Providing Transit Connectivity

R2 could form an iBGP session directly with R4, but R3 would not know where to route traffic from AS 65100 or AS 65300 when traffic from either AS reaches R3, as shown in Figure 11-4, because R3 would not have the appropriate route forwarding information for the destination traffic.



Network	Next-Hop
C 10.12.1.0/24	Gi0/0
B 10.45.1.0/24	10.34.1.4
C 10.23.1.0/24	Gi0/1
S 10.34.1.0/24	10.23.1.3

Network	Next-Hop
C 10.23.1.0/24	Gi0/0
C 10.34.1.0/24	Gi0/1

Network	Next-Hop
B 10.12.1.0/24	10.23.1.2
C 10.45.1.0/24	Gi0/1
S 10.23.1.0/24	10.34.1.3
C 10.34.1.0/24	Gi0/0

Figure 11-4 iBGP Prefix Advertisement Behavior

You might assume that redistributing the BGP table into an IGP overcomes the problem, but this not a viable solution for several reasons:

- **Scalability:** The Internet at the time of this writing has 780,000+ IPv4 network prefixes and continues to increase in size. IGPs cannot scale to that level of routes.

- **Custom routing:** Link-state protocols and distance vector routing protocols use metric as the primary method for route selection. IGP protocols always use this routing pattern for path selection. BGP uses multiple steps to identify the best path and allows for BGP path attributes to manipulate the path for a specific prefix (NLRI). The path could be longer, and that would normally be deemed suboptimal from an IGP's perspective.

- **Path attributes:** All the BGP path attributes cannot be maintained within IGP protocols. Only BGP is capable of maintaining the path attribute as the prefix is advertised from one edge of the AS to the other edge.

Establishing iBGP sessions between all the same routers (R2, R3, and R4) in a full mesh allows for proper forwarding between autonomous systems.

Note

Service providers provide transit connectivity. Enterprise organizations are consumers and should not provide transit connectivity between autonomous systems across the Internet.

Key Topic

eBGP

eBGP peerings are the core component of BGP on the Internet. eBGP involves the exchange of network prefixes between autonomous systems. The following behaviors are different on eBGP sessions than on iBGP sessions:

- Time-to-live (TTL) on eBGP packets is set to 1 by default. eBGP packets drop in transit if a multi-hop BGP session is attempted. (TTL on iBGP packets is set to 255, which allows for multi-hop sessions.)

- The advertising router modifies the BGP next-hop address to the IP address sourcing the BGP connection.

- The advertising router prepends its ASN to the existing AS_Path variable.

- The receiving router verifies that the AS_Path variable does not contain an ASN that matches the local routers. BGP discards the NLRI if it fails the AS_Path loop prevention check.

The configurations for eBGP and iBGP sessions are fundamentally the same except that the ASN in the **remote-as** statement is different from the ASN defined in the BGP process.

Figure 11-5 shows the eBGP and iBGP sessions that would be needed between the routers to allow connectivity between AS 65100 and AS 65300. Notice that AS 65200 R2 establishes an iBGP session with R4 to overcome the loop-prevention behavior of iBGP learned routes.

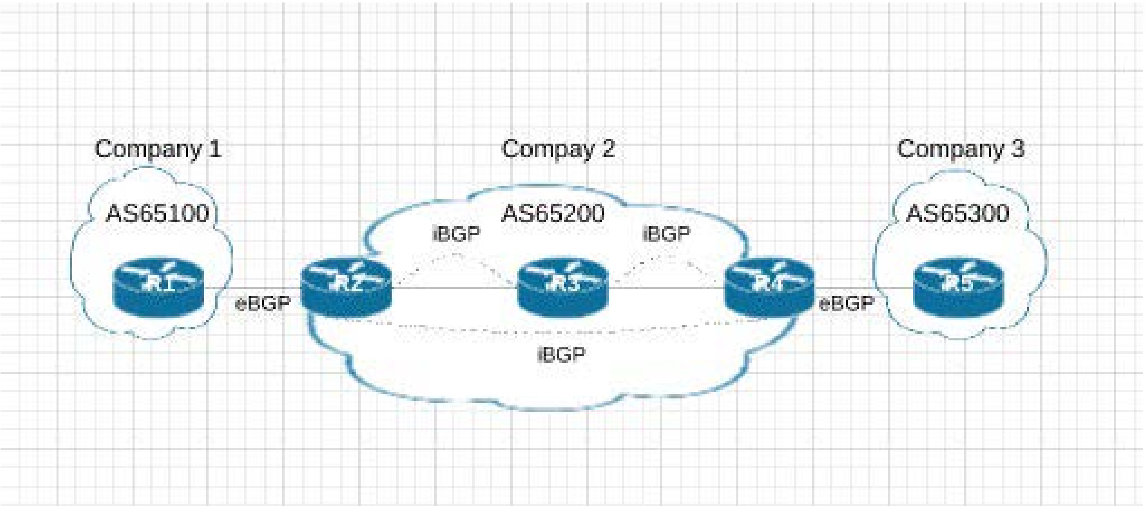


Figure 11-5 eBGP and iBGP Sessions

BGP Messages

BGP communication uses four message types, as shown in Table 11-2.

Table 11-2 BGP Packet Types

Type	Name	Functional Overview
1	OPEN	Sets up and establishes BGP adjacency
2	UPDATE	Advertises, updates, or withdraws routes
3	NOTIFICATION	Indicates an error condition to a BGP neighbor
4	KEEPALIVE	Ensures that BGP neighbors are still alive

- **OPEN:** An OPEN message is used to establish a BGP adjacency. Both sides negotiate session capabilities before BGP peering is established. The OPEN message contains the BGP version number, the ASN of the originating router, the hold time, the BGP identifier, and other optional parameters that establish the session capabilities.

- **Hold time:** The hold time attribute sets the hold timer, in seconds, for each BGP neighbor. Upon receipt of an UPDATE or KEEPALIVE, the hold timer resets to the initial value. If the hold timer reaches zero, the BGP session is torn down, routes from that neighbor are removed, and an appropriate update route withdraw message is sent to other BGP neighbors for the affected prefixes. The hold time is a heartbeat mechanism for BGP neighbors to ensure that a neighbor is healthy and alive.

When establishing a BGP session, the routers use the smaller hold time value contained in the two routers' OPEN messages. The hold time value must be at least 3 seconds, or is set to 0 to disable keepalive messages. For Cisco routers, the default hold timer is 180 seconds.

- **BGP identifier:** The *BGP router ID (RID)* is a 32-bit unique number that identifies the BGP router in the advertised prefixes. The RID can be used as a loop-prevention mechanism for routers advertised within an autonomous system. The RID can be set manually or dynamically for BGP. A nonzero value must be set in order for routers to become neighbors.

- **KEEPALIVE:** BGP does not rely on the TCP connection state to ensure that the neighbors are still alive. KEEPALIVE messages are exchanged every one-third of the hold timer agreed upon between the two BGP routers. Cisco devices have a default hold time of 180 seconds, so the default keepalive interval is 60 seconds. If the hold time is set to 0, then no keepalive messages are sent between the BGP neighbors.

- **UPDATE:** An UPDATE message advertises any feasible routes, withdraws previously advertised routes, or can do both. An UPDATE message includes the Network Layer Reachability Information (NLRI), such as the prefix and associated BGP PAs, when advertising prefixes. Withdrawn NLRIs include only the prefix. An UPDATE message can act as a keepalive to reduce unnecessary traffic.

- **NOTIFICATION:** A NOTIFICATION message is sent when an error is detected with the BGP session, such as a hold timer expiring, neighbor capabilities changing, or a BGP session reset being requested. This causes the BGP connection to close.

BGP Neighbor States

BGP forms a TCP session with neighbor routers called *peers*. BGP uses the finite-state machine (FSM) to maintain a table of all BGP peers and their operational status. The BGP session may report the following states:

- Idle

- Connect

- Active

- OpenSent

- OpenConfirm

- Established

Figure 11-6 shows the BGP FSM and the states, listed in the order used in establishing a BGP session.

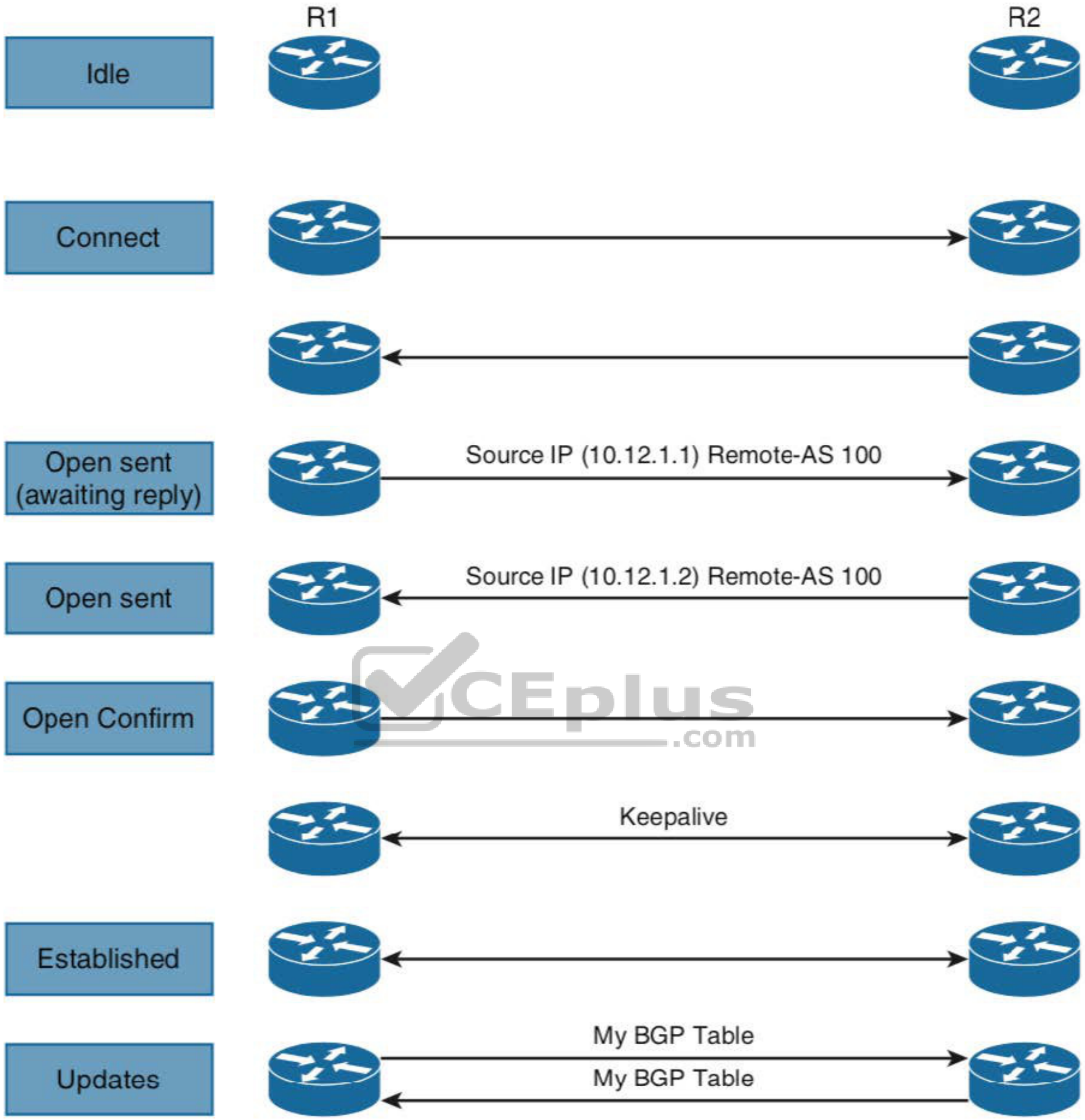


Figure 11-6 BGP Neighbor States with Session Establishment

Idle

Idle is the first stage of the BGP FSM. BGP detects a start event and tries to initiate a TCP connection to the BGP peer and also listens for a new connection from a peer router.

If an error causes BGP to go back to the Idle state for a second time, ConnectRetryTimer is set to 60 seconds and must decrement to zero before the connection can be initiated again. Further failures to leave the Idle state result in the ConnectRetryTimer doubling in length from the previous time.

Connect

In the Connect state, BGP initiates the TCP connection. If the three-way TCP handshake is completed, the established BGP session process resets ConnectRetryTimer and sends the Open message to the neighbor; it then changes to the OpenSent state.

If the ConnectRetryTimer depletes before this stage is complete, a new TCP connection is attempted, ConnectRetryTimer is reset, and the state is moved to Active. If any other input is received, the state is changed to Idle.

During this stage, the neighbor with the higher IP address manages the connection. The router initiating the request uses a dynamic source port, but the destination port is always 179.

Example 11-1 shows an established BGP session using the command **show tcp brief** to displays the active TCP sessions between a router. Notice that the TCP source port is 179 and the destination port is 59884 on R1; the ports are opposite on R2.

Example 11-1 An Established BGP Session

```
R1# show tcp brief
TCB       Local Address                         Foreign Address
F6F84258  10.12.1.1.59884                      10.12.1.2.179

R2# show tcp brief
TCB       Local Address                         Foreign Address
EF153B88  10.12.1.2.59884                       10.12.1.1.179
```

Active

In the Active state, BGP starts a new three-way TCP handshake. If a connection is established, an Open message is sent, the hold timer is set to 4 minutes, and the state moves to OpenSent. If this attempt for TCP connection fails, the state moves back to the Connect state, and ConnectRetryTimer is reset.

OpenSent

In the OpenSent state, an Open message has been sent from the originating router and is awaiting an Open message from the other router. Once the originating router receives the OPEN message from the other router, both OPEN messages are checked for errors. The following items are examined:

- BGP versions must match.

- The source IP address of the OPEN message must match IP address that is configured for the neighbor.

- The AS number in the OPEN message must match what is configured for the neighbor.

- BGP identifiers (RIDs) must be unique. If a RID does not exist, this condition is not met.

- Security parameters (such as password and TTL) must be set appropriately.

If the OPEN messages do not have any errors, the hold time is negotiated (using the lower value), and a KEEPALIVE message is sent (assuming that the value is not set to 0). The connection state is then moved to OpenConfirm. If an error is found in the OPEN message, a NOTIFICATION message is sent, and the state is moved back to Idle.

If TCP receives a disconnect message, BGP closes the connection, resets ConnectRetryTimer, and sets the state to Active. Any other input in this process results in the state moving to Idle.

OpenConfirm

In the OpenConfirm state, BGP waits for a KEEPALIVE or NOTIFICATION message. Upon receipt of a neighbor’s KEEPALIVE message, the state is moved to Established. If the hold timer expires, a stop event occurs, or a NOTIFICATION message is received, the state is moved to Idle.

Established

In the Established state, the BGP session is established. BGP neighbors exchange routes using UPDATE messages. As UPDATE and KEEPALIVE messages are received, the hold timer is reset. If the hold timer expires, an error is detected, and BGP moves the neighbor back to the Idle state.

Key Topic

BASIC BGP CONFIGURATION

When configuring BGP, it is best to think of the configuration from a modular perspective. BGP router configuration requires the following components:

- **BGP session parameters:** BGP session parameters provide settings that involve establishing communication to the remote BGP neighbor. Session settings include the ASN of the BGP peer, authentication, and keepalive timers.

- **Address family initialization:** The address family is initialized under the BGP router configuration mode. Network advertisement and summarization occur within the address family.

- **Activate the address family on the BGP peer:** In order for a session to initiate, one address family for a neighbor must be activated. The router's IP address is added to the neighbor table, and BGP attempts to establish a BGP session or accepts a BGP session initiated from the peer router

The following steps show how to configure BGP:

Step 1. Initialize the BGP routing process with the global command **router bgp** *as-number*.

Step 2. (Optional) Statically define the BGP router ID (RID). The dynamic RID allocation logic uses the highest IP address of the any *up* loopback interfaces. If there is not an *up* loopback interface, then the highest IP address of any active *up* interfaces becomes the RID when the BGP process initializes.

To ensure that the RID does not change, a static RID is assigned (typically representing an IPv4 address that resides on the router, such as a loopback address). Any IPv4 address can be used, including IP addresses not configured on the router. Statically configuring the BGP RID is a best practice and involves using the command **bgp router-id** *router-id*.

When the router ID changes, all BGP sessions reset and need to be reestablished.

Step 3. Identify the BGP neighbor's IP address and autonomous system number with the BGP router configuration command **neighbor** *ip-address* **remote-as** *as-number*. It is important to understand the traffic flow of BGP packets between peers. The source IP address of the BGP packets still reflects the IP address of the outbound interface. When a BGP packet is received, the router correlates the source IP address of the packet to the IP address configured for that neighbor. If the BGP packet source does not match an entry in the neighbor table, the packet cannot be associated to a neighbor and is discarded.

Note

IOS activates the IPv4 address family by default. This can simplify the configuration in an IPv4 environment because steps 4 and 5 are optional but may cause confusion when working with other address families. The BGP router configuration command **no bgp default ip4-unicast** disables the automatic activation of the IPv4 AFI so that steps 4 and 5 are required.

Step 4. Initialize the address family with the BGP router configuration command **address-family** *afi safi*. Examples of *afi* values are IPv4 and IPv6, and examples of *safi* values are unicast and multicast.

Step 5. Activate the address family for the BGP neighbor with the BGP address family configuration command **neighbor** *ip-address* **activate**.

Note

On IOS and IOS XE devices, the default subsequent address family identifier (SAFI) for the IPv4 and IPv6 address families is unicast and is optional.

Figure 11-7 shows a topology for a simple BGP configuration.

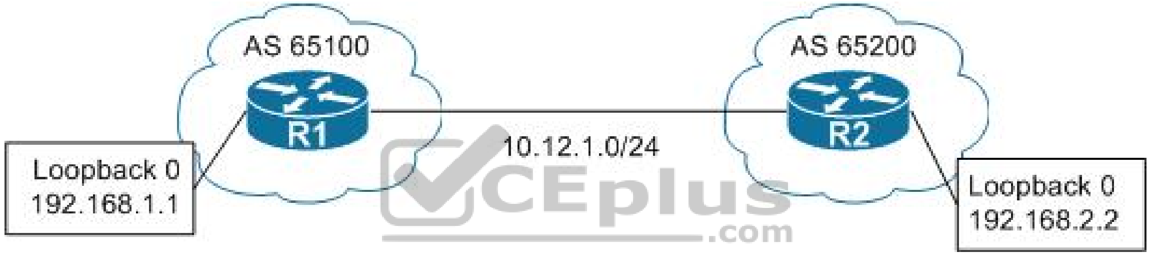


Figure 11-7 Simple BGP Topology

Example 11-2 shows how to configure R1 and R2 using the IOS default and optional IPv4 AFI modifier CLI syntax. R1 is configured with the default IPv4 address family enabled, and R2 disables IOS's default IPv4 address family and manually activates it for the specific neighbor 10.12.1.1.

Example 11-2 Configuring Basic BGP on IOS

```
R1 (Default IPv4 Address-Family Enabled)
router bgp 65100
 neighbor 10.12.1.2 remote-as 65200
```

```
R2 (Default IPv4 Address-Family Disabled)
router bgp 65200
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 !
 address-family ipv4
  neighbor 10.12.1.1 activate
 exit-address-family
```

Key Topic

Verification of BGP Sessions

The BGP session is verified with the command **show bgp** *afi safi* **summary**. Example 11-3 shows the IPv4 BGP unicast summary. Notice that the BGP RID and table version are the first components shown. The Up/Down column indicates that the BGP session is up for over 5 minutes.

Note

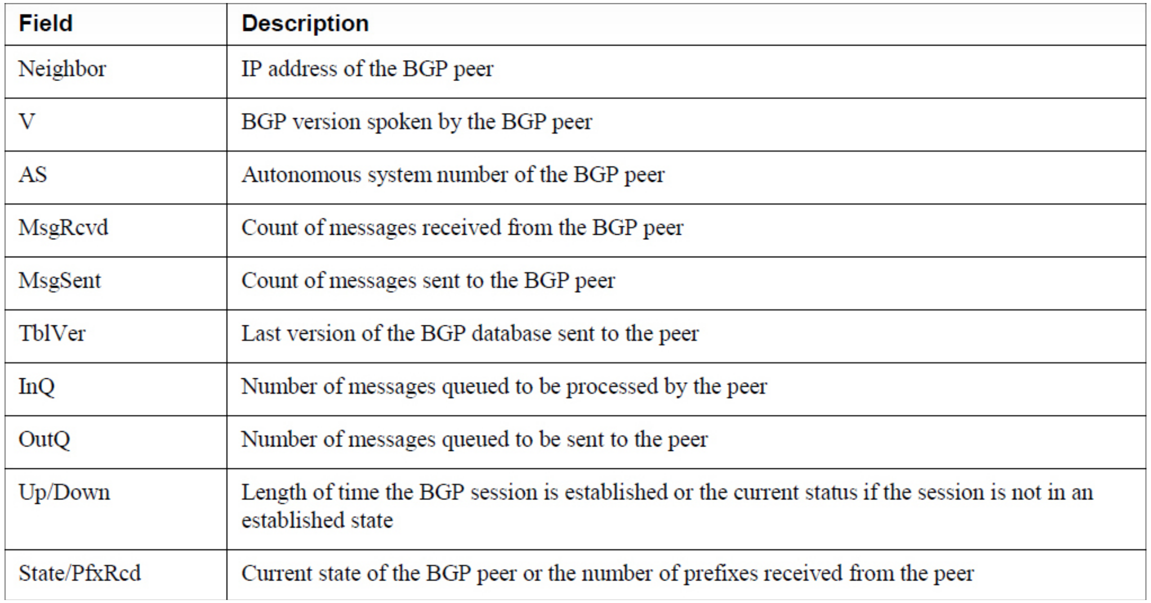
Earlier commands like **show ip bgp summary** came out before MBGP and do not provide a structure for the current multiprotocol capabilities within BGP. Using the AFI and SAFI syntax ensures consistency for the commands, regardless of information exchanged by BGP. This will become more apparent as engineers work with address families like IPv6, VPNv4, and VPNv6.

Example 11-3 Verifying the BGP IPv4 Session Summary

```
R1# show bgp ipv4 unicast summary
BGP router identifier 192.168.2.2, local AS number 65200
BGP table version is 1, main routing table version 1

Neighbor        V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Dow
10.12.1.2       4  65200       8       9        1    0    0 00:05:
```

Table 11-3 explains the fields of output displayed in a BGP table (as in Example 11-3).

Table 11-3 BGP Summary Fields

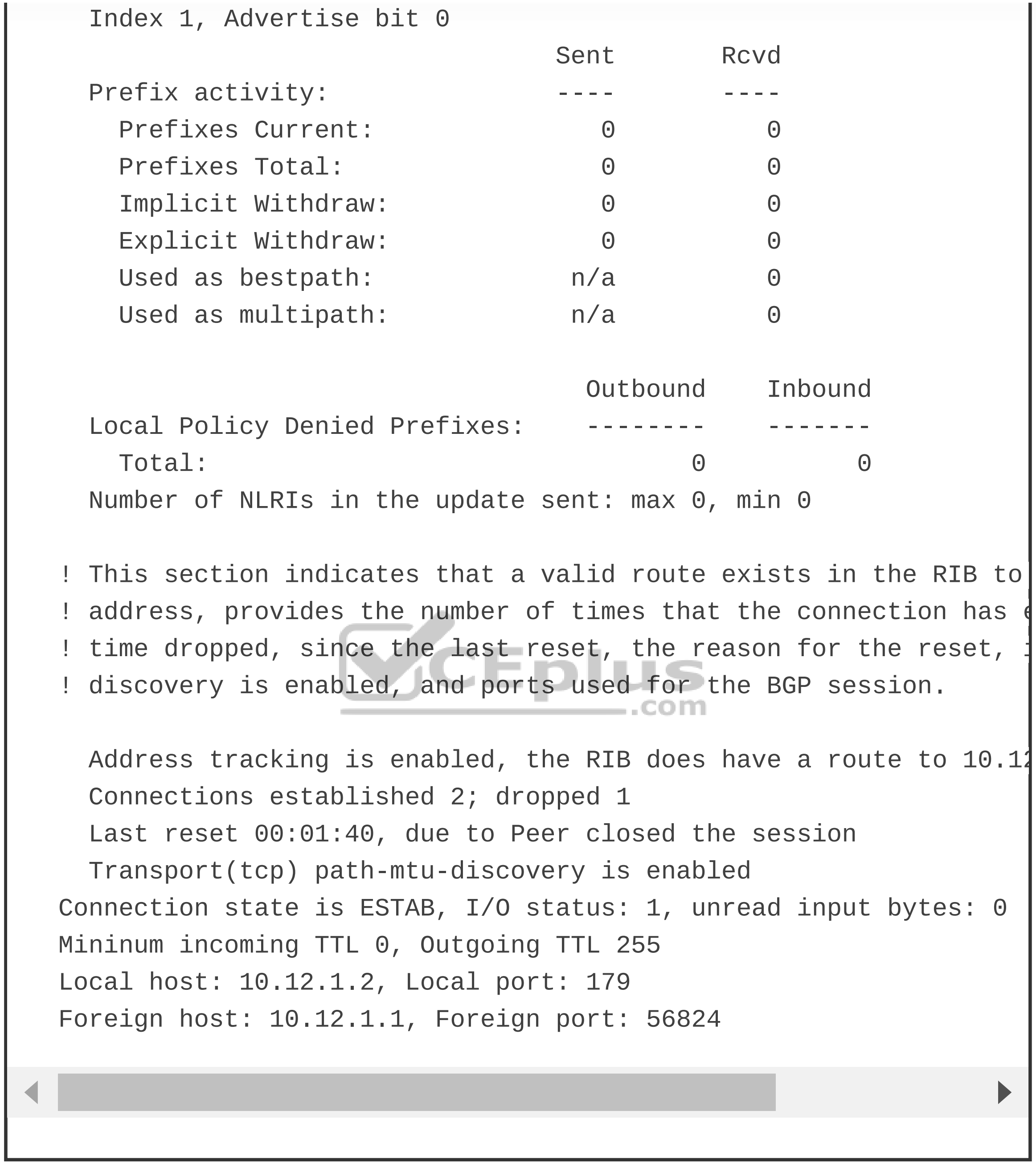


Field	Description
Neighbor	IP address of the BGP peer
V	BGP version spoken by the BGP peer
AS	Autonomous system number of the BGP peer
MsgRcvd	Count of messages received from the BGP peer
MsgSent	Count of messages sent to the BGP peer
TblVer	Last version of the BGP database sent to the peer
InQ	Number of messages queued to be processed by the peer
OutQ	Number of messages queued to be sent to the peer
Up/Down	Length of time the BGP session is established or the current status if the session is not in an established state
State/PfxRcd	Current state of the BGP peer or the number of prefixes received from the peer

BGP neighbor session state, timers, and other essential peering information is available with the command **show bgp** *afi safi* **neighbors** *ip-address*, as shown in Example 11-4.

Example 11-4 BGP IPv4 Neighbor Output

```
R2# show bgp ipv4 unicast neighbors 10.12.1.1
! Output ommitted for brevity

! The first section provides the neighbor's IP address, remote-as
! the neighbor is 'internal' or 'external', the neighbor's BGP ve
! session state, and timers.

BGP neighbor is 10.12.1.1, remote AS65100, external link
  BGP version 4, remote router ID 192.168.1.1
  BGP state = Established, up for 00:01:04
  Last read 00:00:10, last write 00:00:09, hold is 180, keepaliv
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
! This second section indicates the capabilities of the BGP neigh
! address-families configured on the neighbor.
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

! This section provides a list of the BGP packet types that have
! or sent to the neighbor router.
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:             2          2
    Route Refresh:          0          0
    Total:                  4          3
  Default minimum time between advertisement runs is 0 seconds

! This section provides the BGP table version of the IPv4 Unicas
! family. The table version is not a 1-to-1 correlation with rout
! route change can occur during a revision change. Notice the Pr
! columns in this section.

For address family: IPv4 Unicast
  Session: 10.12.1.1
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
```



```
    Index 1, Advertise bit 0
                                    Sent       Rcvd
    Prefix activity:                ----       ----
      Prefixes Current:                0          0
      Prefixes Total:                  0          0
      Implicit Withdraw:               0          0
      Explicit Withdraw:               0          0
      Used as bestpath:              n/a          0
      Used as multipath:             n/a          0

                                      Outbound    Inbound
    Local Policy Denied Prefixes:     --------    -------
      Total:                                 0          0
    Number of NLRIs in the update sent: max 0, min 0

  ! This section indicates that a valid route exists in the RIB to
  ! address, provides the number of times that the connection has e
  ! time dropped, since the last reset, the reason for the reset, i
  ! discovery is enabled, and ports used for the BGP session.

    Address tracking is enabled, the RIB does have a route to 10.12
    Connections established 2; dropped 1
    Last reset 00:01:40, due to Peer closed the session
    Transport(tcp) path-mtu-discovery is enabled
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
  Mininum incoming TTL 0, Outgoing TTL 255
  Local host: 10.12.1.2, Local port: 179
  Foreign host: 10.12.1.1, Foreign port: 56824
```

Key Topic

Prefix Advertisement

BGP **network** statements do not enable BGP for a specific interface; instead, they identify specific network prefixes to be installed into the BGP table, known as the *Loc-RIB table*.

After configuring a BGP **network** statement, the BGP process searches the global RIB for an exact network prefix match. The network prefix can be for a connected network, a secondary connected network, or any route from a routing protocol. After verifying that the **network** statement matches a prefix in the global RIB, the prefix is installed into the BGP Loc-RIB table. As the BGP prefix is installed into the Loc-RIB table, the following BGP PAs are set, depending on the RIB prefix type:

- **Connected network:** The next-hop BGP attribute is set to 0.0.0.0, the BGP origin attribute is set to i (IGP), and the BGP weight is set to 32,768.

- **Static route or routing protocol:** The next-hop BGP attribute is set to the next-hop IP address in the RIB, the BGP origin attribute is set to i (IGP), the BGP weight is set to 32,768, and the MED is set to the IGP metric.

Not every route in the Loc-RIB table is advertised to a BGP peer. All routes in the Loc-RIB table use the following process for advertisement to BGP peers.

Step 1. Pass a validity check. Verify that the NRLI is valid and that the next-hop address is resolvable in the global RIB. If the NRLI fails, the NLRI remains but does not process further.

Step 2. Process outbound neighbor route policies. After processing, if a route was not denied by the outbound policies, the route is maintained in the Adj-RIB-Out table for later reference.

Step 3. Advertise the NLRI to BGP peers. If the NLRI's next-hop BGP PA is 0.0.0.0, then the next-hop address is changed to the IP address of the BGP session.

Figure 11-8 illustrates the concept of installing the network prefix from localized BGP network advertisements to the BGP table.

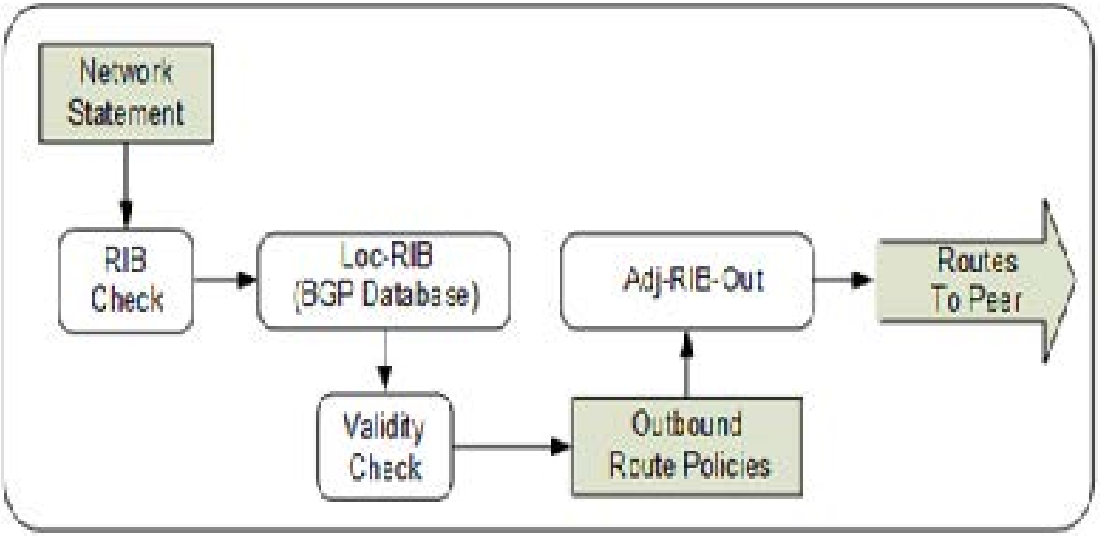


Figure 11-8 BGP Database Processing of Local Route Advertisements

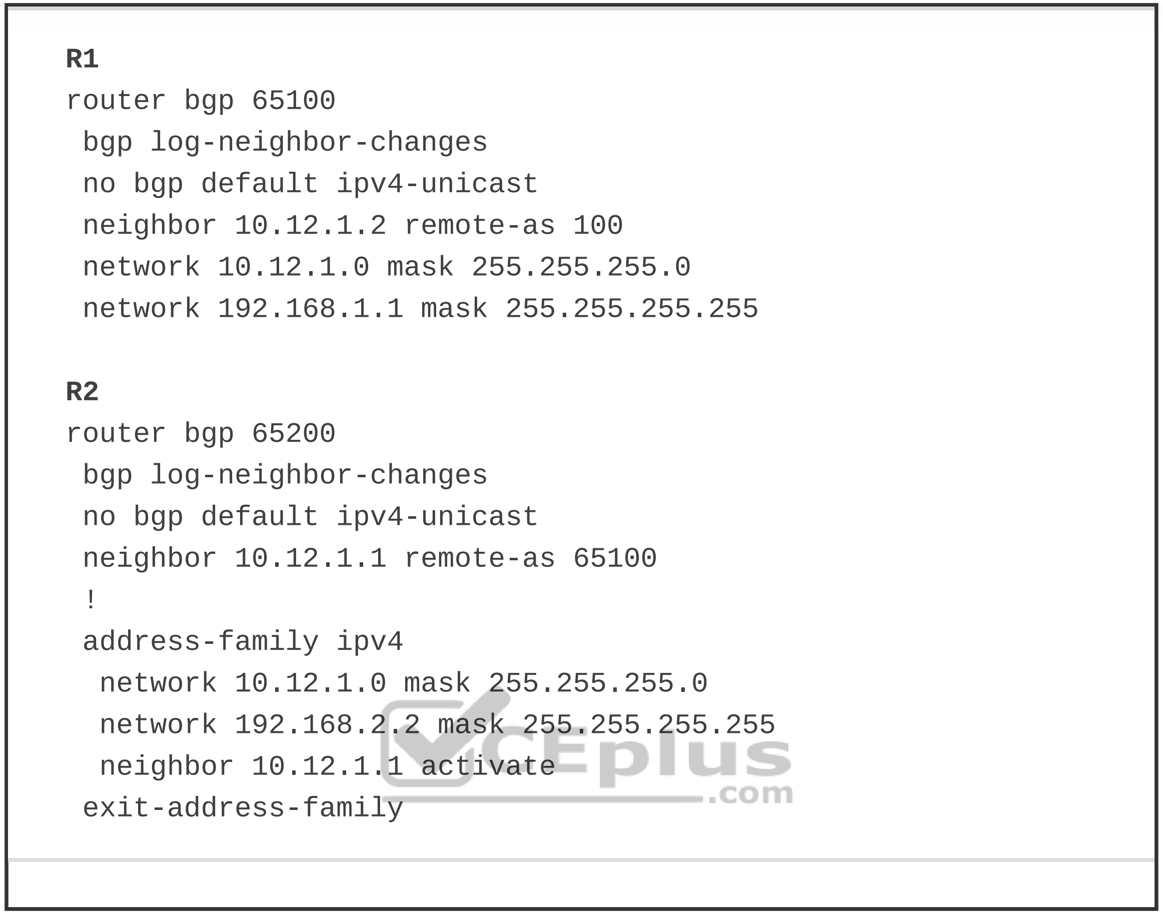
Note

BGP only advertises the best path to other BGP peers, regardless of the number of routes (NLRIs) in the BGP Loc-RIB table.

The **network** statement resides under the appropriate address family within the BGP router configuration. The command **network** *network* **mask** *subnet-mask* [**route-map** *route-map-name*] is used for advertising IPv4 networks. The optional **route-map** provides a method of setting specific BGP PAs when the prefix installs into the Loc-RIB table. Route maps are discussed in more detail in Chapter 12.

Figure 11-7 illustrates R1 and R2 connected through the 10.12.1.0/24 network. Example 11-5 demonstrates the configuration where both routers will advertise the Loopback 0 interfaces (192.168.1.1/32 and 192.168.2.2/32, respectively) and the 10.12.1.0/24 network into BGP. Notice that R1 uses the default IPv4 address family, and R2 explicitly specifies the IPv4 address family.

Example 11-5 Configuring BGP Network Advertisement



R1

```
router bgp 65100
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.2 remote-as 100
 network 10.12.1.0 mask 255.255.255.0
 network 192.168.1.1 mask 255.255.255.255
```

R2

```
router bgp 65200
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 !
 address-family ipv4
  network 10.12.1.0 mask 255.255.255.0
  network 192.168.2.2 mask 255.255.255.255
  neighbor 10.12.1.1 activate
 exit-address-family
```

Receiving and Viewing Routes

BGP uses three tables for maintaining the network prefix and path attributes (PAs) for a route:

- **Adj-RIB-In:** Contains the NLRIs in original form (that is, from before inbound route policies are processed). To save memory, the table is purged after all route policies are processed.

- **Loc-RIB:** Contains all the NLRIs that originated locally or were received from other BGP peers. After NLRIs pass the validity and next-hop reachability check, the BGP best-path algorithm selects the best NLRI for a specific prefix. The Loc-RIB table is the table used for presenting routes to the IP routing table.

- **Adj-RIB-Out:** Contains the NLRIs after outbound route policies have been processed.

Not every prefix in the Loc-RIB table is advertised to a BGP peer or installed into the global RIB when received from a BGP peer. BGP performs the following route processing steps:

Step 1. Store the route in the Adj-RIB-In table in the original state and apply the inbound route policy based on the neighbor on which the route was received.

Step 2. Update the Loc-RIB with the latest entry. The Adj-RIB-In table is cleared to save memory.

Step 3. Pass a validity check to verify that the route is valid and that the next-hop address is resolvable in the global RIB. If the route fails, the route remains in the Loc-RIB table but is not processed further.

Step 4. Identify the BGP best path and pass only the best path and its path attributes to step 5. The BGP best path selection process is covered in Chapter 12.

Step 5. Install the best-path route into the global RIB, process the outbound route policy, store the non-discarded routes in the Adj-RIB-Out table, and advertise to BGP peers.

Figure 11-9 shows the complete BGP route processing logic. It includes the receipt of a route from a BGP peers and the BGP best-path algorithm.

Key Topic

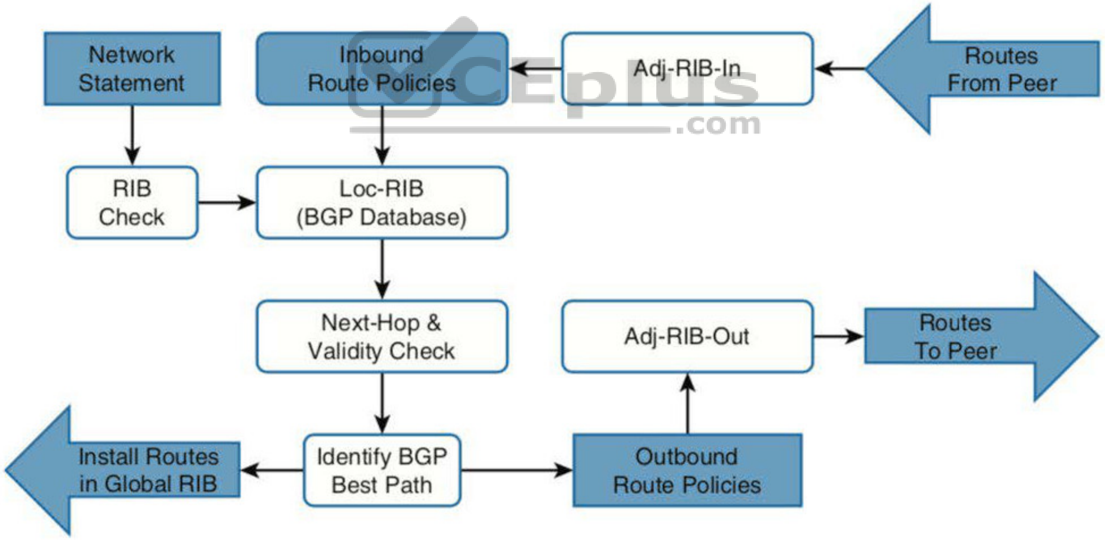


Figure 11-9 BGP Database Processing

The command **show bgp** *afi safi* displays the contents of the BGP database (Loc-RIB) on the router. Every entry in the BGP Loc-RIB table contains at least one path but could contain multiple paths for the same network prefix. Example 11-6 displays the BGP table on R1, which contains received routes and locally generated routes.

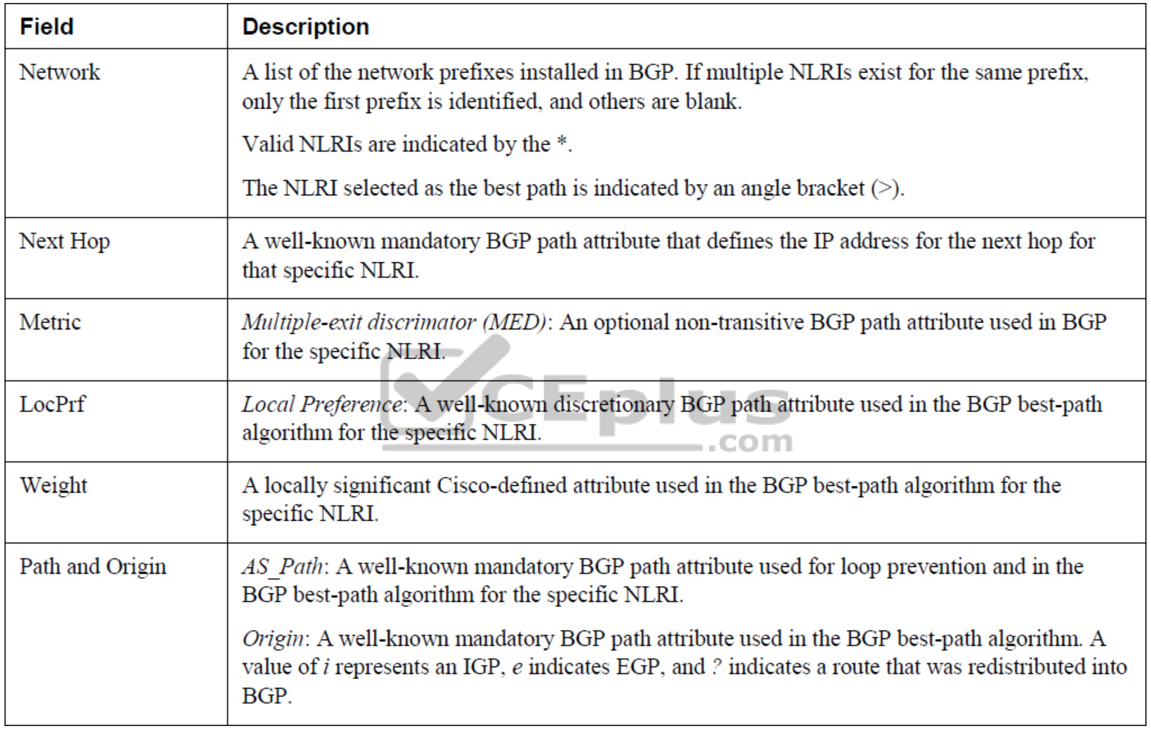
Example 11-6 Displaying the BGP Table

```
R1# show bgp ipv4 unicast
BGP table version is 4, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best,
              r RIB-failure, S Stale, m multipath, b backup-path,
              x best-external, a additional-path, c RIB-compresse
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.1.1/32   0.0.0.0                  0         32768 i
 *>  192.168.2.2/32   10.12.1.2                0             0 65

R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.1.1/32   10.12.1.1                0             0 65
 *>  192.168.2.2/32   0.0.0.0                  0         32768 i
```

Table 11-4 explains the fields of output when displaying the BGP table.

Key Topic

Table 11-4 BGP Table Fields



Field	Description
Network	A list of the network prefixes installed in BGP. If multiple NLRIs exist for the same prefix, only the first prefix is identified, and others are blank. Valid NLRIs are indicated by the *. The NLRI selected as the best path is indicated by an angle bracket (>).
Next Hop	A well-known mandatory BGP path attribute that defines the IP address for the next hop for that specific NLRI.
Metric	*Multiple-exit discrimator (MED)*: An optional non-transitive BGP path attribute used in BGP for the specific NLRI.
LocPrf	*Local Preference*: A well-known discretionary BGP path attribute used in the BGP best-path algorithm for the specific NLRI.
Weight	A locally significant Cisco-defined attribute used in the BGP best-path algorithm for the specific NLRI.
Path and Origin	*AS_Path*: A well-known mandatory BGP path attribute used for loop prevention and in the BGP best-path algorithm for the specific NLRI. *Origin*: A well-known mandatory BGP path attribute used in the BGP best-path algorithm. A value of *i* represents an IGP, *e* indicates EGP, and *?* indicates a route that was redistributed into BGP.

The command **show bgp** *afi safi network* displays all the paths for a specific route and the BGP path attributes for that route. Example 11-7 shows the paths for the 10.12.1.0/24 network. The output includes the number of paths and which path is the best path.

Example 11-7 Viewing Explicit BGP Routes and Path Attributes

```
R1# show bgp ipv4 unicast 10.12.1.0
BGP routing table entry for 10.12.1.0/24, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 1
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin IGP, metric 0, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (192.168.1.1)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, s
      rx pathid: 0, tx pathid: 0x0
```

Note

The command **show bgp** *afi safi* **detail** displays the entire BGP table with all the path attributes, such as those shown in Example 11-7.

The Adj-RIB-Out table is a unique table maintained for each BGP peer. It enables a network engineer to view routes advertised to a specific router. The command **show bgp** *afi safi* **neighbor** *ip-address* **advertised routes** displays the contents of the Adj-RIB-Out table for a neighbor.

Example 11-8 shows the Adj-RIB-Out entries specific to each neighbor. Notice that the next-hop address reflects the local router and will be changed as the route advertises to the peer.

Example 11-8 Neighbor-Specific View of the Adj-RIB-Out Table

```
R1# show bgp ipv4 unicast neighbors 10.12.1.2 advertised-routes
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Pa
*> 10.12.1.0/24     0.0.0.0                  0         32768 i
*> 192.168.1.1/32   0.0.0.0                  0         32768 i

Total number of prefixes 2

R2# show bgp ipv4 unicast neighbors 10.12.1.1 advertised-routes
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Pa
*> 10.12.1.0/24     0.0.0.0                  0         32768 i
*> 192.168.2.2/32   0.0.0.0                  0         32768 i

Total number of prefixes 2
```

The **show bgp ipv4 unicast summary** command can also be used to verify the exchange of NLRIs between nodes, as shown in Example 11-9.

Example 11-9 BGP Summary with Prefixes

```
R1# show bgp ipv4 unicast summary
! Output omitted for brevity
Neighbor         V           AS MsgRcvd MsgSent   TblVer  InQ Out
10.12.1.2        4        65200      11      10        9    0
```

The BGP routes in the global IP routing table (RIB) are displayed with the command **show ip route bgp**. Example 11-10 shows these commands in the sample topology. The prefixes are from an eBGP session and have an AD of 20, and no metric is present.

Example 11-10 Displaying BGP Routes in an IP Routing Table

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

      192.168.2.0/32 is subnetted, 1 subnets
B        192.168.2.2 [20/0] via 10.12.1.2, 00:06:12
```

BGP Route Advertisements from Indirect Sources

As stated earlier, BGP should be thought of as a routing application as the BGP session and route advertisement are two separate components. Figure 11-10 demonstrates a topology where R1 installs multiple routes learned from static routes, EIGRP, and OSPF. R1 can advertise these routes to R2.

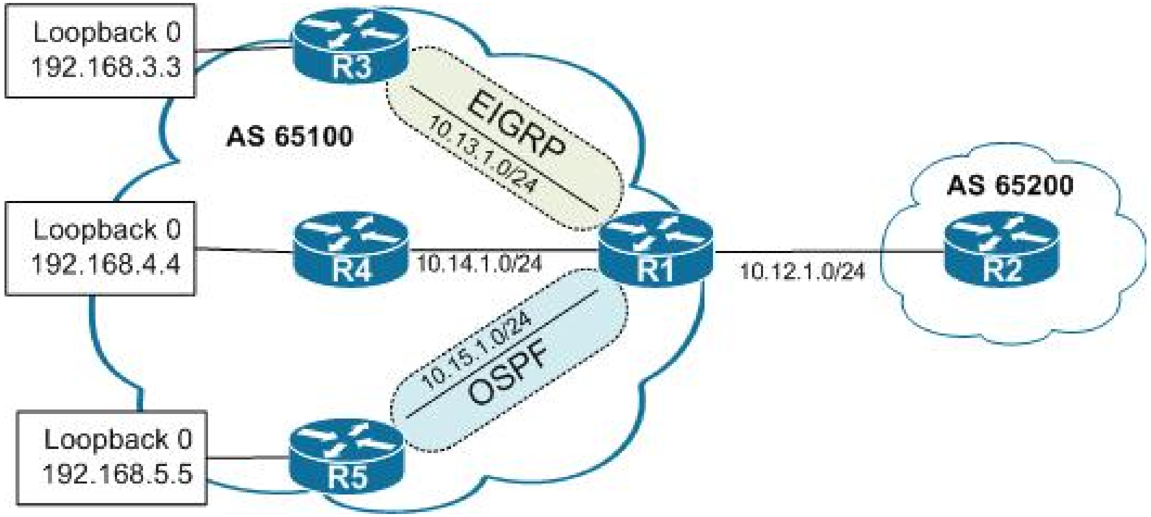


Figure 11-10 Multiple BGP Route Sources

Example 11-11 shows the routing table for R1. Notice that R3’s loopback was learned via EIGRP, R4’s loopback is reached using a static route, and R5’s loopback is learned from OSPF.

Example 11-11 R1’s Routing Table with Loopbacks for R3, R4, and R5

```
R1# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
..
Gateway of last resort is not set
```

```
         10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C           10.12.1.0/24 is directly connected, GigabitEthernet0/0
C           10.13.1.0/24 is directly connected, GigabitEthernet0/1
C           10.14.1.0/24 is directly connected, GigabitEthernet0/2
C           10.15.1.0/24 is directly connected, GigabitEthernet0/3
C           192.168.1.1 is directly connected, Loopback0
B           192.168.2.2 [20/0] via 10.12.1.2, 00:01:17
D           192.168.3.3 [90/3584] via 10.13.1.3, 00:02:10, GigabitEt
S           192.168.4.4 [1/0] via 10.14.1.4
O           192.168.5.5 [110/11] via 10.15.1.5, 00:00:08, GigabitEth
```

Example 11-12 shows the installation of R3’s and R4’s loopback using a **network** statement. Specifying every network prefix that should be advertised might seem tedious. R5’s loopback was learned by redistributing OSPF straight into BGP.

Example 11-12 Configuring Advertising Routes for Non-Connected Routes

```
R1
router bgp 65100
 bgp log-neighbor-changes
 network 10.12.1.0 mask 255.255.255.0
 network 192.168.1.1 mask 255.255.255.255
 network 192.168.3.3 mask 255.255.255.255
 network 192.168.4.4 mask 255.255.255.255
 redistribute ospf 1
 neighbor 10.12.1.2 remote-as 65200
```

Note

Redistributing routes learned from an IGP into BGP is completely safe; however, redistributing routes learned from BGP should be done with caution. BGP is designed for large scale and can handle a routing table the size of the Internet (780,000+ prefixes), whereas IGPs could have stability problems with fewer than 20,000 routes.

Example 11-13 shows the BGP routing tables on R1 and R2. Notice that on R1, the next hop matches the next hop learned from the RIB, the AS_Path is blank, and the origin codes is *IGP* (for routes learned from network statement) or *incomplete* (redistributed). The metric is carried over from R3's and R5's IGP routing protocols and is reflected as the MED. R2 learns the routes strictly from eBGP and sees only the MED and the origin codes.

Example 11-13 BGP Table for Routes from Multiple Sources

```
R1# show bgp ipv4 unicast
BGP table version is 9, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best,
              r RIB-failure, S Stale, m multipath, b backup-path,
              x best-external, a additional-path, c RIB-compresse
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Pa
```

```
 *>  10.12.1.0/24     0.0.0.0                  0         32768 i
 *                    10.12.1.2                0             0 65
 *>  10.15.1.0/24     0.0.0.0                  0         32768 ?
 *>  192.168.1.1/32   0.0.0.0                  0         32768 i
 *>  192.168.2.2/32   10.12.1.2                0             0 65
 ! The following route comes from EIGRP and uses a network statem
 *>  192.168.3.3/32   10.13.1.3             3584         32768 i
! The following route comes from a static route and uses a networ
 *>  192.168.4.4/32   10.14.1.4                0         32768 i
! The following route was redistributed from OSPF statement
 *>  192.168.5.5/32   10.15.1.5               11         32768 ?

R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 i
 *>  10.15.1.0/24     10.12.1.1                0             0 65
 *>  192.168.1.1/32   10.12.1.1                0             0 65
 *>  192.168.2.2/32   0.0.0.0                  0         32768 i
 *>  192.168.3.3/32   10.12.1.1             3584             0 65
 *>  192.168.4.4/32   10.12.1.1                0             0 65
 *>  192.168.5.5/32   10.12.1.1               11             0 65
```

ROUTE SUMMARIZATION

Summarizing prefixes conserves router resources and accelerates best-path calculation by reducing the size of the table. Summarization also provides the benefit of stability by hiding route flaps from downstream routers, thereby reducing routing churn. While most service providers do not accept prefixes larger than /24 for IPv4 (/25 through /32), the Internet, at the time of this writing, still has more than 780,000 routes and continues to grow. Route summarization is required to reduce the size of the BGP table for Internet routers.

BGP route summarization on BGP edge routers reduces route computation on routers in the core for received routes or for advertised routes. In Figure 11-11, R3 summarizes all the eBGP routes received from AS 65100 and AS 65200 to reduce route computation on R4 during link flaps. In the event of a link flap on the 10.13.1.0/24 network, R3 removes all the AS 65100 routes learned directly from R1 and identifies the same network prefixes via R2 with different path attributes (a longer AS_Path). R3 has to advertise new routes to R4 because of these flaps, which is a waste of CPU cycles because R4 only receives connectivity from R3. If R3 summarized the network prefix range, R4 would execute the best-path algorithm once and not need to run during link flaps of the 10.13.1.0/24 link.

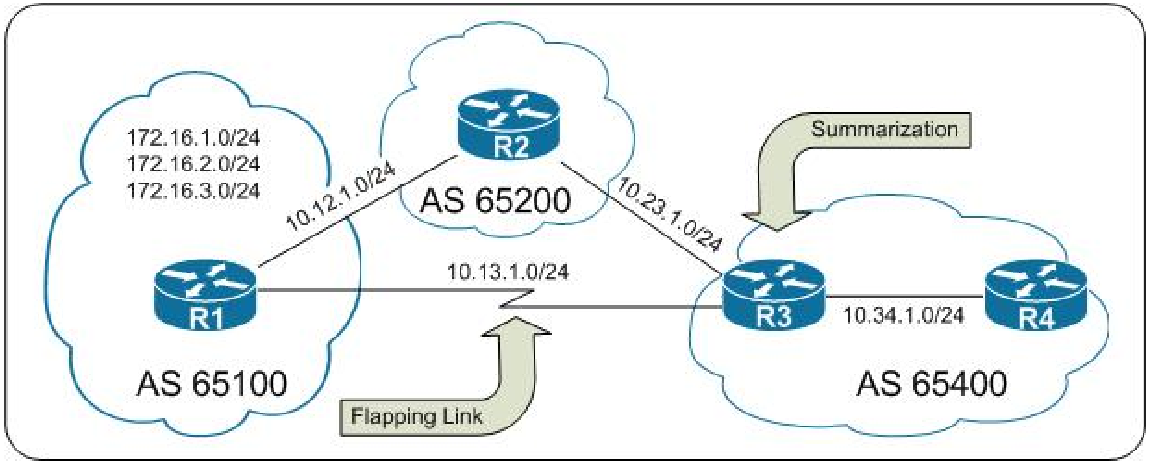


Figure 11-11 BGP Route Summarization Hiding Link Flaps

There are two techniques for BGP summarization:

- **Static:** Create a static route to Null0 for the summary network prefix and then advertise the prefix with a **network** statement. The downfall of this technique is that the summary route is always advertised, even if the networks are not available.

- **Dynamic:** Configure an aggregation network prefix. When viable component routes that match the aggregate network prefix enter the BGP table, then the aggregate prefix is created. The originating router sets the next hop to Null0 as a discard route for the aggregated prefix for loop prevention.

In both methods of route aggregation, a new network prefix with a shorter prefix length is advertised into BGP. Because the aggregated prefix is a new route, the summarizing router is the originator for the new aggregate route.

Aggregate Address

Dynamic route summarization is accomplished with the BGP address family configuration command **aggregate-address** *network subnet-mask* [**summary-only**] [**as-set**].

Figure 11-12 removes the flapping serial link between R1 and R3 to demonstrate BGP route aggregation and the effects of the commands.

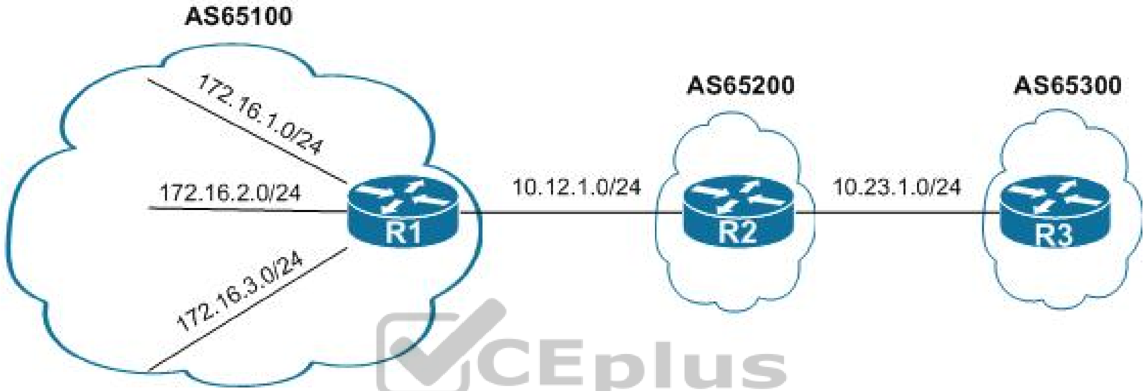
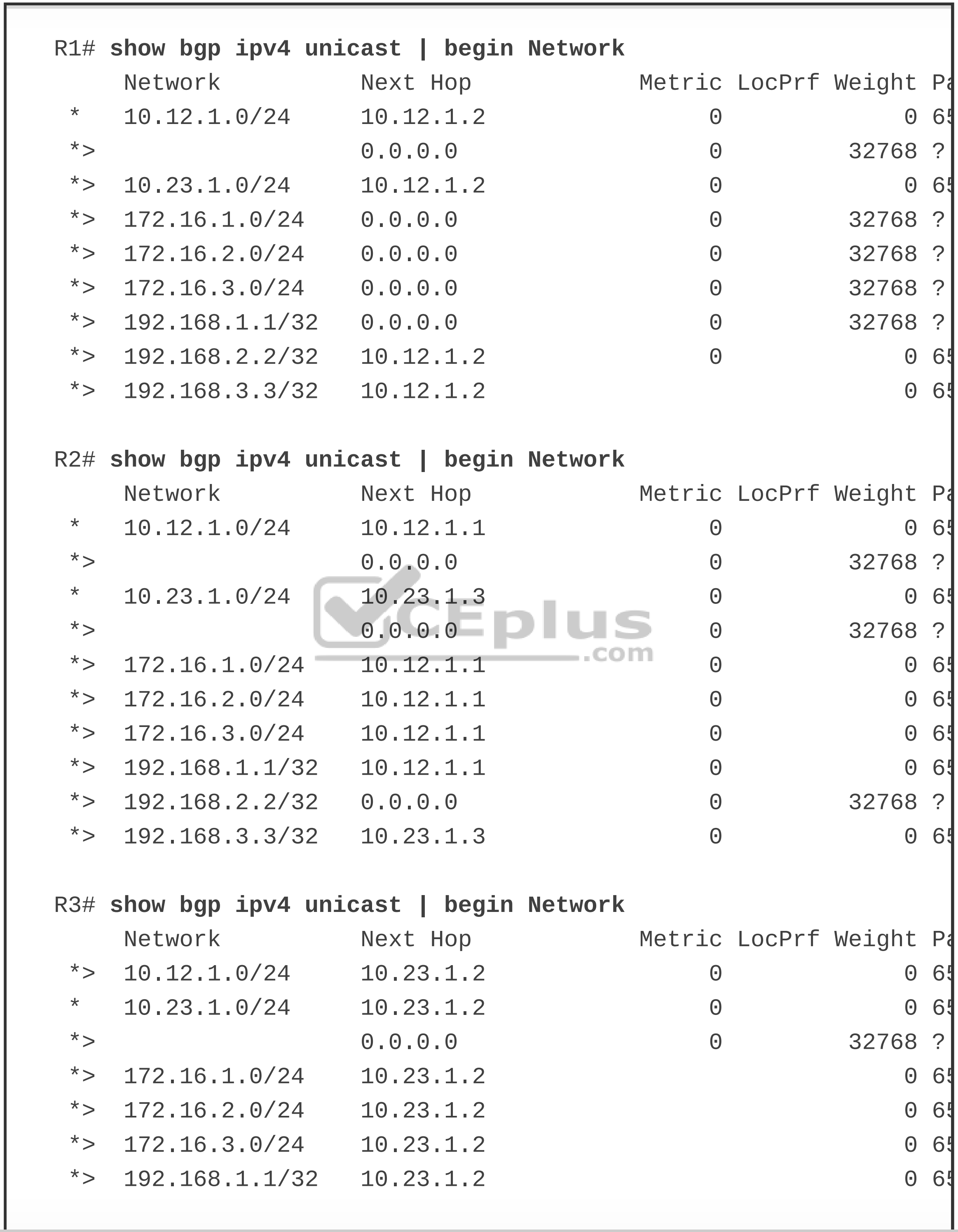


Figure 11-12 BGP Summarization Topology

Example 11-14 shows the BGP tables for R1, R2, and R3 before route aggregation has been performed. R1's stub networks (172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24) are advertised through all the autonomous systems, along with the router's loopback addresses (192.168.1.1/32, 192.168.2.2/32, and 192.168.3.3/32) and the peering links (10.12.1.0/24 and 10.23.1.0/24).

Example 11-14 BGP Tables for R1, R2, and R3 Without Aggregation



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2                0             0 65
 *>  172.16.1.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.2.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.3.0/24    0.0.0.0                  0         32768 ?
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2                0             0 65
 *>  192.168.3.3/32   10.12.1.2                              0 65

R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *   10.23.1.0/24     10.23.1.3                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.1.0/24    10.12.1.1                0             0 65
 *>  172.16.2.0/24    10.12.1.1                0             0 65
 *>  172.16.3.0/24    10.12.1.1                0             0 65
 *>  192.168.1.1/32   10.12.1.1                0             0 65
 *>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 *>  192.168.3.3/32   10.23.1.3                0             0 65

R3# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     10.23.1.2                0             0 65
 *   10.23.1.0/24     10.23.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.1.0/24    10.23.1.2                              0 65
 *>  172.16.2.0/24    10.23.1.2                              0 65
 *>  172.16.3.0/24    10.23.1.2                              0 65
 *>  192.168.1.1/32   10.23.1.2                              0 65
 *>  192.168.2.2/32   10.23.1.2                0            0 65
 *>  192.168.3.3/32   0.0.0.0                  0        32768 ?
```

R1 aggregates all the stub networks (172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24) into a 172.16.0.0/20 network prefix. R2 aggregates all of the router's loopback addresses into a 192.168.0.0/16 network prefix. Example 11-15 shows the configuration for R1 running with the default IPv4 address family and R2 running without the default IPv4 address family.

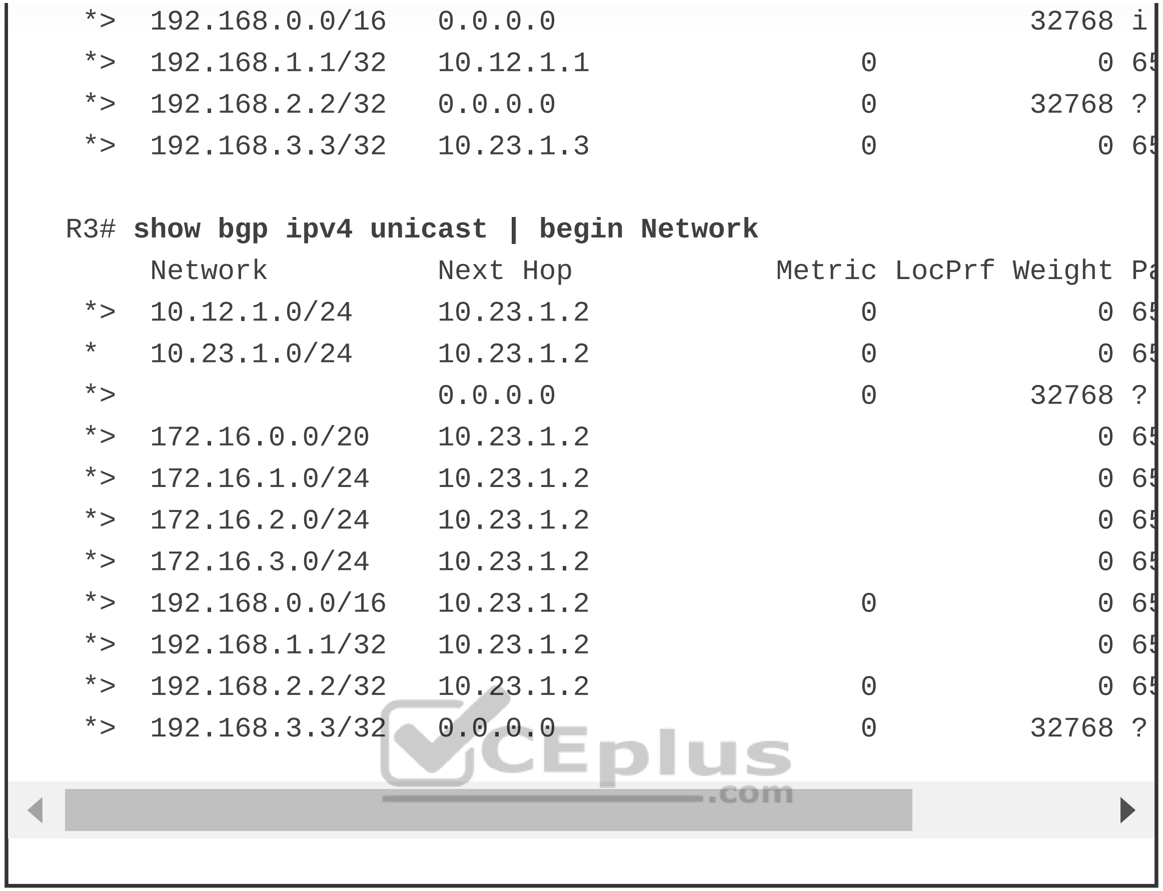
Example 11-15 Configuring BGP Route Aggregation

```
R1# show running-config | section router bgp
router bgp 65100
 bgp log-neighbor-changes
 aggregate-address 172.16.0.0 255.255.240.0
 redistribute connected
 neighbor 10.12.1.2 remote-as 65200

R2# show running-config | section router bgp
router bgp 65200
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 neighbor 10.23.1.3 remote-as 65300
 !
 address-family ipv4
  aggregate-address 192.168.0.0 255.255.0.0
  redistribute connected
  neighbor 10.12.1.1 activate
  neighbor 10.23.1.3 activate
 exit-address-family
```

Example 11-16 shows the routing tables for R1, R2, and R3 after aggregation is configured on R1 and R2.

Example 11-16 BGP Tables for R1, R2, and R3 with Aggregation

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2                0             0 65
 *>  172.16.0.0/20    0.0.0.0                            32768 i
 *>  172.16.1.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.2.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.3.0/24    0.0.0.0                  0         32768 ?
 *>  192.168.0.0/16   10.12.1.2                0             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2                0             0 65
 *>  192.168.3.3/32   10.12.1.2                              0 65

R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *   10.23.1.0/24     10.23.1.3                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.12.1.1                0             0 65
 *>  172.16.1.0/24    10.12.1.1                0             0 65
 *>  172.16.2.0/24    10.12.1.1                0             0 65
 *>  172.16.3.0/24    10.12.1.1                0             0 65
```



```
 *>  192.168.0.0/16   0.0.0.0                            32768 i
 *>  192.168.1.1/32   10.12.1.1                0             0 65
 *>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 *>  192.168.3.3/32   10.23.1.3                0             0 65

R3# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     10.23.1.2                0             0 65
 *   10.23.1.0/24     10.23.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.23.1.2                              0 65
 *>  172.16.1.0/24    10.23.1.2                              0 65
 *>  172.16.2.0/24    10.23.1.2                              0 65
 *>  172.16.3.0/24    10.23.1.2                              0 65
 *>  192.168.0.0/16   10.23.1.2                0             0 65
 *>  192.168.1.1/32   10.23.1.2                              0 65
 *>  192.168.2.2/32   10.23.1.2                0             0 65
 *>  192.168.3.3/32   0.0.0.0                  0         32768 ?
```

Notice that the 172.16.0.0/20 and 192.168.0.0/16 network prefixes are visible, but the smaller component network prefixes still exist on all the routers. The **aggregate-address** command advertises the aggregated route in addition to the original component network prefixes. Using the optional **summary-only** keyword suppresses the component network prefixes in the summarized network range. Example 11-17 shows the configuration with the **summary-only** keyword.

Example 11-17 BGP Route Aggregation Configuration with Suppression

```
R1# show running-config | section router bgp
router bgp 65100
 bgp log-neighbor-changes
 aggregate-address 172.16.0.0 255.255.240.0 summary-only
 redistribute connected
 neighbor 10.12.1.2 remote-as 65200

R2# show running-config | section router bgp
router bgp 65200
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 neighbor 10.23.1.3 remote-as 65300
 !
 address-family ipv4
  aggregate-address 192.168.0.0 255.255.0.0 summary-only
  redistribute connected
  neighbor 10.12.1.1 activate
  neighbor 10.23.1.3 activate
 exit-address-family
```

Example 11-18 shows the BGP table for R3 after the **summary-only** keyword is added to the aggregation command. R1’s stub network has been aggregated in the 172.16.0.0/20 network prefix, while R1’s and R2’s loopback has been aggregated into the 192.168.0.0/16 network prefix. None of R1's stub networks or the loopback addresses from R1 or R2 are visible on R3.

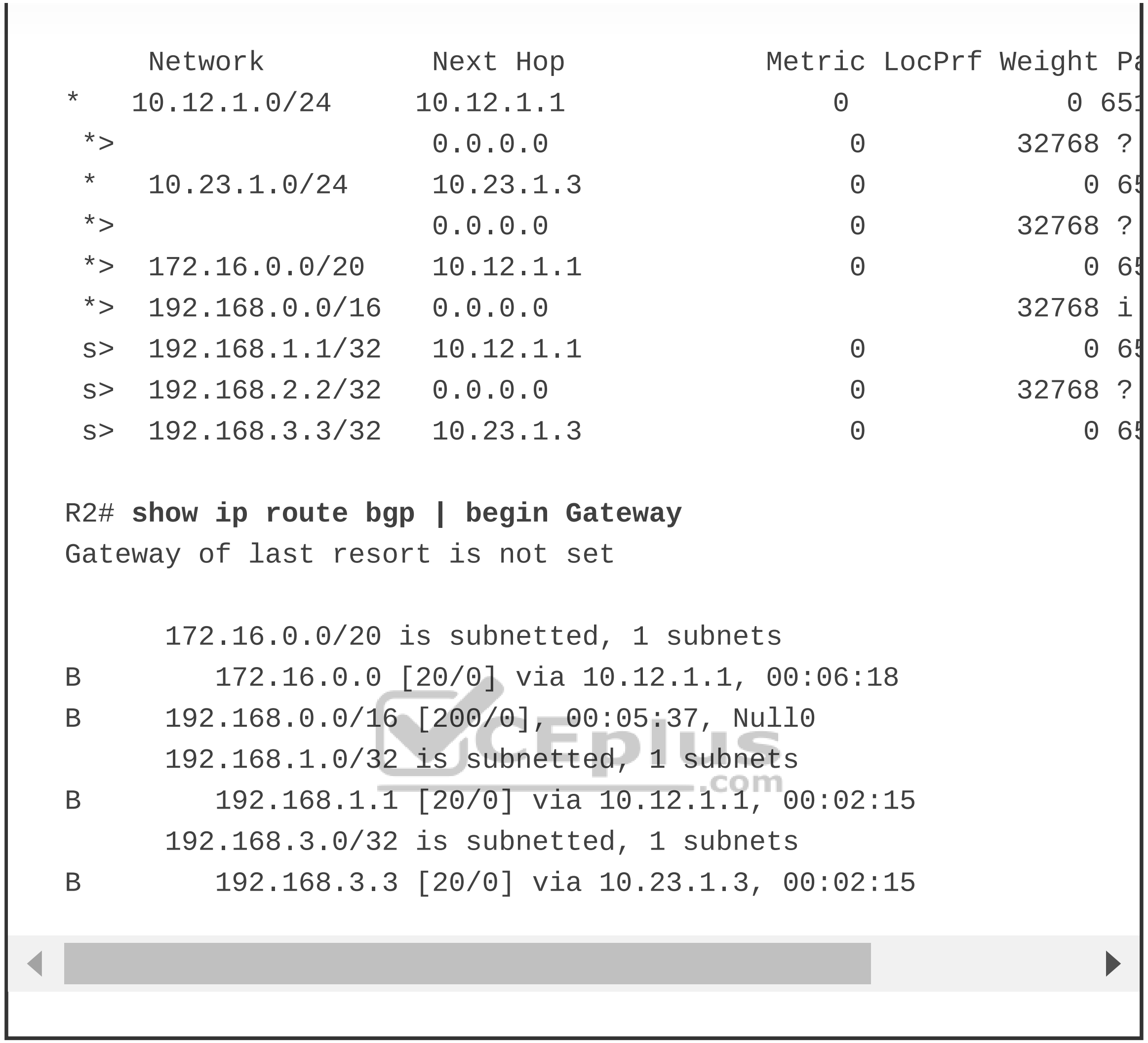
Example 11-18 BGP Tables for R3 with Aggregation and Suppression

```
R3# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     10.23.1.2                0             0 65
 *   10.23.1.0/24     10.23.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.23.1.2                              0 65
 *>  192.168.0.0/16   10.23.1.2                0             0 65
 *>  192.168.3.3/32   0.0.0.0                  0         32768 ?
```

Example 11-19 shows the BGP table and RIB for R2. Notice that the component loopback networks have been suppressed by BGP and are not advertised by R2. In addition, a summary discard route has been installed to Null0 as a loop-prevention mechanism.

Example 11-19 R2's BGP and RIB After Aggregation with Suppression

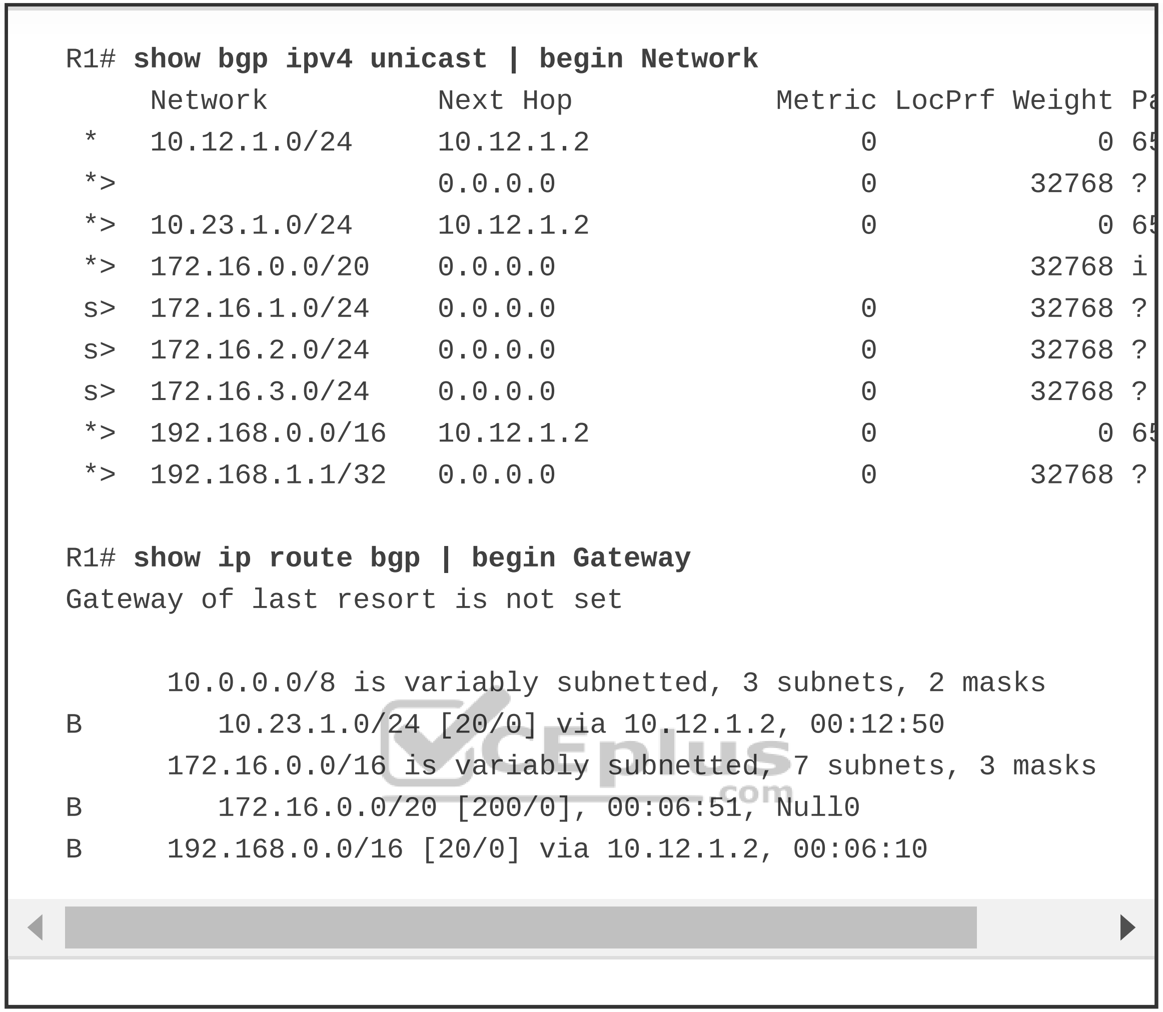
```
R2# show bgp ipv4 unicast
BGP table version is 10, local router ID is 192.168.2.2
Status codes: s suppressed, d damped, h history, * valid, > best,
              r RIB-failure, S Stale, m multipath, b backup-path,
              x best-external, a additional-path, c RIB-compresse
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```



```
     Network          Next Hop            Metric LocPrf Weight Pa
*   10.12.1.0/24     10.12.1.1                0             0 65
 *>                  0.0.0.0                  0         32768 ?
 *   10.23.1.0/24     10.23.1.3                0             0 65
 *>                  0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.12.1.1                0             0 65
 *>  192.168.0.0/16   0.0.0.0                            32768 i
 s>  192.168.1.1/32   10.12.1.1                0             0 65
 s>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 s>  192.168.3.3/32   10.23.1.3                0             0 65

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

      172.16.0.0/20 is subnetted, 1 subnets
B        172.16.0.0 [20/0] via 10.12.1.1, 00:06:18
B     192.168.0.0/16 [200/0], 00:05:37, Null0
      192.168.1.0/32 is subnetted, 1 subnets
B        192.168.1.1 [20/0] via 10.12.1.1, 00:02:15
      192.168.3.0/32 is subnetted, 1 subnets
B        192.168.3.3 [20/0] via 10.23.1.3, 00:02:15
```

Example 11-20 shows that R1's stub networks have been suppressed, and the summary discard route for the 172.16.0.0/20 network has been installed in the RIB as well.

Example 11-20 R1's BGP and RIB After Aggregation with Suppression



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2                0             0 65
 *>  172.16.0.0/20    0.0.0.0                            32768 i
 s>  172.16.1.0/24    0.0.0.0                  0         32768 ?
 s>  172.16.2.0/24    0.0.0.0                  0         32768 ?
 s>  172.16.3.0/24    0.0.0.0                  0         32768 ?
 *>  192.168.0.0/16   10.12.1.2                0             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?

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

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.23.1.0/24 [20/0] via 10.12.1.2, 00:12:50
      172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
B        172.16.0.0/20 [200/0], 00:06:51, Null0
B     192.168.0.0/16 [20/0] via 10.12.1.2, 00:06:10
```

Key Topic

Atomic Aggregate

Aggregated routes act like new BGP routes with a shorter prefix length. When a BGP router summarizes a route, it does not advertise the AS_Path information from before the aggregation. BGP path attributes like AS_Path, MED, and BGP communities are not included in the new BGP advertisement.

The atomic aggregate attribute indicates that a loss of path information has occurred. To demonstrate this best, the previous BGP route aggregation on R1 has been removed and added to R2 so that R2 is now aggregating the 172.16.0.0/20 and 192.168.0.0/16 networks with suppression. Example 11-21 shows the configuration on R2.

Example 11-21 Configuring Aggregation for 172.16.0.0/20 and 192.168.0.0/16

```
R2# show running-config | section router bgp
router bgp 65200
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 neighbor 10.23.1.3 remote-as 65300
 !
 address-family ipv4
  aggregate-address 192.168.0.0 255.255.0.0 summary-only
  aggregate-address 172.16.0.0 255.255.240.0 summary-only
  redistribute connected
  neighbor 10.12.1.1 activate
  neighbor 10.23.1.3 activate
 exit-address-family
```

Example 11-22 shows R2’s and R3’s BGP tables. R2 is aggregating and suppressing R1’s stub networks (172.16.1.0/24, 172.16.2.0/24, and 172.16.3.0/24) into the 172.16.0.0/20 network. The component network prefixes maintain an AS_Path of 65100 on R2, while the aggregate 172.16.0.0/20 network appears locally generated on R2.

From R3’s perspective, R2 does not advertise R1’s stub networks; instead, it advertises the 172.16.0.0/20 network as its own. The AS_Path for the 172.16.0.0/20 network prefix on R3 is simply AS 65200 and does not include AS 65100.

Example 11-22 R2’s and R3’s BGP Tables with Path Attribute Loss

```
R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *   10.23.1.0/24     10.23.1.3                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    0.0.0.0                            32768 i
 s>  172.16.1.0/24    10.12.1.1                0             0 65
 s>  172.16.2.0/24    10.12.1.1                0             0 65
 s>  172.16.3.0/24    10.12.1.1                0             0 65
 *>  192.168.0.0/16   0.0.0.0                            32768 i
 s>  192.168.1.1/32   10.12.1.1                0             0 65
 s>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 s>  192.168.3.3/32   10.23.1.3                0             0 65

R3# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     10.23.1.2                0             0 65
 *   10.23.1.0/24     10.23.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.23.1.2                0             0 65
```

```
  *>  192.168.0.0/16   10.23.1.2                0            0 65
  *>  192.168.3.3/32   0.0.0.0                  0        32768 ?
```

Example 11-23 shows the explicit 172.16.0.0/20 prefix entry on R3. The route's NLRI information indicates that the routes were aggregated in AS 65200 by the router with the RID 192.168.2.2. In addition, the atomic aggregate attribute has been set to indicate a loss of path attributes, such as AS_Path in this scenario.

Example 11-23 Examining the BGP Attribute for the Atomic Aggregate Attribute

```
R3# show bgp ipv4 unicast 172.16.0.0
BGP routing table entry for 172.16.0.0/20, version 25
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 2
  65200, (aggregated by 65200 192.168.2.2)
    10.23.1.2 from 10.23.1.2 (192.168.2.2)
      Origin IGP, metric 0, localpref 100, valid, external, atomi
      rx pathid: 0, tx pathid: 0x0
```

Key Topic

Route Aggregation with AS_SET

To keep the BGP path information history, the optional **as-set** keyword may be used with the **aggregate-address** command. As the router generates the aggregate route, BGP attributes from the component aggregate routes are copied over to it. The AS_Path settings from the original prefixes are stored in the AS_SET portion of the AS_Path. The AS_SET, which is displayed within brackets, only counts as one hop, even if multiple ASs are listed.

Example 11-24 shows R2's updated BGP configuration for summarizing both networks with the **as-set** keyword.

Example 11-24 Configuring Aggregation While Preserving BGP Attributes

```
R2# show running-config | section router bgp
router bgp 65200
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.12.1.1 remote-as 65100
 neighbor 10.23.1.3 remote-as 65300
 !
 address-family ipv4
  aggregate-address 192.168.0.0 255.255.0.0 as-set summary-only
  aggregate-address 172.16.0.0 255.255.240.0 as-set summary-only
  redistribute connected
  neighbor 10.12.1.1 activate
  neighbor 10.23.1.3 activate
 exit-address-family
```

Example 11-25 shows the 172.16.0.0/20 network again, now that BGP attributes will be propagated into the new route. Notice that the AS_Path information now contains AS 65100 as part of the information.

Example 11-25 Verifying That Path Attributes Are Injected into the BGP Aggregate

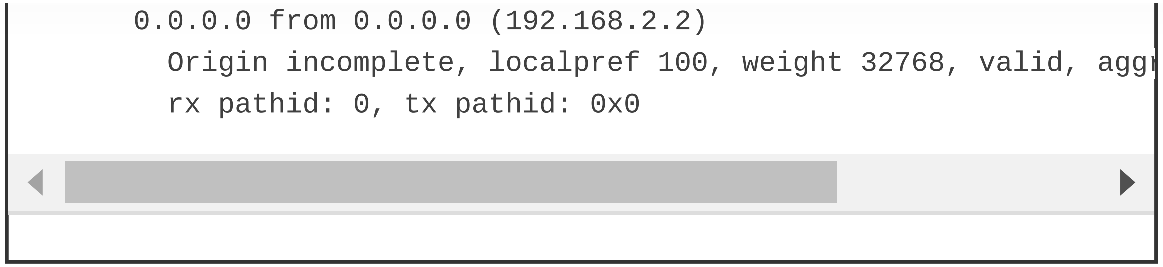
```
R3# show bgp ipv4 unicast 172.16.0.0
BGP routing table entry for 172.16.0.0/20, version 30
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 2
  65200 65100, (aggregated by 65200 192.168.2.2)
    10.23.1.2 from 10.23.1.2 (192.168.2.2)
      Origin incomplete, metric 0, localpref 100, valid, externa
      rx pathid: 0, tx pathid: 0x0

R3# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     10.23.1.2                0             0 65
 *   10.23.1.0/24     10.23.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    10.23.1.2                0             0 65
 *>  192.168.3.3/32   0.0.0.0                  0         32768 ?
```

Did you notice that the 192.168.0.0/16 network is no longer present in R3's BGP table? The reason for this is that on R2, R2 is aggregating all of the loopback networks from R1 (AS 65100), R2 (AS 65200), and R3 (AS 65300). And now that R2 is copying all component routes' BGP path attributes into the AS_SET information, the AS_Path for the 192.168.0.0/16 network contains AS 65300. When the aggregate is advertised to R3, R3 discards that route because it sees its own AS_Path in the advertisement and thinks that it is a loop.

Example 11-26 shows R2's BGP table and the path attributes for the aggregated 192.168.0.0/16 network entry.

Example 11-26 Viewing the Aggregated Properties of 192.168.0.0/16

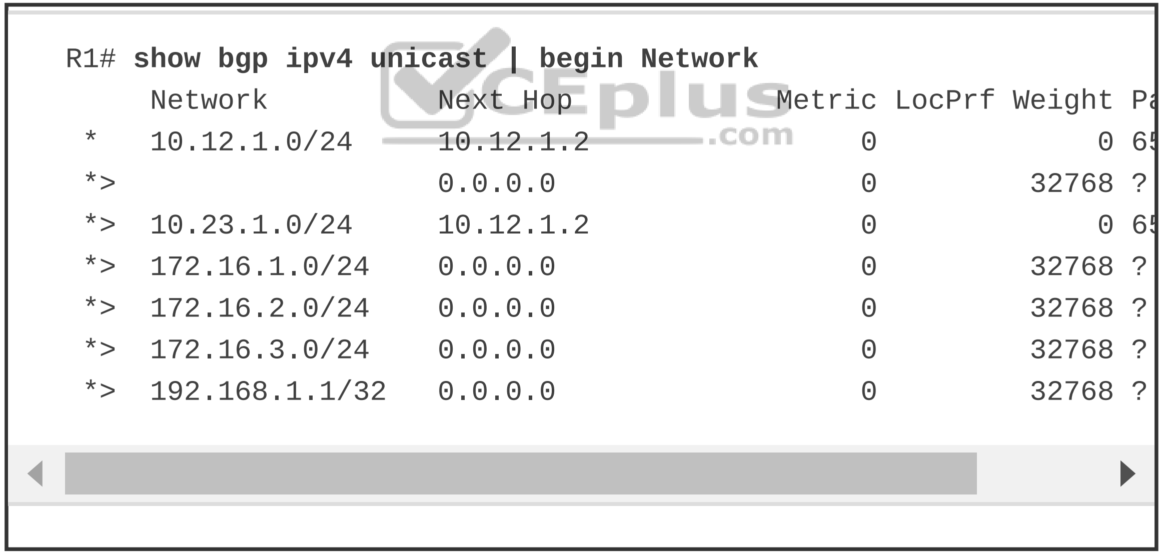
```
R2# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.1                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *   10.23.1.0/24     10.23.1.3                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  172.16.0.0/20    0.0.0.0                       100  32768 65
 s>  172.16.1.0/24    10.12.1.1                0             0 65
 s>  172.16.2.0/24    10.12.1.1                0             0 65
 s>  172.16.3.0/24    10.12.1.1                0             0 65
 *>  192.168.0.0/16   0.0.0.0                       100  32768 {6
 s>  192.168.1.1/32   10.12.1.1                0             0 65
 s>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 s>  192.168.3.3/32   10.23.1.3                0             0 65

R2# show bgp ipv4 unicast 192.168.0.0
BGP routing table entry for 192.168.0.0/16, version 28
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  {65100,65300}, (aggregated by 65200 192.168.2.2)
```



```
     0.0.0.0 from 0.0.0.0 (192.168.2.2)
       Origin incomplete, localpref 100, weight 32768, valid, agg
       rx pathid: 0, tx pathid: 0x0
```

R1 does not install the 192.168.0.0/16 network for the same reasons that R3 does not install the 192.168.0.0/16 network. R1 thinks that the advertisement is a loop because it detects AS65100 in the advertisement. This can be confirmed by examining R1’s BGP table, as shown in Example 11-27.

Example 11-27 R1’s BGP Table, with 192.168.0.0/16 Discarded



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *   10.12.1.0/24     10.12.1.2                0             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2                0             0 65
 *>  172.16.1.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.2.0/24    0.0.0.0                  0         32768 ?
 *>  172.16.3.0/24    0.0.0.0                  0         32768 ?
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
```

Key Topic

MULTIPROTOCOL BGP FOR IPV6

Multiprotocol BGP (MP-BGP) enables BGP to carry NLRI for multiple protocols, such as IPv4, IPv6, and Multiprotocol Label Switching (MPLS) Layer 3 virtual private networks (L3VPNs).

RFC 4760 defines the following new features:

- A new address family identifier (AFI) model

- New BGPv4 optional and nontransitive attributes:

- Multiprotocol reachable NLRI

- Multiprotocol unreachable NLRI

The new multiprotocol reachable NLRI attribute describes IPv6 route information, and the multiprotocol unreachable NLRI attribute withdraws the IPv6 route from service. The attributes are optional and nontransitive, so if an older router does not understand the attributes, the information can just be ignored.

All the same underlying IPv4 path vector routing protocol features and rules also apply to MP-BGP for IPv6. MP-BGP for IPv6 continues to use the same well-known TCP port 179 for session peering as BGP uses for IPv4. During the initial open message negotiation, the BGP peer routers exchange capabilities. The MP-BGP extensions include an address family identifier (AFI) that describes the supported protocols, along with subsequent address family identifier (SAFI) attribute fields that describe whether the prefix applies to the unicast or multicast routing table:

- **IPv4 unicast:** AFI: 1, SAFI: 1

- **IPv6 unicast:** AFI: 2, SAFI: 1

Figure 11-13 demonstrates a simple topology with three different ASs and R2 forming an eBGP session with R1 and R3. The link-local addresses have been configured from the defined link-local range FE80::/10. All of R1's links are configured to FE80::1, all of R2's links are set to FE80::2, and all of R3's links are configured for FE80::3. This topology is used throughout this section.

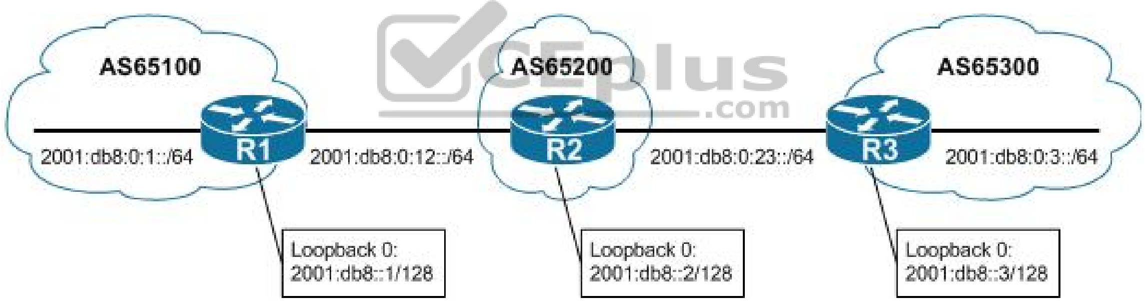


Figure 11-13 IPv6 Sample Topology

Key Topic

IPv6 Configuration

All the BGP configuration rules demonstrated earlier apply with IPv6, except that the IPv6 address family must be initialized, and the neighbor is activated. Routers with only IPv6 addressing must statically define the BGP RID to allow sessions to form.

The protocol used to establish the BGP session is independent of the AFI/SAFI route advertisements. The TCP session used by BGP is a Layer 4 protocol, and it can use either an IPv4 or IPv6 address to form a session adjacency and exchange routes. Advertising IPv6 prefixes over an IPv4 BGP session is feasible but beyond the scope of this book as additional configuration is required.

Note

Unique global unicast addressing is the recommended method for BGP peering to avoid operational complexity. BGP peering using the link-local address may introduce risk if the address is not manually assigned to an interface. A hardware failure or cabling move will change the MAC address, resulting in a new link-local address. This will cause the session to fail because the stateless address autoconfiguration will generate a new IP address.

Example 11-28 shows the IPv6 BGP configuration for R1, R2, and R3. The peering uses global unicast addressing for establishing the session. The BGP RID has been set to the IPv4 loopback format used throughout this book. R1 advertises all its networks through redistribution, and R2 and R3 use the **network** statement to advertise all their connected networks.

Example 11-28 Configuring IPv6 BGP

```
R1
router bgp 65100
 bgp router-id 192.168.1.1
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:DB8:0:12::2 remote-as 65200
 !
address-family ipv6
  neighbor 2001:DB8:0:12::2 activate
  redistribute connected

R2
router bgp 65200
 bgp router-id 192.168.2.2
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 2001:DB8:0:12::1 remote-as 65100
 neighbor 2001:DB8:0:23::3 remote-as 65300
!
 address-family ipv6
  neighbor 2001:DB8:0:12::1 activate
  neighbor 2001:DB8:0:23::3 activate
  network 2001:DB8::2/128
  network 2001:DB8:0:12::/64
  network 2001:DB8:0:23::/64

R3
router bgp 65300
```

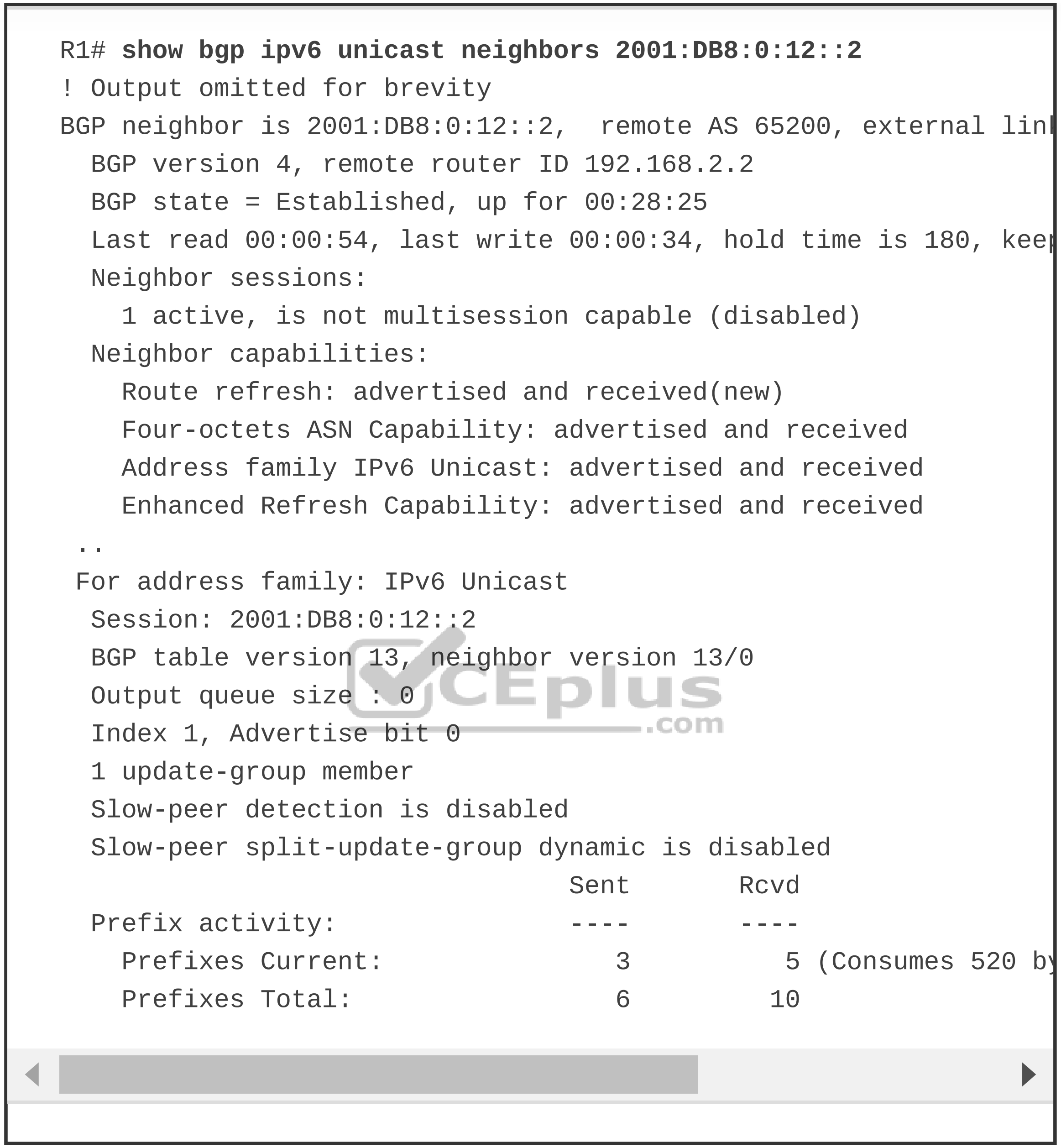
```
bgp router-id 192.168.3.3
bgp log-neighbor-changes
no bgp default ipv4-unicast
neighbor 2001:DB8:0:23::2 remote-as 65200
!
address-family ipv6
 neighbor 2001:DB8:0:23::2 activate
 network 2001:DB8::3/128
 network 2001:DB8:0:3::/64
 network 2001:DB8:0:23::/64
```

Note

IPv4 unicast routing capability is advertised by default in IOS unless the neighbor is specifically shut down within the IPv4 address family or globally within the BGP process with the command **no bgp default ipv4-unicast**.

Routers exchange AFI capabilities during the initial BGP session negotiation. The command **show bgp ipv6 unicast neighbors** *ip-address* [**detail**] displays detailed information on whether the IPv6 capabilities were negotiated successfully. Example 11-29 shows the fields that should be examined for IPv6 session establishment and route advertisement

Example 11-29 Viewing BGP Neighbors for IPv6 Capabilities



```
R1# show bgp ipv6 unicast neighbors 2001:DB8:0:12::2
! Output omitted for brevity
BGP neighbor is 2001:DB8:0:12::2,  remote AS 65200, external lin
  BGP version 4, remote router ID 192.168.2.2
  BGP state = Established, up for 00:28:25
  Last read 00:00:54, last write 00:00:34, hold time is 180, kee
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv6 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
 ..
 For address family: IPv6 Unicast
  Session: 2001:DB8:0:12::2
  BGP table version 13, neighbor version 13/0
  Output queue size : 0
  Index 1, Advertise bit 0
  1 update-group member
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             3          5 (Consumes 520 by
    Prefixes Total:               6         10
```

The command **show bgp ipv6 unicast summary** displays a status summary of the sessions, including the number of routes that have been exchanged and the session uptime.

Example 11-30 highlights the IPv6 AFI neighbor status for R2. Notice that the two neighbor adjacencies have been up for about 25 minutes. Neighbor 2001:db8:0:12::1 is advertising three routes, and neighbor 2001:db8:0:23::3 is advertising three routes.

Example 11-30 Verifying an IPv6 BGP Session

```
R2# show bgp ipv6 unicast summary
BGP router identifier 192.168.2.2, local AS number 65200
BGP table version is 19, main routing table version 19
7 network entries using 1176 bytes of memory
8 path entries using 832 bytes of memory
3/3 BGP path/bestpath attribute entries using 456 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2512 total bytes of memory
BGP activity 7/0 prefixes, 8/0 paths, scan interval 60 secs

Neighbor         V     AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down
2001:DB8:0:12::1 4  65100      35      37     19   0    0 00:25:0
2001:DB8:0:23::3 4  65300      32      37     19   0    0 00:25:1
```

Example 11-31 shows the IPv6 unicast BGP tables for R1, R2, and R3. Notice that some of the routes include an unspecified address (::) as the next hop. An unspecified address indicates that the local router is generating the prefix for the BGP table. The weight value 32,768 also indicates that the prefix is locally originated by the router.

Example 11-31 Viewing the IPv6 BGP Tables

```
R1# show bgp ipv6 unicast
BGP table version is 13, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best,
              r RIB-failure, S Stale, m multipath, b backup-path,
              x best-external, a additional-path, c RIB-compresse
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network             Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::1/128    ::                       0         32768 ?
 *>  2001:DB8::2/128    2001:DB8:0:12::2         0             0 65
 *>  2001:DB8::3/128    2001:DB8:0:12::2                       0 65
 *>  2001:DB8:0:1::/64  ::                       0         32768 ?
 *>  2001:DB8:0:3::/64  2001:DB8:0:12::2                       0 65
 *   2001:DB8:0:12::/64 2001:DB8:0:12::2         0             0 65
 *>                     ::                       0         32768 ?
 *>  2001:DB8:0:23::/64 2001:DB8:0:12::2                       0 65

R2# show bgp ipv6 unicast | begin Network
     Network             Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::1/128    2001:DB8:0:12::1         0             0 65
 *>  2001:DB8::2/128    ::                       0         32768 i
 *>  2001:DB8::3/128    2001:DB8:0:23::3         0             0 65
 *>  2001:DB8:0:1::/64  2001:DB8:0:12::1         0             0 65
 *>  2001:DB8:0:3::/64  2001:DB8:0:23::3         0             0 65
 *>  2001:DB8:0:12::/64 ::                       0         32768 i
 *                      2001:DB8:0:12::1         0             0 65
 *>  2001:DB8:0:23::/64 ::                         0         32768
                        2001:DB8:0:23::3         0             0 65

R3# show bgp ipv6 unicast | begin Network
     Network             Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::1/128    2001:DB8:0:23::2                     0 65
 *>  2001:DB8::2/128    2001:DB8:0:23::2       0             0 65
 *>  2001:DB8::3/128    ::                     0         32768 i
 *>  2001:DB8:0:1::/64  2001:DB8:0:23::2                     0 65
 *>  2001:DB8:0:3::/64  ::                     0         32768 i
 *>  2001:DB8:0:12::/64 2001:DB8:0:23::2       0             0 65
 *>  2001:DB8:0:23::/64 ::                     0         32768 i
```

The BGP path attributes for an IPv6 route are displayed with the command **show bgp ipv6 unicast** *prefix/prefix-length*. Example 11-32 shows R3 examining R1's loopback address. Some of the common fields, such as AS_Path, origin, and local preference, are identical to those for IPv4 routes.

Example 11-32 Viewing the BGP Path Attributes for an IPv6 Route

```
R3# show bgp ipv6 unicast 2001:DB8::1/128
BGP routing table entry for 2001:DB8::1/128, version 9
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 2
  65200 65100
    2001:DB8:0:23::2 (FE80::2) from 2001:DB8:0:23::2 (192.168.2.2
      Origin incomplete, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

Example 11-33 shows the IPv6 BGP route entries for R2. Notice that the next-hop address is the link-local address for the next-hop forwarding address, which is resolved through a recursive lookup.

Example 11-33 Global RIB for BGP Learned IPv6 Routes

```
R2# show ipv6 route bgp
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NE
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr -
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA e
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn
       a - Application
B   2001:DB8::1/128 [20/0]
     via FE80::1, GigabitEthernet0/0
B   2001:DB8::3/128 [20/0]
     via FE80::3, GigabitEthernet0/1
B   2001:DB8:0:1::/64 [20/0]
     via FE80::1, GigabitEthernet0/0
B   2001:DB8:0:3::/64 [20/0]
     via FE80::3, GigabitEthernet0/1
```

Key Topic

IPv6 Summarization

The same process for summarizing or aggregating IPv4 routes occurs with IPv6 routes, and the format is identical except that the configuration is placed under the IPv6 address family using the command **aggregate-address** *prefix/prefix-length* [**summary-only**] [**as-set**].

Let's revisit the previous IPv6 deployment but now want to summarize all the loopback addresses (2001:db8:0:1/128, 2001:db8:0:2/128, and 2001:db8:0:3/128) along with the peering link between R1 and R2 (2001:db8:0:12/64) on R2. The configuration would look as shown in Example 11-34.

Example 11-34 Configuring IPv6 BGP Aggregation on R2

```
router bgp 65200
 bgp router-id 192.168.2.2
 bgp log-neighbor-changes
 neighbor 2001:DB8:0:12::1 remote-as 65100
 neighbor 2001:DB8:0:23::3 remote-as 65300
 !
 address-family ipv4
  no neighbor 2001:DB8:0:12::1 activate
  no neighbor 2001:DB8:0:23::3 activate
 exit-address-family
 !
```

```
    address-family ipv6
     bgp scan-time 6
     network 2001:DB8::2/128
     network 2001:DB8:0:12::/64
     aggregate-address 2001:DB8::/59 summary-only
     neighbor 2001:DB8:0:12::1 activate
     neighbor 2001:DB8:0:23::3 activate
    exit-address-family
```

Example 11-35 shows the BGP tables on R1 and R3. You can see that all the smaller routes have been aggregated and suppressed into 2001:db8::/59, as expected.

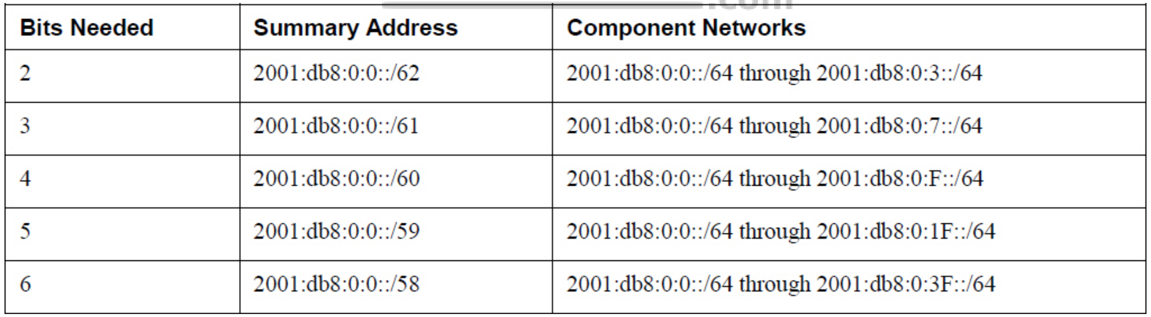
Example 11-35 Verifying IPv6 Route Aggregation

```
R3# show bgp ipv6 unicast | b Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::/59      2001:DB8:0:23::2         0              0 65
 *>  2001:DB8::3/128    ::                       0          32768 i
 *>  2001:DB8:0:3::/64  ::                       0          32768 i
 *>  2001:DB8:0:23::/64 ::                       0          32768 i

R1# show bgp ipv6 unicast | b Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::/59      2001:DB8:0:12::2         0              0 65
 *>  2001:DB8::1/128    ::                       0          32768 ?
 *>  2001:DB8:0:1::/64  ::                       0          32768 ?
 *>  2001:DB8:0:12::/64 ::                       0          32768 ?
 *>  2001:DB8:0:23::/64 2001:DB8:0:12::2         0 65200 65300 i
```

The summarization of the IPv6 loopback addresses (2001:db8:0:1/128, 2001:db8:0:2/128, and 2001:db8:0:3/128) is fairly simple as they all fall into the base IPv6 summary range 2001:db8:0:0::/64. The fourth hextet beginning with a decimal value of 1, 2, or 3 would consume only 2 bits; the range could be summarized easily into the 2001:db8:0:0::/62 (or 2001:db8::/62) network range.

The peering link between R1 and R2 (2001:db8:0:12::/64) requires thinking in hex first, rather than in decimal values. The fourth hextet carries a decimal value of 18 (not 12), which requires 5 bits minimum. Table 11-5 lists the bits needed for summarization, the IPv6 summary address, and the component networks in the summary range.

Table 11-5 IPv6 Summarization Table



Bits Needed	Summary Address	Component Networks
2	2001:db8:0:0::/62	2001:db8:0:0::/64 through 2001:db8:0:3::/64
3	2001:db8:0:0::/61	2001:db8:0:0::/64 through 2001:db8:0:7::/64
4	2001:db8:0:0::/60	2001:db8:0:0::/64 through 2001:db8:0:F::/64
5	2001:db8:0:0::/59	2001:db8:0:0::/64 through 2001:db8:0:1F::/64
6	2001:db8:0:0::/58	2001:db8:0:0::/64 through 2001:db8:0:3F::/64

Currently the peering link between R2 and R3 (2001:db8:0:23::/64) is not being summarized and suppressed, as it is still visible in R1’s routing table in Example 11-35. The hex value of 23 (i.e. 0x23) converts to a decimal value of 35, which requires 6 bits. The summarized network range must be changed to 2001:db8::/58 for summarization of the 2001:db9:0:23::/64 network to occur. Example 11-36 shows the configuration change being made to R2.

Example 11-36 Configuring a Change to Summarize the 2001:db8:0:23::/64 Network

```
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# router bgp 65200
R2(config-router)# address-family ipv6 unicast
R2(config-router-af)# no aggregate-address 2001:DB8::/59 summary-
R2(config-router-af)# aggregate-address 2001:DB8::/58 summary-onl
```

Example 11-37 verifies that the 2001:db8:0:23::/64 is now within the aggregate address space and is no longer being advertised to R1.

Example 11-37 Verifying Summarization of the 2001:db8:0:23::/64 Network

```
R1# show bgp ipv6 unicast | b Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  2001:DB8::/58      2001:DB8:0:12::2         0             0 65
 *>  2001:DB8::1/128    ::                       0         32768 ?
 *>  2001:DB8:0:1::/64  ::                       0         32768 ?
 *>  2001:DB8:0:12::/64 ::                       0         32768 ?
```

EXAM PREPARATION TASKS

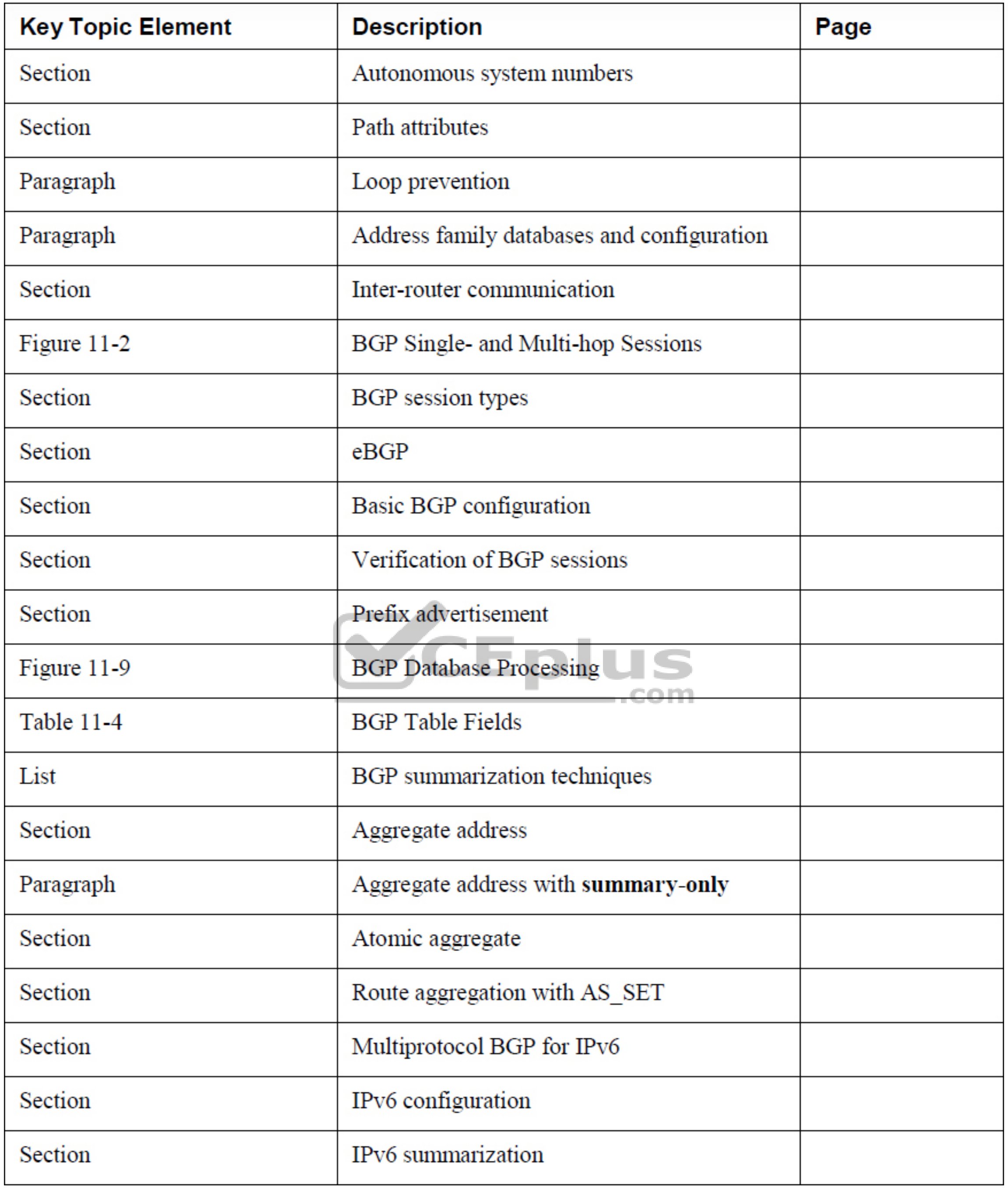
As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 11-6 lists these key topics and the page number on which each is found.

Key Topic

Table 11-6 Key Topics for Chapter 11



Key Topic Element	Description	Page
Section	Autonomous system numbers	
Section	Path attributes	
Paragraph	Loop prevention	
Paragraph	Address family databases and configuration	
Section	Inter-router communication	
Figure 11-2	BGP Single- and Multi-hop Sessions	
Section	BGP session types	
Section	eBGP	
Section	Basic BGP configuration	
Section	Verification of BGP sessions	
Section	Prefix advertisement	
Figure 11-9	BGP Database Processing	
Table 11-4	BGP Table Fields	
List	BGP summarization techniques	
Section	Aggregate address	
Paragraph	Aggregate address with **summary-only**	
Section	Atomic aggregate	
Section	Route aggregation with AS_SET	
Section	Multiprotocol BGP for IPv6	
Section	IPv6 configuration	
Section	IPv6 summarization	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

address family

AS_Path

atomic aggregate

autonomous system (AS)

eBGP session

iBGP session

Loc-RIB table

optional non-transitive

optional transitive

path vector routing protocol

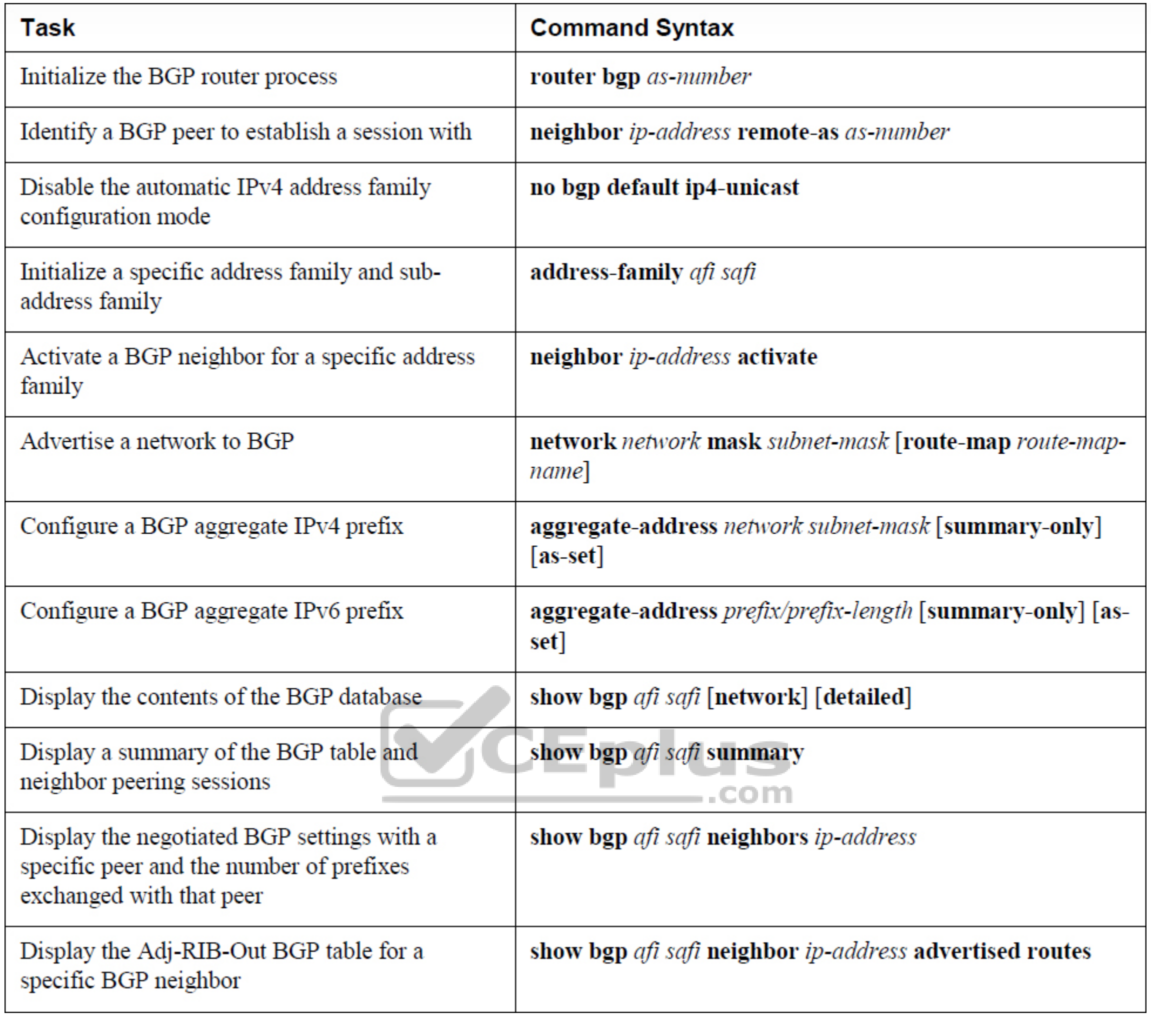
well-known mandatory

well-known discretionary

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 11-7 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 11-7 Command Reference



Task	Command Syntax
Initialize the BGP router process	**router bgp** *as-number*
Identify a BGP peer to establish a session with	**neighbor** *ip-address* **remote-as** *as-number*
Disable the automatic IPv4 address family configuration mode	**no bgp default ip4-unicast**
Initialize a specific address family and sub-address family	**address-family** *afi safi*
Activate a BGP neighbor for a specific address family	**neighbor** *ip-address* **activate**
Advertise a network to BGP	**network** *network* **mask** *subnet-mask* [**route-map** *route-map-name*]
Configure a BGP aggregate IPv4 prefix	**aggregate-address** *network subnet-mask* [**summary-only**] [**as-set**]
Configure a BGP aggregate IPv6 prefix	**aggregate-address** *prefix/prefix-length* [**summary-only**] [**as-set**]
Display the contents of the BGP database	**show bgp** *afi safi* [**network**] [**detailed**]
Display a summary of the BGP table and neighbor peering sessions	**show bgp** *afi safi* **summary**
Display the negotiated BGP settings with a specific peer and the number of prefixes exchanged with that peer	**show bgp** *afi safi* **neighbors** *ip-address*
Display the Adj-RIB-Out BGP table for a specific BGP neighbor	**show bgp** *afi safi* **neighbor** *ip-address* **advertised routes**

REFERENCES IN THIS CHAPTER

RFC 1654, *A Border Gateway Protocol 4 (BGP-4)*, by Yakov Rekhter and Tony Li, https://www.ietf.org/rfc/rfc1654.txt, July 1994.

RFC 2858, *Multiprotocol Extensions for BGP-4*, by Yakov Rekhter, Tony Bates, Ravi Chandra, and Dave Katz, https://www.ietf.org/rfc/rfc2858.txt, June 2000.

Chapter 12. Advanced BGP

This chapter covers the following subjects:

- **BGP Multihoming:** This section reviews the methods of providing resiliency through redundant BGP connections, along with desired and undesired design considerations for Internet and MPLS connections (branch and data center).

- **Conditional Matching:** This section provides an overview of how network prefixes can be conditionally matched with ACLs, prefix lists, and regular expressions.

- **Route Maps:** This section explains the structure of a route map and how conditional matching and conditional actions can be combined to filter or manipulate routes.

- **BGP Route Filtering and Manipulation:** This section expands on how conditional matching and route maps work by applying real-world use cases to demonstrate the filtering or manipulation of BGP routes.

- **BGP Communities:** This section explains the BGP well-known mandatory path attribute and how it can be used to tag a prefix to have route policies applied by routers in the same autonomous system or in an external autonomous system.

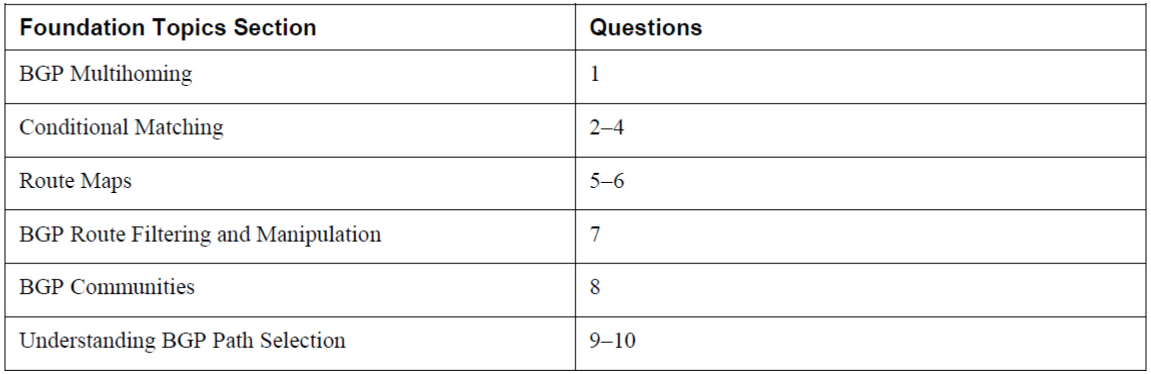
- **Understanding BGP Path Selection:** This section describes the logic used by BGP to identify the best path when multiple routes are installed in the BGP table.

Border Gateway Protocol (BGP) can support hundreds of thousands of routes, making it the ideal choice for the Internet. Organizations also use BGP for its flexibility and traffic engineering properties. This chapter expands on Chapter 11, "Border Gateway Protocol (BGP)," explaining BGP's advanced features and concepts involved with the BGP routing protocol, such as BGP multihoming, route filtering, BGP communities, and the logic for identifying the best path for a specific network prefix.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 12-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 12-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping



Foundation Topics Section	Questions
BGP Multihoming	1
Conditional Matching	2–4
Route Maps	5–6
BGP Route Filtering and Manipulation	7
BGP Communities	8
Understanding BGP Path Selection	9–10

1. Transit routing between a multihomed enterprise network and a service provider is generally not recommend in which scenarios? (Choose all that apply.)

a. Internet connections at data centers

b. Internet connections at branch locations

c. MPLS data centers

d. MPLS branch locations

2. True or false: An extended ACL used to match routes changes behavior if the routing protocol is an IGP rather than BGP.

a. True

b. False

3. Which network prefixes match the prefix match pattern 10.168.0.0/13 ge 24? (Choose two.)

a. 10.168.0.0/13

b. 10.168.0.0/24

c. 10.173.1.0/28

d. 10.104.0.0/24

4. What is the correct regular expression syntax for matching a route that originated in AS 300?

a. ^300_

b. $300!

c. _300_

d. _300$

5. What happens when the route map **route-map QUESTION permit 20** does not contain a conditional match statement?

a. The routes are discarded, and a syslog message is logged.

b. All routes are discarded.

c. All routes are accepted.

d. An error is assigned when linking the route map to a BGP peer.

6. What happens to a route that does not match the PrefixRFC1918 prefix list when using the following route map?

```
route-map QUESTION deny 10
  match ip address prefix-list PrefixRFC1918
route-map QUESTION permit 20
  set metric 200
```

a. The route is allowed, and the metric is set to 200.

b. The route is denied.

c. The route is allowed.

d. The route is allowed, and the default metric is set to 100.

7. True or false: A BGP AS_Path ACL and a prefix list can be applied to a neighbor at the same time.

a. True

b. False

8. Which of the following is not a well-known BGP community?

a. No_Advertise

b. Internet

c. No_Export

d. Private_Route

9. Which of the following techniques is the second selection criterion for the BGP best path?

a. Weight

b. Local preference

c. Origin

d. MED

10. True or false: For MED to be used as a selection criterion, the routes must come from different autonomous systems.

a. True

b. False

Answers to the "Do I Know This Already?" quiz:

1. A, B, D

2. A

3. B, C

4. D

5. C

6. A

7. A

8. D

9. B

10. B

FOUNDATION TOPICS

The Internet has become a vital component for businesses today. Internet connectivity required for email and research at a minimum. In addition, some organizations host e-commerce servers, use Voice over IP (VoIP) telephony, or terminate VPN tunnels through private MPLS connections. An organization must incorporate redundancies in the network architecture to ensure that there are not any single points of failure (SPOF) with network connectivity to support the needs of the business.

A company can connect to the Internet with a simple default route using a single connection. However, if a company wants to use multiple service providers (SPs) for redundancy or additional throughput, BGP is required. BGP is the routing protocol used on the Internet.

A company's use of BGP is not limited to Internet connectivity. If the company uses MPLS L3VPN from a service provider, it is probably using BGP to exchange the LAN networks with the service provider. Routes are typically redistributed between BGP and the LAN-based routing protocol. In both of these scenarios, BGP is used at the edge of the network (Internet or WAN) and has redundant connections to ensure a reliable network. It provides advanced path selection and connectivity for an organization. This chapter focuses on troubleshooting BGP edge architectures.

BGP MULTIHOMING

The simplest method of providing redundancy is to provide a second circuit. Adding a second circuit and establishing a second BGP session across that peering link is known as *BGP multihoming* because there are multiple sessions to learn routes and establish connectivity. BGP's default behavior is to advertise only the best path to the RIB, which means that only one path for a network prefix is used when forwarding network traffic to a destination.

Key Topic

Resiliency in Service Providers

Routing failures can occur within a service provider network, and some organizations chose to use a different SP for each circuit. A second service provider could be selected for a variety of reasons, but the choice typically comes down to cost, circuit availability for remote locations, or separation of the control plane.

By using a different SP, if one SP has problems in its network, network traffic can still flow across the other SP. In addition, adding more SPs means traffic can select an optimal path between devices due to the BGP best-path algorithm, discussed later in this chapter.

Figure 12-1 illustrates four common multihoming scenarios:

- **Scenario 1:** R1 connects to R3 with the same SP. This design accounts for link failures; however, a failure on either router or within SP1’s network results in a network failure.

- **Scenario 2:** R1 connects to R3 and R4 with the same SP. This design accounts for link failures; however, a failure on R1 or within SP1’s network results in a network failure.

- **Scenario 3:** R1 connects to R3 and R4 with the different SPs. This design accounts for link failures and failures in either SP’s network, and it can optimize routing traffic. However, a failure on R1 results in a network failure.

- **Scenario 4:** R1 and R2 form an iBGP session with each other. R3 connects to SP1, and R4 connects to SP2. This design accounts for link failures and failures in either SP's network, and it can optimize routing traffic.

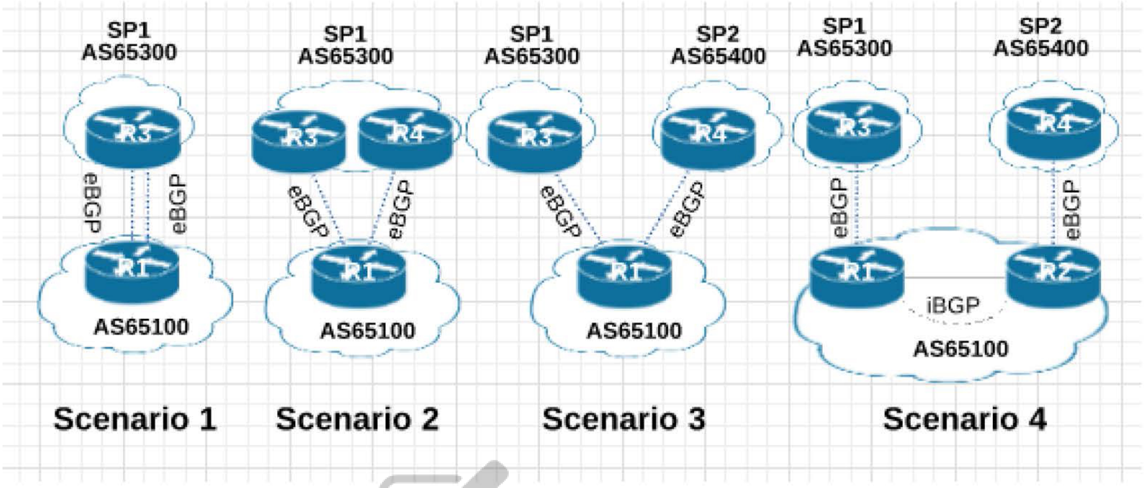


Figure 12-1 Common BGP Multihoming Scenarios

Key Topic

Internet Transit Routing

If an enterprise uses BGP to connect with more than one service provider, it runs the risk of its autonomous system (AS) becoming a transit AS. In Figure 12-2, AS 500 is connecting to two different service providers (SP3 and SP4) for resiliency.

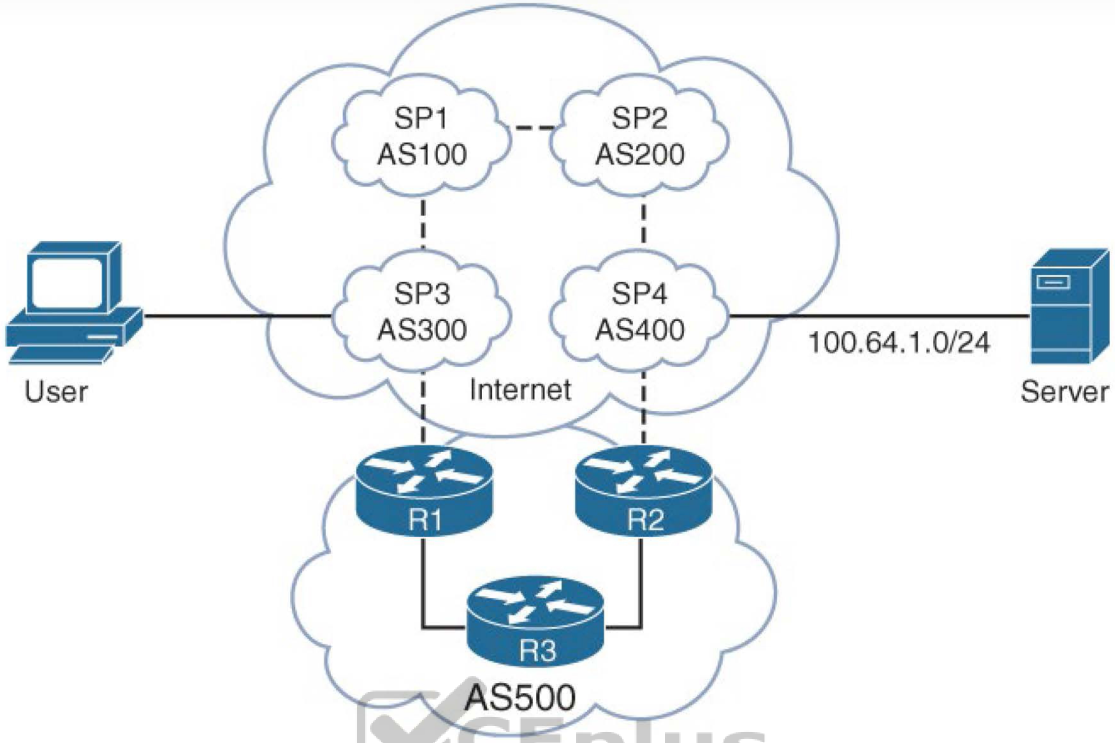


Figure 12-2 Enterprise Transit Routing

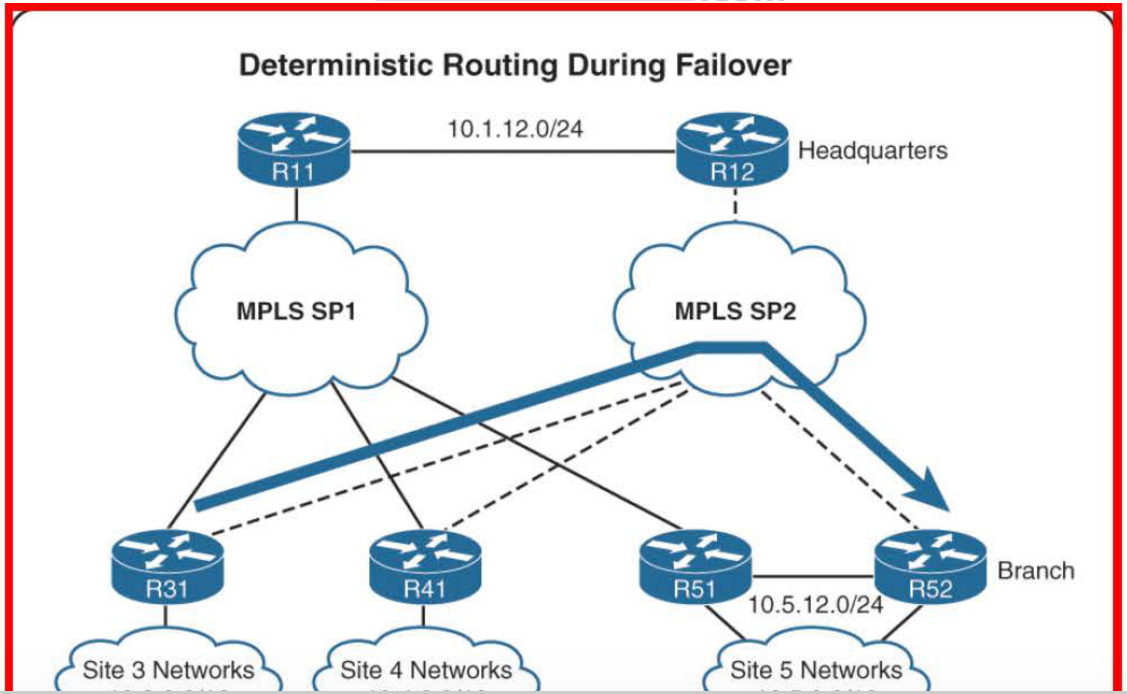
Problems can arise if R1 and R2 use the default BGP routing policy. A user that connects to SP3 (AS 300) routes through the enterprise network (AS 500) to reach a server that attaches to SP4 (AS 400). SP3 receives the 100.64.1.0/24 prefix from AS 100 and AS 500. SP3 selects the path through AS 500 because the AS_Path is much shorter than going through SP1 and SP2's networks.

The AS 500 network is providing transit routing to everyone on the Internet, which can saturate AS 500's peering links. In addition to causing problems for the users in AS 500, this situation has an impact on traffic from the users that are trying to transverse AS 500.

Transit routing can be avoided by applying outbound BGP route policies that only allow for local BGP routes to be advertised to other autonomous systems. This is discussed later in this chapter, in the section “BGP Route Filtering and Manipulation.”

Branch Transit Routing

Proper network design should take traffic patterns into account to prevent suboptimal routing or routing loops. Figure 12-3 shows a multihomed design using multiple transports for all the sites. All the routers are configured so that they prefer the MPLS SP2 transport over the MPLS SP1 transport (active/passive). All the routers peer and advertise all the routes via eBGP to the SP routers. The routers do not filter any of the prefixes, and the set the local preference for MPLS SP2 to a higher value to route traffic through it.



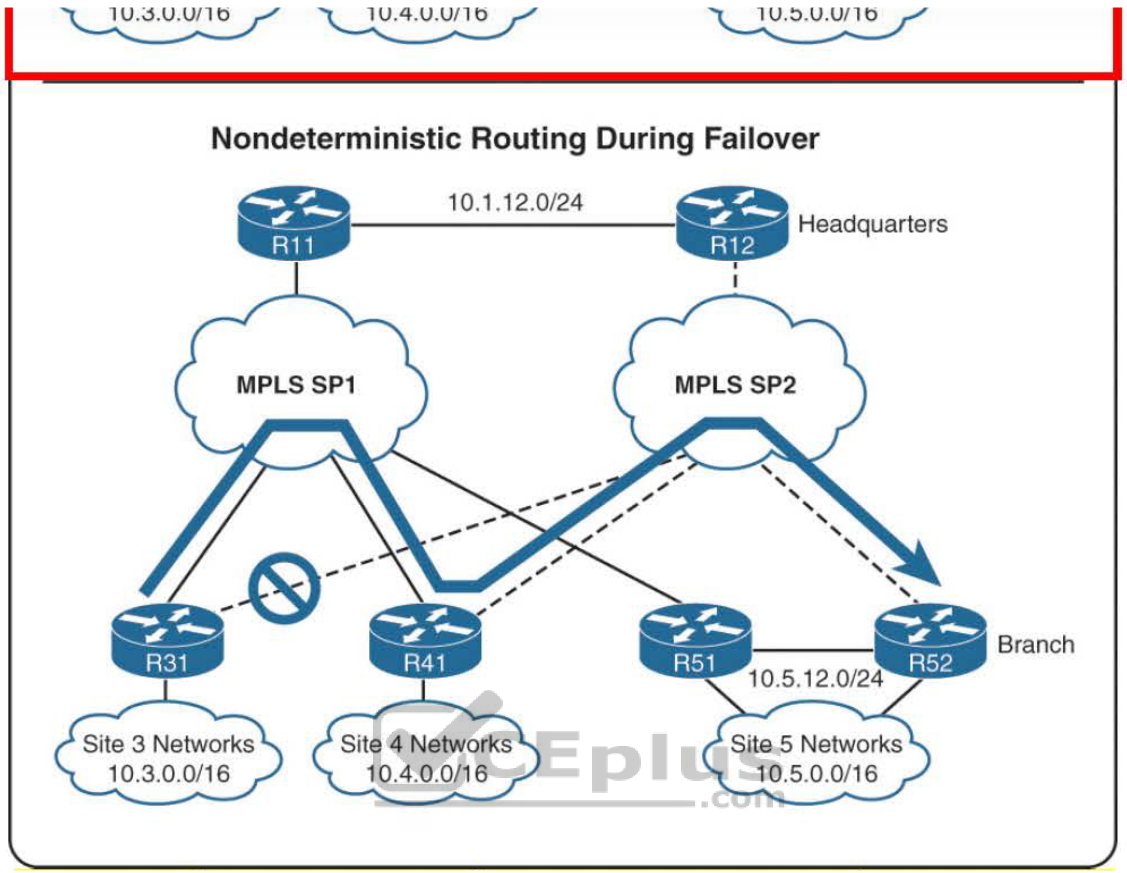
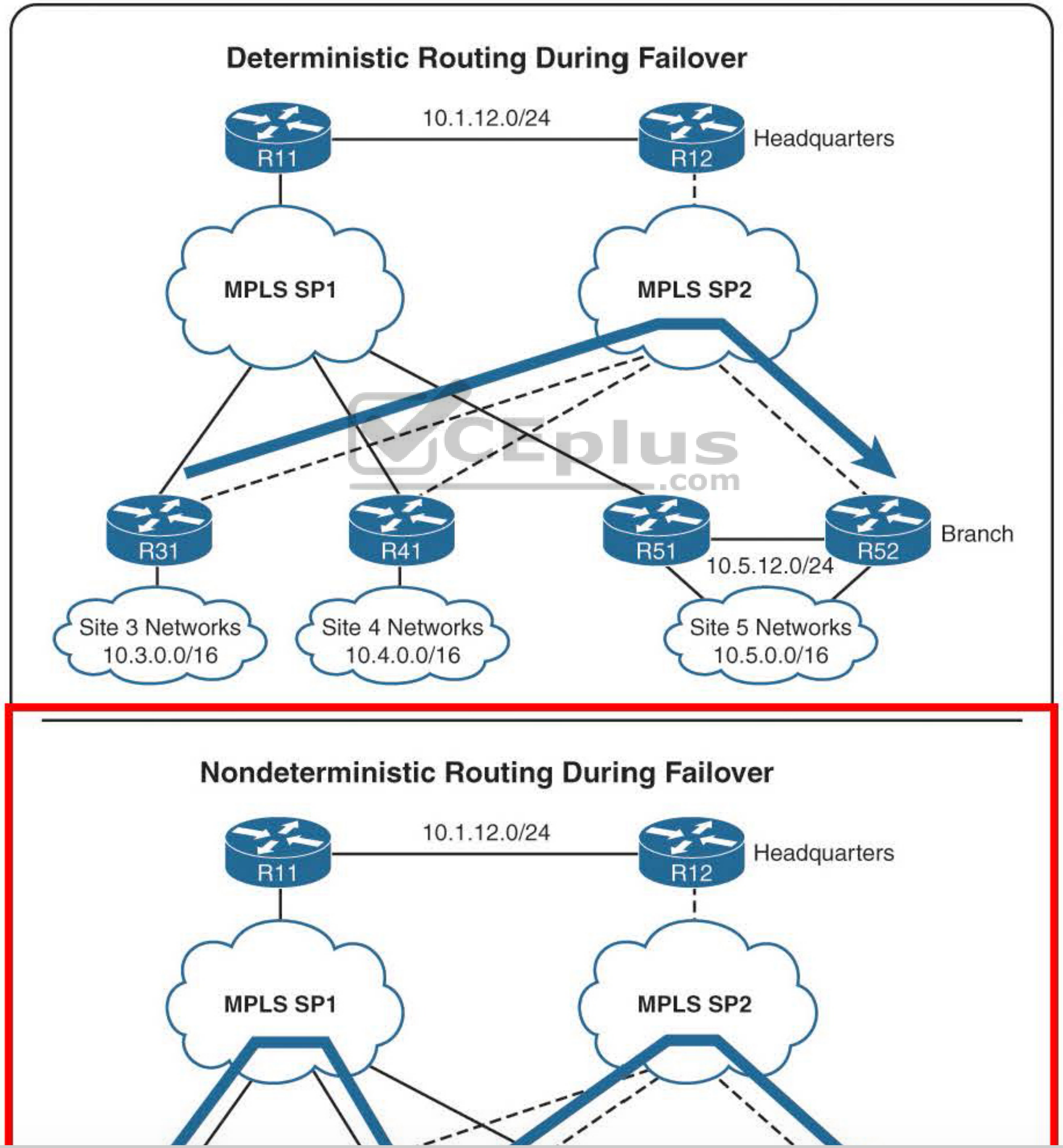


Figure 12-3 Deterministic Routing

When the network is working as intended, traffic between the sites uses the preferred SP network (MPLS SP2) in both directions. This simplifies troubleshooting when the traffic flow is symmetric (same path in both directions) as opposed to asymmetric forwarding (a different path for each direction) because the full path has to be discovered in both directions. The path is considered *deterministic* when the flow between sites is predetermined and predictable.

During a link failure within the SP network, there is a possibility of a branch router connecting to the destination branch router through an intermediary branch router. Figure 12-4 shows the failure scenario with R41 providing transit connectivity between Site 3 and Site 5.



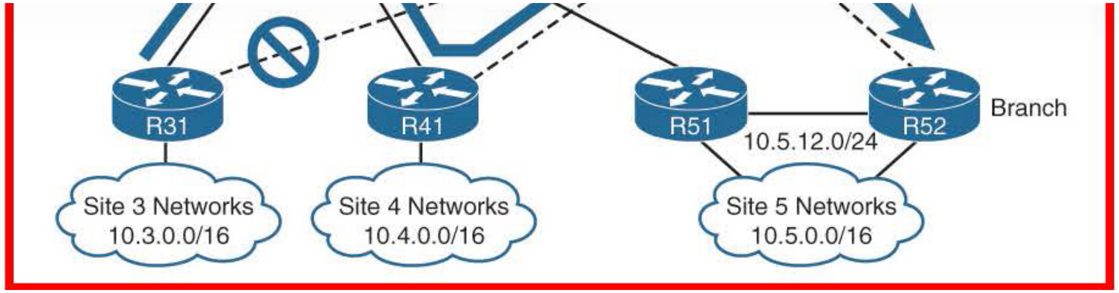


Figure 12-4 Nondeterministic Routing During Failover

Unplanned transit connectivity presents the following issues:

- The transit router's circuits can become oversaturated because they were sized only for that site's traffic and not the traffic crossing through them.

- The routing patterns can become unpredictable and nondeterministic. In this scenario, traffic from R31 may flow through R41, but the return traffic may take a different return path. The path might be very different if the traffic were sourced from a different router. This prevents deterministic routing, complicates troubleshooting, and can make your NOC staff feel as if they are playing whack-a-mole when troubleshooting network issues.

Multihomed environments should be configured so that branch routers cannot act as transit routers. In most designs, transit routing of traffic from another branch is undesirable, as WAN bandwidth may not be sized accordingly. Transit routing can be avoided by configuring outbound route filtering at each branch site. In essence, the branch sites do not advertise what they learn from the WAN but advertise only networks that face the LAN. If transit behavior is required, it is restricted to the data centers or specific locations as follows:

- Proper routing design can accommodate outages.

- Bandwidth can be sized accordingly.

- The routing pattern is bidirectional and predictable.

Note

Transit routing at the data center or other planned locations is normal in enterprise designs as they have accounted for the bandwidth. Typically, this is done when a portion of branches are available only with one SP, and the other branches connect with a different SP.

CONDITIONAL MATCHING

Applying bulk changes to routes on a neighbor-by-neighbor basis (or interface-by-interface basis for IGPs) does not easily allow for tuning of the network. This section reviews some of the common techniques used to conditionally matching a route—using access control lists (ACLs), prefix lists, regular expressions (regex), and AS path ACLs.

Access Control Lists

Originally, access control lists (ACLs) were intended to provide filtering of packets flowing into or out of a network interface, similar to the functionality of a basic firewall. Today, ACLs provide packet classification for a variety of features, such as quality of service (QoS), or for identifying networks within routing protocols.

ACLs are composed of *access control entries (ACEs)*, which are entries in the ACL that identify the action to be taken (permit or deny) and the relevant packet classification. Packet classification starts at the top (lowest sequence) and proceeds down (higher sequence) until a matching pattern is identified. Once a match is found, the appropriate action (permit or deny) is taken, and processing stops. At the end of every ACLs is an implicit deny ACE, which denies all packets that did not match earlier in the ACL.

Note

ACE placement within an ACL is important, and unintended consequences may result from ACEs being out of order.

ACLs are classified into two categories:

• **Standard ACLs:** Define packets based solely on the source network.

• **Extended ACLs:** Define packets based on source, destination, protocol, port, or a combination of other packet attributes. This book is concerned with routing and limits the scope of ACLs to source, destination, and protocol.

Standard ACLS use a numbered entry 1–99, 1300–1999, or a named ACL. Extended ACLs use a numbered entry 100–199, 2000–2699, or a named ACL. Named ACLs provide relevance to the functionality of the ACL, can be used with standard or extended ACLs, and are generally preferred.

Standard ACLs

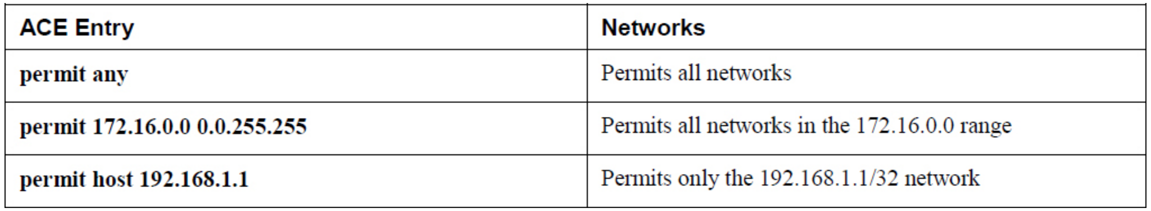
The following is the process for defining a standard ACL:

Step 1. Define the ACL by using the command **ip access-list standard** {*acl-number* | *acl-name*} and placing the CLI in ACL configuration mode.

Step 2. Configure the specific ACE entry with the command [*sequence*] {**permit** | **deny** } *source source-wildcard*. In lieu of using the *source source-wildcard*, the keyword **any** replaces 0.0.0.0 0.0.0.0, and use of the **host** keyword refers to a /32 IP address so that the *source-wildcard* can be omitted.

Table 12-2 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with a standard ACL.

Table 12-2 Standard ACL-to-Network Entries



ACE Entry	Networks
permit any	Permits all networks
permit 172.16.0.0 0.0.255.255	Permits all networks in the 172.16.0.0 range
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

Extended ACLs

The following is the process for defining an extended ACL:

Step 1. Define the ACL by using the command **ip access-list extended** {*acl-number* | *acl-name*} and placing the CLI in ACL configuration mode.

Step 2. Configure the specific ACE entry with the command [*sequence*] {**permit** | **deny** } *protocol source source-wildcard destination destination-wildcard*. The behavior for selecting a network prefix with an extended ACL varies depending on whether the protocol is an IGP (EIGRP, OSPF, or IS-IS) or BGP.

Key Topic

IGP Network Selection

When ACLS are used for IGP network selection, the source fields of the ACL are used to identify the network, and the destination fields identify the smallest prefix length allowed in the network range. Table 12-3 provides sample ACL entries from within the ACL configuration mode and specifies the networks that would match with the extended ACL. Notice that the subtle difference in the destination wildcard for the 172.16.0.0 network affects the network ranges that are permitted in the second and third rows of the table.

Table 12-3 Extended ACL for IGP Route Selection

ACE Entry	Networks
permit ip any any	Permits all networks
permit ip host 172.16.0.0 host 255.240.0.0	Permits all networks in the 172.16.0.0/12 range
permit ip host 172.16.0.0 host 255.255.0.0	Permits all networks in the 172.16.0.0/16 range
permit host 192.168.1.1	Permits only the 192.168.1.1/32 network

Key Topic

BGP Network Selection

Extended ACLs react differently when matching BGP routes than when matching IGP routes. The source fields match against the network portion of the route, and the destination fields match against the network mask, as shown in Figure 12-5. Until the introduction of prefix lists, extended ACLs were the only match criteria used with BGP.

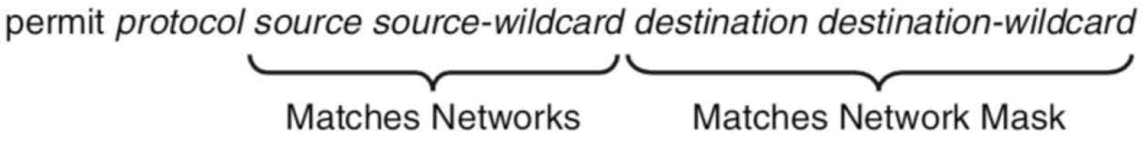
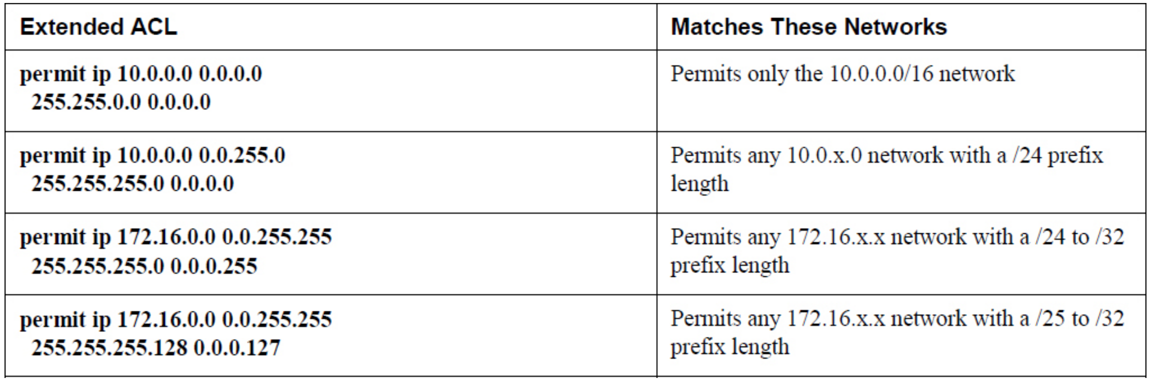

permit *protocol source source-wildcard destination destination-wildcard*
Matches Networks
Matches Network Mask

Figure 12-5 BGP Extended ACL Matches

Table 12-4 demonstrates the concept of the wildcard for the network and subnet mask.

Table 12-4 Extended ACL for BGP Route Selection



Extended ACL	Matches These Networks
permit ip 10.0.0.0 0.0.0.0 255.255.0.0 0.0.0.0	Permits only the 10.0.0.0/16 network
permit ip 10.0.0.0 0.0.255.0 255.255.255.0 0.0.0.0	Permits any 10.0.x.0 network with a /24 prefix length
permit ip 172.16.0.0 0.0.255.255 255.255.255.0 0.0.0.255	Permits any 172.16.x.x network with a /24 to /32 prefix length
permit ip 172.16.0.0 0.0.255.255 255.255.255.128 0.0.0.127	Permits any 172.16.x.x network with a /25 to /32 prefix length

Prefix Matching

Prefix lists provide another method of identifying networks in a routing protocol. A prefix list identifies a specific IP address, network, or network range and allows for the selection of multiple networks with a variety of prefix lengths by using a prefix match specification. Many network engineers prefer this over the ACL network selection method.

Key Topic

A prefix match specification contains two parts: a high-order bit pattern and a high-order bit count, which determines the high-order bits in the bit pattern that are to be matched. Some documentation refers to the high-order bit pattern as the address or network and the high-order bit count as the length or mask length.

In Figure 12-6, the prefix match specification has the high-order bit pattern 192.168.0.0 and the high-order bit count 16. The high-order bit pattern has been converted to binary to demonstrate where the high-order bit count lies. Because there are not additional matching length parameters included, the high-order bit count is an exact match.

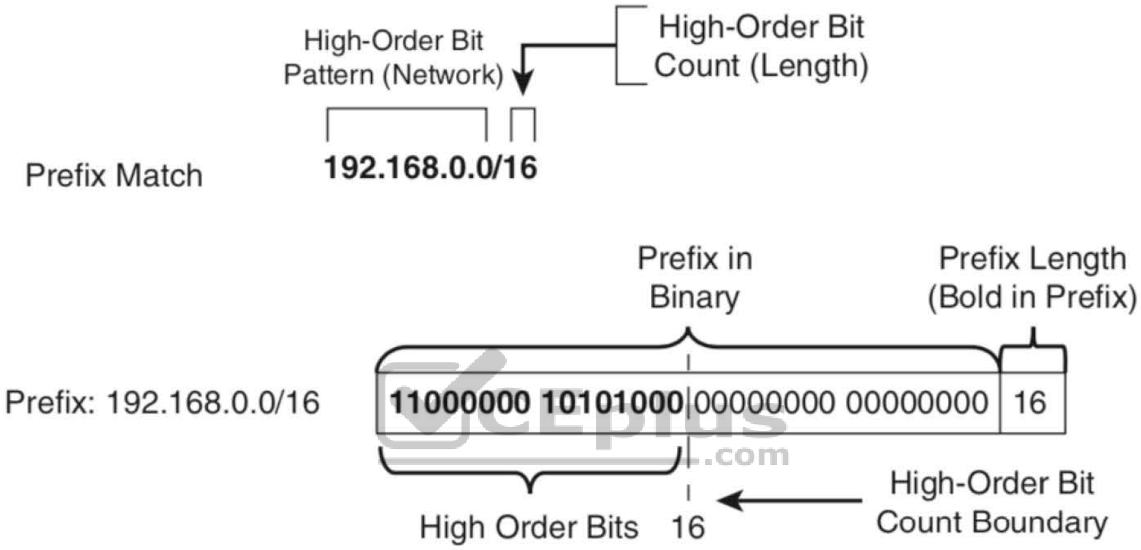

High-Order Bit Pattern (Network)
High-Order Bit Count (Length)
Prefix Match
192.168.0.0/16
Prefix in Binary
Prefix Length (Bold in Prefix)
Prefix: 192.168.0.0/16
11000000 10101000 00000000 00000000
16
High Order Bits
16
High-Order Bit Count Boundary

Figure 12-6 Basic Prefix Match Pattern

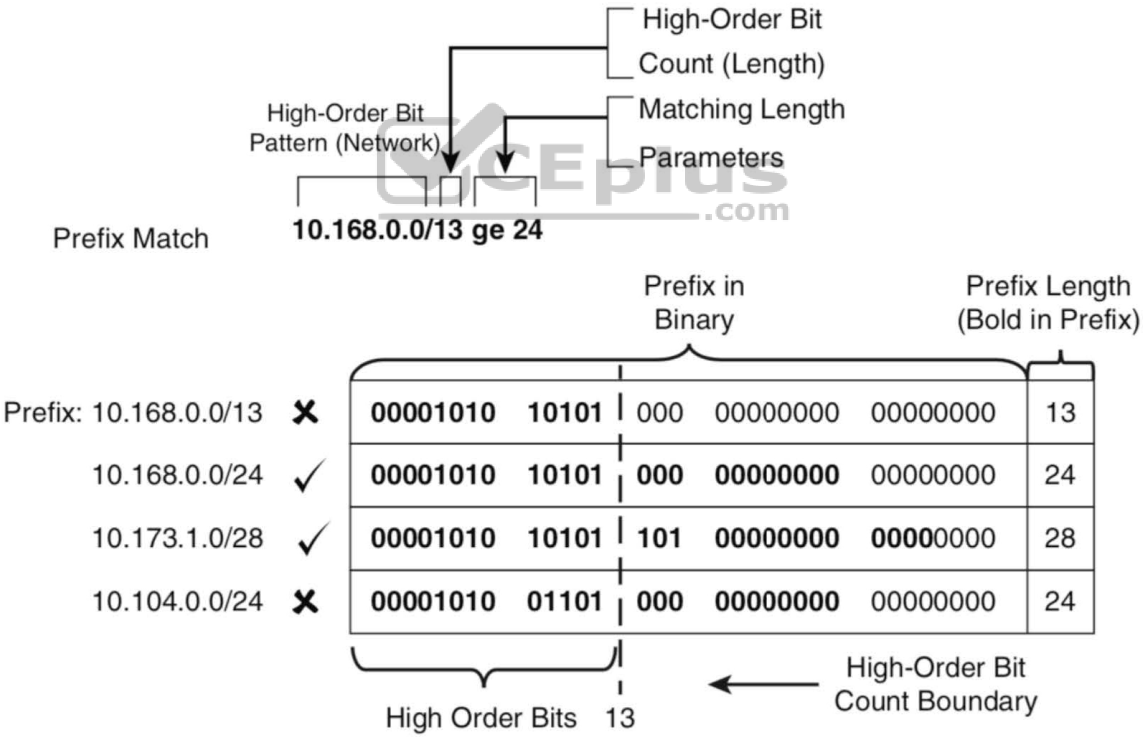
Key Topic

At this point, the prefix match specification logic looks identical to the functionality of an access list. The true power and flexibility comes in using matching length parameters to identify multiple networks with specific prefix lengths with one statement. The matching length parameter options are:

- **le:** Less than or equal to, <=

- **ge:** Greater than or equal to, >=

Figure 12-7 demonstrates the prefix match specification with the high-order bit pattern 10.168.0.0 and high-order bit count 13; the matching length of the prefix must be greater than or equal to 24.



Prefix: 10.168.0.0/13	✗	**00001010**	**10101**	000	00000000	00000000	13
10.168.0.0/24	✓	**00001010**	**10101**	**000**	**00000000**	00000000	24
10.173.1.0/28	✓	**00001010**	**10101**	**101**	**00000000**	**0000**0000	28
10.104.0.0/24	✗	**00001010**	**01101**	**000**	**00000000**	00000000	24

Figure 12-7 Prefix Match Pattern with Matching Length Parameters

The 10.168.0.0/13 prefix does not meet the matching length parameter because the prefix length is less than the minimum of 24 bits, whereas the 10.168.0.0/24 prefix does meet the matching length parameter. The 10.173.1.0/28 prefix qualifies because the first 13 bits match the high-order bit pattern, and the prefix length is within the matching length parameter. The 10.104.0.0/24 prefix does not qualify because the high-order bit pattern does not match within the high-order bit count.

Figure 12-8 demonstrates a prefix match specification with the high-order bit pattern 10.0.0.0, high-order bit count 8, and matching length between 22 and 26.

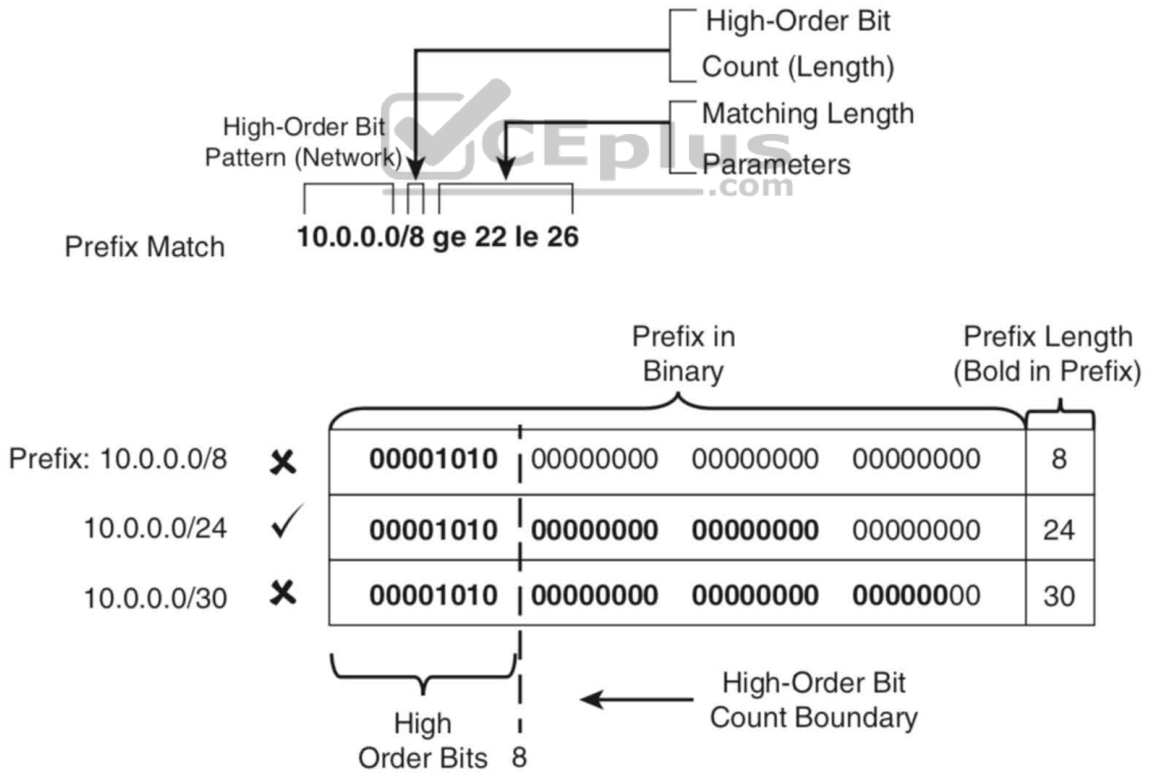


Figure 12-8 Prefix Match with Ineligible Matched Prefixes

The 10.0.0.0/8 prefix does not match because the prefix length is too short. The 10.0.0.0/24 network qualifies because the bit pattern matches, and the prefix length is between 22 and 26. The 10.0.0.0/30 prefix does not match because the bit pattern is too long. Any prefix that starts with 10 in the first octet and has a prefix length between 22 and 26 will match.

> **Note**
>
> Matching to a specific prefix length that is higher than the high-order bit count requires that the *ge-value* and *le-value* match.

Key Topic

Prefix Lists

Prefix lists can contain multiple prefix matching specification entries that contain a permit or deny action. Prefix lists process in sequential order in a top-down fashion, and the first prefix match processes with the appropriate permit or deny action.

Prefix lists are configured with the global configuration command **ip prefix-list** *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern/high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*].

If a sequence is not provided, the sequence number auto-increments by 5, based on the highest sequence number. The first entry is 5. Sequencing enables the deletion of a specific entry. Because prefix lists cannot be resequenced, it is advisable to leave enough space for insertion of sequence numbers at a later time.

IOS and IOS XE require that the *ge-value* be greater than the high-order bit count and that the *le-value* be greater than or equal to the *ge-value*:

high-order bit count < ge-value <= le-value

Example 12-1 provides a sample prefix list named RFC1918 for all of the networks in the RFC 1918 address range. The prefix list only allows /32 prefixes to exist in the 192.168.0.0 network range and not exist in any other network range in the prefix list.

Notice that sequence 5 permits all /32 prefixes in the 192.168.0.0/13 bit pattern, and sequence 10 denies all /32 prefixes in any bit pattern, and sequences 15, 20, and 25 permit routes in the appropriate network ranges. The sequence order is important for the first two entries to ensure that only /32 prefixes exist in the 192.168.0.0 network in the prefix list.

Example 12-1 Sample Prefix List

```
ip prefix-list RFC1918 seq 5 permit 192.168.0.0/13 ge 32
ip prefix-list RFC1918 seq 10 deny 0.0.0.0/0 ge 32
ip prefix-list RFC1918 seq 15 permit 10.0.0.0/7 ge 8
ip prefix-list RFC1918 seq 20 permit 172.16.0.0/11 ge 12
ip prefix-list RFC1918 seq 25 permit 192.168.0.0/15 ge 16
```

IPv6 Prefix Lists

The prefix matching logic works exactly the same for IPv6 networks as for IPv4 networks. The most important thing to remember is that IPv6 networks are notated in hex and not in binary when identifying ranges. Ultimately, however, everything functions at the binary level.

IPv6 prefix lists are configured with the global configuration command **ipv6 prefix-list** *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern/high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*].

Example 12-2 provides a sample prefix list named PRIVATE-IPV6 for all the networks in the documentation and benchmarking IPv6 space.

Example 12-2 Sample IPv6 Prefix List

```
ipv6 prefix-list PRIVATE-IPV6 seq 5 permit 2001:2::/48 ge 48
ipv6 prefix-list PRIVATE-IPV6 seq 10 permit 2001:db8::/32 ge 32
```

Key Topic

Regular Expressions (regex)

There may be times when conditionally matching on network prefixes may be too complicated, and identifying all routes from a specific organization is preferred. In such a case, path selection can be made by using a BGP AS_Path.

Regular expressions (regex) are used to parse through the large number of available ASNs (4,294,967,295). Regular expressions are based on query modifiers used to select the appropriate content. The BGP table can be parsed with regex by using the command **show bgp** *afi safi* **regexp** *regex-pattern*.

Table 12-5 provides a brief list and description of the common regex query modifiers.

Table 12-5 RegEx Query Modifiers

Modifier	Description
_ (underscore)	Matches a space
^ (caret)	Indicates the start of a string
$ (dollar sign)	Indicates the end of a string
[] (brackets)	Matches a single character or nesting within a range
- (hyphen)	Indicates a range of numbers in brackets
[^] (caret in brackets)	Excludes the characters listed in brackets
() (parentheses)	Used for nesting of search patterns
| (pipe)	Provides OR functionality to the query
. (period)	Matches a single character, including a space
* (asterisk)	Matches zero or more characters or patterns
+ (plus sign)	Matches one or more instances of the character or pattern
? (question mark)	Matches one or no instances of the character or pattern

Learning regex can take time, but the most common ones used in BGP involve the ^, $, and _. Table 12-6 displays some common BGP regex.

Table 12-6 Common BGP Regular Expressions

Regular Expression	Meaning
^$	Local originating routes
permit ^200_	Only routes from neighbor AS 200
permit _200$	Only routes originating from AS 200
permit _200_	Only routes that pass through AS 200
permit ^[0-9]+ [0-9]+ [0-9]+?	Routes with three or fewer AS_Path entries

Note

Hands-on experience is helpful when learning technologies such as regex. There are public servers called *looking glasses* that allow users to log in and view BGP tables. Most of these devices are Cisco routers, but some are from other vendors. These servers allow network engineers to see if they are advertising their routes to the Internet as they had intended and provide a great method to try out regular expressions on the Internet BGP table.

A quick search on the Internet will provide website listings of looking glass and route servers. We suggest http://www.bgp4.as (http://www.bgp4.as).

ROUTE MAPS

Route maps provide many different features to a variety of routing protocols. At the simplest level, route maps can filter networks much the same way as ACLs, but they also provide additional capability through the addition or modification of network attributes. To influence a routing protocol, a route map must be referenced from the routing protocol. Route maps are critical to BGP because they are the main component in modifying a unique routing policy on a neighbor-by-neighbor basis.

Key Topic

A route map has four components:

- **Sequence number:** Dictates the processing order of the route map.

- **Conditional matching criteria:** Identifies prefix characteristics (network, BGP path attribute, next hop, and so on) for a specific sequence.

- **Processing action:** Permits or denies the prefix.

- **Optional action:** Allows for manipulations, depending on how the route map is referenced on the router. Actions can include modification, addition, or removal of route characteristics.

Key Topic

A route map uses the command syntax **route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]. The following rules apply to route map statements:

- If a processing action is not provided, the default value **permit** is used.

- If a sequence number is not provided, the sequence number is incremented by 10 automatically.

- If a matching statement is not included, an implied *all prefixes* is associated with the statement.

- Processing within a route map stops after all optional actions have processed (if configured) after matching a conditional matching criterion.

Example 12-3 provides a sample route map to demonstrate the four components of a route map shown earlier. The conditional matching criterion is based on network ranges specified in an ACL. Comments have been added to this example to explain the behavior of the route map in each sequence.

Example 12-3 Sample Route map

```
route-map EXAMPLE permit 10
 match ip address ACL-ONE
! Prefixes that match ACL-ONE are permitted. Route-map completes

route-map EXAMPLE deny 20
 match ip address ACL-TWO
! Prefixes that match ACL-TWO are denied. Route-map completes pr

route-map EXAMPLE permit 30
 match ip address ACL-THREE
 set metric 20
! Prefixes that match ACL-THREE are permitted and modify the met
! processing upon a match
```

```
route-map EXAMPLE permit 40
! Because a matching criteria was not specified, all other prefix
! If this sequence was not configured, all other prefixes would d
! implicit deny  for all route-maps
```

Note

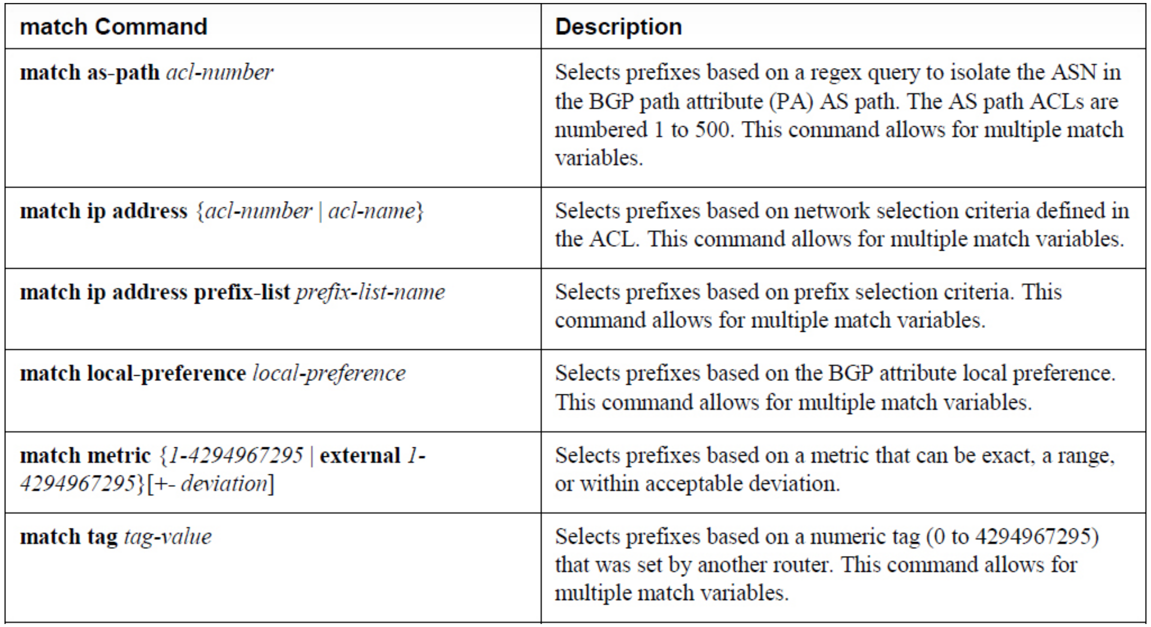
When deleting a specific route-map statement, include the sequence number to prevent deleting the entire route map.

Key Topic

Conditional Matching

Now that the components and processing order of a route map have been explained, this section expands on how a route can be matched. Table 12-7 provides the command syntax for the most common methods for conditionally matching prefixes and describes their usage. As you can see, there are a number of options available.

Table 12-7 Conditional Match Options



match Command	Description
match as-path *acl-number*	Selects prefixes based on a regex query to isolate the ASN in the BGP path attribute (PA) AS path. The AS path ACLs are numbered 1 to 500. This command allows for multiple match variables.
match ip address {*acl-number* | *acl-name*}	Selects prefixes based on network selection criteria defined in the ACL. This command allows for multiple match variables.
match ip address prefix-list *prefix-list-name*	Selects prefixes based on prefix selection criteria. This command allows for multiple match variables.
match local-preference *local-preference*	Selects prefixes based on the BGP attribute local preference. This command allows for multiple match variables.
match metric {*1-4294967295* | **external** *1-4294967295*}[+- *deviation*]	Selects prefixes based on a metric that can be exact, a range, or within acceptable deviation.
match tag *tag-value*	Selects prefixes based on a numeric tag (0 to 4294967295) that was set by another router. This command allows for multiple match variables.

Key Topic

Multiple Conditional Match Conditions

If there are multiple variables (ACLs, prefix lists, tags, and so on) configured for a specific route map sequence, only one variable must match for the prefix to qualify. The Boolean logic uses an OR operator for this configuration.

In Example 12-4, sequence 10 requires that a prefix pass ACL-ONE or ACL-TWO. Notice that sequence 20 does not have a match statement, so all prefixes that are not passed in sequence 10 will qualify and are denied.

Example 12-4 Multiple Match Variables Route Map Example

```
route-map EXAMPLE permit 10
 match ip address ACL-ONE ACL-TWO
!
route-map EXAMPLE deny 20
```

Note

In Example 12-4, sequence 20 is redundant because of the implicit deny for any prefixes that are not matched in sequence 10.

If there are multiple match options configured for a specific route map sequence, both match options must be met for the prefix to qualify for that sequence. The Boolean logic uses an AND operator for this configuration.

In Example 12-5, sequence 10 requires that the prefix match ACL-ONE and that the metric be a value between 500 and 600. If the prefix does not qualify for both match options, the prefix does not qualify for sequence 10 and is denied because another sequence does not exist with a permit action.

Example 12-5 Multiple Match Options Route Map Example

```
route-map EXAMPLE permit 10
 match ip address ACL-ONE
 match metric 550 +- 50
```

Complex Matching

Some network engineers find route maps too complex if the conditional matching criteria use an ACL, an AS path ACL, or a prefix list that contains a **deny** statement. Example 12-6 shows a configuration where the ACL uses a **deny** statement for the 172.16.1.0/24 network range.

Reading configurations like this should follow the sequence order first and conditional matching criteria second, and only after a match occurs should the processing action and optional action be used. Matching a **deny** statement in the conditional match criteria excludes the route from that sequence in the route map.

The prefix 172.16.1.0/24 is denied by ACL-ONE, which implies that there is not a match in sequences 10 and 20; therefore, the processing action (**permit** or **deny**) is not needed. Sequence 30 does not contain a match clause, so any remaining routes are permitted. The prefix 172.16.1.0/24 would pass on sequence 30 with the metric set to 20. The prefix 172.16.2.0/24 would match ACL-ONE and would pass in sequence 10.

Example 12-6 Complex Matching Route Maps

```
ip access-list standard ACL-ONE
 deny   172.16.1.0 0.0.0.255
 permit 172.16.0.0 0.0.255.255

route-map EXAMPLE permit 10
 match ip address ACL-ONE
!
route-map EXAMPLE deny 20
 match ip address ACL-ONE
!
route-map EXAMPLE permit 30
 set metric 20
```

Note

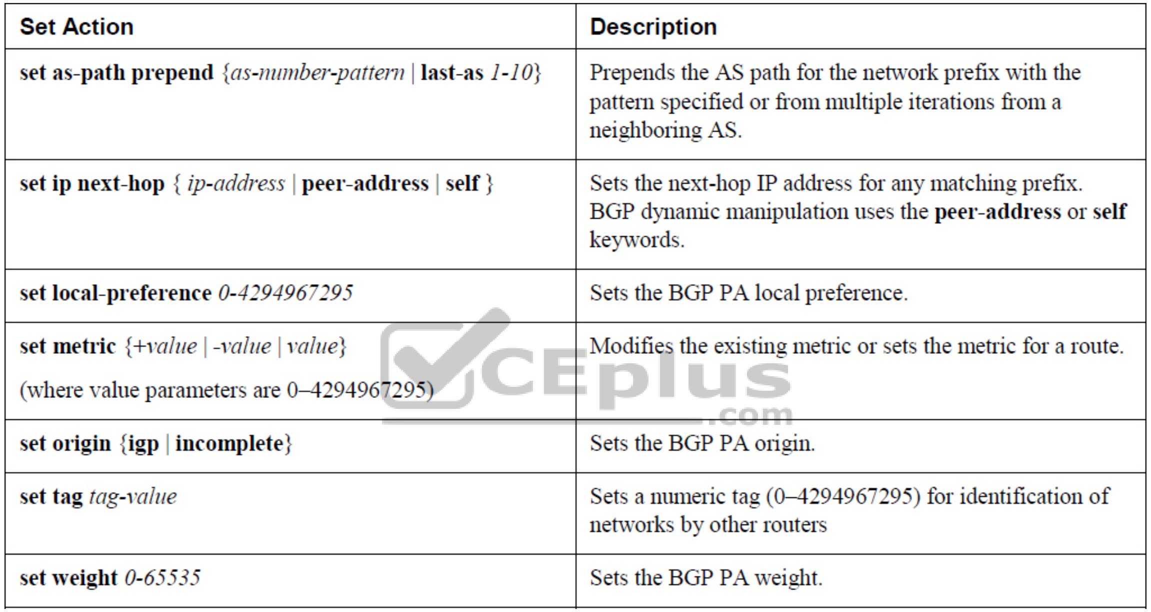
Route maps process using a particular order of evaluation: the sequence, conditional match criteria, processing action, and optional action in that order. Any **deny** statements in the match component are isolated from the route map sequence action.

Key Topic

Optional Actions

In addition to permitting the prefix to pass, route maps can modify route attributes. Table 12-8 provides a brief overview of the most popular attribute modifications.

Table 12-8 Route map Set Actions



Set Action	Description
set as-path prepend {*as-number-pattern* | **last-as** *1-10*}	Prepends the AS path for the network prefix with the pattern specified or from multiple iterations from a neighboring AS.
set ip next-hop { *ip-address* | **peer-address** | **self** }	Sets the next-hop IP address for any matching prefix. BGP dynamic manipulation uses the **peer-address** or **self** keywords.
set local-preference *0-4294967295*	Sets the BGP PA local preference.
set metric {+*value* | -*value* | *value*} (where value parameters are 0–4294967295)	Modifies the existing metric or sets the metric for a route.
set origin {**igp** | **incomplete**}	Sets the BGP PA origin.
set tag *tag-value*	Sets a numeric tag (0–4294967295) for identification of networks by other routers
set weight *0-65535*	Sets the BGP PA weight.

The continue Keyword

Default route map behavior processes the route map sequences in order, and upon the first match, it executes the processing action, performs any optional action (if feasible), and stops processing. This prevents multiple route map sequences from processing.

Adding the keyword **continue** to a route map allows the route map to continue processing other route map sequences. Example 12-7 provides a basic configuration. The network prefix 192.168.1.1 matches in sequences 10, 20, and 30. Because the keyword **continue** was added to sequence 10, sequence 20 processes, but sequence 30 does not because a **continue** command was not present in sequence 20. The 192.168.1.1 prefix is permitted, and it is modified so that the metric is 20, with the next-hop address 10.12.1.1.

Example 12-7 Route Map with the **continue** Keyword

```
ip access-list standard ACL-ONE
 permit 192.168.1.1 0.0.0.0
 permit 172.16.0.0 0.0.255.255
 !
ip access-list standard ACL-TWO
 permit 192.168.1.1 0.0.0.0
 permit 172.31.0.0 0.0.255.255
!
route-map EXAMPLE permit 10
 match ip address ACL-ONE
 set metric 20
 continue
!
route-map EXAMPLE permit 20
 match ip address ACL-TWO
 set ip next-hop 10.12.1.1
!
route-map EXAMPLE permit 30
 set ip next-hop 10.13.1.3
```

Note

The **continue** command is not commonly used because it adds complexity when troubleshooting route maps.

BGP ROUTE FILTERING AND MANIPULATION

Route filtering is a method of selectively identifying routes that are advertised or received from neighbor routers. Route filtering may be used to manipulate traffic flows, reduce memory utilization, or improve security. For example, it is common for ISPs to deploy route filters on BGP peerings to customers. Ensuring that only the customer routes are allowed over the peering link prevents the customer from accidentally becoming a transit AS on the Internet.

Figure 12-9 shows the complete BGP route processing logic. Notice that the routing policies occur on inbound route receipt and outbound route advertisement.

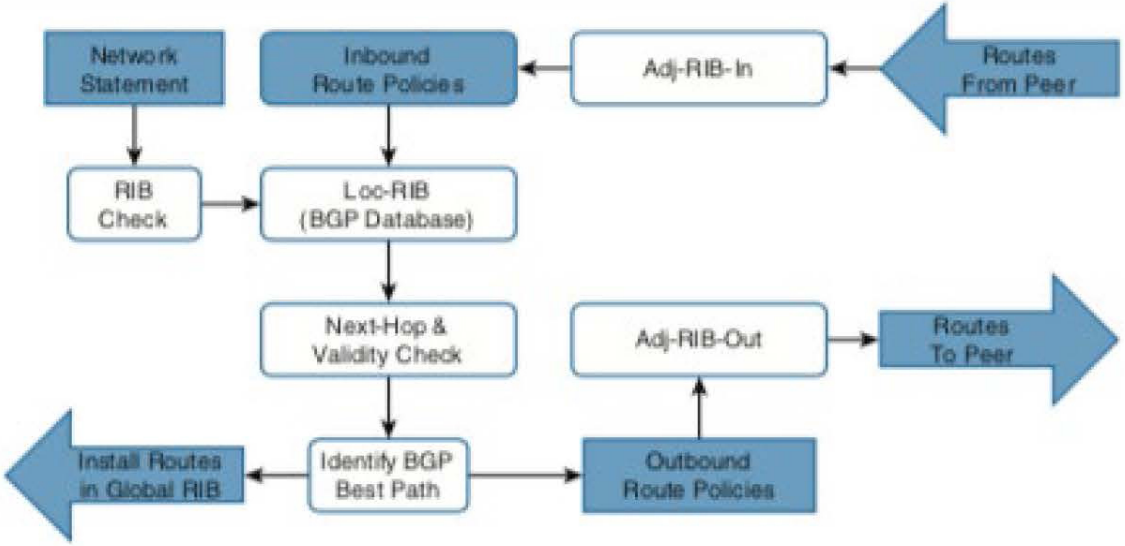


Figure 12-9 BGP Route Policy Processing

IOS XE provides four methods of filtering routes inbound or outbound for a specific BGP peer. These methods can be used individually or simultaneously with other methods:

- **Distribute list:** A distribute list involves the filtering of network prefixes based on a standard or extended ACL. An implicit deny is associated with any prefix that is not permitted.

- **Prefix list:** A list of prefix-matching specifications permit or deny network prefixes in a top-down fashion, similar to an ACL. An implicit deny is associated with any prefix that is not permitted.

- **AS path ACL/filtering:** A list of regex commands allow for the permit or deny of a network prefix based on the current AS path values. An implicit deny is associated with any prefix that is not permitted.

- **Route maps:** Route maps provide a method of conditional matching on a variety of prefix attributes and taking a variety of actions. Actions could be a simple permit or deny; or could include the modification of BGP path attributes. An implicit deny is associated with any prefix that is not permitted.

Note

A BGP neighbor cannot use a distribute list and prefix list at the same time for receiving or advertising routes.

The following sections explain each of these filtering techniques in more detail. Imagine a simple scenario with R1 (AS 65100) that has a single eBGP peering with R2 (AS 65200), which then may peer with other autonomous systems (such as AS 65300). The relevant portion of the topology is that R1 peers with R2 and focuses on R1’s BGP table, as shown in Example 12-8, with an emphasis on the network prefix and the AS path.

Example 12-8 Reference BGP Table

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.3.3.0/24      10.12.1.2               33             0 65
 *   10.12.1.0/24     10.12.1.2               22             0 65
```

```
    *>                    0.0.0.0                  0         32768 ?
    *>  10.23.1.0/24      10.12.1.2              333             0 65
    *>  100.64.2.0/25     10.12.1.2               22             0 65
    *>  100.64.2.192/26   10.12.1.2               22             0 65
    *>  100.64.3.0/25     10.12.1.2               22             0 65
    *>  192.168.1.1/32    0.0.0.0                  0         32768 ?
    *>  192.168.2.2/32    10.12.1.2               22             0 65
    *>  192.168.3.3/32    10.12.1.2             3333             0 65
```

Key Topic

Distribute List Filtering

Distribute lists allow the filtering of network prefixes on a neighbor-by-neighbor basis, using standard or extended ACLs. Configuring a distribute list requires using the BGP address-family configuration command **neighbor** *ip-address* **distribute-list** {*acl-number* | *acl-name*} {**in**|**out**}. Remember that extended ACLs for BGP use the source fields to match the network portion and the destination fields to match against the network mask.

Example 12-9 provides R1's BGP configuration, which demonstrates filtering with distribute lists. The configuration uses an extended ACL named ACL-ALLOW that contains two entries. The first entry allows for any network that starts in the 192.168.0.0 to 192.168.255.255 range with any length of network.

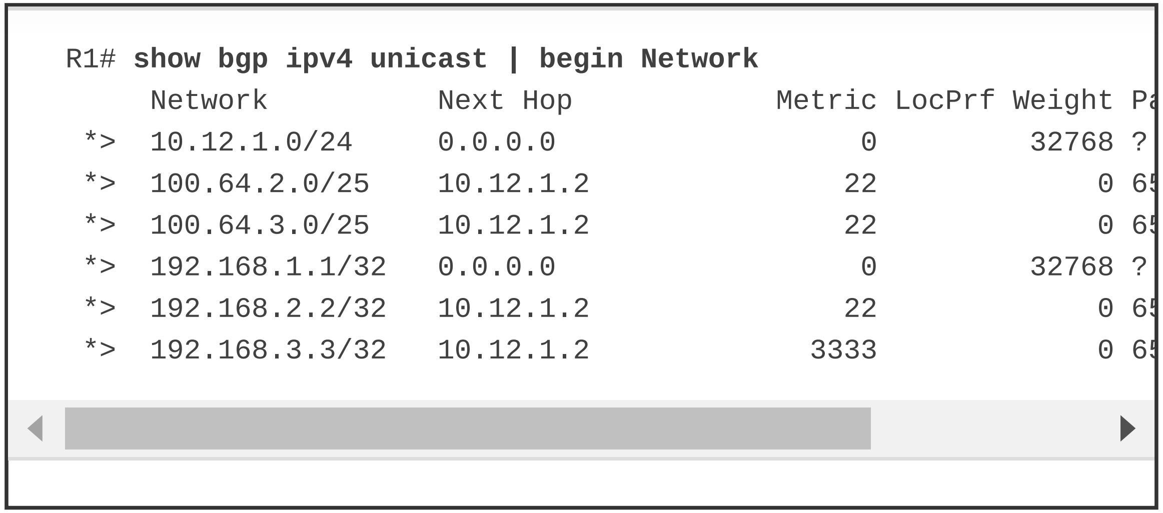
The second entry allows for networks that contain 100.64.x.0 pattern with a prefix length of /26 to demonstrate the wildcard abilities of an extended ACL with BGP. The distribute list is then associated with R2’s BGP session.

Example 12-9 BGP Distribute List Configuration

```
R1
ip access-list extended ACL-ALLOW
 permit ip 192.168.0.0 0.0.255.255 host 255.255.255.255
 permit ip 100.64.0.0 0.0.255.0 host 255.255.255.128
!
router bgp 65100
 address-family ipv4
  neighbor 10.12.1.2 distribute-list ACL-ALLOW in
```

Example 12-10 displays the routing table of R1. Two local routes are injected into the BGP table by R1 (10.12.1.0/24 and 192.168.1.1/32). The two loopback networks from R2 (AS 65200) and R3 (AS 65300) are allowed because they are within the first ACL-ALLOW entry, and two of the networks in the 100.64.x.0 pattern (100.64.2.0/25 and 100.64.3.0/25) are accepted. The 100.64.2.192/26 network is rejected because the prefix length does not match the second ACL-ALLOW entry. Example 12-8 can be referenced to identify the routes before the BGP distribute list was applied.

Example 12-10 Viewing Routes Filtered by BGP Distribute List



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     0.0.0.0                  0         32768 ?
 *>  100.64.2.0/25    10.12.1.2               22             0 65
 *>  100.64.3.0/25    10.12.1.2               22             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2               22             0 65
 *>  192.168.3.3/32   10.12.1.2             3333             0 65
```

Prefix List Filtering

Prefix lists allow the filtering of network prefixes on a neighbor-by-neighbor basis, using a prefix list. Configuring a prefix list involves using the BGP address family configuration command **neighbor** *ip-address* **prefix-list** *prefix-list-name* {**in** | **out**}.

To demonstrate the use of a prefix list, we can use the same initial BGP table from Example 12-8 and filter it to allow only routes within the RFC 1918 space. The same prefix list from Example 12-1 is used and will be applied on R1's peering to R2 (AS 65200). Example 12-11 shows the configuration of the prefix list and application to R2.

Example 12-11 Prefix List Filtering Configuration

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip prefix-list RFC1918 seq 5 permit 192.168.0.0/13 ge
R1(config)# ip prefix-list RFC1918 seq 10 deny 0.0.0.0/0 ge 32
R1(config)# ip prefix-list RFC1918 seq 15 permit 10.0.0.0/7 ge 8
R1(config)# ip prefix-list RFC1918 seq 20 permit 172.16.0.0/11 ge
R1(config)# ip prefix-list RFC1918 seq 25 permit 192.168.0.0/15 g
R1(config)# router bgp 65100
R1(config-router)# address-family ipv4 unicast
R1(config-router-af)# neighbor 10.12.1.2 prefix-list RFC1918 in
```

Now that the prefix list has been applied, the BGP table can be examined on R1, as shown in Example 12-12. Notice that the 100.64.2.0/25, 100.64.2.192/26, and 100.64.3.0/25 networks were filtered as they did not fall within the prefix list matching criteria. Example 12-8 can be referenced to identify the routes before the BGP prefix list was applied.

Example 12-12 Verification of Filtering with a BGP Prefix List

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.3.3.0/24      10.12.1.2               33             0 65
 *   10.12.1.0/24     10.12.1.2               22             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2              333             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2                22             0 65
 *>  192.168.3.3/32   10.12.1.2              3333             0 65
```

AS Path ACL Filtering

Selecting routes from a BGP neighbor by using the AS path requires the definition of an *AS path access control list (AS path ACL)*. Regular expressions, introduced earlier in this chapter, are a component of AS_Path filtering.

Example 12-13 shows the routes that R2 (AS 65200) is advertising toward R1 (AS 65100).

Example 12-13 AS Path Access List Configuration

```
R2# show bgp ipv4 unicast neighbors 10.12.1.1 advertised-routes
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.3.3.0/24      10.23.1.3               33             0 65
 *>  10.12.1.0/24     0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     0.0.0.0                  0         32768 ?
 *>  100.64.2.0/25    0.0.0.0                  0         32768 ?
 *>  100.64.2.192/26  0.0.0.0                  0         32768 ?
 *>  100.64.3.0/25    10.23.1.3                3             0 65
 *>  192.168.2.2/32   0.0.0.0                  0         32768 ?
 *>  192.168.3.3/32   10.23.1.3              333             0 65

Total number of prefixes 8
```

R2 is advertising the routes learned from R3 (AS 65300) to R1. In essence, R2 provides transit connectivity between the autonomous systems. If this were an Internet connection and R2 were an enterprise, it would not want to advertise routes learned from other ASs. Using an AS path access list to restrict the advertisement of only AS 65200 routes is recommended.

Key Topic

Processing is peformed in a sequential top-down order, and the first qualifying match processes against the appropriate **permit** or **deny** action. An implicit deny exists at the end of the AS path ACL. IOS supports up to 500 AS path ACLs and uses the command **ip as-path access-list** *acl-number* {**deny** | **permit**} *regex-query* for creating an AS path ACL. The ACL is then applied with the command **neighbor** *ip-address* **filter-list** *acl-number* {**in**|**out**}.

Example 12-14 shows the configuration on R2 using an AS path ACL to restrict traffic to only locally originated traffic, using the regex pattern ^$ (refer to Table 12-5). To ensure completeness, the AS path ACL is applied on all eBGP neighborships.

Example 12-14 AS Path Access List Configuration

```
R2
ip as-path access-list 1 permit ^$
```

```
!
router bgp 65200
 address-family ipv4 unicast
  neighbor 10.12.1.1 filter-list 1 out
  neighbor 10.23.1.3 filter-list 1 out
```

Now that the AS path ACL has been applied, the advertised routes can be checked again. Example 12-15 displays the routes being advertised to R1. Notice that all the routes do not have an AS path confirming that only locally originating routes are being advertised externally. Example 12-13 can be referenced to identify the routes before the BGP AS path ACL was applied.

Example 12-15 Verification of Local Route Advertisements with an AS Path ACL

```
R2# show bgp ipv4 unicast neighbors 10.12.1.1 advertised-routes
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.12.1.0/24     0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     0.0.0.0                  0         32768 ?
 *>  100.64.2.0/25    0.0.0.0                  0         32768 ?
 *>  100.64.2.192/26  0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   0.0.0.0                  0         32768 ?

Total number of prefixes 5
```

Key Topic

Route Maps

As explained earlier, route maps provide additional functionality over pure filtering. Route maps provide a method to manipulate BGP path attributes as well. Route maps are applied on a BGP neighbor basis for routes that are advertised or received. A different route map can be used for each direction. The route map is associated with the BGP neighbor with the command **neighbor** *ip-address* **route-map** *route-map-name* {**in**|**out**} under the specific address family.

Example 12-16 shows the BGP routing table of R1, which is used here to demonstrate the power of a route map.

Example 12-16 BGP Table Before Applying a Route Map

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      10.12.1.2               33             0 65
 *   10.12.1.0/24     10.12.1.2               22             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2              333             0 65
 *>  100.64.2.0/25    10.12.1.2               22             0 65
 *>  100.64.2.192/26  10.12.1.2               22             0 65
 *>  100.64.3.0/25    10.12.1.2               22             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
```

```
   *>  192.168.2.2/32   10.12.1.2                22             0 65
   *>  192.168.3.3/32   10.12.1.2              3333             0 65
```

Route maps allow for multiple steps in processing as well. To demonstrate this concept, our route map will consist of four steps:

1. Deny any routes that are in the 192.168.0.0/16 network by using a prefix list.

2. Match any routes originating from AS 65200 that are within the 100.64.0.0/10 network range and set the BGP local preference to 222.

3. Match any routes originating from AS 65200 that did not match step 2 and set the BGP weight to 65200.

4. Permit all other routes to process.

Example 12-17 demonstrates R1’s configuration, where multiple prefix lists are referenced along with an AS path ACL.

Example 12-17 R1’s Route Map Configuration for Inbound AS 65200 Routes

```
R1
ip prefix-list FIRST-RFC1918 permit  192.168.0.0/15 ge 16
ip as-path access-list 1 permit _65200$
ip prefix-list SECOND-CGNAT permit 100.64.0.0/10 ge 11
!
route-map AS65200IN deny 10
```

```
 description Deny any RFC1918 networks via Prefix List Matching
 match ip address prefix-list FIRST-RFC1918
!
route-map AS65200IN permit 20
 description Change local preference for AS65200 originate route
 match ip address prefix-list SECOND-CGNAT
 match as-path 1
 set local-preference 222
!
route-map AS65200IN permit 30
 description Change the weight for AS65200 originate routes
 match as-path 1
 set weight 65200
!
route-map AS65200IN permit 40
 description Permit all other routes un-modified
!
router bgp 65100
 address-family ipv4 unicast
  neighbor 10.12.1.1 route-map AS65200IN in
```

Example 12-18 displays R1’s BGP routing table. The following actions have occurred:

- The 192.168.2.2/32 and 192.168.3.3/32 routes were discarded. The 192.168.1.1/32 route is a locally generated route.

- The 100.64.2.0/25 and 100.64.2.192/26 networks had the local preference modified to 222 because they originated from AS 65200 and are within the 100.64.0.0/10 network range.

- The 10.12.1.0/24 and 10.23.1.0/24 routes from R2 were assigned the locally significant BGP attribute weight 65200.

- All other routes were received and not modified.

Example 12-18 Verifying Changes from R1's Route Map to AS 65200

```
R1# show bgp ipv4 unicast | b Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      10.12.1.2               33             0 65
 r>  10.12.1.0/24     10.12.1.2               22         65200 65
 r                    0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2              333         65200 65
 *>  100.64.2.0/25    10.12.1.2               22    222      0 65
 *>  100.64.2.192/26  10.12.1.2               22    222      0 65
 *>  100.64.3.0/25    10.12.1.2               22             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
```

Note

It is considered a best practice to use a different route policy for inbound and outbound prefixes for each BGP neighbor.

Clearing BGP Connections

Depending on the change to the BGP route manipulation technique, a BGP session may need to be refreshed in order to take effect. BGP supports two methods of clearing a BGP session. The first method is a *hard reset*, which tears down the BGP session, removes BGP routes from the peer, and is the most disruptive. The second method is a *soft reset*, which invalidates the BGP cache and requests a full advertisement from its BGP peer.

Routers initiate a hard reset with the command **clear ip bgp** *ip-address* [**soft**] and a soft reset by using the optional **soft** keyword. All of a router's BGP sessions can be cleared by using an asterisk * in lieu of the peer's IP address.

When a BGP policy changes, the BGP table must be processed again so that the neighbors can be notified accordingly. Routes received by a BGP peer must be processed again. If the BGP session supports *route refresh* capability, the peer re-advertises (refreshes) the prefixes to the requesting router, allowing for the inbound policy to process using the new policy changes. The route refresh capability is negotiated for each address family when the session is established.

Performing a soft reset on sessions that support route refresh capability actually initiates a route refresh. Soft resets can be performed for a specific address family with the command **clear bgp** *afi safi* {*ip-address*|*} **soft** [**in** | **out**]. Soft resets reduce the number of routes that must be exchanged if multiple address families are configured with a single BGP peer. Changes to the outbound routing policies use the optional **out** keyword, and changes to inbound routing policies use the optional **in** keyword. You can use an * in lieu of specifying a peer's IP address to perform that action for all BGP peers.

Key Topic

BGP COMMUNITIES

BGP communities provide additional capability for tagging routes and for modifying BGP routing policy on upstream and downstream routers. BGP communities can be appended, removed, or modified selectively on each attribute as a route travels from router to router.

BGP communities are an optional transitive BGP attribute that can traverse from AS to AS. A BGP community is a 32-bit number that can be included with a route. A BGP community can be displayed as a full 16-bit number (0–4,294,967,295) or as two 16-bit numbers (0–65535):(0–65535), commonly referred to as *new format*.

Private BGP communities follow a particular convention where the first 16 bits represent the AS of the community origination, and the second 16 bits represent a pattern defined by the originating AS. A private BGP community pattern can vary from organization to organization, does not need to be registered, and can signify geographic locations for one AS while signifying a method of route advertisement in another AS. Some organizations publish their private BGP community patterns on websites such as http://www.onesc.net/communities/ (http://www.onesc.net/communities/).

In 2006, RFC 4360 expanded BGP communities' capabilities by providing an extended format. *Extended BGP communities* provide structure for various classes of information and are commonly used for VPN services. RFC 8092 provides support for communities larger than 32 bits (which are beyond the scope of this book).

Well-Known Communities

RFC 1997 defines a set of global communities (known as *well-known communities*) that use the community range 4,294,901,760 (0xFFFF0000) to 4,294,967,295 (0xFFFFFFFF). All routers that are capable of sending/receiving BGP communities must implement well-known communities. Following are three common well-known communities:

- **Internet:** This is a standardized community for identifying routes should be advertised on the Internet. In larger networks that deploy BGP into the core, advertised routes should be advertised to the Internet and should have this community set. This allows for the edge BGP routers to only allow the advertisement of BGP routes with the Internet community to the Internet. Filtering is not automatic but can be done with an outbound route map.

- **No_Advertise:** Routes with this community should not be advertised to any BGP peer (iBGP or eBGP).

- **No_Export:** When a route with this community is received, the route is not advertised to any eBGP peer. Routes with this community can be advertised to iBGP peers.

Key Topic

Enabling BGP Community Support

IOS and IOS XE routers do not advertise BGP communities to peers by default. Communities are enabled on a neighbor-by-neighbor basis with the BGP address family configuration command **neighbor** *ip-address* **send-community** [**standard** | **extended** | **both**] under the neighbor's address family configuration. If a keyword is not specified, standard communities are sent by default.

IOS XE nodes can display communities in new format, which is easier to read, with the global configuration command **ip bgp-community new-format**. Example 12-19 displays the BGP community in decimal format first, followed by the new format.

Example 12-19 BGP Community Formats

```
! Decimal Format
R3# show bgp 192.168.1.1
! Output omitted for brevity
BGP routing table entry for 192.168.1.1/32, version 6
```

```
Community: 6553602 6577023

! New-Format
R3# show bgp 192.168.1.1
! Output omitted for brevity
BGP routing table entry for 192.168.1.1/32, version 6
Community: 100:2 100:23423
```

Conditionally Matching BGP Communities

Conditionally matching BGP communities allows for selection of routes based on the BGP communities within the route’s path attributes so that selective processing can occur in route maps. Example 12-20 demonstrates the BGP table for R1, which has received multiple routes from R2 (AS 65200).

Example 12-20 BGP Routes from R2 (AS 65200)

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
 *   10.12.1.0/24     10.12.1.2               22             0 65
 *>                   0.0.0.0                  0         32768 ?
 *>  10.23.1.0/24     10.12.1.2              333             0 65
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2               22             0 65
 *>  192.168.3.3/32   10.12.1.2             3333             0 65
```

In this example, say that you want to conditionally match for a specific community. The entire BGP table can be displayed with the command **show bgp** *afi safi* **detail** and then you can manually select a route with a specific community. However, if the BGP community is known, all the routes can be displayed with the command **show bgp** *afi safi* **community** *community*, as shown in Example 12-21.

Example 12-21 Displaying the BGP Routes with a Specific Community

```
R1# show bgp ipv4 unicast community 333:333 | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.23.1.0/24     10.12.1.2              333             0 65
```

Example 12-22 displays the explicit path entry for the 10.23.1.0/24 network and all the BGP path attributes. Notice that two BGP communities (333:333 and 65300:333) are added to the path.

Example 12-22 Viewing BGP Path Attributes for the 10.23.1.0/24 Network

```
R1# show ip bgp 10.23.1.0/24
BGP routing table entry for 10.23.1.0/24, version 15
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 3
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
```

```
        Origin incomplete, metric 333, localpref 100, valid, exter
        Community: 333:333 65300:333
        rx pathid: 0, tx pathid: 0x0
```

Key Topic

Conditionally matching requires the creation of a community list that shares a similar structure to an ACL, can be standard or expanded, and can be referenced by number or name. Standard community lists are numbered 1 to 99 and match either well-known communities or a private community number (*as-number:16-bit-number*). Expanded community lists are numbered 100 to 500 and use regex patterns.

The configuration syntax for a community list is **ip community-list** {*1-500* | **standard** *list-name* | **expanded** *list-name*} {**permit** | **deny**} *community-pattern*. After defining the community list, the community list is referenced in the route map with the command **match community** *1-500*.

Note

When multiple communities are on the same **ip community list** statement, all communities for that statement must exist in the route. If only one out of many communities is required, you can use multiple **ip community list** statements.

Example 12-23 demonstrates the creation of a BGP community list that matches on the community 333:333. The BGP community list is then used in the first sequence of *route-map COMMUNITY-CHECK*, which denies any routes with that community. The second route map sequence allows for all other BGP routes and sets the BGP weight (locally significant) to 111. The route map is then applied on routes advertised from R2 toward R1.

Example 12-23 Conditionally Matching BGP Communities

```
R1
ip community-list 100 permit 333:333
!
route-map COMMUNITY-CHECK deny 10
 description Block Routes with Community 333:333 in it
 match community 100
route-map COMMUNITY-CHECK permit 20
 description Allow routes with either community in it
 set weight 111
!
router bgp 65100
```

```
   address-family ipv4 unicast
    neighbor 10.12.1.2 route-map COMMUNITY-CHECK in
```

Example 12-24 shows the BGP table after the route map has been applied to the neighbor. The 10.23.1.0/24 network prefix was discarded, and all the other routes learned from AS 65200 had the BGP weight set to 111.

Example 12-24 R1's BGP Table After Applying the Route Map

```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Pa
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
 *   10.12.1.0/24     10.12.1.2               22           111 65
 *>                   0.0.0.0                  0         32768 ?
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2               22           111 65
 *>  192.168.3.3/32   10.12.1.2             3333           111 65
```

Key Topic

Setting Private BGP Communities

A private BGP community is set in a route map with the command **set community** *bgp-community* [**additive**]. By default, when setting a community, any existing communities are over-written but can be preserved by using the optional **additive** keyword.

Example 12-25 shows the BGP table entries for the 10.23.1.0/24 network, which has the 333:333 and 65300:333 BGP communities. The 10.3.3.0/24 network has the 65300:300 community.

Example 12-25 Viewing the BGP Communities for Two Network Prefixes

```
R1# show bgp ipv4 unicast 10.23.1.0/24
! Output omitted for brevity
BGP routing table entry for 10.23.1.0/24, version 15
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 333, localpref 100, valid, exter
      Community: 333:333 65300:333

R1# show bgp ipv4 unicast 10.3.3.0/24
! Output omitted for brevity
BGP routing table entry for 10.3.3.0/24, version 12
  65200 65300 3003
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 33, localpref 100, valid, externa
      Community: 65300:300
```

Example 12-26 shows the configuration where the BGP community is set to the 10.23.1.0/24 network. The **additive** keyword is not used, so the previous community values 333:333 and 65300:333 are overwritten with the 10:23

community. The 10.3.3.0/24 network has the communities 3:0, 3:3, and 10:10 added to the existing communities. The route map is then associated to R2 (AS 65200)

Example 12-26 Setting Private BGP Community Configuration

```
ip prefix-list PREFIX10.23.1.0 seq 5 permit 10.23.1.0/24
ip prefix-list PREFIX10.3.3.0 seq 5 permit 10.3.3.0/24
!
route-map SET-COMMUNITY permit 10
 match ip address prefix-list PREFIX10.23.1.0
 set community 10:23
route-map SET-COMMUNITY permit 20
 match ip address prefix-list PREFIX10.3.3.0
 set community 3:0 3:3 10:10 additive
route-map SET-COMMUNITY permit 30
!
router bgp 65100
 address-family ipv4
  neighbor 10.12.1.2 route-map SET-COMMUNITY in
```

Now that the route map has been applied and the routes have been refreshed, the path attributes can be examined, as demonstrated in Example 12-27. As anticipated, the previous BGP communities were removed for the 10.23.1.0/24 network but were maintained for the 10.3.3.0/24 network.

Example 12-27 Verifying BGP Community Changes

```
R1# show bgp ipv4 unicast 10.23.1.0/24
! Output omitted for brevity
BGP routing table entry for 10.23.1.0/24, version 22
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 333, localpref 100, valid, extern
      Community: 10:23

R1# show bgp ipv4 unicast 10.3.3.0/24
BGP routing table entry for 10.3.3.0/24, version 20
  65200 65300 3003
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 33, localpref 100, valid, externa
      Community: 3:0 3:3 10:10 65300:300
```

UNDERSTANDING BGP PATH SELECTION

The BGP best-path selection algorithm influences how traffic enters or leaves an AS. Some router configurations modify the BGP attributes to influence inbound traffic, outbound traffic, or inbound and outbound traffic, depending on the network design requirements. A lot of network engineers do not understand BGP best-path selection, which can often result in suboptimal routing. This section explains the logic used by a router that uses BGP when forwarding packets.

Key Topic

Routing Path Selection Using Longest Match

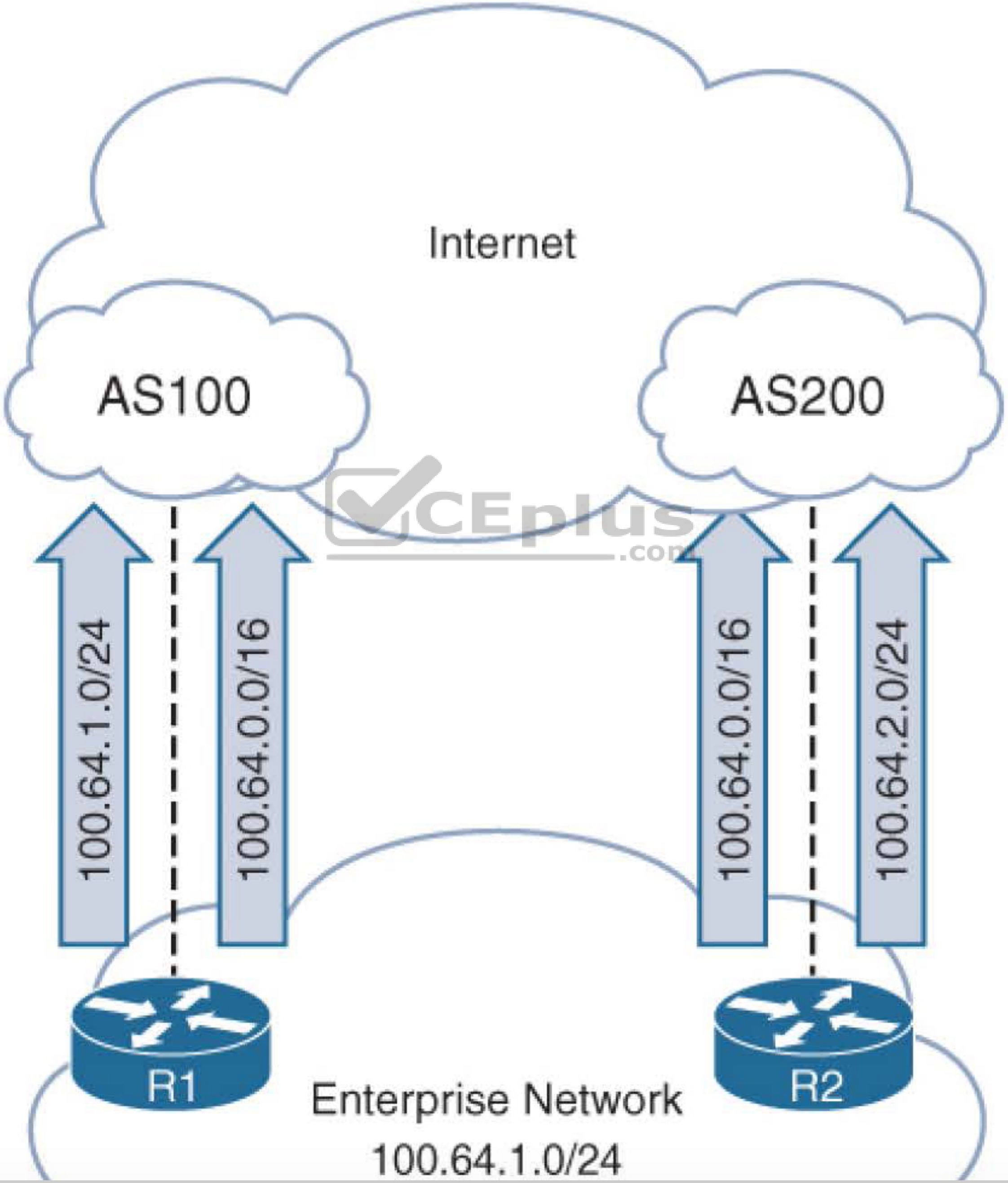
Routers always select the path a packet should take by examining the prefix length of a network entry. The path selected for a packet is chosen based on the prefix length, where the longest prefix length is always preferred. For example, /28 is preferred over /26, and /26 is preferred over /24.

This logic can be used to influence path selection in BGP. Assume that an organization owns the 100.64.0.0/16 network range but only needs to advertise two subnets (100.64.1.0/24 and 100.64.2.0/24). It could advertise both prefixes (100.64.1.0/24 and 100.64.2.0/24) from all its routers, but how can it distribute the load for each subnet if all traffic comes in on one router (such as R1)?

The organization could modify various BGP path attributes (PAs) that are advertised externally, but an SP could have a BGP routing policy that ignores those path attributes, resulting in random receipt of network traffic.

A more elegant way that guarantees that paths are selected deterministically outside the organization is to advertise a summary prefix (100.64.0.0/16) out both routers. Then the organization can advertise a longer matching prefix out the router that should receive network traffic for that prefix. Figure 12-10 shows the concept, with R1 advertising the 100.64.1.0/24 prefix, R2 advertising the

100.64.2.0/24 prefix, and both routers advertising the 100.64.0.0/16 summary network prefix.



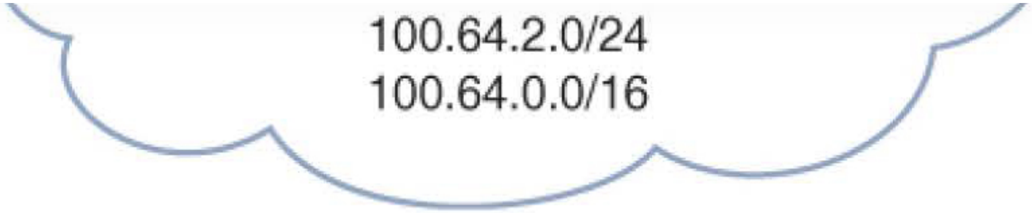


Figure 12-10 BGP Path Selection Using Longest Match

Regardless of an SP's routing policy, the more specific prefixes are advertised out only one router. Redundancy is provided by advertising the summary address. If R1 crashes, devices use R2's route advertisement of 100.64.0.016 to reach the 100.64.1.0/24 network.

Note

Ensure that the network summaries that are being advertised from your organization are within only your network range. In addition, service providers typically do not accept IPv4 routes longer than /24 (for example, /25 or /26) or IPv6 routes longer than /48. Routes are restricted to control the size of the Internet routing table.

BGP Best Path Overview

In BGP, route advertisements consist of Network Layer Reachability Information (NLRI) and path attributes (PAs). The NLRI consists of the network prefix and prefix length, and the BGP attributes such as AS_Path, origin, and so on are stored in the PAs. A BGP route may contain multiple paths to the same destination network. Every path's attributes impact the desirability of the route when a router selects the best path. A BGP router advertises only the best path to the neighboring routers.

Inside the BGP Loc-RIB table, all the routes and their path attributes are maintained with the best path calculated. The best path is then installed in the RIB of the router. If the best path is no longer available, the router can use the existing paths to quickly identify a new best path. BGP recalculates the best path for a prefix upon four possible events:

- BGP next-hop reachability change

- Failure of an interface connected to an eBGP peer

- Redistribution change

- Reception of new or removed paths for a route

BGP automatically installs the first received path as the best path. When additional paths are received for the same network prefix length, the newer paths are compared against the current best path. If there is a tie, processing continues until a best-path winner is identified.

Key Topic

The BGP best-path algorithm uses the following attributes, in the order shown, for the best-path selection:

1. Weight

2. Local preference

3. Local originated (network statement, redistribution, or aggregation)

4. AIGP

5. Shortest AS_Path

6. Origin type

7. Lowest MED

8. eBGP over iBGP

9. Lowest IGP next hop

10. If both paths are external (eBGP), prefer the first (oldest)

11. Prefer the route that comes from the BGP peer with the lower RID

12. Prefer the route with the minimum cluster list length

13. Prefer the path that comes from the lowest neighbor address

The BGP routing policy can vary from organization to organization, based on the manipulation of the BGP PAs. Because some PAs are transitive and carry from one AS to another AS, those changes could impact downstream routing for other SPs, too. Other PAs are non-transitive and only influence the routing policy within the organization. Network prefixes are conditionally matched on a variety of factors, such as AS_Path length, specific ASN, BGP communities, or other attributes.

The best-path algorithm is explained in the following sections.

Weight

BGP weight is a Cisco-defined attribute and the first step for selecting the BGP best path. Weight is a 16-bit value (0 to 65,535) assigned locally on the router; it is not advertised to other routers. The path with the higher weight is preferred. Weight can be set for specific routes with an inbound route map or for all routes learned from a specific neighbor. Weight is not advertised to peers and only influences outbound traffic from a router or an AS. Because it is the first step in the best-path algorithm, it should be used when other attributes should not influence the best path for a specific network.

Example 12-28 displays the BGP table for the 172.16.1.0/24 network prefix on R2. On the third line of the output, the router indicates that two paths exist, and the first path is the best path. By examining the output of each path, the path learned through AS 65300 has a weight of 123. The path through AS 65100 does not have the weight, which equates to a value of 0; therefore, the route through AS 65300 is the best path.

Example 12-28 An Example of a BGP Best-Path Choice Based on Weight

```
R2# show bgp ipv4 unicast 172.16.1.0/24
BGP routing table entry for 172.16.1.0/24, version 3
Paths: (2 available, best #1, table default)
  Refresh Epoch 2
  65300
    10.23.1.3 from 10.23.1.3 (192.18.3.3)
      Origin IGP, metric 0, localpref 100, weight 123, valid, ext
  Refresh Epoch 2
  65100
    10.12.1.1 from 10.12.1.1 (192.168.1.1)
      Origin IGP, metric 0, localpref 100, valid, external
```

Local Preference

Local preference (LOCAL_PREF) is a well-known discretionary path attribute and is included with path advertisements throughout an AS. The local preference attribute is a 32-bit value (0 to 4,294,967,295) that indicates the preference for exiting the AS to the destination network. The local preference is not advertised between eBGP peers and is typically used to influence the next-hop address for outbound traffic (that is, leaving an autonomous system). Local preference can be set for specific routes by using a route map or for all routes received from a specific neighbor.

A higher value is preferred over a lower value. If an edge BGP router does not define the local preference upon receipt of a prefix, the default local preference value of 100 is used during best-path calculation, and it is included in advertisements to other iBGP peers. Modifying the local preference can influence the path selection on other iBGP peers without impacting eBGP peers because local preference is not advertised outside the autonomous system.

Example 12-29 shows the BGP table for the 172.16.1.0/24 network prefix on R2. On the third line of the output, the router indicates that two paths exist, and the first path is the best path. The BGP weight does not exist, so then the local preference is used. The path learned through AS 65300 is the best path because it has a local preference of 333, while the path through AS 65200 has a local preference of 111.

Example 12-29 An Example of a BGP Best-Path Choice Based on Local Preference

```
R2# show bgp ipv4 unicast 172.16.1.0/24
BGP routing table entry for 172.16.1.0/24, version 4
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 4
  65300
    10.23.1.3 from 10.23.1.3 (192.18.3.3)
```

```
      Origin IGP, metric 0, localpref 333, valid, external, best
  Refresh Epoch 4
  65100
    10.12.1.1 from 10.12.1.1 (192.168.1.1)
      Origin IGP, metric 0, localpref 111, valid, external
```

Locally Originated via Network or Aggregate Advertisement

The third decision point in the best-path algorithm is to determine whether the route originated locally. Preference is given in the following order:

- Routes that were advertised locally
- Networks that have been aggregated locally
- Routes received by BGP peers

Accumulated Interior Gateway Protocol

Accumulated Interior Gateway Protocol (AIGP) is an optional nontransitive path attribute that is included with advertisements throughout an AS. IGPs typically use the lowest-path metric to identify the shortest path to a destination but cannot provide the scalability of BGP. BGP uses an AS to identify a single domain of control for a routing policy. BGP does not use path metric due to scalability issues combined with the notion that each AS may use a different routing policy to calculate metrics.

AIGP provides the ability for BGP to maintain and calculate a conceptual path metric in environments that use multiple ASs with unique IGP routing domains in each AS. The ability for BGP to make routing decisions based on a path metric is a viable option because all the ASs are under the control of a single domain, with consistent routing policies for BGP and IGPs.

In Figure 12-11, AS 100, AS 200, and AS 300 are all under the control of the same service provider. AIGP has been enabled on the BGP sessions between all the routers, and the IGPs are redistributed into BGP. The AIGP metric is advertised between AS 100, AS 200, and AS 300, allowing BGP to use the AIGP metric for best-path calculations between the autonomous systems.

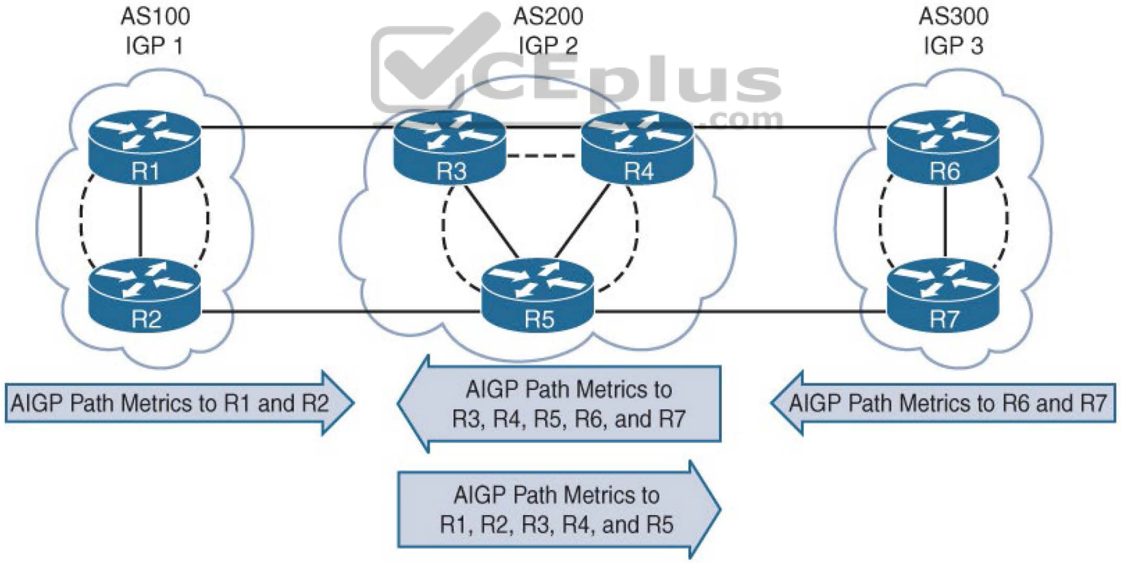

AS100
IGP 1
AS200
IGP 2
AS300
IGP 3
R1
R2
R3
R4
R5
R6
R7
AIGP Path Metrics to R1 and R2
AIGP Path Metrics to R3, R4, R5, R6, and R7
AIGP Path Metrics to R6 and R7
AIGP Path Metrics to R1, R2, R3, R4, and R5

Figure 12-11 AIGP Path Attribute Exchange Between Autonomous Systems

The following guidelines apply to AIGP metrics:

- A path with an AIGP metric is preferred to a path without an AIGP metric.

- If the next-hop address requires a recursive lookup, the AIGP path needs to calculate a derived metric to include the distance to the next-hop address. This ensures that the cost to the BGP edge router is included. The formula is

Derived AIGP metric = (Original AIGP metric + Next-hop AIGRP metric)

- If multiple AIGP paths exist and one next-hop address contains an AIGP metric and the other does not, the non-AIGP path is not used.

- The next-hop AIGP metric is recursively added if multiple lookups are performed.

- AIGP paths are compared based on the derived AIGP metric (with recursive next hops) or the actual AIGP metric (non-recursive next hop). The path with the lower AIGP metric is preferred.

- When a router R2 advertises an AIGP-enabled path that was learned from R1, if the next-hop address changes to an R2 address, R2 increments the AIGP metric to reflect the distance (the IGP path metric) between R1 and R2.

Shortest AS Path

The next decision factor for the BGP best-path algorithm is the AS path length. The path length typically correlates to the AS hop count. A shorter AS path is preferred over a longer AS path.

Prepending ASNs to the AS path makes it longer, thereby making that path less desirable compared to other paths. Typically, the AS path is prepended with the network owner’s ASN.

In general, a path that has had the AS path prepended is not selected as the BGP best path because the AS path is longer than the non-prepended path advertisement. Inbound traffic is influenced by prepending AS path length in advertisements to other ASs, and outbound traffic is influenced by prepending advertisements received from other ASs.

Example 12-30 shows the BGP table for the 172.16.1.0/24 network prefix on R2. The second route learned through AS 65100 is the best path. There is not a weight set on either path, and the local preference is identical. The second path has an AS path length of 1, while the first path has an AS path length of 2 (65300 and 65300).

Example 12-30 An Example of a BGP Best-Path Choice Based on AS Path Length

```
R2# show bgp ipv4 unicast 172.16.1.0/24
BGP routing table entry for 172.16.1.0/24, version 6
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 8
  65300 65300
    10.23.1.3 from 10.23.1.3 (192.18.3.3)
      Origin IGP, metric 0, localpref 100, valid, external
  Refresh Epoch 8
```

```
   65100
     10.12.1.1 from 10.12.1.1 (192.168.1.1)
       Origin IGP, metric 0, localpref 100, valid, external, best
```

Note

The ASNs are repeated in the first entry, which indicates that AS 65300 prepended its BGP advertisement to steer network traffic.

Note

Peering with different Internet providers provides optimal routing to most companies because one SP may be one AS path hop away (or provide connectivity to other tier 2/3 SPs), while a different SP may have a shorter AS path to other customers.

Origin Type

The next BGP best-path decision factor is the well-known mandatory BGP attribute named *origin*. By default, networks that are advertised through the **network** statement are set with the IGP or i origin, and redistributed networks are assigned the Incomplete or ? origin attribute. The origin preference order is

1. IGP origin (most)

2. EGP origin

3. Incomplete origin (least)

Example 12-31 shows the BGP table for the 172.16.1.0/24 network prefix on R2. The second path learned through AS 65100 is the best path because it has an origin of IGP, while first path has an origin of incomplete, which is the least preferred.

Example 12-31 An Example of a BGP Best-Path Choice Based on Origin Type

```
R2# show bgp ipv4 unicast 172.16.1.0/24
BGP routing table entry for 172.16.1.0/24, version 6
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 10
  65300
    10.23.1.3 from 10.23.1.3 (192.18.3.3)
      Origin incomplete, metric 0, localpref 100, valid, external
  Refresh Epoch 10
  65100
```

```
      10.12.1.1 from 10.12.1.1 (192.168.1.1)
        Origin IGP, metric 0, localpref 100, valid, external, best
```

Multi-Exit Discriminator

The next BGP best-path decision factor is the non-transitive BGP attribute named *multiple-exit discriminator (MED)*. MED uses a 32-bit value (0 to 4,294,967,295) called a *metric*. BGP sets the MED automatically to the IGP path metric during network advertisement or redistribution. If the MED is received from an eBGP session, it can be advertised to other iBGP peers, but it should not be sent outside the AS that received it. MED's purpose is to influence traffic flows inbound from a different AS. A lower MED is preferred over a higher MED.

Note

For MED to be an effective decision factor, the paths being decided upon must come from the same ASN.

RFC 4451 guidelines state that a prefix without a MED value should be given priority and, in essence, should be compared with a value of 0. Some organizations require that a MED be set to a specific value for all the prefixes and declare that paths without the MED should be treated as the least preferred. By default, if the MED is missing from a prefix learned from an eBGP peer, devices use a MED of 0 for the best-path calculation. IOS routers advertise a MED of 0 to iBGP peers.

Example 12-32 shows the BGP table for the 172.16.1.0/24 network prefix on R2. Notice that R2 is peering only with AS 65300 for MED to be eligible for the best-path selection process. The first path has a MED of 0, and the second path has a MED of 33. The first path is preferred as the MED is lower.

Example 12-32 An Example of a BGP Best-Path Choice Based on MED

```
R2# show bgp ipv4 unicast 172.16.1.0
BGP routing table entry for 172.16.1.0/24, version 9
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     2
  Refresh Epoch 4
  65300
    10.12.1.1 from 10.12.1.1 (192.168.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
  Refresh Epoch 14
  65300
    10.23.1.3 from 10.23.1.3 (192.18.3.3)
      Origin IGP, metric 33, localpref 100, valid, external
```

Note

It is possible for the SP forgot to advertise the MED from both peers and configure only one. This might have unintended consequences and can be easily fixed.

eBGP over iBGP

The next BGP best-path decision factor is whether the route comes from an iBGP, eBGP, or confederation member AS (sub-AS) peering. The best-path selection order is

1. eBGP peers (most desirable)

2. Confederation member AS peers

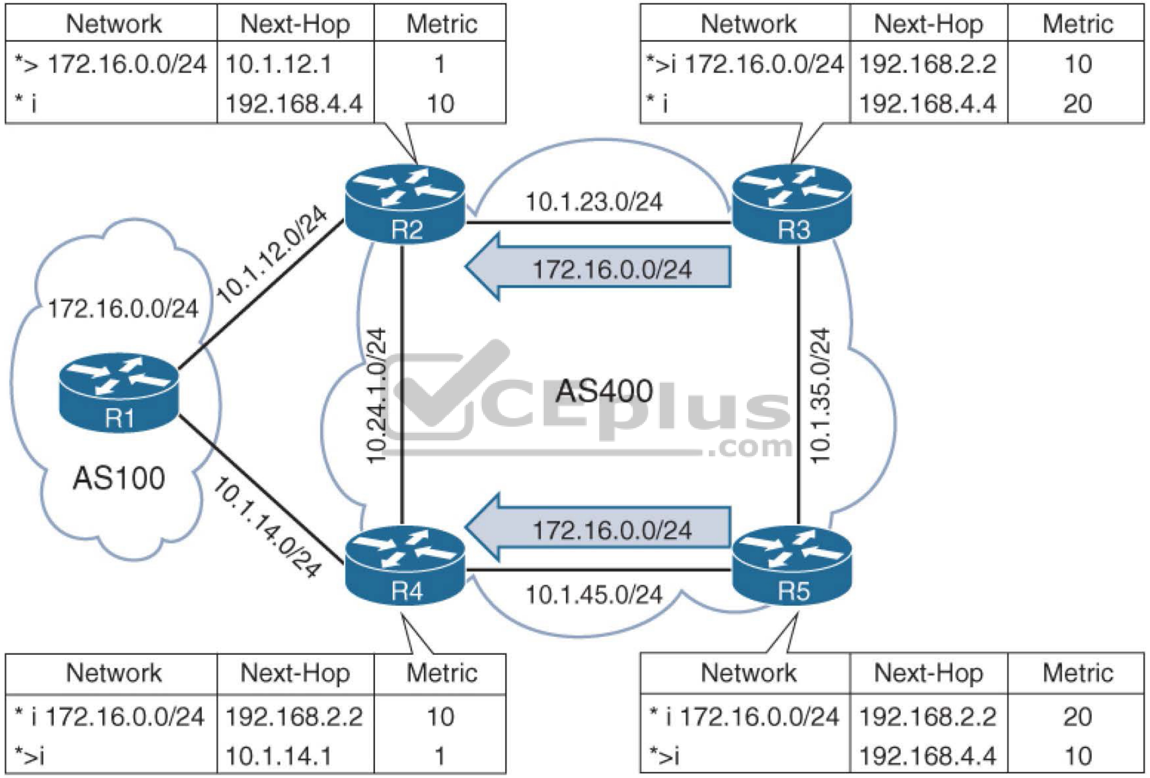
3. iBGP peers (least desirable)

Note

BGP confederations are beyond the scope of the CCNP and CCIE Enterprise Core ENCOR 300-401 exam and are not discussed in this book.

Lowest IGP Metric

The next decision step is to use the lowest IGP cost to the BGP next-hop address. Figure 12-12 illustrates a topology where R2, R3, R4, and R5 are in AS 400. AS 400 peers in a full mesh and establishes BGP sessions using Loopback 0 interfaces. R1 advertises the 172.16.0.0/24 network prefix to R2 and R4.



Network	Next-Hop	Metric
*> 172.16.0.0/24	10.1.12.1	1
* i	192.168.4.4	10

Network	Next-Hop	Metric
*>i 172.16.0.0/24	192.168.2.2	10
* i	192.168.4.4	20

Network	Next-Hop	Metric
* i 172.16.0.0/24	192.168.2.2	10
*>i	10.1.14.1	1

Network	Next-Hop	Metric
* i 172.16.0.0/24	192.168.2.2	20
*>i	192.168.4.4	10

Figure 12-12 Lowest IGP Metric Topology

R3 prefers the path from R2 compared to the iBGP path from R4 because the metric to reach the next-hop address is lower. R5 prefers the path from R4 compared to the iBGP path from R2 because the metric to reach the next-hop address is lower.

Prefer the Oldest eBGP Path

BGP can maintain large routing tables, and unstable sessions result in the BGP best-path calculation executing frequently. BGP maintains stability in a network by preferring the path from the oldest (established) BGP session.

The downfall of this technique is that it does not lead to a deterministic method of identifying the BGP best path from a design perspective.

Router ID

The next step for the BGP best-path algorithm is to select the best path using the lowest router ID of the advertising eBGP router. If the route was received by a route reflector, then the originator ID is substituted for the router ID.

Minimum Cluster List Length

The next step in the BGP best-path algorithm is to select the best path using the lowest cluster list length. The *cluster list* is a non-transitive BGP attribute that is appended (not overwritten) by a route reflector with its cluster ID. Route reflectors use the cluster ID attribute as a loop-prevention mechanism. The cluster ID is not advertised between ASs and is locally significant. In simplest terms, this step locates the path that has traveled the lowest number of iBGP advertisement hops.

Note

BGP route reflectors are beyond the scope of the CCNP and CCIE Enterprise Core ENCOR 300-401exam and are not discussed in this book.

Lowest Neighbor Address

The last step of the BGP best-path algorithm is to select the path that comes from the lowest BGP neighbor address. This step is limited to iBGP peerings because eBGP peerings used the oldest received path as the tie breaker.

Figure 12-13 demonstrates the concept of choosing the router with the lowest neighbor address. R1 is advertising the 172.16.0.0/24 network prefix to R2. R1 and R2 have established two BGP sessions using the 10.12.1.0/24 and 10.12.2.0/24 networks. R2 selects the path advertised from 10.12.1.1 as it is the lower IP address.

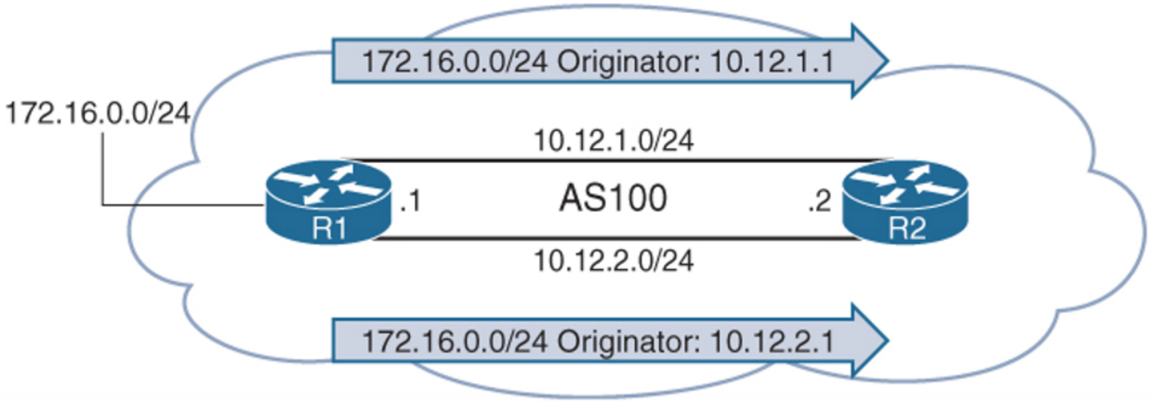

172.16.0.0/24 Originator: 10.12.1.1
172.16.0.0/24
10.12.1.0/24
R1
.1
AS100
.2
R2
10.12.2.0/24
172.16.0.0/24 Originator: 10.12.2.1

Figure 12-13 Lowest IP Address

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 12-9 lists these key topics and the page number on which each is found.

Key Topic

Table 12-9 Key Topics for Chapter 12

Key Topic Element	Description	Page
Section	Resiliency in service providers	
Section	Transit routing	
Section	Extended ACL IGP network selection	
Section	Extended ACL BGP network selection	
Paragraph	Prefix match specifications	
Paragraph	Prefix matching with length parameters	
Section	Prefix lists	
Section	Regular expressions	
List	Route map components	
Paragraph	Route map syntax and processing	
Section	Route map conditional matching	
Section	Route map matching with multiple conditions	
Section	Route map optional actions	
Section	BGP distribute list filtering	
Section	BGP prefix list filtering	
Paragraph	BGP AS path ACL	
Section	BGP route maps for neighbors	
Section	BGP communities	
Section	Enabling BGP community support	
Paragraph	BGP community list	
Section	Setting private BGP communities	
Section	Route path selection with longest match	
List	BGP best-path algorithm	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

AS path access control list (ACL)

BGP community

BGP multihoming

distribute list

prefix list

regular expression (regex)

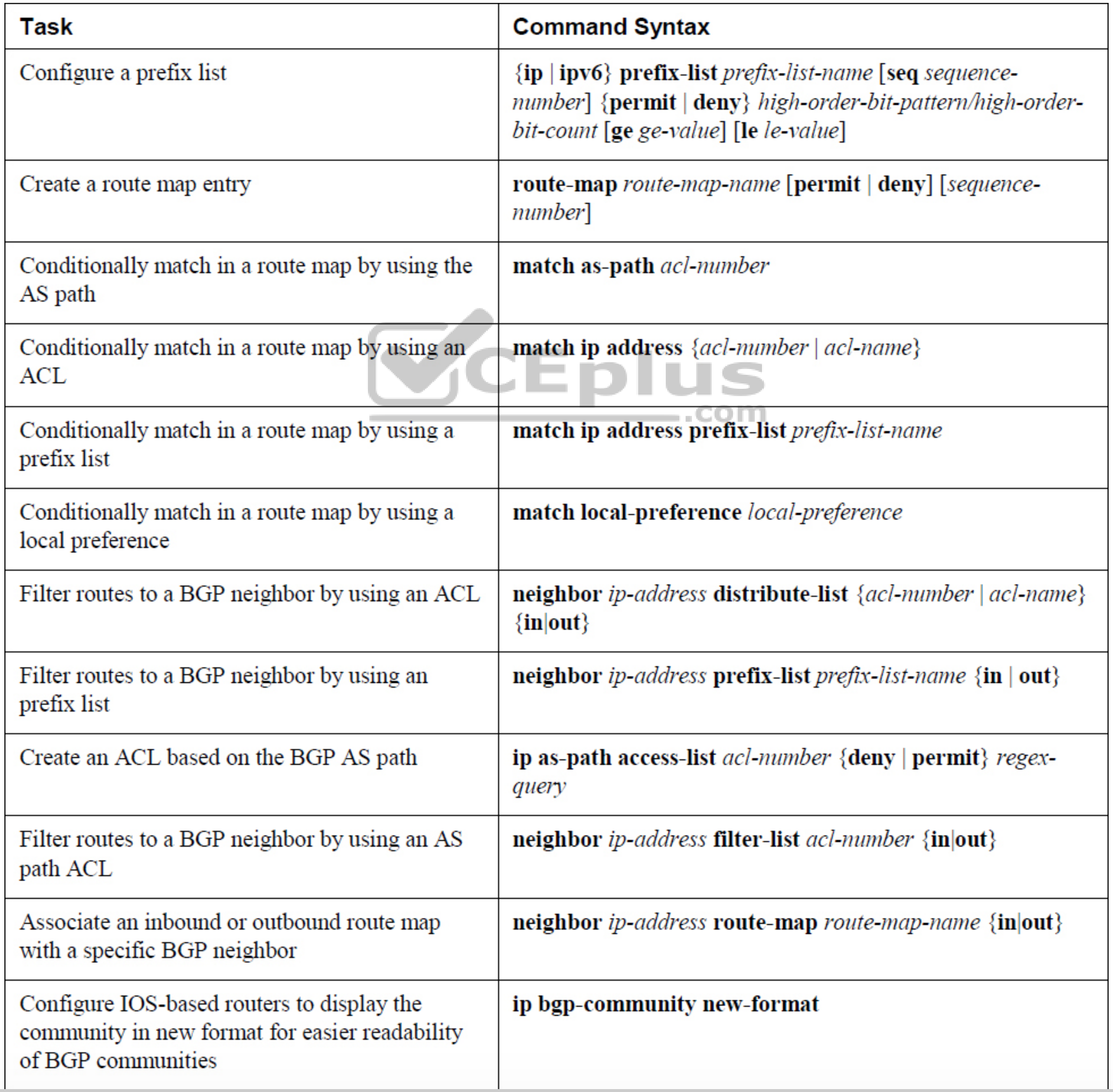
route map

transit routing

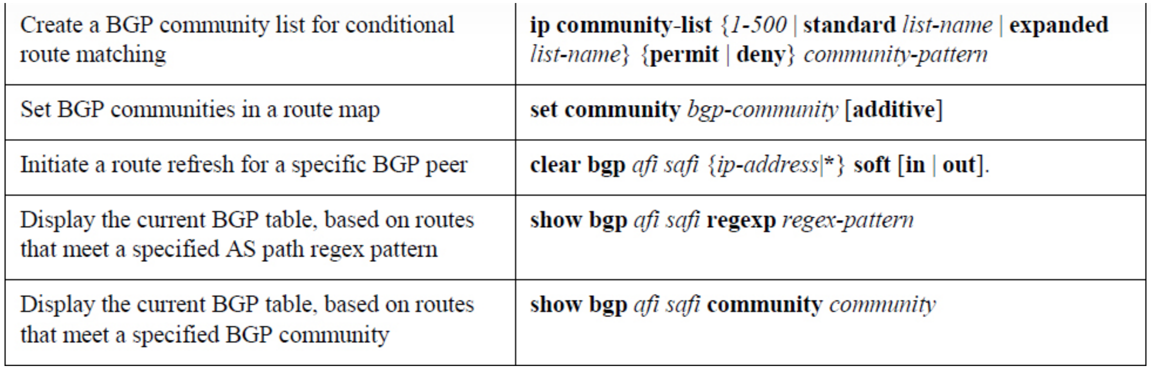
USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 12-10 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 12-10 Command Reference



Task	Command Syntax
Configure a prefix list	{**ip** | **ipv6**} **prefix-list** *prefix-list-name* [**seq** *sequence-number*] {**permit** | **deny**} *high-order-bit-pattern/high-order-bit-count* [**ge** *ge-value*] [**le** *le-value*]
Create a route map entry	**route-map** *route-map-name* [**permit** | **deny**] [*sequence-number*]
Conditionally match in a route map by using the AS path	**match as-path** *acl-number*
Conditionally match in a route map by using an ACL	**match ip address** {*acl-number* | *acl-name*}
Conditionally match in a route map by using a prefix list	**match ip address prefix-list** *prefix-list-name*
Conditionally match in a route map by using a local preference	**match local-preference** *local-preference*
Filter routes to a BGP neighbor by using an ACL	**neighbor** *ip-address* **distribute-list** {*acl-number* | *acl-name*} {**in**|**out**}
Filter routes to a BGP neighbor by using an prefix list	**neighbor** *ip-address* **prefix-list** *prefix-list-name* {**in** | **out**}
Create an ACL based on the BGP AS path	**ip as-path access-list** *acl-number* {**deny** | **permit**} *regex-query*
Filter routes to a BGP neighbor by using an AS path ACL	**neighbor** *ip-address* **filter-list** *acl-number* {**in**|**out**}
Associate an inbound or outbound route map with a specific BGP neighbor	**neighbor** *ip-address* **route-map** *route-map-name* {**in**|**out**}
Configure IOS-based routers to display the community in new format for easier readability of BGP communities	**ip bgp-community new-format**



Create a BGP community list for conditional route matching	**ip community-list** {*1-500* | **standard** *list-name* | **expanded** *list-name*} {**permit** | **deny**} *community-pattern*
Set BGP communities in a route map	**set community** *bgp-community* [**additive**]
Initiate a route refresh for a specific BGP peer	**clear bgp** *afi safi* {*ip-address*|*} **soft** [**in** | **out**].
Display the current BGP table, based on routes that meet a specified AS path regex pattern	**show bgp** *afi safi* **regexp** *regex-pattern*
Display the current BGP table, based on routes that meet a specified BGP community	**show bgp** *afi safi* **community** *community*

REFERENCES IN THIS CHAPTER

RFC 4360, *BGP Extended Communities Attribute*, by Yakov Rekhter, Dan Tappan, and Srihari R. Sangli, https://www.ietf.org/rfc/rfc4360.txt, February 2006.

RFC 8092, *BGP Large Communities Attribute*, by John Heasley, et. al, https://www.ietf.org/rfc/rfc2858.txt, February 2017.

Chapter 13. Multicast

This chapter covers the following subjects:

- **Multicast Fundamentals:** This section describes multicast concepts as well as the need for multicast.

- **Multicast Addressing:** This section describes the multicast address scopes used by multicast to operate at Layer 2 and Layer 3.

- **Internet Group Management Protocol:** This section explains how multicast receivers join multicast groups to start receiving multicast traffic using IGMPv2 or IGMPv3. It also describes how multicast flooding on Layer 2 switches is prevented using a feature called IGMP snooping.

- **Protocol Independent Multicast:** This section describes the concepts, operation, and features of PIM. PIM is the protocol used to route multicast traffic across network segments from a multicast source to a group of receivers.

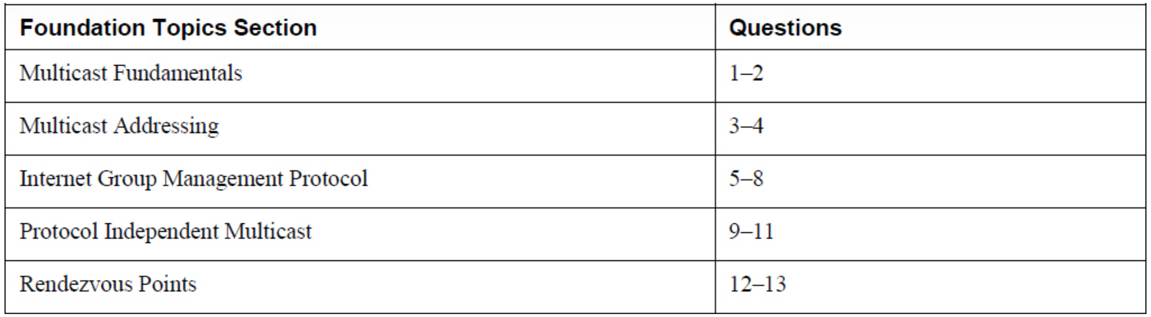
- **Rendezvous Points:** This section describes the purpose, function, and operation of rendezvous points in a multicast network.

Multicast is deployed on almost every type of network. It allows a source host to send data packets to a group of destination hosts (receivers) in an efficient manner that conserves bandwidth and system resources. This chapter describes the need for multicast as well as the fundamental protocols that are required to understand its operation, such as IGMP, PIM dense mode/sparse mode, and rendezvous points (RPs).

“DO I KNOW THIS ALREADY?” QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the “Exam Preparation Tasks” section. Table 13-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

Table 13-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping



Foundation Topics Section	Questions
Multicast Fundamentals	1–2
Multicast Addressing	3–4
Internet Group Management Protocol	5–8
Protocol Independent Multicast	9–11
Rendezvous Points	12–13

1. Which of the following transmission methods is multicast known for?

a. One-to-one

b. One-to-all

c. One-for-all

d. All-for-one

e. One-to-many

2. Which protocols are essential to multicast operation? (Choose two.)

a. Open Shortest Path First (OSPF)

b. Protocol Independent Multicast (PIM)

c. Internet Group Management Protocol (IGMP)

d. Auto-RP and BSR

3. Which of the following multicast address ranges match the administratively scoped block? (Choose two.)

a. 239.0.0.0 to 239.255.255.255

b. 232.0.0.0 to 232.255.255.255

c. 224.0.0.0 to 224.0.0.255

d. 239.0.0.0/8

e. 224.0.1.0/24

4. The first 24 bits of a multicast MAC address always start with ______.

a. 01:5E:00

b. 01:00:53

c. 01:00:5E

d. 01:05:E0

e. none of the above

5. What does a host need to do to start receiving multicast traffic?

a. Send an IGMP join

b. Send an unsolicited membership report

c. Send an unsolicited membership query

d. Send an unsolicited group specific query

6. What is the main difference between IGMPv2 and IGMPv3?

a. IGMPv3's max response time is 10 seconds by default.

b. IGMPv3 sends periodic IGMP membership queries.

c. IGMPv3 introduced a new IGMP membership report with source filtering support.

d. IGMPv3 can only work with SSM, while IGMPv2 can only work with PIM-SM/DM.

7. True or false: IGMPv3 was designed to work exclusively with SSM and is not backward compatible with PIM-SM.

a. True

b. False

8. How can you avoid flooding of multicast frames in a Layer 2 network?

a. Disable unknown multicast flooding

b. Enable multicast storm control

c. Enable IGMP snooping

d. Enable control plane policing

9. Which of the following best describe SPT and RPT? (Choose two.)

a. RPT is a source tree where the rendezvous point is the root of the tree.

b. SPT is a source tree where the source is the root of the tree.

c. RPT is a shared tree where the rendezvous point is the root of the tree.

d. SPT is a shared tree where the source is the root of the tree.

10. What does an LHR do after it receives an IGMP join from a receiver?

a. It sends a PIM register message toward the RP.

b. It sends a PIM join toward the RP.

c. It sends a PIM register message toward the source.

d. It sends a PIM join message toward the source.

11. What does an FHR do when an attached source becomes active and there are no interested receivers?

a. It unicasts register messages to the RP and stops after a register stop from the RP.

b. It unicasts encapsulated register messages to the RP and stops after a register stop from the RP.

c. It waits for the RP to send register message indicating that there are interested receivers.

d. It multicasts encapsulated register messages to the RP and stops after a register stop from the RP.

e. It unicasts encapsulated register messages to the RP until there are interested receivers.

12. Which of the following is a group-to-RP mapping mechanism developed by Cisco?

a. BSR

b. Static RP

c. Auto-RP

d. Phantom RP

e. Anycast-RP

13. True or false: When PIM is configured in dense mode, it is mandatory to choose one or more routers to operate as rendezvous points (RPs)

a. True

b. False

Answers to the "Do I Know This Already?" quiz:

1. E

2. B, C

3. A, D

4. C

5. B

6. C

7. B

8. C

9. B, C

10. B

11. B

12. C

13. B

FOUNDATION TOPICS

MULTICAST FUNDAMENTALS

Traditional IP communication between network hosts typically uses one of the following transmission methods:

- Unicast (one-to-one)

- Broadcast (one-to-all)
- Multicast (one-to-many)

Multicast communication is a technology that optimizes network bandwidth utilization and conserves system resources. It relies on Internet Group Management Protocol (IGMP) for its operation in Layer 2 networks and Protocol Independent Multicast (PIM) for its operation in Layer 3 networks.

Figure 13-1 illustrates how IGMP operates between the receivers and the local multicast router and how PIM operates between routers. These two technologies work hand-in-hand to allow multicast traffic to flow from the source to the receivers, and they are explained in this chapter.

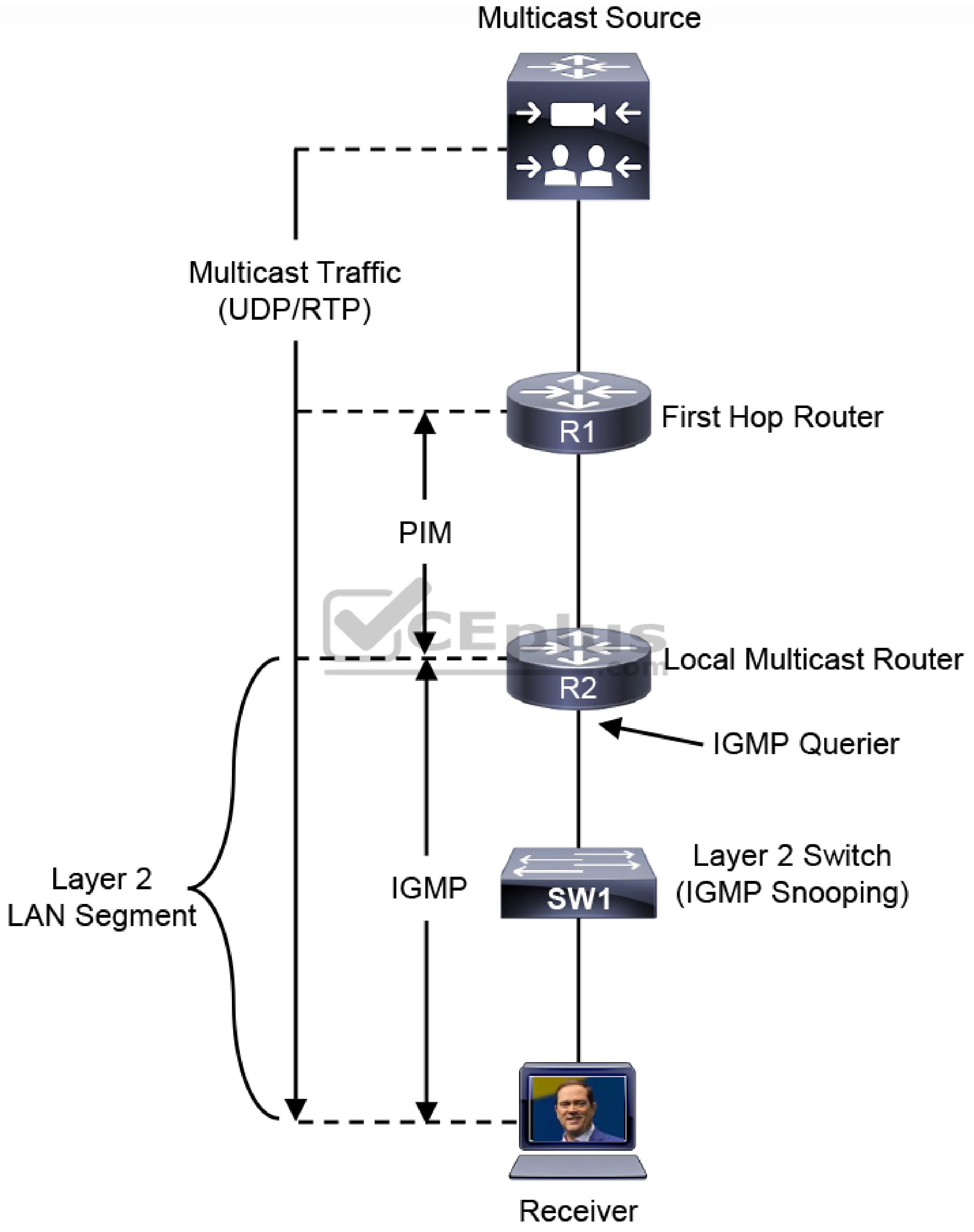

Multicast Source
Multicast Traffic
(UDP/RTP)
R1
First Hop Router
PIM
R2
Local Multicast Router
IGMP Querier
Layer 2
LAN Segment
IGMP
SW1
Layer 2 Switch
(IGMP Snooping)
Receiver

Figure 13-1 Multicast Architecture

Figure 13-2 shows an example where six workstations are watching the same video that is advertised by a server using unicast traffic (one-to-one). Each arrow represents a data stream of the same video going to five different hosts. If each stream is 10 Mbps, the network link between R1 and R2 needs 50 Mbps of bandwidth. The network link between R2 and R4 requires 30 Mbps of bandwidth, and the link between R2 and R5 requires 20 Mbps of bandwidth. The server must maintain session state information for all the sessions between the hosts. The bandwidth and load on the server increase as more receivers request the same video feed.

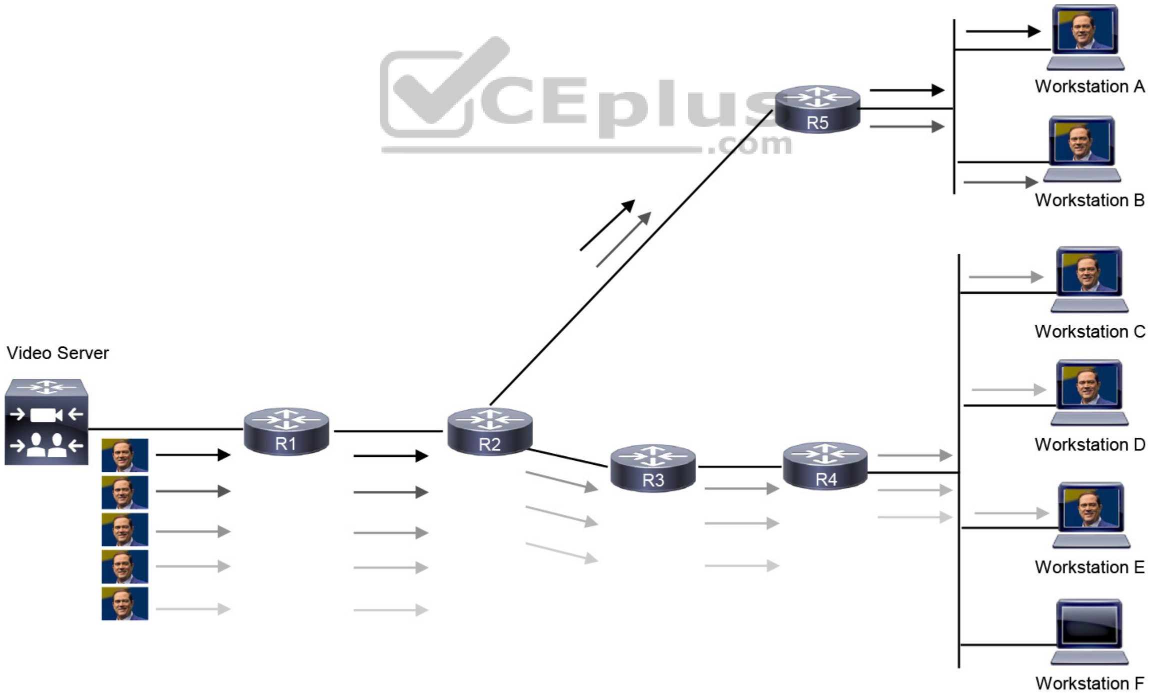


Figure 13-2 Unicast Video Feed

An alternative method for all five workstations to receive the video is to send it from the server using broadcast traffic (one-to-all). Figure 13-3 shows an example of how the same video stream is transmitted using IP directed broadcasts. The load on the server is reduced because it needs to maintain only one session state rather than many. The same video stream consumes only 10 Mbps of bandwidth on all network links. However, this approach does have disadvantages:

- IP directed broadcast functionality is not enabled by default on Cisco routers, and enabling it exposes the router to distributed denial-of-service (DDoS) attacks.

- The network interface cards (NICs) of uninterested workstations must still process the broadcast packets and send them on to the workstation’s CPU, which wastes processor resources. In Figure 13-3, Workstation F is processing unwanted packets.

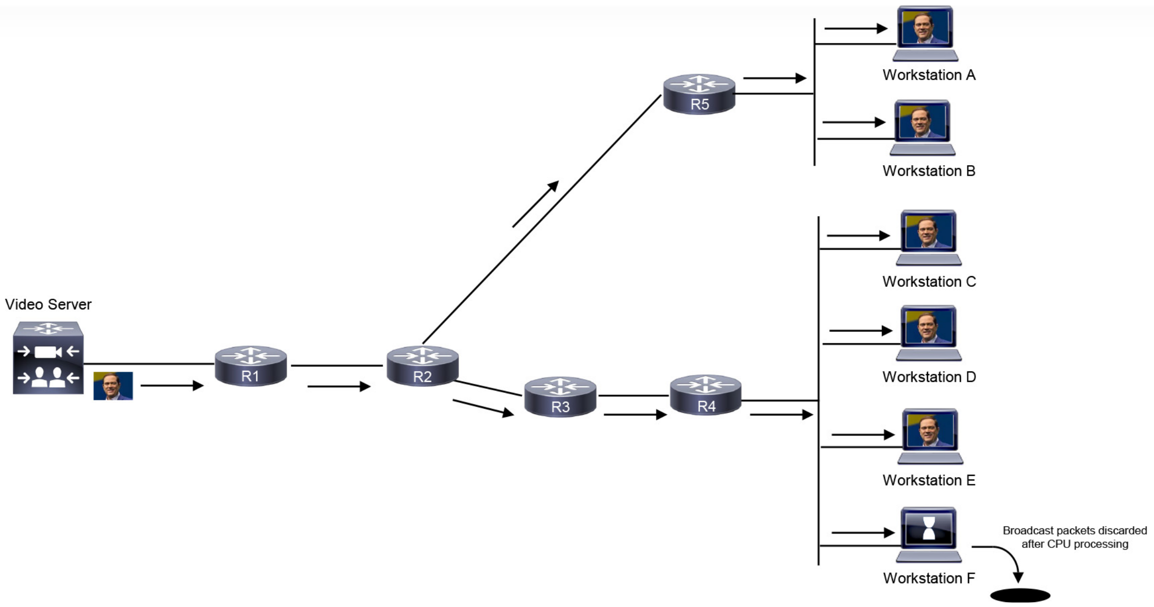


Figure 13-3 Broadcast Video Feed

For these reasons, broadcast traffic is generally not recommended.

Key Topic

Multicast traffic provides one-to-many communication, where only one data packet is sent on a link as needed and then is replicated between links as the data forks (splits) on a network device along the multicast distribution tree (MDT). The data packets are known as a *stream* that use a special destination IP address, known as a *group address*. A server for a stream still manages only one session, and network devices selectively request to receive the stream. Recipient devices of a multicast stream are known as *receivers*. Common applications that take advantage of multicast traffic include Cisco TelePresence, real-time video, IPTV, stock tickers, distance learning, video/audio conferencing, music on hold, and gaming.

Figure 13-4 shows an example of the same video feed using multicast. Each of the network links consumes only 10 Mbps of bandwidth, as much as with broadcast traffic, but only receivers that are interested in the video stream process the multicast traffic. For example, Workstation F would drop the multicast traffic at the NIC level because it would not be programmed to accept the multicast traffic.

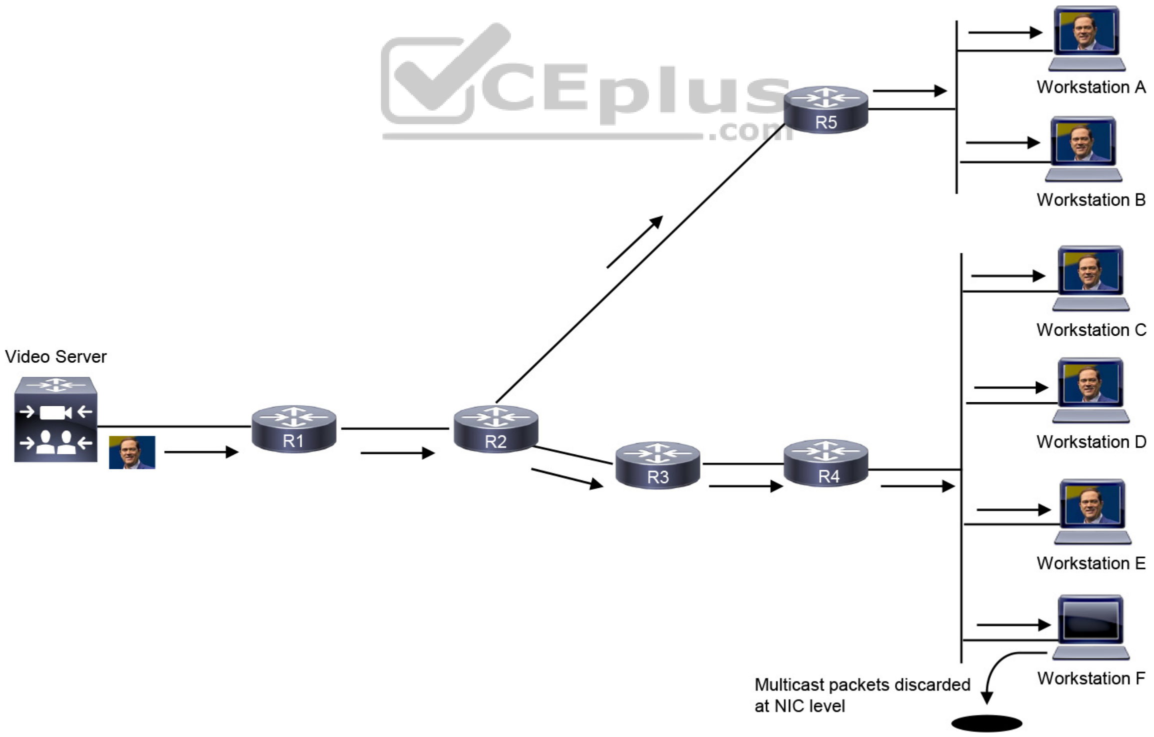


Figure 13-4 Multicast Video Feed

Note

Workstation F would not receive any multicast traffic if the switch for that network segment enabled Internet Group Management Protocol (IGMP) snooping. IGMP and IGMP snooping are covered in the next section.

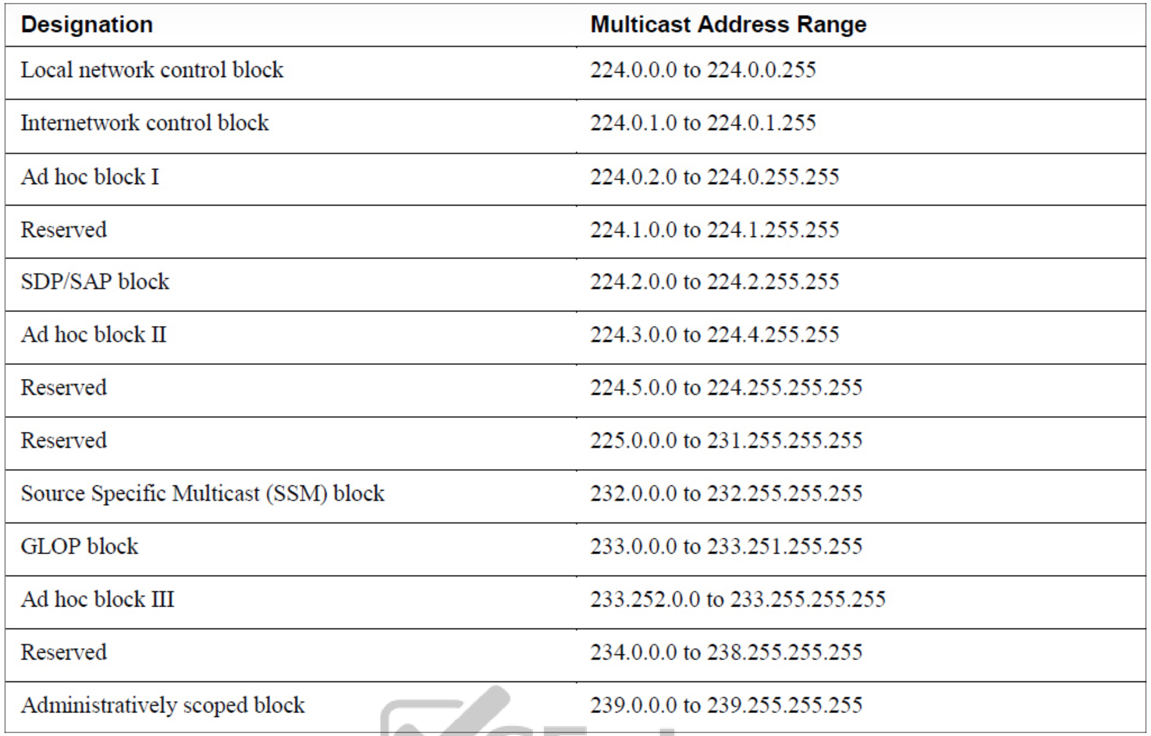
MULTICAST ADDRESSING

The Internet Assigned Number Authority (IANA) assigned the IP Class D address space 224.0.0.0/4 for multicast addressing; it includes addresses ranging from 224.0.0.0 to 239.255.255.255. The first 4 bits of this whole range start with 1110.

In the multicast address space, multiple blocks of addressing are reserved for specific purposes, as shown in Table 13-2.

Key Topic

Table 13-2 IP Multicast Addresses Assigned by IANA



Designation	Multicast Address Range
Local network control block	224.0.0.0 to 224.0.0.255
Internetwork control block	224.0.1.0 to 224.0.1.255
Ad hoc block I	224.0.2.0 to 224.0.255.255
Reserved	224.1.0.0 to 224.1.255.255
SDP/SAP block	224.2.0.0 to 224.2.255.255
Ad hoc block II	224.3.0.0 to 224.4.255.255
Reserved	224.5.0.0 to 224.255.255.255
Reserved	225.0.0.0 to 231.255.255.255
Source Specific Multicast (SSM) block	232.0.0.0 to 232.255.255.255
GLOP block	233.0.0.0 to 233.251.255.255
Ad hoc block III	233.252.0.0 to 233.255.255.255
Reserved	234.0.0.0 to 238.255.255.255
Administratively scoped block	239.0.0.0 to 239.255.255.255

Out of the multicast blocks mentioned in Table 13-2, the most important are discussed in the list that follows:

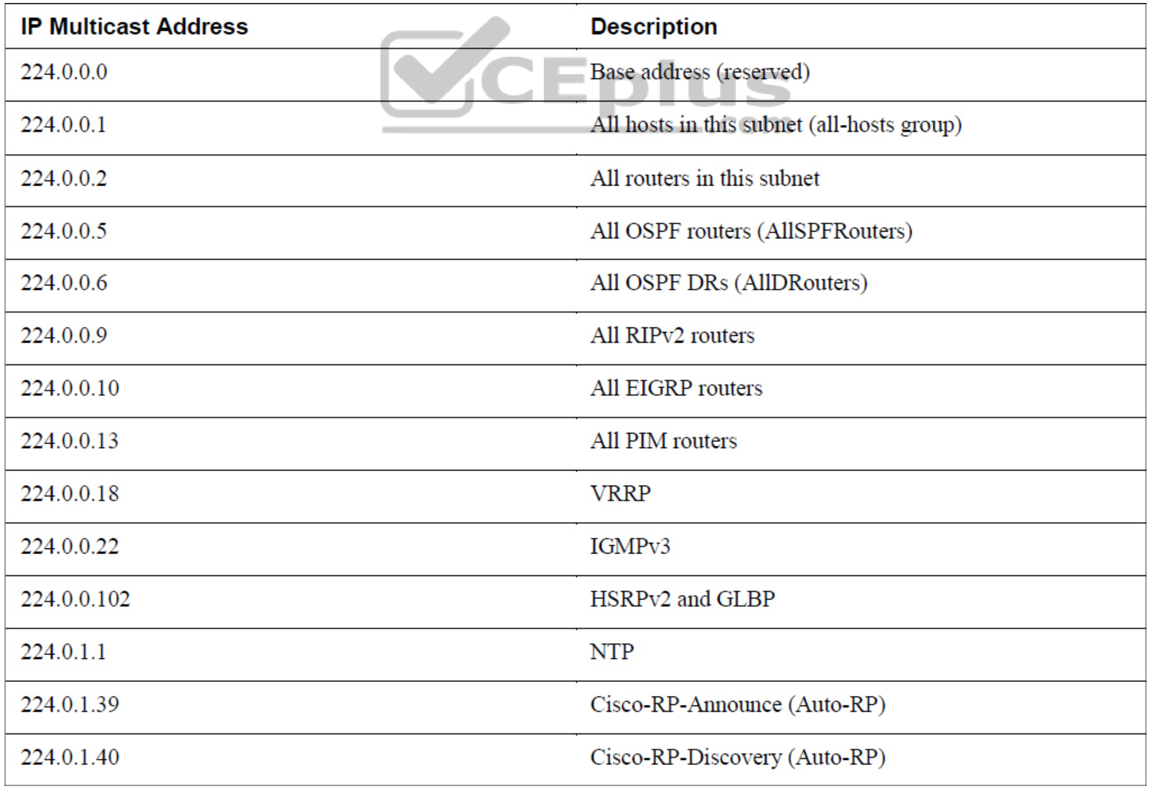
- **Local network control block (224.0.0/24):** Addresses in the local network control block are used for protocol control traffic that is not forwarded out a broadcast domain. Examples of this type of multicast control traffic are all hosts in this subnet (224.0.0.1), all routers in this subnet (224.0.0.2), and all PIM routers (224.0.0.13).

- **Internetwork control block (224.0.1.0/24):** Addresses in the internetwork control block are used for protocol control traffic that may be forwarded through the Internet. Examples include Network Time Protocol (NTP) (224.0.1.1), Cisco-RP-Announce (224.0.1.39), and Cisco-RP-Discovery (224.0.1.40).

Table 13-3 lists some of the well-known local network control block and internetwork control block multicast addresses.

Key Topic

Table 13-3 Well-Known Reserved Multicast Addresses



IP Multicast Address	Description
224.0.0.0	Base address (reserved)
224.0.0.1	All hosts in this subnet (all-hosts group)
224.0.0.2	All routers in this subnet
224.0.0.5	All OSPF routers (AllSPFRouters)
224.0.0.6	All OSPF DRs (AllDRouters)
224.0.0.9	All RIPv2 routers
224.0.0.10	All EIGRP routers
224.0.0.13	All PIM routers
224.0.0.18	VRRP
224.0.0.22	IGMPv3
224.0.0.102	HSRPv2 and GLBP
224.0.1.1	NTP
224.0.1.39	Cisco-RP-Announce (Auto-RP)
224.0.1.40	Cisco-RP-Discovery (Auto-RP)

- **Source Specific Multicast (SSM) block (232.0.0.0/8):** This is the default range used by SSM. SSM is a PIM extension described in RFC 4607. SSM forwards traffic to receivers from only those multicast sources for which the receivers have explicitly expressed interest; it is primarily targeted to one-to-many applications.

- **GLOP block (233.0.0.0/8):** Addresses in the GLOP block are globally scoped statically assigned addresses. The assignment is made for domains with a 16-bit autonomous system number (ASN) by mapping the domain's ASN, expressed in octets as X.Y, into the middle two octets of the GLOP block, yielding an assignment of 233.X.Y.0/24. The mapping and assignment are defined in RFC 3180. Domains with a 32-bit ASN may apply for space in ad-hoc block III or can consider using IPv6 multicast addresses.

- **Administratively scoped block (239.0.0.0/8):** These addresses, described in RFC 2365, are limited to a local group or organization. These addresses are similar to the reserved IP unicast ranges (such as 10.0.0.0/8) defined in RFC 1918 and will not be assigned by the IANA to any other group or protocol. In other words, network administrators are free to use multicast addresses in this range inside of their domain without worrying about conflicting with others elsewhere on the Internet. Even though SSM is assigned to the 232.0.0.0/8 range by default, it is typically deployed in private networks using the 239.0.0.0/8 range.

Key Topic

Layer 2 Multicast Addresses

Historically, NICs on a LAN segment could receive only packets destined for their burned-in MAC address or the broadcast MAC address. Using this logic can cause a burden on routing resources during packet replication for LAN segments. Another method for multicast traffic was created so that replication of multicast traffic did not require packet manipulation, and a method of using a common destination MAC address was created.

A MAC address is a unique value associated with a NIC that is used to uniquely identify the NIC on a LAN segment. MAC addresses are 12-digit hexadecimal numbers (48 bits in length), and they are typically stored in 8-bit segments separated by hyphens (-) or colons (:) (for example, 00-12-34-56-78-00 or 00:12:34:56:78:00).

Every multicast group address (IP address) is mapped to a special MAC address that allows Ethernet interfaces to identify multicast packets to a specific group. A LAN segment can have multiple streams, and a receiver knows which traffic to send to the CPU for processing based on the MAC address assigned to the multicast traffic.

The first 24 bits of a multicast MAC address always start with 01:00:5E. The low-order bit of the first byte is the *individual/group bit (I/G)* bit, also known as the unicast/multicast bit, and when it is set to 1, it indicates that the frame is a multicast frame, and the 25th bit is always 0. The lower 23 bits of the multicast MAC address are copied from the lower 23 bits of the multicast group IP address.

Figure 13-5 shows an example of mapping the multicast IP address 239.255.1.1 into multicast MAC address 01:00:5E:7F:01:01. The first 25 bits are always fixed; the last 23 bits that are copied directly from the multicast IP address vary.

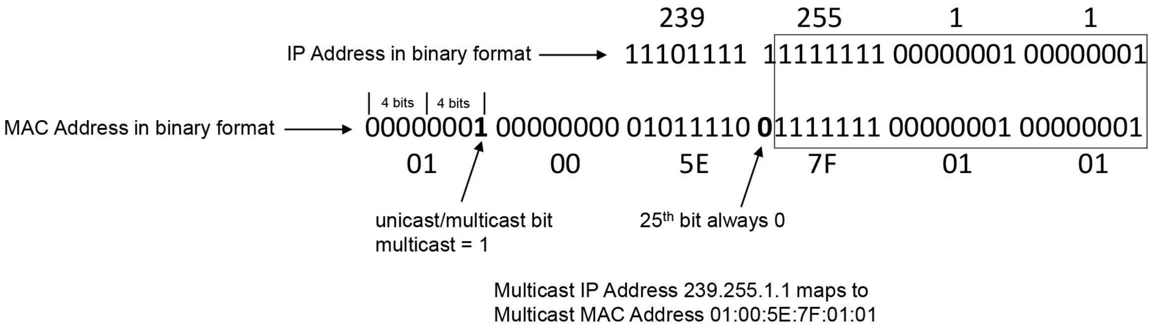
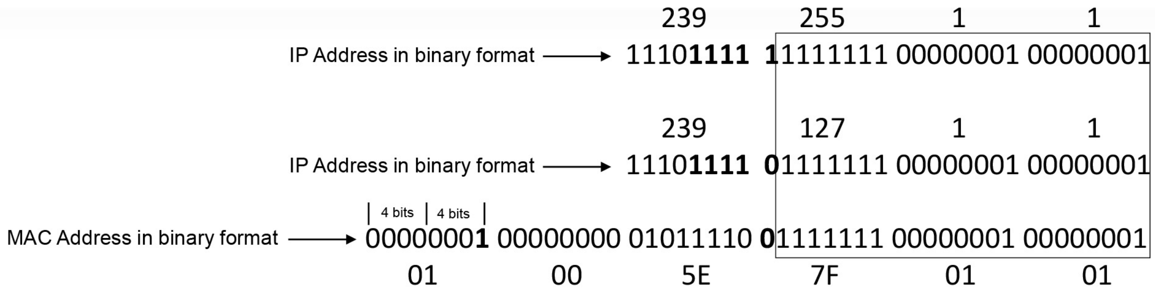


Figure 13-5 Multicast IP Address-to-Multicast MAC Address Mapping

Out of the 9 bits from the multicast IP address that are not copied into the multicast MAC address, the high-order bits 1110 are fixed; that leaves 5 bits that are variable that are not transferred into the MAC address. Because of this, there are 32 (2^5) multicast IP addresses that are not universally unique and could correspond to a single MAC address; in other words, they overlap. Figure 13-6 shows an example of two multicast IP addresses that overlap because they map to the same multicast MAC address.



Multicast IP Addresses 239.255.1.1 and 239.127.1.1 map to
Multicast MAC Address 01:00:5E:7F:01:01

Figure 13-6 Multicast IP Address to Multicast MAC Address Mapping Overlap

When a receiver wants to receive a specific multicast feed, it sends an IGMP join using the multicast IP group address for that feed. The receiver reprograms its interface to accept the multicast MAC group address that correlates to the group address. For example, a PC could send a join to 239.255.1.1 and would reprogram its NIC to receive 01:00:5E:7F:01:01. If the PC were to receive an OSPF update sent to 224.0.0.5 and its corresponding multicast MAC 01:00:5E:00:00:05, it would ignore it and eliminate wasted CPU cycles by avoiding the processing of undesired multicast traffic.

INTERNET GROUP MANAGEMENT PROTOCOL

Key Topic

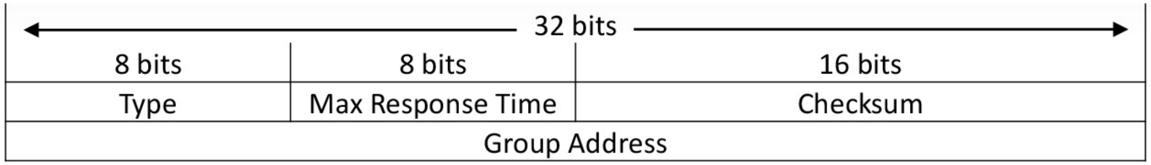
Internet Group Management Protocol (IGMP) is the protocol that receivers use to join multicast groups and start receiving traffic from those groups. IGMP must be supported by receivers and the router interfaces facing the receivers. When a receiver wants to receive multicast traffic from a source, it sends an IGMP join to its router. If the router does not have IGMP enabled on the interface, the request is ignored.

Three versions of IGMP exist. RFC 1112 defines IGMPv1, which is old and rarely used. RFC 2236 defines IGMPv2, which is common in most multicast networks, and RFC 3376 defines IGMPv3, which is used by SSM. Only IGMPv2 and IGMPv3 are described in this chapter.

IGMPv2

Key Topic

IGMPv2 uses the message format shown in Figure 13-7. This message is encapsulated in an IP packet with a protocol number of 2. Messages are sent with the IP router alert option set, which indicates that the packets should be examined more closely, and a time-to-live (TTL) of 1. TTL is an 8-bit field in an IP packet header that is set by the sender of the IP packet and decremented by every router on the route to its destination. If the TTL reaches 0 before reaching the destination, the packet is discarded. IGMP packets are sent with a TTL of 1 so that packets are processed by the local router and not forwarded by any router.



32 bits		
8 bits	8 bits	16 bits
Type	Max Response Time	Checksum
Group Address		

Figure 13-7 IGMP Message Format

The IGMP message format fields are defined as follows:

- **Type:** This field describes five different types of IGMP messages used by routers and receivers:

- **Version 2 membership report** (type value 0x16) is a message type also commonly referred to as an IGMP join; it is used by receivers to join a multicast group or to respond to a local router’s membership query message.

- **Version 1 membership report** (type value 0x12) is used by receivers for backward compatibility with IGMPv1.

- **Version 2 leave group** (type value 0x17) is used by receivers to indicate they want to stop receiving multicast traffic for a group they joined.

- **General membership query** (type value 0x11) is sent periodically sent to the all-hosts group address 224.0.0.1 to see whether there are any receivers in the attached subnet. It sets the group address field to 0.0.0.0.

• **Group specific query** (type value 0x11) is sent in response to a leave group message to the group address the receiver requested to leave. The group address is the destination IP address of the IP packet and the group address field.

• **Max response time:** This field is set only in general and group-specific membership query messages (type value 0x11); it specifies the maximum allowed time before sending a responding report in units of one-tenth of a second. In all other messages, it is set to 0x00 by the sender and ignored by receivers.

• **Checksum:** This field is the 16-bit 1s complement of the 1s complement sum of the IGMP message. This is the standard checksum algorithm used by TCP/IP.

• **Group address:** This field is set to 0.0.0.0 in general query messages and is set to the group address in group-specific messages. Membership report messages carry the address of the group being reported in this field; group leave messages carry the address of the group being left in this field.

Key Topic

When a receiver wants to receive a multicast stream, it sends an unsolicited membership report, commonly referred to as an IGMP join, to the local router for the group it wants to join (for example, 239.1.1.1). The local router then sends this request upstream toward the source using a PIM join message. When the local router starts receiving the multicast stream, it forwards it downstream to the subnet where the receiver that requested it resides.

Note

IGMP join is not a valid message type in the IGMP RFC specifications, but the term is commonly used in the field in place of *IGMP membership reports* because it is easier to say and write.

The router then starts periodically sending general membership query messages into the subnet, to the all-hosts group address 224.0.0.1, to see whether any members are in the attached subnet. The general query message contains a max response time field that is set to 10 seconds by default.

In response to this query, receivers set an internal random timer between 0 and 10 seconds (which can change if the max response time is using a non-default value). When the timer expires, receivers send membership reports for each group they belong to. If a receiver receives another receiver’s report for one of the groups it belongs to while it has a timer running, it stops its timer for the specified group and does not send a report; this is meant to suppress duplicate reports.

When a receiver wants to leave a group, if it was the last receiver to respond to a query, it sends a leave group message to the all-routers group address 224.0.0.2. Otherwise, it can leave quietly because there must be another receiver in the subnet.

When the leave group message is received by the router, it follows with a specific membership query to the group multicast address to determine whether there are any receivers interested in the group remaining in the subnet. If there are none, the router removes the IGMP state for that group.

If there is more than one router in a LAN segment, an IGMP querier election takes place to determine which router will be the querier. IGMPv2 routers send general membership query messages with their interface address as the source IP address and destined to the 224.0.0.1 multicast address. When an IGMPv2 router receives such a message, it checks the source IP address and compares it to its own interface IP address. The router with the lowest interface IP address in the LAN subnet is elected as the IGMP querier. At this point, all the non-querier routers start a timer that resets each time they receive a membership query report from the querier router.

If the querier router stops sending membership queries for some reason (for instance, if it is powered down), a new querier election takes place. A non-querier router waits twice the query interval, which is by default 60 seconds, and if it has heard no queries from the IGMP querier, it triggers IGMP querier election.

IGMPv3

Key Topic

In IGMPv2, when a receiver sends a membership report to join a multicast group, it does not specify which source it would like to receive multicast traffic from. IGMPv3 is an extension of IGMPv2 that adds support for multicast source filtering, which give the receivers the capability to pick the source they wish to accept multicast traffic from.

IGMPv3 is designed to coexist with IGMPv1 and IGMPv2.

IGMPv3 supports all IGMPv2's IGMP message types and is backward compatible with IGMPv2. The differences between the two are that IGMPv3 added new fields to the IGMP membership query and introduced a new IGMP message type called Version 3 membership report to support source filtering.

IGMPv3 supports applications that explicitly signal sources from which they want to receive traffic. With IGMPv3, receivers signal membership to a multicast group address using a membership report in the following two modes:

- **Include mode:** In this mode, the receiver announces membership to a multicast group address and provides a list of source addresses (the *include list*) from which it wants to receive traffic.

- **Exclude mode:** In this mode, the receiver announces membership to a multicast group address and provides a list of source addresses (the *exclude list*) from which it does not want to receive traffic. The receiver then receives traffic only from sources whose IP addresses are not listed on the exclude list. To receive traffic from all sources, which is the behavior of IGMPv2, a receiver uses exclude mode membership with an empty exclude list.

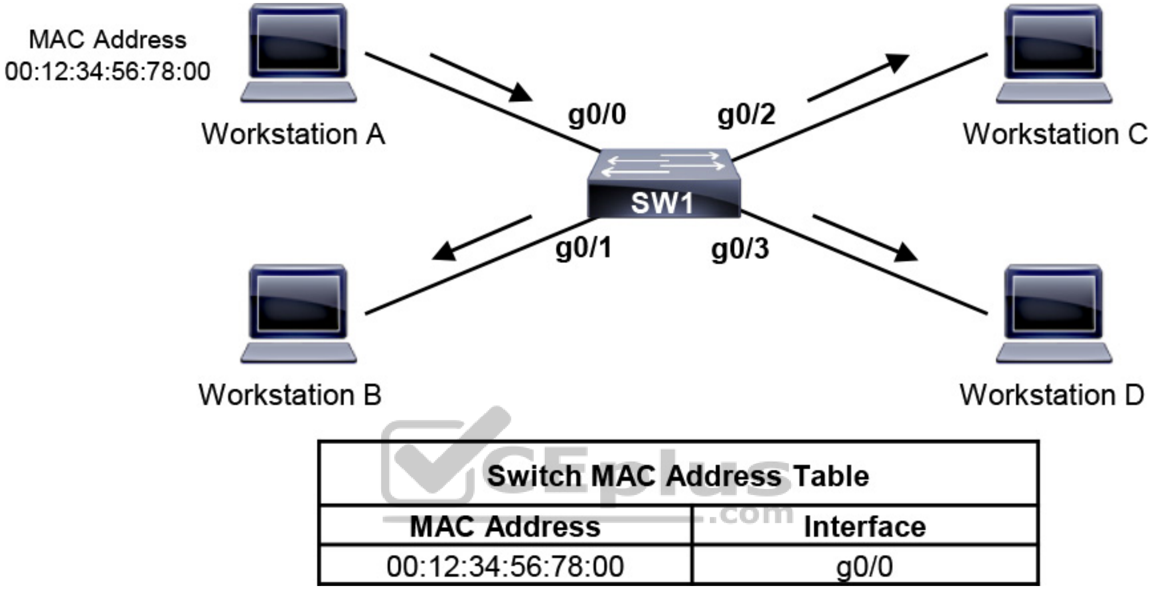
Note

IGMPv3 is used to provide source filtering for Source Specific Multicast (SSM).

IGMP Snooping

To optimize forwarding and minimize flooding, switches need a method of sending traffic only to interested receivers. In the case of unicast traffic, Cisco switches learn about Layer 2 MAC addresses and what ports they belong to by inspecting the Layer 2 MAC address source; they store this information in the MAC address table. If they receive a Layer 2 frame with a destination MAC address that is not in this table, they treat it as an unknown frame and flood it out all the ports within the same VLAN except the interface the frame was received on. Uninterested workstations will notice that the destination MAC address in the frame is not theirs and will discard the packet.

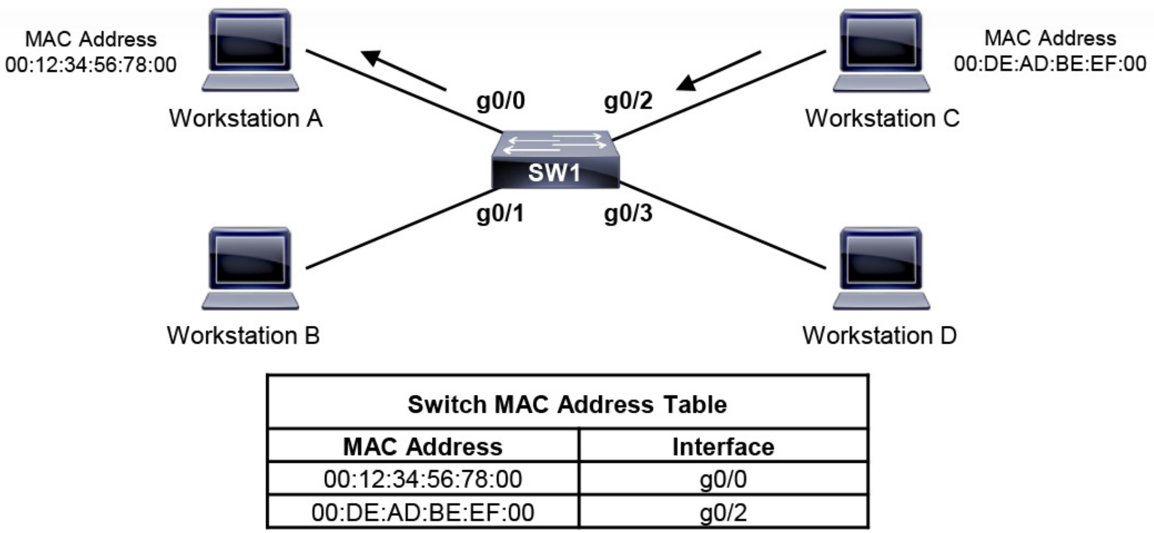
In Figure 13-8, SW1 starts with an empty MAC address table. When Workstation A sends a frame, it stores its source MAC address and interface in the MAC address table and floods the frame it received out all ports (except the port it received the frame on).



Switch MAC Address Table	
MAC Address	Interface
00:12:34:56:78:00	g0/0

Figure 13-8 Unknown Frame Flooding

If any other workstation sends a frame destined to the MAC address of Workstation A, the frame is not flooded anymore because it's already in the MAC address table, and it is sent only to Workstation A, as shown in Figure 13-9.



Switch MAC Address Table	
MAC Address	**Interface**
00:12:34:56:78:00	g0/0
00:DE:AD:BE:EF:00	g0/2

Figure 13-9 Known Destination Is Not Flooded

In the case of multicast traffic, a multicast MAC address is never used as a source MAC address. Switches treat multicast MAC addresses as unknown frames and flood them out all ports; all workstations then process these frames. It is then up to the workstations to select interested frames for processing and select the frames that should be discarded. The flooding of multicast traffic on a switch wastes bandwidth utilization on each LAN segment.

Cisco switches use two methods to reduce multicast flooding on a LAN segment:

- IGMP snooping

- Static MAC address entries

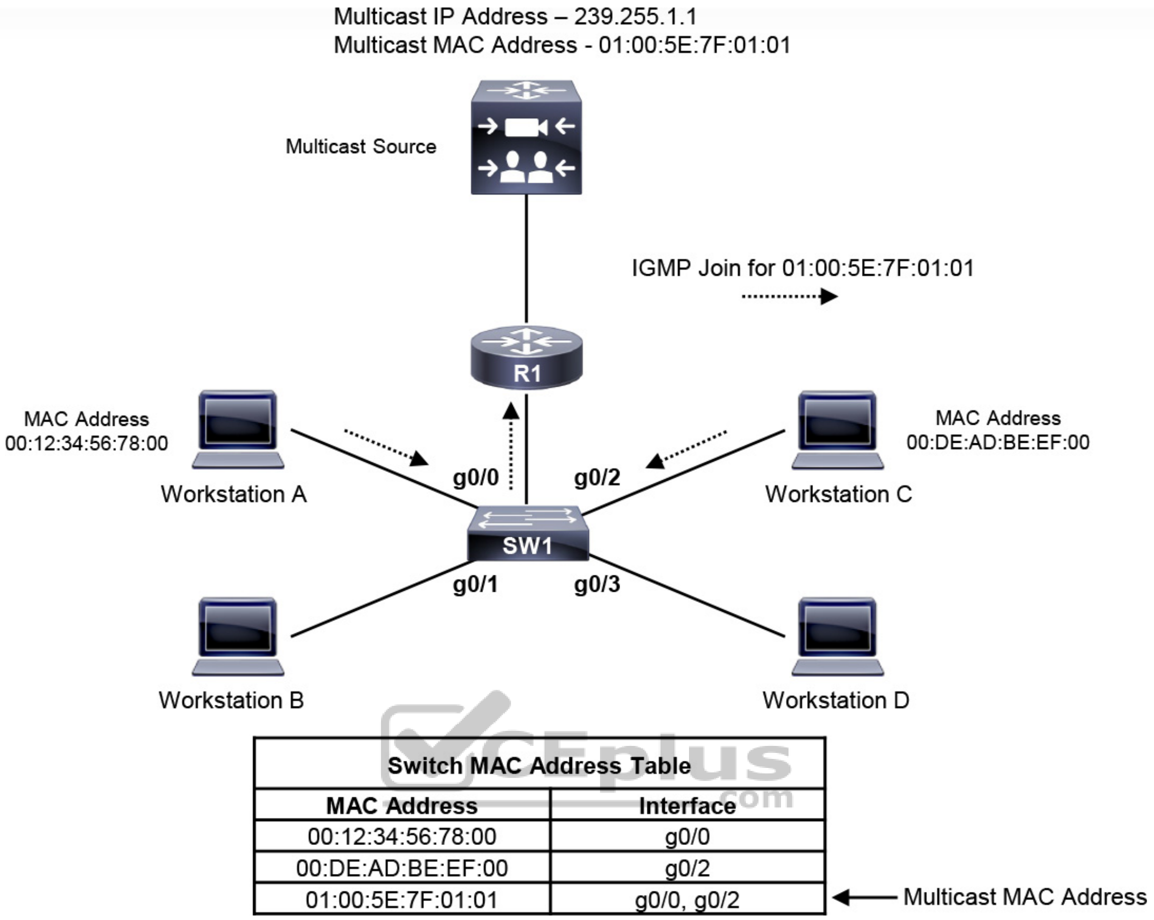
Key Topic

IGMP snooping, defined in RFC 4541, is the most widely used method and works by examining IGMP joins sent by receivers and maintaining a table of interfaces to IGMP joins. When the switch receives a multicast frame destined for a multicast group, it forwards the packet only out the ports where IGMP joins were received for that specific multicast group.

Figure 13-10 illustrates Workstation A and Workstation C sending IGMP joins to 239.255.1.1, which translates to the multicast MAC address 01:00:5E:7F:01:01. Switch 1 has IGMP snooping enabled and populates the MAC address table with this information.

Note

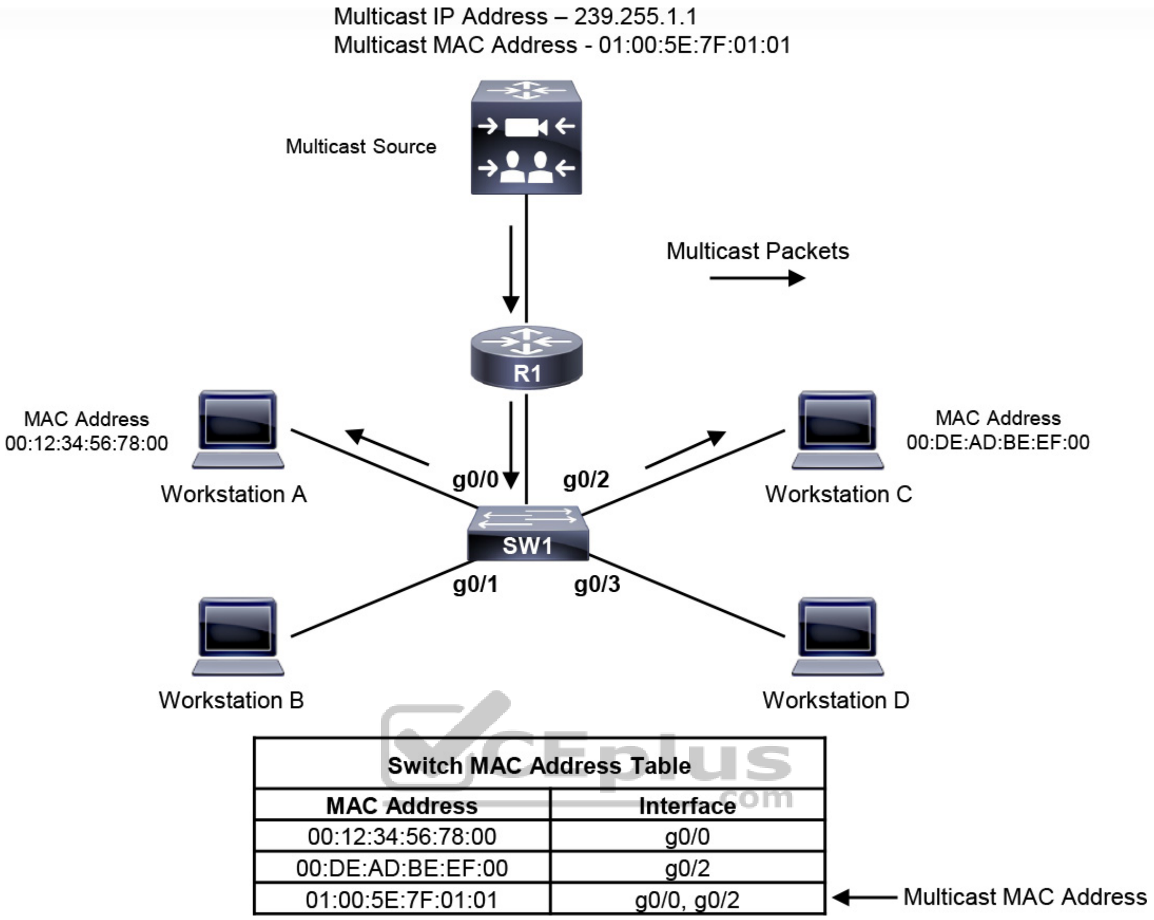
Even with IGMP snooping enabled, some multicast groups are still flooded on all ports (for example, 224.0.0.0/24 reserved addresses).



Switch MAC Address Table	
MAC Address	**Interface**
00:12:34:56:78:00	g0/0
00:DE:AD:BE:EF:00	g0/2
01:00:5E:7F:01:01	g0/0, g0/2

Figure 13-10 IGMP Snooping Example

Figure 13-11 illustrates the source sending traffic to 239.255.1.1(01:00:5E:7F:01:01). Switch 1 receives this traffic, and it forwards it out only the g0/0 and g0/2 interfaces because those are the only ports that received IGMP joins for that group.



Switch MAC Address Table	
MAC Address	**Interface**
00:12:34:56:78:00	g0/0
00:DE:AD:BE:EF:00	g0/2
01:00:5E:7F:01:01	g0/0, g0/2

Figure 13-11 No Flooding with IGMP Snooping

A multicast static entry can also be manually programmed into the MAC address table, but this is not a scalable solution because it cannot react dynamically to changes; for this reason, it is not a recommended approach.

PROTOCOL INDEPENDENT MULTICAST

Key Topic

Receivers use IGMP to join a multicast group, which is sufficient if the group's source connects to the same router to which the receiver is attached. A multicast routing protocol is necessary to route the multicast traffic throughout the network so that routers can locate and request multicast streams from other routers. Multiple multicast routing protocols exist, but Cisco fully supports only Protocol Independent Multicast (PIM).

PIM is a multicast routing protocol that routes multicast traffic between network segments. PIM can use any of the unicast routing protocols to identify the path between the source and receivers.

PIM Distribution Trees

Multicast routers create distribution trees that define the path that IP multicast traffic follows through the network to reach the receivers. The two basic types of multicast distribution trees are source trees, also known as *shortest path trees (SPTs)*, and shared trees.

Source Trees

Key Topic

A *source tree* is a multicast distribution tree where the source is the root of the tree, and branches form a distribution tree through the network all the way down to the receivers. When this tree is built, it uses the shortest path through the network from the source to the leaves of the tree; for this reason, it is also referred to as a shortest path tree (SPT).

The forwarding state of the SPT is known by the notation (S,G), pronounced “S comma G,” where S is the source of the multicast stream (server), and G is the multicast group address. Using this notation, the SPT state for the example shown in Figure 13-12 is (10.1.1.2, 239.1.1.1), where the multicast source S is 10.1.1.2, and the multicast group G is 239.1.1.1, joined by Receivers A and B.

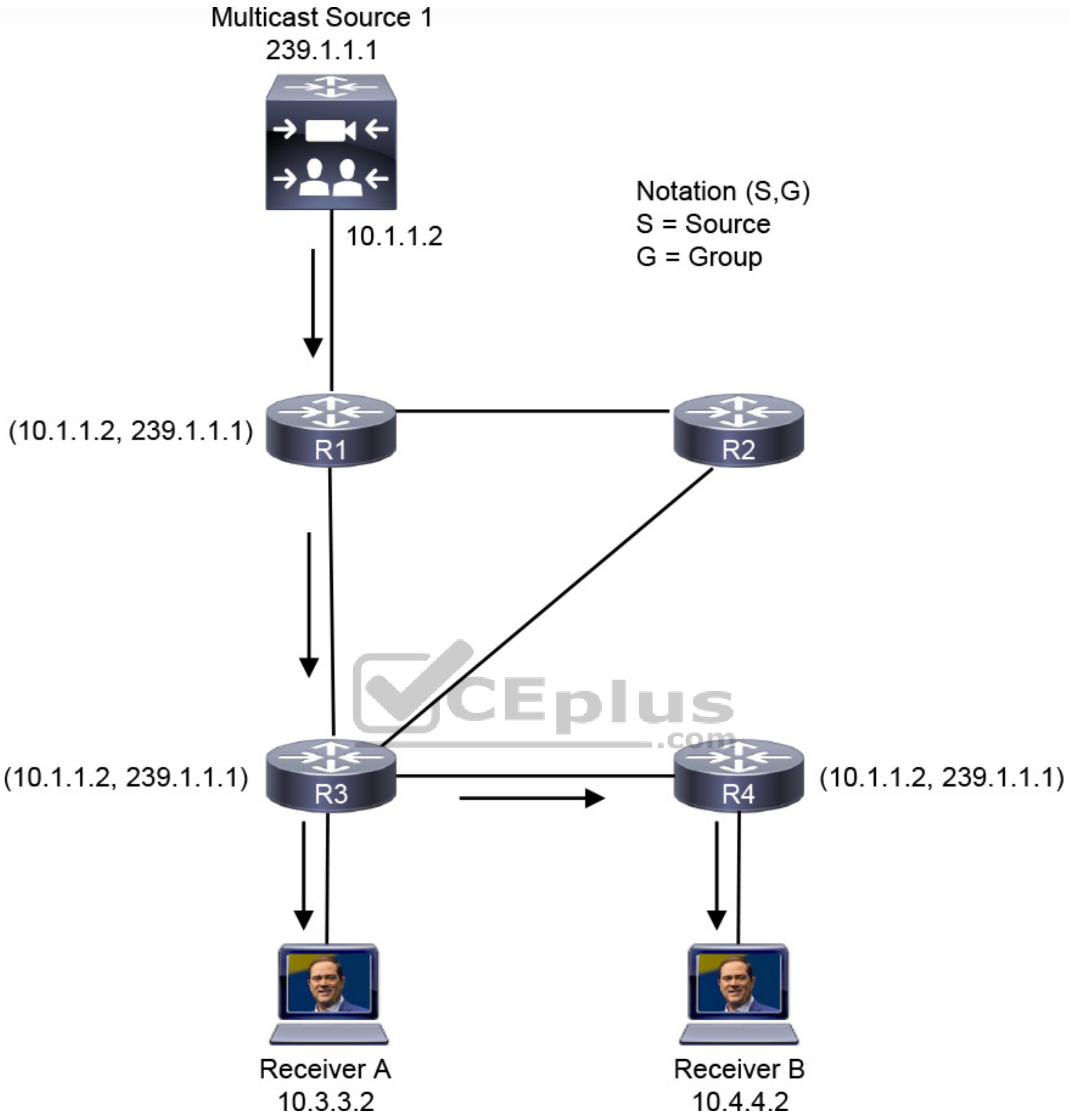


Figure 13-12 Source Tree Example

Because every SPT is rooted at the source S, every source sending to a multicast group requires an SPT.

Shared Trees

Key Topic

A shared tree is a multicast distribution tree where the root of the shared tree is not the source but a router designated as the rendezvous point (RP). For this reason, shared trees are also referred to as *RP trees (RPTs)*. Multicast traffic is forwarded down the shared tree according to the group address G that the packets are addressed to, regardless of the source address. For this reason, the forwarding state on the shared tree is referred to by the notation (*,G), pronounced “star comma G.” Figure 13-13 illustrates a shared tree where R2 is the RP, and the (*,G) is (*,239.1.1.1).

Note

In any-source multicast (ASM), the (S,G) state requires a parent (*,G). For this reason, Figure 13-13 illustrates R1 and R2 as having (*,G) state.

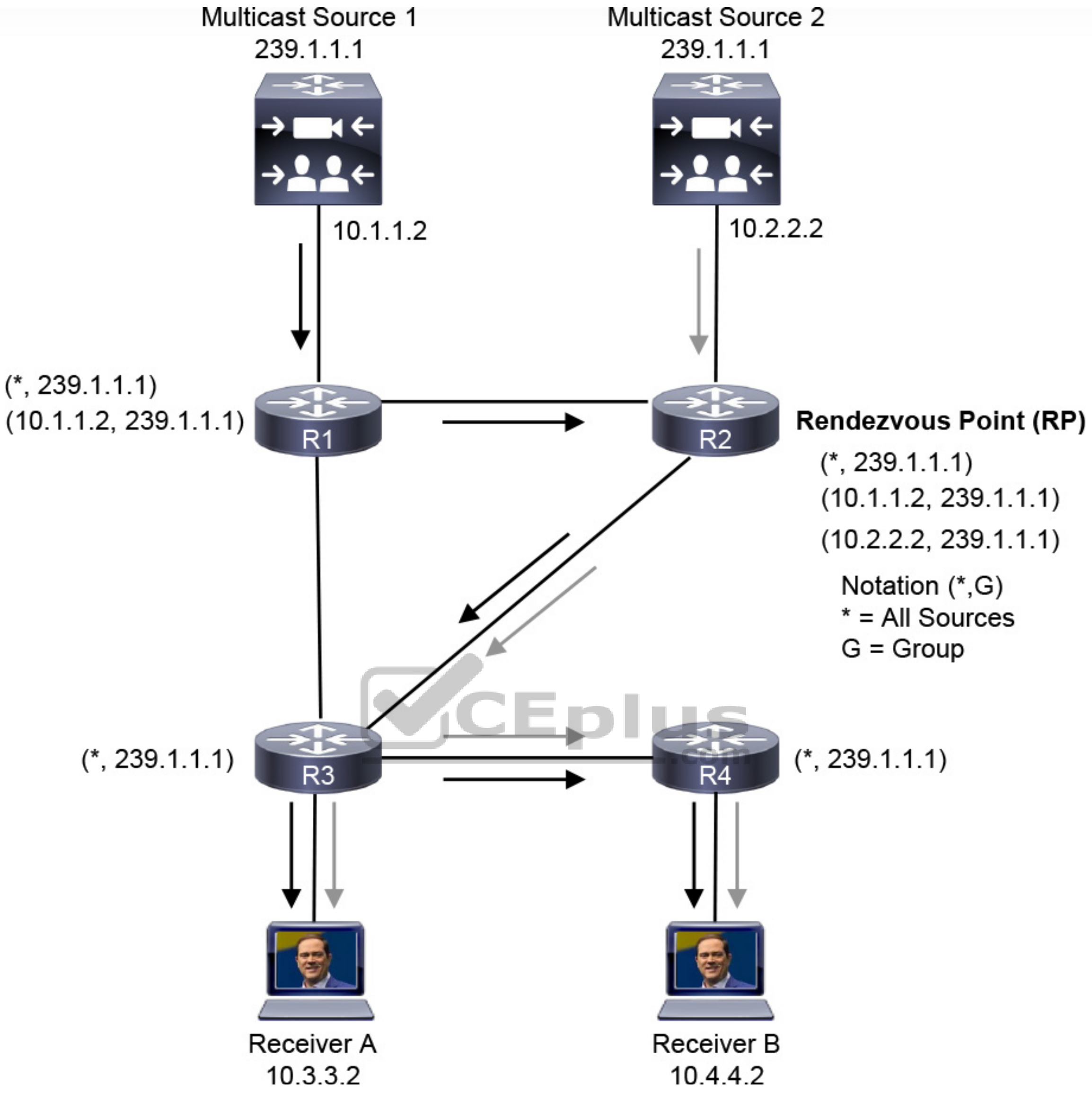


Figure 13-13 Shared Tree Between RP and LHRs

One of the benefits of shared trees over source trees is that they require fewer multicast entries (for example, S,G and *,G). For instance, as more sources are introduced into the network, sending traffic to the same multicast group, the number of multicast entries for R3 and R4 always remains the same: (*,239.1.1.1).

The major drawback of shared trees is that the receivers receive traffic from all the sources sending traffic to the same multicast group. Even though the receiver's applications can filter out the unwanted traffic, this situation still generates a lot of unwanted network traffic, wasting bandwidth. In addition, because shared trees can allow multiple sources in an IP multicast group, there is a potential network security issue because unintended sources could send unwanted packets to receivers.

PIM Terminology

Figure 13-14 provides a reference topology for some multicast routing terminology.

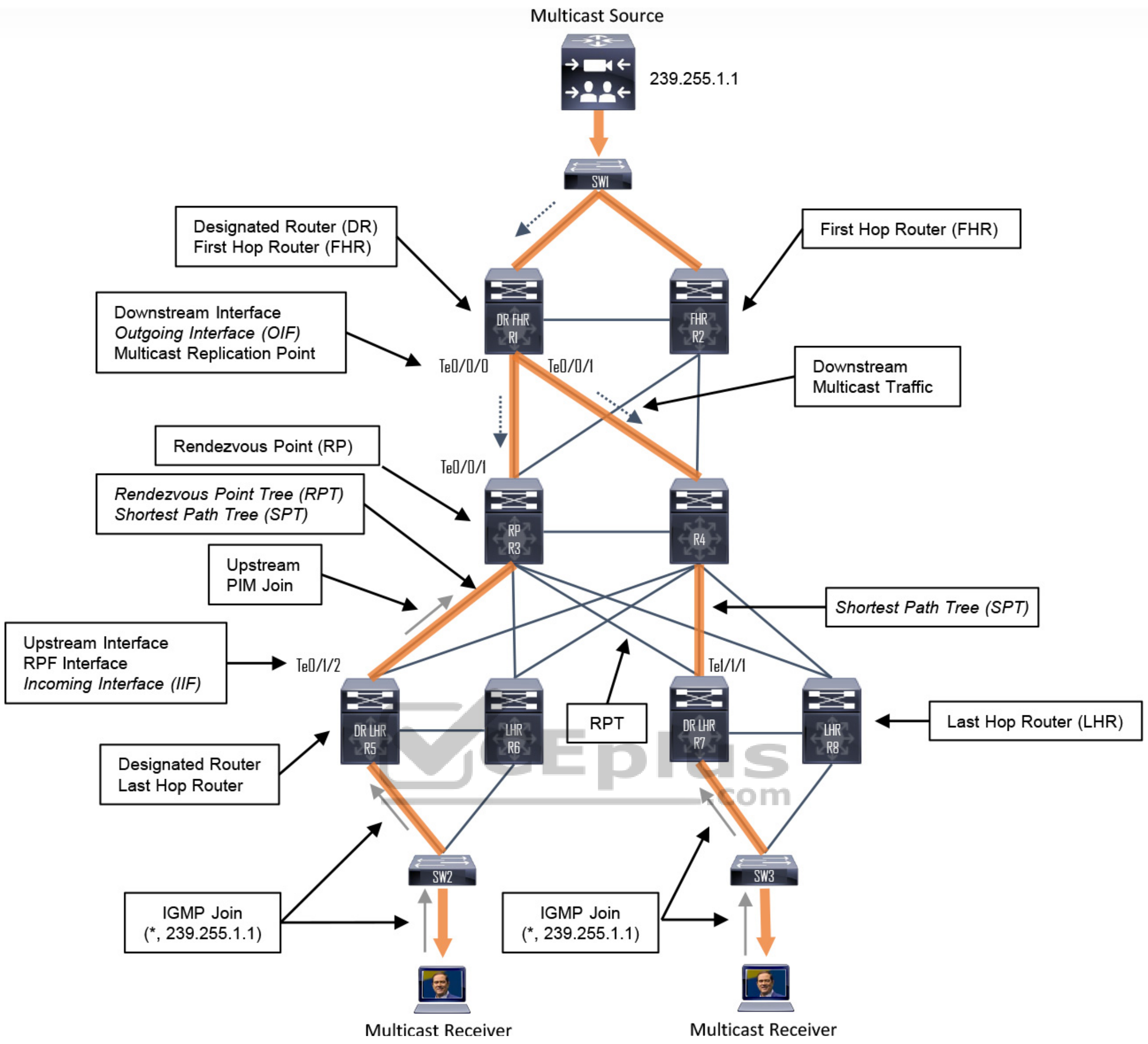


Figure 13-14 PIM Terminology Illustration

Key Topic

The following list defines the common PIM terminology illustrated in Figure 13-14:

- **Reverse Path Forwarding (RPF) interface:** The interface with the lowest-cost path (based on administrative distance [AD] and metric) to the IP address of the source (SPT) or the RP, in the case of shared trees. If multiple interfaces have the same cost, the interface with the highest IP address is chosen as the tiebreaker. An example of this type of interface is Te0/1/2 on R5 because it is the shortest path to the source. Another example is Te1/1/1 on R7 because the shortest path to the source was determined to be through R4.

- **RPF neighbor:** The PIM neighbor on the RPF interface. For example, if R7 is using the RPT shared tree, the RPF neighbor would be R3, which is the lowest-cost path to the RP. If it is using the SPT, R4 would be its RPF neighbor because it offers the lowest cost to the source.

- **Upstream:** Toward the source of the tree, which could be the actual source in source-based trees or the RP in shared trees. A PIM join travels upstream toward the source.

- **Upstream interface:** The interface toward the source of the tree. It is also known as the RPF interface or the incoming interface (IIF). An example of an upstream interface is R5’s Te0/1/2 interface, which can send PIM joins upstream to its RPF neighbor.

- **Downstream:** Away from the source of the tree and toward the receivers.

- **Downstream interface:** Any interface that is used to forward multicast traffic down the tree, also known as an outgoing interface (OIF). An example of a downstream interface is R1’s Te0/0/0 interface, which forwards multicast traffic to R3’s Te0/0/1 interface.

- **Incoming interface (IIF):** The only type of interface that can accept multicast traffic coming from the source, which is the same as the RPF interface. An example of this type of interface is Te0/0/1 on R3 because the shortest path to the source is known through this interface.

- **Outgoing interface (OIF):** Any interface that is used to forward multicast traffic down the tree, also known as the downstream interface.

- **Outgoing interface list (OIL):** A group of OIFs that are forwarding multicast traffic to the same group. An example of this is R1’s Te0/0/0 and Te0/0/1 interfaces sending multicast traffic downstream to R3 and R4 for the same multicast group.

- **Last-hop router (LHR):** A router that is directly attached to the receivers, also known as a leaf router. It is responsible for sending PIM joins upstream toward the RP or to the source.

- **First-hop router (FHR):** A router that is directly attached to the source, also known as a root router. It is responsible for sending register messages to the RP.

- **Multicast Routing Information Base (MRIB):** A topology table that is also known as the multicast route table (mroute), which derives from the unicast routing table and PIM. MRIB contains the source S, group G, incoming interfaces (IIF), outgoing interfaces (OIFs), and RPF neighbor information for each multicast route as well as other multicast-related information.

- **Multicast Forwarding Information Base (MFIB):** A forwarding table that uses the MRIB to program multicast forwarding information in hardware for faster forwarding.

- **Multicast state:** The multicast traffic forwarding state that is used by a router to forward multicast traffic. The multicast state is composed of the entries found in the mroute table (S, G, IIF, OIF, and so on).

Key Topic

There are currently five PIM operating modes:

- PIM Dense Mode (PIM-DM)

- PIM Sparse Mode (PIM-SM)

- PIM Sparse Dense Mode

- PIM Source Specific Multicast (PIM-SSM)

- PIM Bidirectional Mode (Bidir-PIM)

Note

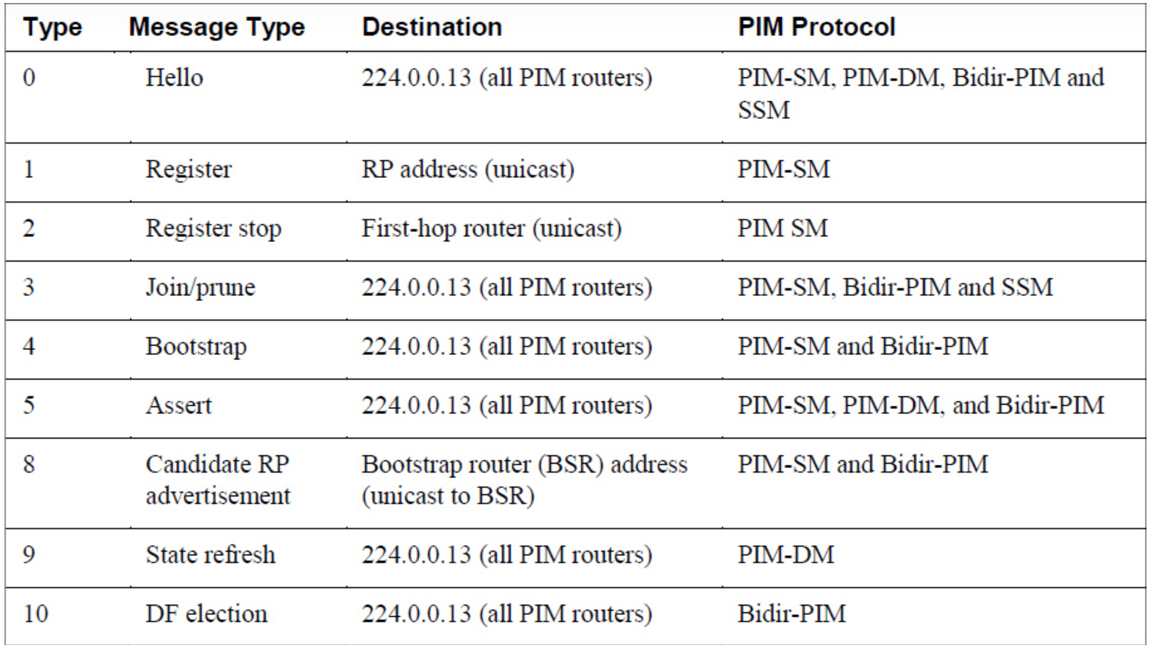
PIM-DM and PIM-SM are also commonly referred to as any-source multicast (ASM).

All PIM control messages use the IP protocol number 103; they are either unicast (that is, register and register stop messages) or multicast, with a TTL of 1 to the all PIM routers address 224.0.0.13.

Table 13-4 lists the PIM control messages.

Key Topic

Table 13-4 PIM Control Message Types



Type	Message Type	Destination	PIM Protocol
0	Hello	224.0.0.13 (all PIM routers)	PIM-SM, PIM-DM, Bidir-PIM and SSM
1	Register	RP address (unicast)	PIM-SM
2	Register stop	First-hop router (unicast)	PIM SM
3	Join/prune	224.0.0.13 (all PIM routers)	PIM-SM, Bidir-PIM and SSM
4	Bootstrap	224.0.0.13 (all PIM routers)	PIM-SM and Bidir-PIM
5	Assert	224.0.0.13 (all PIM routers)	PIM-SM, PIM-DM, and Bidir-PIM
8	Candidate RP advertisement	Bootstrap router (BSR) address (unicast to BSR)	PIM-SM and Bidir-PIM
9	State refresh	224.0.0.13 (all PIM routers)	PIM-DM
10	DF election	224.0.0.13 (all PIM routers)	Bidir-PIM

PIM hello messages are sent by default every 30 seconds out each PIM-enabled interface to learn about the neighboring PIM routers on each interface to the *all PIM routers* address shown in Table 13-4. Hello messages are also the mechanism used to elect a designated router (DR), as described later in this chapter, and to negotiate additional capabilities. All PIM routers must record the hello information received from each PIM neighbor.

PIM Dense Mode

Key Topic

PIM routers can be configured for PIM Dense Mode (PIM-DM) when it is safe to assume that the receivers of a multicast group are located on every subnet within the network—in other words, when the multicast group is densely populated across the network.

For PIM-DM, the multicast tree is built by flooding traffic out every interface from the source to every Dense Mode router in the network. The tree is grown from the root toward the leaves. As each router receives traffic for the multicast group, it must decide whether it already has active receivers wanting to receive the multicast traffic. If so, the router remains quiet and lets the multicast flow continue. If no receivers have requested the multicast stream for the multicast group on the LHR, the router sends a prune message toward the source. That branch of the tree is then pruned off so that the unnecessary traffic does not continue. The resulting tree is a source tree because it is unique from the source to the receivers.

Figure 13-15 shows the flood and prune operation of Dense Mode. The multicast traffic from the source is flooding throughout the entire network. As each router receives the multicast traffic from its upstream neighbor via its RPF interface, it forwards the multicast traffic to all its PIM-DM neighbors. This results in some traffic arriving via a non-RPF interface, as in the case of R3 receiving traffic from R2 on its non-RPF interface. Packets arriving via the non-RPF interface are discarded.

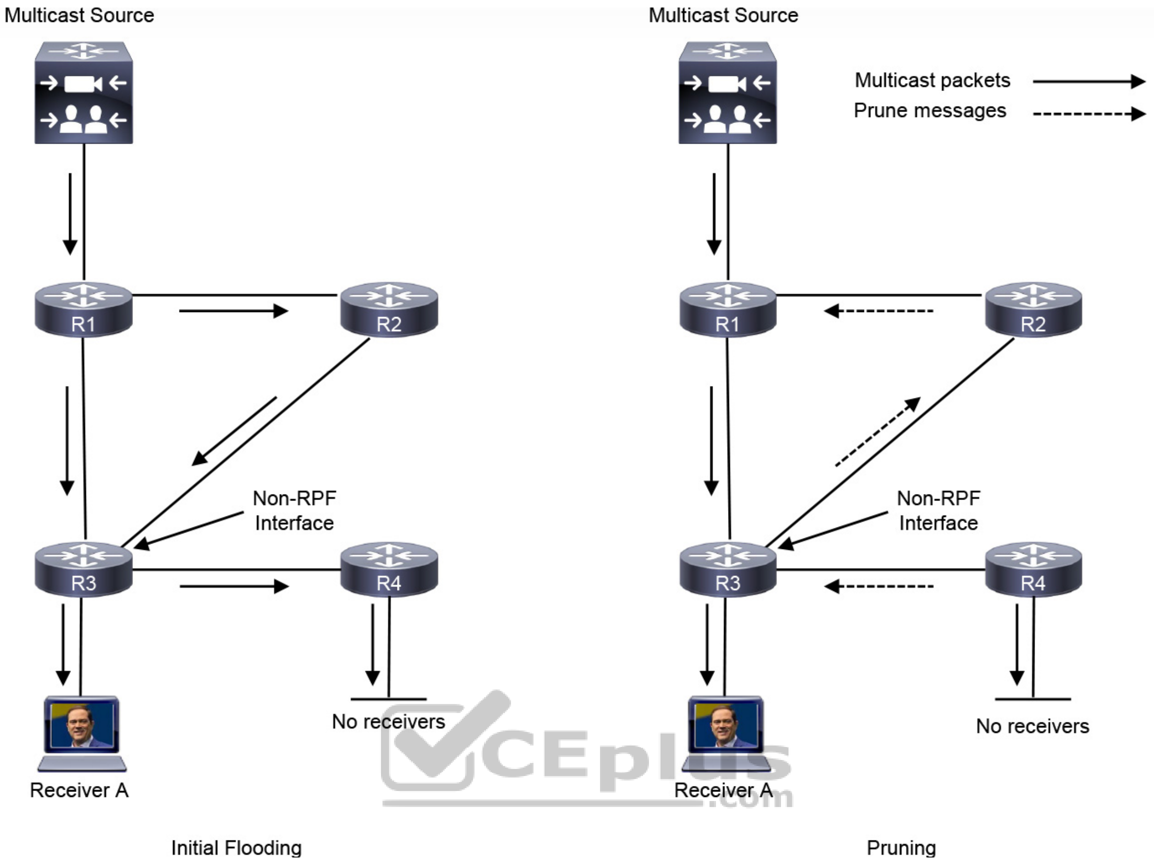
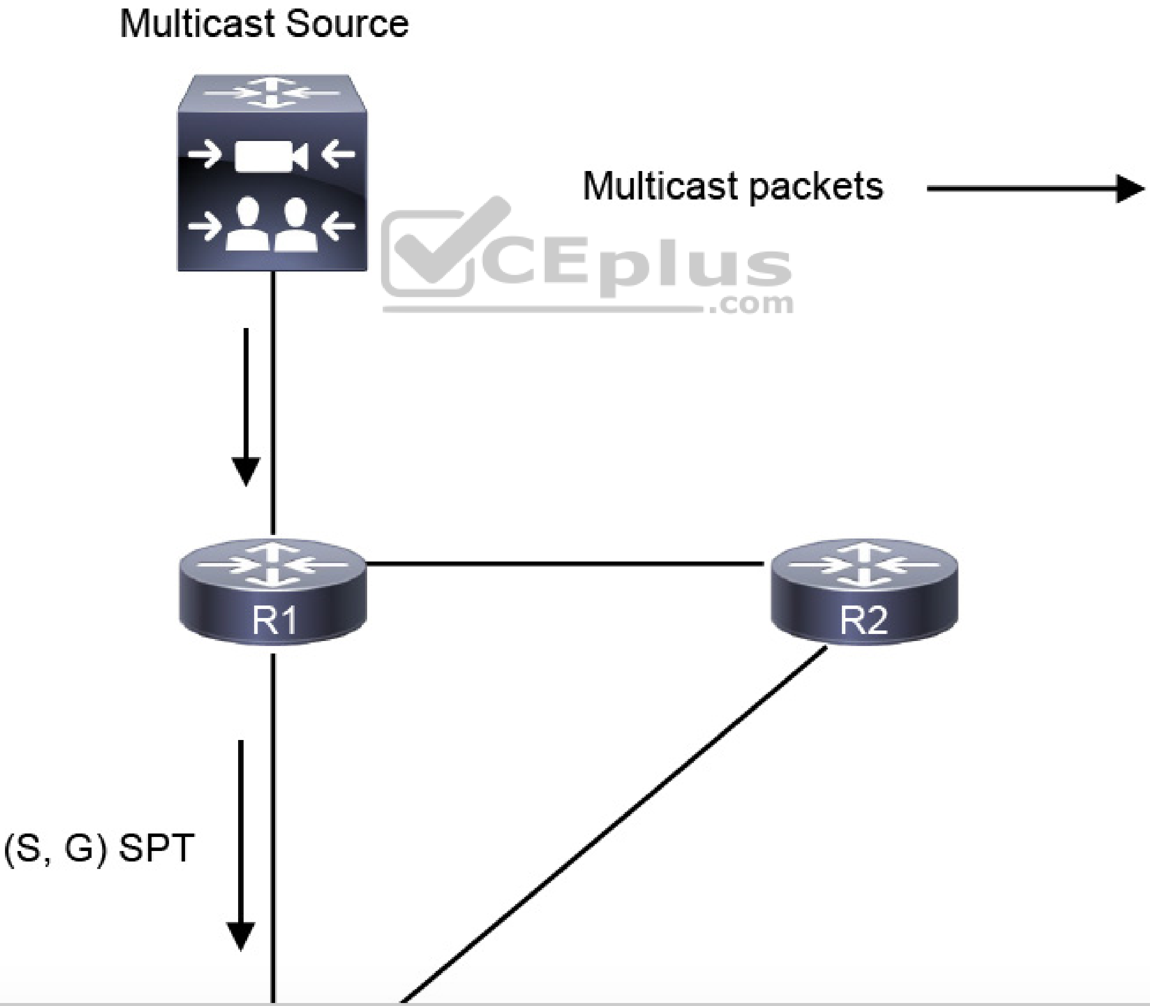


Figure 13-15 PIM-DM Flood and Prune Operation

These non-RPF multicast flows are normal for the initial flooding of multicast traffic and are corrected by the normal PIM-DM pruning mechanism. The pruning mechanism is used to stop the flow of unwanted traffic. Prunes (denoted by the dashed arrows) are sent out the RPF interface when the router has no downstream members that need the multicast traffic, as is the case for R4, which has no interested receivers, and they are also sent out non-RPF interfaces to stop the flow of multicast traffic that is arriving through the non-RPF interface, as is the case for R3, where multicast traffic is arriving through a non-RPF interface from R2, which results in a prune message.

Figure 13-16 illustrates the resulting topology after all unnecessary links have been pruned off. This results in an SPT from the source to the receiver. Even though the flow of multicast traffic is no longer reaching most of the routers in the network, the (S,G) state still remains in all routers in the network. This (S,G) state remains until the source stops transmitting.



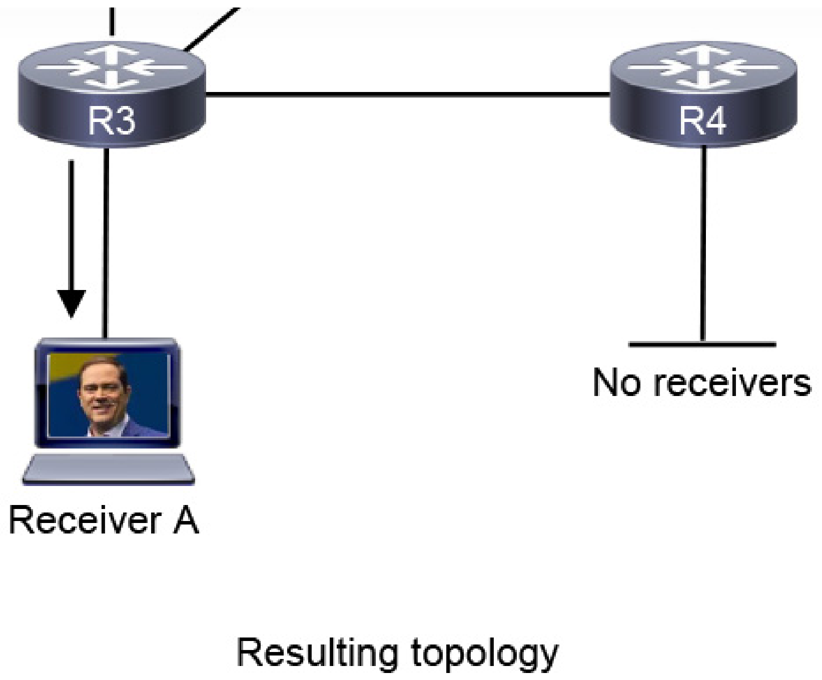


Figure 13-16 PIM-DM Resulting Topology After Pruning

In PIM-DM, prunes expire after three minutes. This causes the multicast traffic to be reflooded to all routers just as was done during the initial flooding. This periodic (every three minutes) flood and prune behavior is normal and must be taken into account when a network is designed to use PIM-DM.

PIM-DM is applicable to small networks where there are active receivers on every subnet of the network. Because this is rarely the case, PIM-DM is not generally recommended for production environments; however, it can be useful for a lab environment because it is easy to set up.

PIM Sparse Mode

Key Topic

PIM Sparse Mode (PIM-SM) was designed for networks with multicast application receivers scattered throughout the network—in other words, when the multicast group is sparsely populated across the network. However, PIM-SM also works well in densely populated networks. It also assumes that no receivers are interested in multicast traffic unless they explicitly request it.

Just like PIM-DM, PIM-SM uses the unicast routing table to perform RPF checks, and it does not care which routing protocol (including static routes) populates the unicast routing table; therefore, it is protocol independent.

PIM Shared and Source Path Trees

PIM-SM uses an explicit join model where the receivers send an IGMP join to their locally connected router, which is also known as the *last-hop router (LHR)*, and this join causes the LHR to send a PIM join in the direction of the root of the tree, which is either the RP in the case of a shared tree (RPT) or the first-hop router (FHR) where the source transmitting the multicast streams is connected in the case of an SPT.

A multicast forwarding state is created as the result of these explicit joins; it is very different from the flood and prune or implicit join behavior of PIM-DM, where the multicast packet arriving on the router dictates the forwarding state.

Figure 13-17 illustrates a multicast source sending multicast traffic to the FHR. The FHR then sends this multicast traffic to the RP, which makes the multicast source known to the RP. It also illustrates a receiver sending an IGMP join to the LHR to join the multicast group. The LHR then sends a PIM join (*,G) to the RP, and this forms a shared tree from the RP to the LHR. The RP then sends a PIM join (S,G) to the FHR, forming a source tree between the source and the RP. In essence, two trees are created: an SPT from the FHR to the RP (S,G) and a shared tree from the RP to the LHR (*,G).

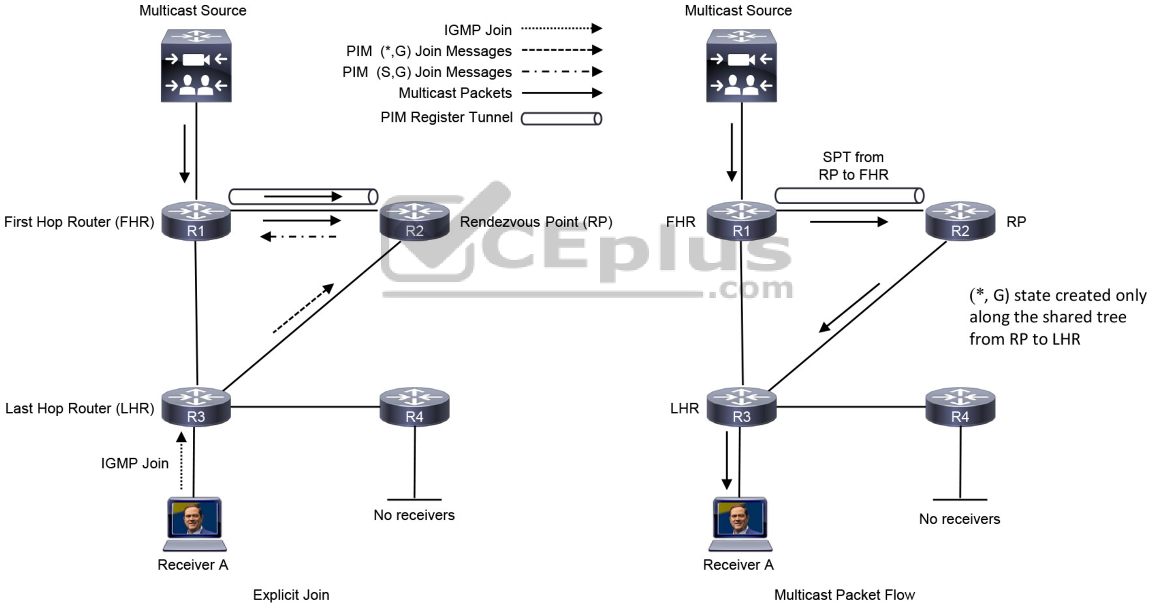


Figure 13-17 PIM-SM Multicast Distribution Tree Building

At this point, multicast starts flowing down from the source to the RP and from the RP to the LHR and then finally to the receiver. This is an oversimplified view of how PIM-SM achieves multicast forwarding. The following sections explain it in more detail.

Shared Tree Join

Key Topic

Figure 13-17 shows Receiver A attached to the LHR joining multicast group G. The LHR knows the IP address of the RP for group G, and it then sends a (*,G) PIM join for this group to the RP. If the RP were not directly connected, this (*,G) PIM join would travel hop-by-hop to the RP, building a branch of the shared tree that would extend from the RP to the LHR. At this point, group G multicast traffic arriving at the RP can flow down the shared tree to the receiver.

Source Registration

Key Topic

In Figure 13-17, as soon as the source for a group G sends a packet, the FHR that is attached to this source is responsible for registering this source with the RP and requesting the RP to build a tree back to that router.

The FHR encapsulates the multicast data from the source in a special PIM-SM message called the *register message* and unicasts that data to the RP using a unidirectional PIM tunnel.

When the RP receives the register message, it decapsulates the multicast data packet inside the register message, and if there is no active shared tree because there are no interested receivers, the RP sends a register stop message directly to the registering FHR, without traversing the PIM tunnel, instructing it to stop sending the register messages.

If there is an active shared tree for the group, it forwards the multicast packet down the shared tree, and it sends an (S,G) join back toward the source network S to create an (S,G) SPT. If there are multiple hops (routers) between the RP and the source, this results in an (S,G) state being created in all the routers along the SPT, including the RP. There will also be a (*,G) in R1 and all of the routers between the FHR and the RP.

As soon as the SPT is built from the source router to the RP, multicast traffic begins to flow natively from the source S to the RP.

Once the RP begins receiving data natively (that is, down the SPT) from source S, it sends a register stop message to the source’s FHR to inform it that it can stop sending the unicast register messages. At this point, multicast traffic from the source is flowing down the SPT to the RP and, from there, down the shared tree (RPT) to the receiver.

The PIM register tunnel from the FHR to the RP remains in an active up/up state even when there are no active multicast streams, and it remains active as long as there is a valid RPF path for the RP.

PIM SPT Switchover

Key Topic

PIM-SM allows the LHR to switch from the shared tree to an SPT for a specific source. In Cisco routers, this is the default behavior, and it happens immediately after the first multicast packet is received from the RP via the shared tree, even if the shortest path to the source is through the RP. Figure 13-18 illustrates the SPT switchover concept. When the LHR receives the first multicast packet from the RP, it becomes aware of the IP address of the multicast source. At this point, the LHR checks its unicast routing table to see which is the shortest path to the source, and it sends an (S,G) PIM join hop-by-hop to the FHR to form an SPT. Once it receives a multicast packet from the FHR through the SPT, if necessary, it switches the RPF interface to be the one in the direction of the SPT to the FHR, and it then sends a PIM prune message to the RP to shut off the duplicate multicast traffic coming from it through the shared tree. In Figure 13-18, the shortest path to the source is between R1 and R3, if that link were shut down or not present, the shortest path would be through the RP, in which case an SPT switchover would still take place.

Note

The PIM SPT switchover mechanism can be disabled for all groups or for specific groups.

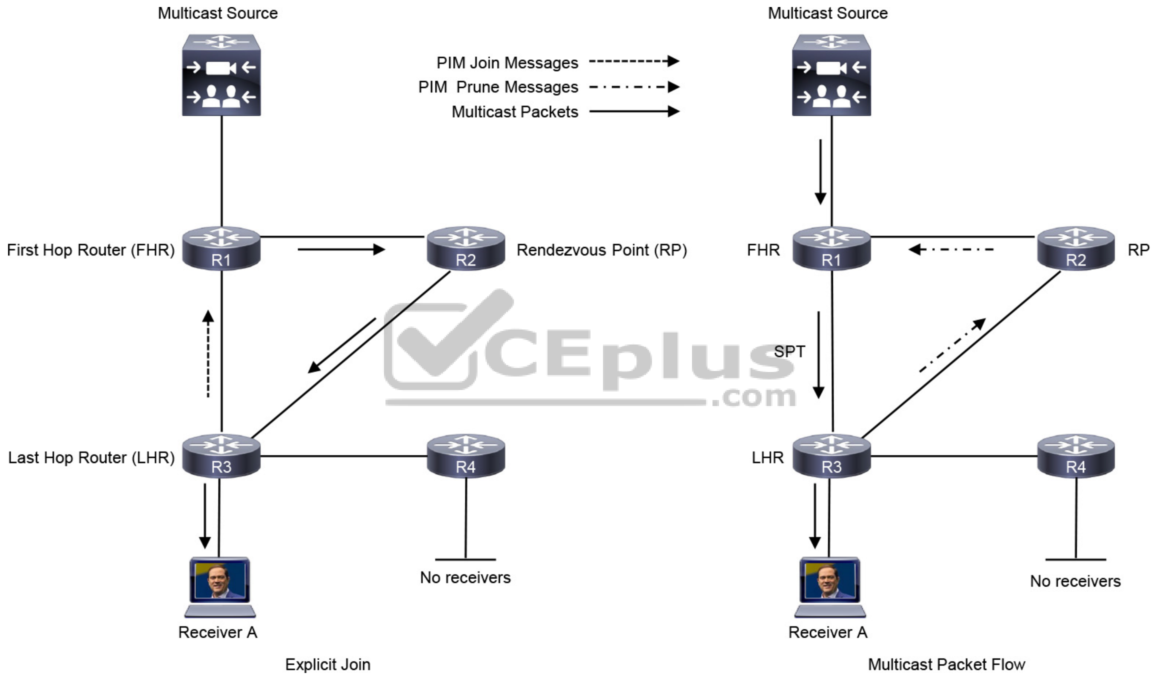


Figure 13-18 PIM-SM SPT Switchover Example

If the RP has no other interfaces that are interested in the multicast traffic, it sends a PIM prune message in the direction of the FHR. If there are any routers between the RP and the FHR, this prune message would travel hop-by-hop until it reaches the FHR.

Designated Routers

Key Topic

When multiple PIM-SM routers exist on a LAN segment, PIM hello messages are used to elect a designated router (DR) to avoid sending duplicate multicast traffic into the LAN or the RP. By default, the DR priority value of all PIM routers is 1, and it can be changed to force a particular router to become the DR during the DR election process, where a higher DR priority is preferred. If a router in the subnet does not support the DR priority option or if all routers have the same DR priority, the highest IP address in the subnet is used as a tiebreaker.

On an FHR, the designated router is responsible for encapsulating in unicast register messages any multicast packets originated by a source that are destined to the RP. On an LHR, the designated router is responsible for sending PIM join and prune messages toward the RP to inform it about host group membership, and it is also responsible for performing a PIM STP switchover.

Without DRs, all LHRs on the same LAN segment would be capable of sending PIM joins upstream, which could result in duplicate multicast traffic arriving on the LAN. On the source side, if multiple FHRs exist on the LAN, they all send register messages to the RP at the same time.

The default DR hold time is 3.5 times the hello interval, or 105 seconds. If there are no hellos after this interval, a new DR is elected. To reduce DR failover time, the hello query interval can be reduced to speed up failover with a trade-off of more control plane traffic and CPU resource utilization of the router.

Reverse Path Forwarding

Key Topic

Reverse Path Forwarding (RPF) is an algorithm used to prevent loops and ensure that multicast traffic is arriving on the correct interface. RPF functions as follows:

- If a router receives a multicast packet on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.

- If the packet arrives on the RPF interface, a router forwards the packet out the interfaces present in the outgoing interface list (OIL) of a multicast routing table entry.

- If the packet does not arrive on the RPF interface, the packet is discarded to prevent loops.

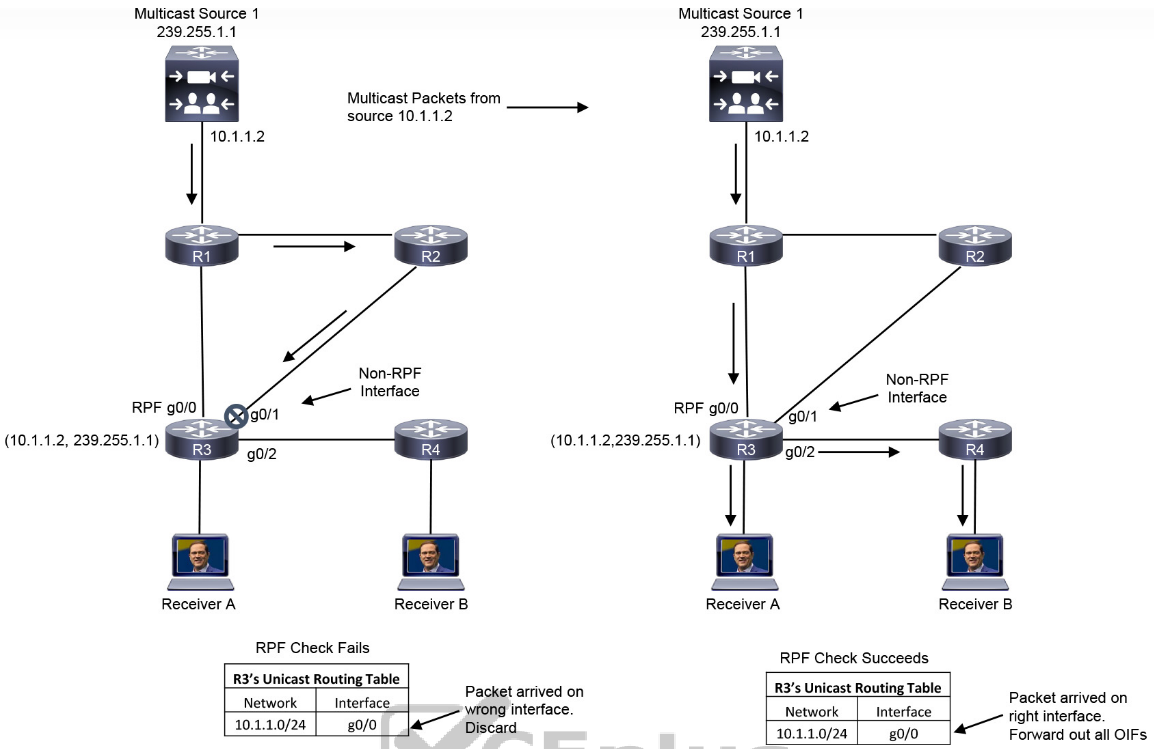
PIM uses multicast source trees between the source and the LHR and between the source and the RP. It also uses multicast shared trees between the RP and the LHRs. The RPF check is performed differently for each, as follows:

- If a PIM router has an (S,G) entry present in the multicast routing table (an SPT state), the router performs the RPF check against the IP address of the source for the multicast packet.

- If a PIM router has no explicit source-tree state, this is considered a shared-tree state. The router performs the RPF check on the address of the RP, which is known when members join the group.

PIM-SM uses the RPF lookup function to determine where it needs to send joins and prunes. (S,G) joins (which are SPT states) are sent toward the source. (*,G) joins (which are shared tree states) are sent toward the RP.

The topology on the left side of Figure 13-19 illustrates a failed RPF check on R3 for the (S,G) entry because the packet is arriving via a non-RPF interface. The topology on the right shows the multicast traffic arriving on the correct interface on R3; it is then forwarded out all the OIFs.



R3's Unicast Routing Table

Network	Interface
10.1.1.0/24	g0/0

R3's Unicast Routing Table

Network	Interface
10.1.1.0/24	g0/0

Figure 13-19 RPF Check

PIM Forwarder

There are certain scenarios in which duplicate multicast packets could flow onto a multi-access network. The PIM assert mechanism stops these duplicate flows.

Figure 13-20 illustrates R2 and R3 both receiving the same (S,G) traffic via their RPF interfaces and forwarding the packets on to the LAN segment. R2 and R3 therefore receive an (S,G) packet via their downstream OIF that is in the OIF of their (S,G) entry. In other words, they detect a multicast packet for a specific (S,G) coming into their OIF that is also going out the same OIF for the same (S,G). This triggers the assert mechanism.

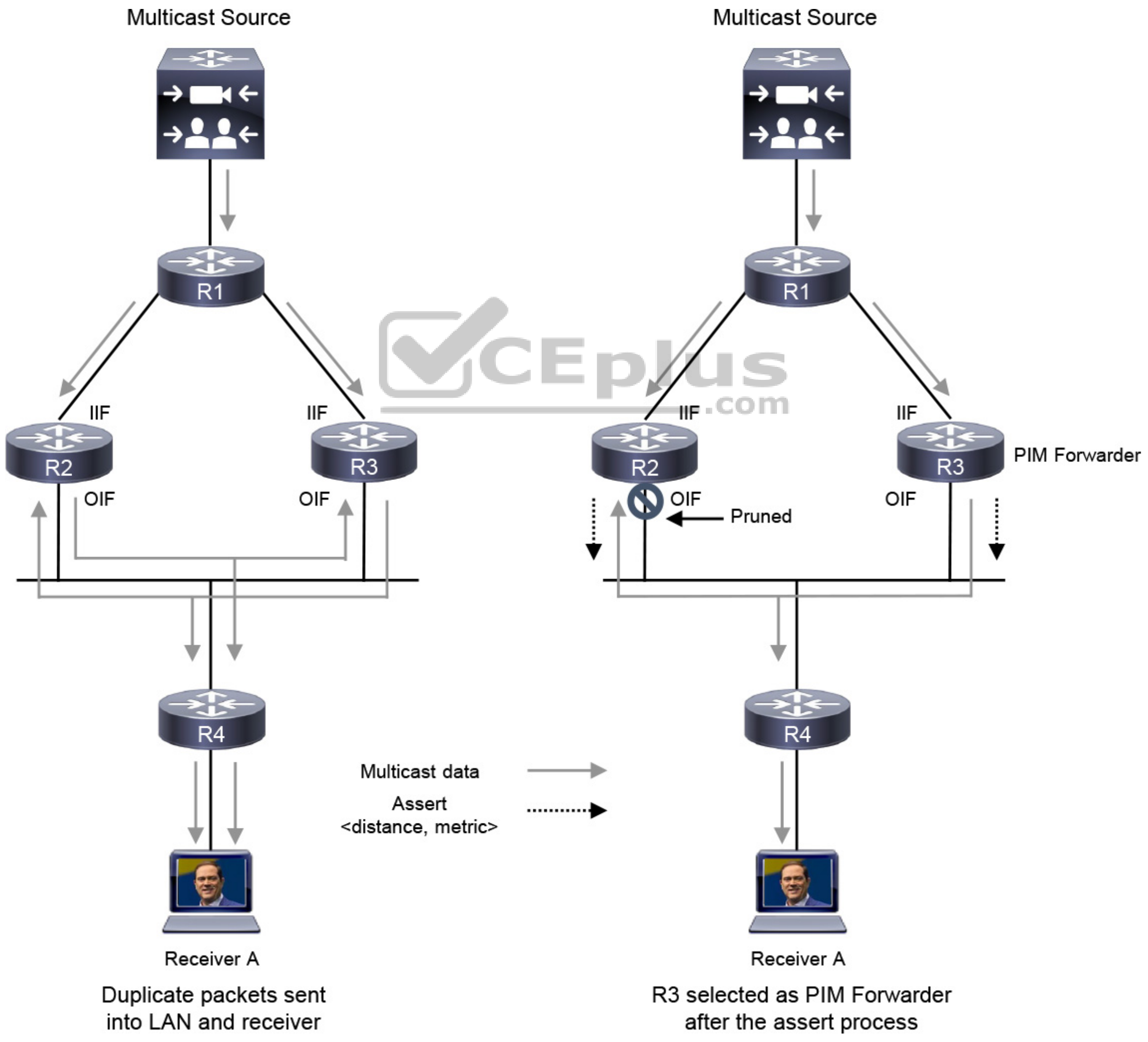


Figure 13-20 PIM Forwarder Example

R2 and R3 both send PIM assert messages into the LAN. These assert messages send their administrative distance (AD) and route metric back to the source to determine which router should forward the multicast traffic to that network segment.

Each router compares its own values with the received values. Preference is given to the PIM message with the lowest AD to the source. If a tie exists, the lowest route metric for the protocol wins; and as a final tiebreaker, the highest IP address is used.

The losing router prunes its interface just as if it had received a prune on this interface, and the winning router is the PIM forwarder for the LAN.

Note

The prune times out after three minutes on the losing router and causes it to begin forwarding on the interface again. This triggers the assert process to repeat. If the winning router were to go offline, the loser would take over the job of forwarding on to this LAN segment after its prune timed out.

The PIM forwarder concept applies to PIM-DM and PIM-SM. It is commonly used by PIM-DM but rarely used by PIM-SM because the only time duplicate packets can end up in a LAN is if there is some sort of routing inconsistency.

With the topology shown in Figure 13-20, PIM-SM would not send duplicate flows into the LAN as PIM-DM would because of the way PIM-SM operates. For example, assuming that R1 is the RP, when R4 sends a PIM join message upstream toward it, it sends it to the all PIM routers address 224.0.0.13, and R2 and R3 receive it. One of the fields of the PIM join message includes the IP address of the upstream neighbor, also known as the RPF neighbor. Assuming that R3 is the RPF neighbor, R3 is the only one that will send a PIM join to R1. R2 will not because the PIM join was not meant for it. At this point, a shared tree exists between R1, R3, and R4, and no traffic duplication exists.

Figure 13-21 illustrates how duplicate flows could exist in a LAN using PIM-SM. On the topology on the left side, R2 and R4 are running Open Shortest Path First (OSPF) Protocol, and R3 and R4 are running Enhanced Interior Gateway Routing Protocol (EIGRP). R4 learns about the RP (R1) through R2, and R5 learns about the RP through R3. R4’s RPF neighbor is R2, and R5’s RPF neighbor is R3. Assuming that Receiver A and Receiver B join the same group, R4 would send a PIM join to its upstream neighbor R2, which would in turn send a PIM join to R1. R5 would send a PIM join to its upstream neighbor R3, which would send a PIM join to R1. At this point, traffic starts flowing downstream from R1 into R2 and R3, and duplicate packets are then sent out into the LAN and to the receivers. At this point, the PIM assert mechanism kicks in, R3 is elected as the PIM forwarder, and R2’s OIF interface is pruned, as illustrated in the topology on the right side.

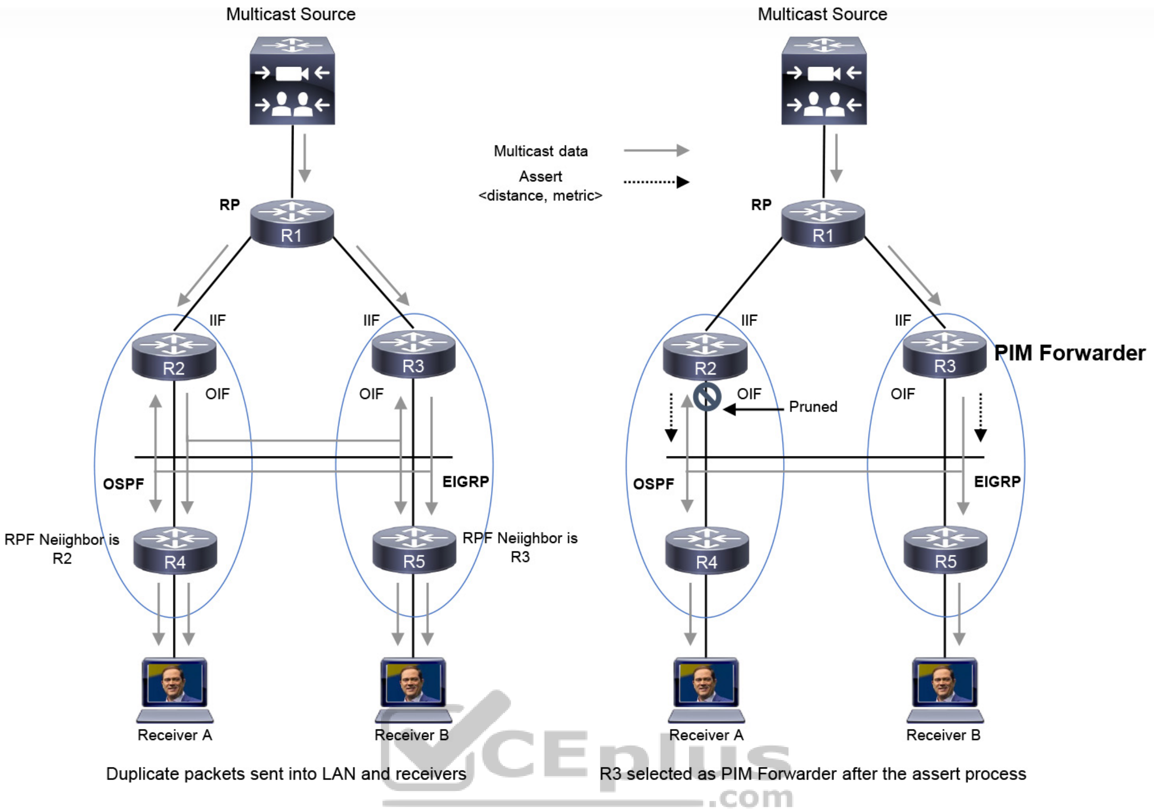


Figure 13-21 PIM-SM PIM Forwarder Example

RENDEZVOUS POINTS

Key Topic

In PIM-SM, it is mandatory to choose one or more routers to operate as *rendezvous points (RPs)*. An RP is a single common root placed at a chosen point of a shared distribution tree, as described earlier in this chapter. An RP can be either configured statically in each router or learned through a dynamic mechanism. A PIM router can be configured to function as an RP either statically in each router in the multicast domain or dynamically by configuring Auto-RP or a PIM bootstrap router (BSR), as described in the following sections.

Note

BSR and Auto-RP were not designed to work together and may introduce unnecessary complexities when deployed in the same network. The recommendation is not to use them concurrently.

Static RP

Key Topic

It is possible to statically configure RP for a multicast group range by configuring the address of the RP on every router in the multicast domain. Configuring static RPs is relatively simple and can be achieved with one or two lines of configuration on each router. If the network does not have many different RPs defined or if the RPs do not change very often, this could be the simplest method for defining RPs. It can also be an attractive option if the network is small.

However, static configuration can increase administrative overhead in a large and complex network. Every router must have the same RP address. This means changing the RP address requires reconfiguring every router. If several RPs are active for different groups, information about which RP is handling which multicast group must be known by all routers. To ensure this information is complete, multiple configuration commands may be required. If a manually configured RP fails, there is no failover procedure for another router to take over the function performed by the failed RP, and this method by itself does not provide any kind of load splitting.

Auto-RP

Key Topic

Auto-RP is a Cisco proprietary mechanism that automates the distribution of group-to-RP mappings in a PIM network. Auto-RP has the following benefits:

- It is easy to use multiple RPs within a network to serve different group ranges.

- It allows load splitting among different RPs.

- It simplifies RP placement according to the locations of group participants.

- It prevents inconsistent manual static RP configurations that might cause connectivity problems.

- Multiple RPs can be used to serve different group ranges or to serve as backups for each other.

- The Auto-RP mechanism operates using two basic components, candidate RPs (C-RPs) and RP mapping agents (MAs).

Candidate RPs

Key Topic

A C-RP advertises its willingness to be an RP via RP announcement messages. These messages are sent by default every RP announce interval, which is 60 seconds by default, to the reserved well-known multicast group 224.0.1.39 (Cisco-RP-Announce). The RP announcements contain the default group range 224.0.0.0/4, the C-RP's address, and the hold time, which is three times the RP announce interval. If there are multiple C-RPs, the C-RP with the highest IP address is preferred.

RP Mapping Agents

Key Topic

RP MAs join group 224.0.1.39 to receive the RP announcements. They store the information contained in the announcements in a group-to-RP mapping cache, along with hold times. If multiple RPs advertise the same group range, the C-RP with the highest IP address is elected.

The RP MAs advertise the RP mappings to another well-known multicast group address, 224.0.1.40 (Cisco-RP-Discovery). These messages are advertised by default every 60 seconds or when changes are detected. The MA announcements contain the elected RPs and the group-to-RP mappings. All PIM-enabled routers join 224.0.1.40 and store the RP mappings in their private cache.

Multiple RP MAs can be configured in the same network to provide redundancy in case of failure. There is no election mechanism between them, and they act independently of each other; they all advertise identical group-to-RP mapping information to all routers in the PIM domain.

Figure 13-22 illustrates the Auto-RP mechanism where the MA periodically receives the C-RP Cisco RP announcements to build a group-to-RP mapping cache and then periodically multicasts this information to all PIM routers in the network using Cisco RP discovery messages.

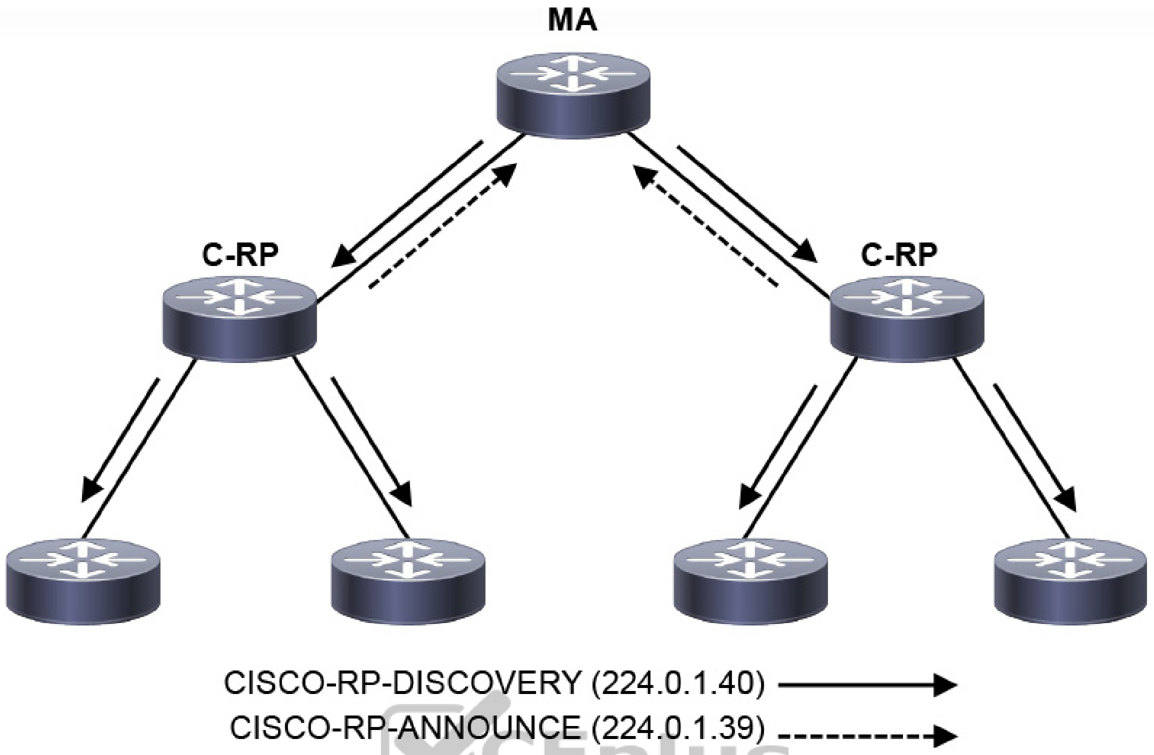

MA
C-RP
C-RP
CISCO-RP-DISCOVERY (224.0.1.40)
CISCO-RP-ANNOUNCE (224.0.1.39)

Figure 13-22 Auto-RP Mechanism

With Auto-RP, all routers automatically learn the RP information, which makes it easier to administer and update RP information. Auto-RP permits backup RPs to be configured, thus enabling an RP failover mechanism.

PIM Bootstrap Router

Key Topic

The *bootstrap router (BSR)* mechanism, described in RFC 5059, is a nonproprietary mechanism that provides a fault-tolerant, automated RP discovery and distribution mechanism.

PIM uses the BSR to discover and announce RP set information for each group prefix to all the routers in a PIM domain. This is the same function accomplished by Auto-RP, but the BSR is part of the PIM Version 2 specification. The RP set is a group-to-RP mapping that contains the following components:

- Multicast group range
- RP priority
- RP address
- Hash mask length
- SM/Bidir flag

Generally, BSR messages originate on the BSR, and they are flooded hop-by-hop by intermediate routers. When a bootstrap message is forwarded, it is forwarded out of every PIM-enabled interface that has PIM neighbors (including the one over which the message was received). BSR messages use the all PIM routers address 224.0.0.13 with a TTL of 1.

To avoid a single point of failure, multiple candidate BSRs (C-BSRs) can be deployed in a PIM domain. All C-BSRs participate in the BSR election process by sending PIM BSR messages containing their BSR priority out all interfaces.

The C-BSR with the highest priority is elected as the BSR and sends BSR messages to all PIM routers in the PIM domain. If the BSR priorities are equal or if the BSR priority is not configured, the C-BSR with the highest IP address is elected as the BSR.

Candidate RPs

Key Topic

A router that is configured as a candidate RP (C-RP) receives the BSR messages, which contain the IP address of the currently active BSR. Because it knows the IP address of the BSR, the C-RP can unicast candidate RP advertisement (C-RP-Adv) messages directly to it. A C-RP-Adv message carries a list of group address and group mask field pairs. This enables a C-RP to specify the group ranges for which it is willing to be the RP.

The active BSR stores all incoming C-RP advertisements in its group-to-RP mapping cache. The BSR then sends the entire list of C-RPs from its group-to-RP mapping cache in BSR messages every 60 seconds by default to all PIM routers in the entire network. As the routers receive copies of these BSR messages, they update the information in their local group-to-RP mapping caches, and this allows them to have full visibility into the IP addresses of all C-RPs in the network.

Unlike with Auto-RP, where the mapping agent elects the active RP for a group range and announces the election results to the network, the BSR does not elect the active RP for a group. Instead, it leaves this task to each individual router in the network.

Each router in the network uses a well-known hashing algorithm to elect the currently active RP for a particular group range. Because each router is running the same algorithm against the same list of C-RPs, they will all select the same RP for a particular group range. C-RPs with a lower priority are preferred. If the priorities are the same, the C-RP with the highest IP address is elected as the RP for the particular group range.

Figure 13-23 illustrates the BSR mechanism, where the elected BSR receives candidate RP advertisement messages from all candidate RPs in the domain, and it then sends BSR messages with RP set information out all PIM-enabled interfaces, which are flooded hop-by-hop to all routers in the network.

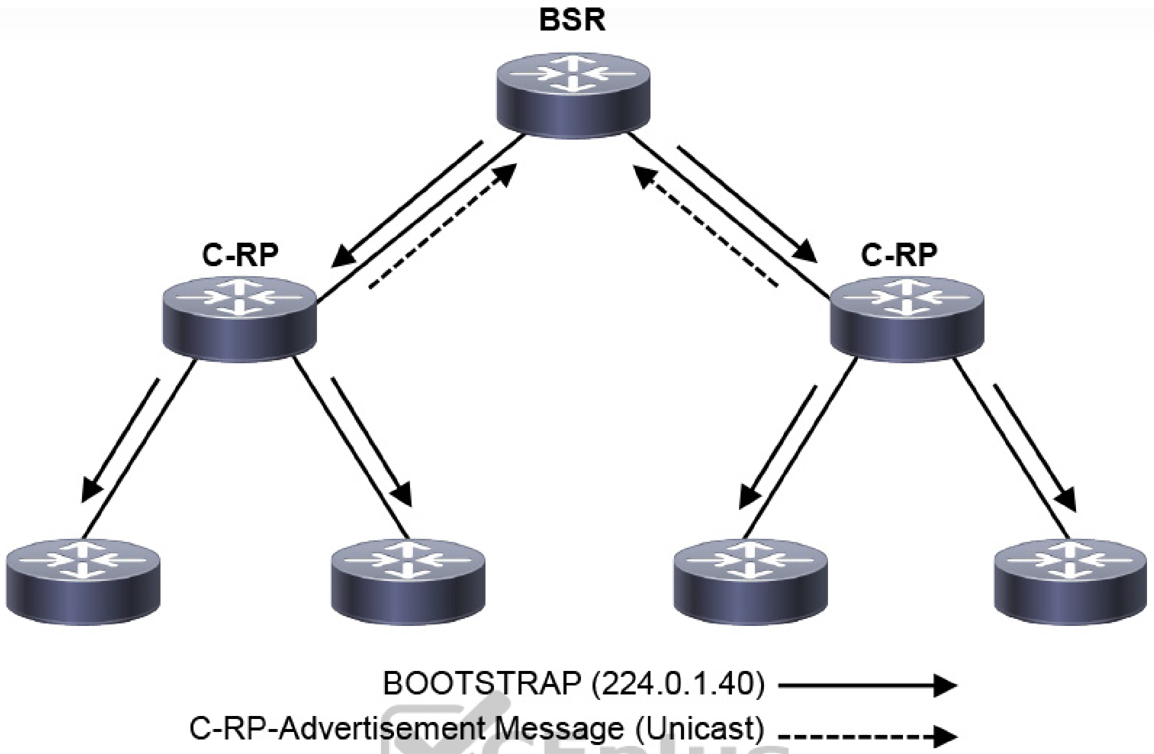


Figure 13-23 BSR Mechanism

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the key topics icon in the outer margin of the page. Table 13-5 lists these key topics and the page number on which each is found.

Key Topic

Table 13-5 Key Topics for Chapter 13

Key Topic Element	Description	Page
Paragraph	Multicast fundamentals	
Table 13-2	IP Multicast Addresses Assigned by IANA	
Table 13-3	Well-Known Reserved Multicast Addresses	
Section	Layer 2 multicast addressing	
Paragraph	IGMP description	
Paragraph	IGMPv2 definition	
List	IGMP message format field definitions	
Paragraph	IGMPv2 operation	
Paragraph	IGMPv3 definition	
Paragraph	IGMP snooping definition	
Paragraph	PIM definition	
Paragraph	PIM source tree definition	
Paragraph	PIM shared tree definition	
List	PIM terminology	
List	PIM operating modes	
Table 13-4	PIM control message types	

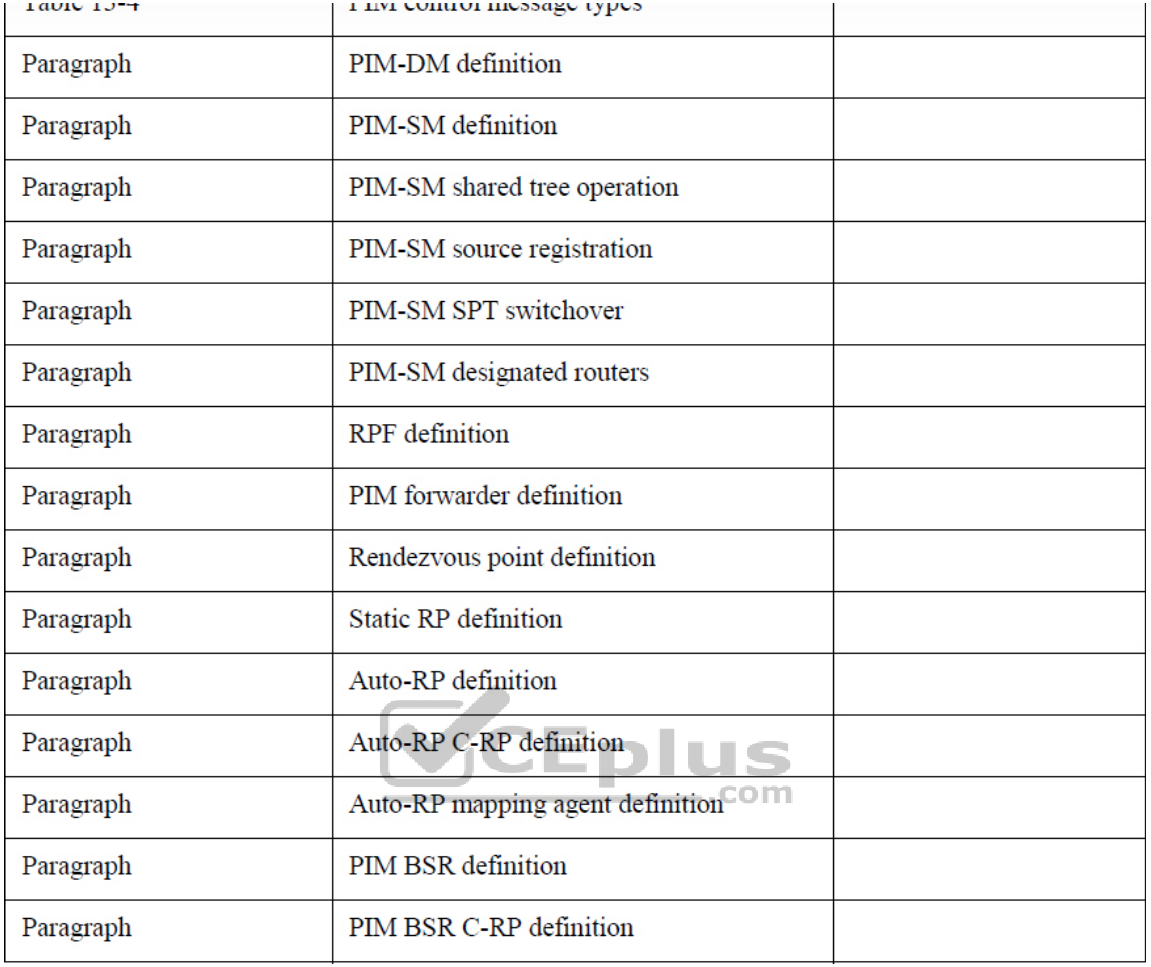


Table 13-4	PIM control message types	
Paragraph	PIM-DM definition	
Paragraph	PIM-SM definition	
Paragraph	PIM-SM shared tree operation	
Paragraph	PIM-SM source registration	
Paragraph	PIM-SM SPT switchover	
Paragraph	PIM-SM designated routers	
Paragraph	RPF definition	
Paragraph	PIM forwarder definition	
Paragraph	Rendezvous point definition	
Paragraph	Static RP definition	
Paragraph	Auto-RP definition	
Paragraph	Auto-RP C-RP definition	
Paragraph	Auto-RP mapping agent definition	
Paragraph	PIM BSR definition	
Paragraph	PIM BSR C-RP definition	

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix B, “Memory Tables” (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix C, “Memory Tables Answer Key,” also on the companion website, includes completed tables and lists you can use to check your work.

DEFINE KEY TERMS

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

designated router (DR)

downstream

downstream interface

first-hop router (FHR)

incoming interface (IIF)

Internet Group Management Protocol (IGMP)

IGMP snooping

last-hop router (LHR)

Multicast Forwarding Information Base (MFIB)

Multicast Routing Information Base (MRIB)

multicast state

outgoing interface (OIF)

outgoing interface list (OIL)

Protocol Independent Multicast (PIM)

rendezvous point (RP)

rendezvous point tree (RPT)

Reverse Path Forwarding (RPF) interface

RPF neighbor

shortest path tree (SPT)

upstream

upstream interface

REFERENCES IN THIS CHAPTER

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

Part IV: Services

Chapter 14. QoS

This chapter covers the following subjects:

- **The Need for QoS:** This section describes the leading causes of poor quality of service and how they can be alleviated by using QoS tools and mechanisms.

- **QoS Models:** This section describes the three different models available for implementing QoS in a network: best effort, Integrated Services (IntServ), and Differentiated Services (DiffServ).

- **Classification and Marking:** This section describes classification, which is used to identify and assign IP traffic into different traffic classes, and marking, which is used to mark packets with a specified priority based on classification or traffic conditioning policies.

- **Policing and Shaping:** This section describes how policing is used to enforce rate limiting, where excess IP traffic is either dropped, marked, or delayed.

- **Congestion Management and Avoidance:** This section describes congestion management, which is a queueing mechanism used to prioritize and protect IP traffic. It also describes congestion avoidance, which involves discarding IP traffic to avoid network congestion.

QoS is a network infrastructure technology that relies on a set of tools and mechanisms to assign different levels of priority to different IP traffic flows and provides special treatment to higher-priority IP traffic flows. For higher-priority IP traffic flows, it reduces packet loss during times of network congestion and also helps control delay (latency) and delay variation (jitter); for low-priority IP traffic flows, it provides a best-effort delivery service. This is analogous to how a high-occupancy vehicle (HOV) lane, also referred to as a carpool lane, works: A special high-priority lane is reserved for use of carpools (high-priority traffic), and those who carpool can flow freely by bypassing the heavy traffic congestion in the adjacent general-purpose lanes.

These are the primary goals of implementing QoS on a network:

- Expediting delivery for real-time applications

- Ensuring business continuance for business-critical applications

- Providing fairness for non-business-critical applications when congestion occurs

- Establishing a trust boundary across the network edge to either accept or reject traffic markings injected by the endpoints

QoS uses the following tools and mechanisms to achieve its goals:

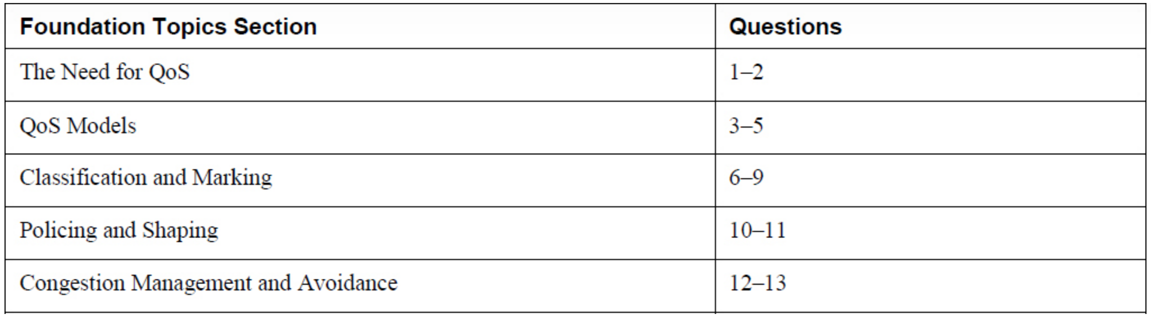
- Classification and marking
- Policing and shaping
- Congestion management and avoidance

All of these QoS mechanisms are described in this chapter.

"DO I KNOW THIS ALREADY?" QUIZ

The "Do I Know This Already?" quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the "Exam Preparation Tasks" section. Table 14-1 lists the major headings in this chapter and the "Do I Know This Already?" quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the "Do I Know This Already?" quiz appear in Appendix A, "Answers to the 'Do I Know This Already?' Quiz Questions."

Table 14-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping



Foundation Topics Section	Questions
The Need for QoS	1–2
QoS Models	3–5
Classification and Marking	6–9
Policing and Shaping	10–11
Congestion Management and Avoidance	12–13

1. Which of the following are the leading causes of quality of service issues? (Choose all that apply.)

a. Bad hardware

b. Lack of bandwidth

c. Latency and Jitter

d. Copper cables

e. Packet loss

2. Network latency can be broken down into which of the following types? (Choose all that apply.)

a. Propagation delay (fixed)

b. Time delay (variable)

c. Serialization delay (fixed)

d. Processing delay (fixed)

e. Packet delay (fixed)

f. Delay variation (variable)

3. Which of the following is *not* a QoS implementation model?

a. IntServ

b. Expedited forwarding

c. Best effort

d. DiffServ

4. Which of the following is the QoS implementation model that requires a signaling protocol?

a. IntServ

b. Best Effort

c. DiffServ

d. RSVP

5. Which of the following is the most popular QoS implementation model?

a. IntServ

b. Best effort

c. DiffServ

d. RSVP

6. True or false: Traffic classification should always be performed in the core of the network.

a. True

b. False

7. The 16-bit TCI field is composed of which fields? (Choose three.)

a. Priority Code Point (PCP)

b. Canonical Format Identifier (CFI)

c. User Priority (PRI)

d. Drop Eligible Indicator (DEI)

e. VLAN Identifier (VLAN ID)

8. True or false: The DiffServ field is an 8-bit Differentiated Services Code Point (DSCP) field that allows for classification of up to 64 values (0 to 63).

a. True

b. False

9. Which of the following is *not* a QoS PHB?

a. Best Effort (BE)

b. Class Selector (CS)

c. Default Forwarding (DF)

d. Assured Forwarding (AF)

e. Expedited Forwarding (EF)

10. Which traffic conditioning tool can be used to drop or mark down traffic that goes beyond a desired traffic rate?

a. Policers

b. Shapers

c. WRR

d. None of the above

11. What does Tc stand for? (Choose two.)

a. Committed time interval

b. Token credits

c. Bc bucket token count

d. Traffic control

12. Which of the following are the recommended congestion management mechanisms for modern rich-media networks? (Choose two.)

a. Class-based weighted fair queuing (CBWFQ)

b. Priority queuing (PQ)

c. Weighted RED (WRED)

d. Low-latency queuing (LLQ)

13. Which of the following is a recommended congestion-avoidance mechanism for modern rich-media networks?

a. Weighted RED (WRED)

b. Tail drop

c. FIFO

d. RED

Answers to the “Do I Know This Already?” quiz:

1. B, C, E

2. A, C, D, F

3. B

4. A

5. C

6. B

7. A, D, E

8. B

9. A

10. A

11. A, C

12. A, D

13. A

FOUNDATION TOPICS

THE NEED FOR QOS

Modern real-time multimedia applications such as IP telephony, telepresence, broadcast video, Cisco Webex, and IP video surveillance are extremely sensitive to delivery delays and create unique quality of service (QoS) demands on a network. When packets are delivered using a best-effort delivery model, they may not arrive in order or in a timely manner, and they may be dropped. For video, this can result in pixelization of the image, pausing, choppy video, audio and video being out of sync, or no video at all. For audio, it could cause echo, talker overlap (a walkie-talkie effect where only one person can speak at a time), unintelligible and distorted speech, voice breakups, longs silence gaps, and call drops. The following are the leading causes of quality issues:

- Lack of bandwidth
- Latency and jitter
- Packet loss

Lack of Bandwidth

The available bandwidth on the data path from a source to a destination equals the capacity of the lowest-bandwidth link. When the maximum capacity of the lowest-bandwidth link is surpassed, link congestion takes place, resulting in traffic drops. The obvious solution to this type of problem is to increase the link bandwidth capacity, but this is not always possible, due to budgetary or technological constraints. Another option is to implement QoS mechanisms such as policing and queueing to prioritize traffic according to level of importance. Voice, video, and business-critical traffic should get prioritized forwarding and sufficient bandwidth to support their application requirements, and the least important traffic should be allocated the remaining bandwidth.

Latency and Jitter

One-way end-to-end delay, also referred to as *network latency*, is the time it takes for packets to travel across a network from a source to a destination. ITU Recommendation G.114 recommends that, regardless of the application type, a network latency of 400 ms should not be exceeded, and for real-time traffic, network latency should be less than 150 ms; however, ITU and Cisco have demonstrated that real-time traffic quality does not begin to significantly degrade until network latency exceeds 200 ms. To be able to implement these recommendations, it is important to understand what causes network latency. Network latency can be broken down into fixed and variable latency:

- Propagation delay (fixed)

- Serialization delay (fixed)

- Processing delay (fixed)

- Delay variation (variable)

Propagation Delay

Propagation delay is the time it takes for a packet to travel from the source to a destination at the speed of light over a medium such as fiber-optic cables or copper wires. The speed of light is 299,792,458 meters per second in a vacuum. The lack of vacuum conditions in a fiber-optic cable or a copper wire slows down the speed of light by a ratio known as the *refractive index*; the larger the refractive index value, the slower light travels.

The average refractive index value of an optical fiber is about 1.5. The speed of light through a medium v is equal to the speed of light in a vacuum c divided by the refractive index n, or $v = c / n$. This means the speed of light through a fiber-optic cable with a refractive index of 1.5 is approximately 200,000,000 meters per second (that is, 300,000,000 / 1.5).

If a single fiber-optic cable with a refractive index of 1.5 were laid out around the equatorial circumference of Earth, which is about 40,075 km, the propagation delay would be equal to the equatorial circumference of Earth divided by 200,000,000 meters per second. This is approximately 200 ms, which would be an acceptable value even for real-time traffic.

Keep in mind that optical fibers are not always physically placed over the shortest path between two points. Fiber-optic cables may be hundreds or even thousands of miles longer than expected. In addition, other components required by fiber-optic cables, such as repeaters and amplifiers, may introduce additional delay. A provider’s service-level agreement (SLA) can be reviewed to estimate and plan for the minimum, maximum, and average latency for a circuit.

> **Note**
>
> Sometimes it is necessary to use satellite communication for hard-to-reach locations. The propagation delay for satellite circuits is the time it takes a radio wave traveling at the speed of light from the Earth’s surface to a satellite (which could mean multiple satellite hops) and back to the Earth’s surface; depending on the number of hops, this may surpass the recommended maximum 400 ms. For cases like this, there is nothing that can be done to reduce the delay other than to try to find a satellite provider that offers lower propagation delays.

Serialization Delay

Serialization delay is the time it takes to place all the bits of a packet onto a link. It is a fixed value that depends on the link speed; the higher the link speed, the lower the delay. The serialization delay s is equal to the packet size in bits divided by the line speed in bits per second. For example, the serialization delay for a 1500-byte packet over a 1 Gbps interface is 12 μs and can be calculated as follows:

s = packet size in bits / line speed in bps

s = (1500 bytes × 8) / 1 Gbps

s = 12,000 bits / 1000,000,000 bps = 0.000012 s × 1000 = .012 ms × 1000 = 12 µs

Processing Delay

Processing delay is the fixed amount of time it takes for a networking device to take the packet from an input interface and place the packet onto the output queue of the output interface. The processing delay depends on factors such as the following:

- CPU speed (for software-based platforms)

- CPU utilization (load)

- IP packet switching mode (process switching, software CEF, or hardware CEF)

- Router architecture (centralized or distributed)

- Configured features on both input and output interfaces

Delay Variation

Delay variation, also referred to as *jitter*, is the difference in the latency between packets in a single flow. For example, if one packet takes 50 ms to traverse the network from the source to destination, and the following packet takes 70 ms, the jitter is 20 ms. The major factors affecting variable delays are queuing delay, dejitter buffers, and variable packet sizes.

Jitter is experienced due to the queueing delay experienced by packets during periods of network congestion. Queuing delay depends on the number and sizes of packets already in the queue, the link speed, and the queuing mechanism. Queuing introduces unequal delays for packets of the same flow, thus producing jitter.

Voice and video endpoints typically come equipped with de-jitter buffers that can help smooth out changes in packet arrival times due to jitter. A de-jitter buffer is often dynamic and can adjust for approximately 30 ms changes in arrival times of packets. If a packet is not received within the 30 ms window allowed for by the de-jitter buffer, the packet is dropped, and this affects the overall voice or video quality.

To prevent jitter for high-priority real-time traffic, it is recommended to use queuing mechanisms such as low-latency queueing (LLQ) that allow matching packets to be forwarded prior to any other low priority traffic during periods of network congestion.

Packet Loss

Packet loss is usually a result of congestion on an interface. Packet loss can be prevented by implementing one of the following approaches:

• Increase link speed

• Implement QoS congestion-avoidance and congestion-management mechanism

- Implement traffic policing to drop low-priority packets and allow high-priority traffic through.

- Implement traffic shaping to delay packets instead of dropping them since traffic may burst and exceed the capacity of an interface buffer. Traffic shaping is not recommended for real-time traffic because it relies on queuing that can cause jitter.

Note

Standard traffic shaping is unable to handle data bursts that occur on a microsecond time interval (that is, micro-bursts). Microsecond or low-burst shaping is required for cases where micro-bursts need to be smoothed out by a shaper.

QOS MODELS

Key Topic

There are three different QoS implementation models:

- **Best effort:** QoS is not enabled for this model. It is used for traffic that does not require any special treatment.

- **Integrated Services (IntServ):** Applications signal the network to make a bandwidth reservation and to indicate that they require special QoS treatment.

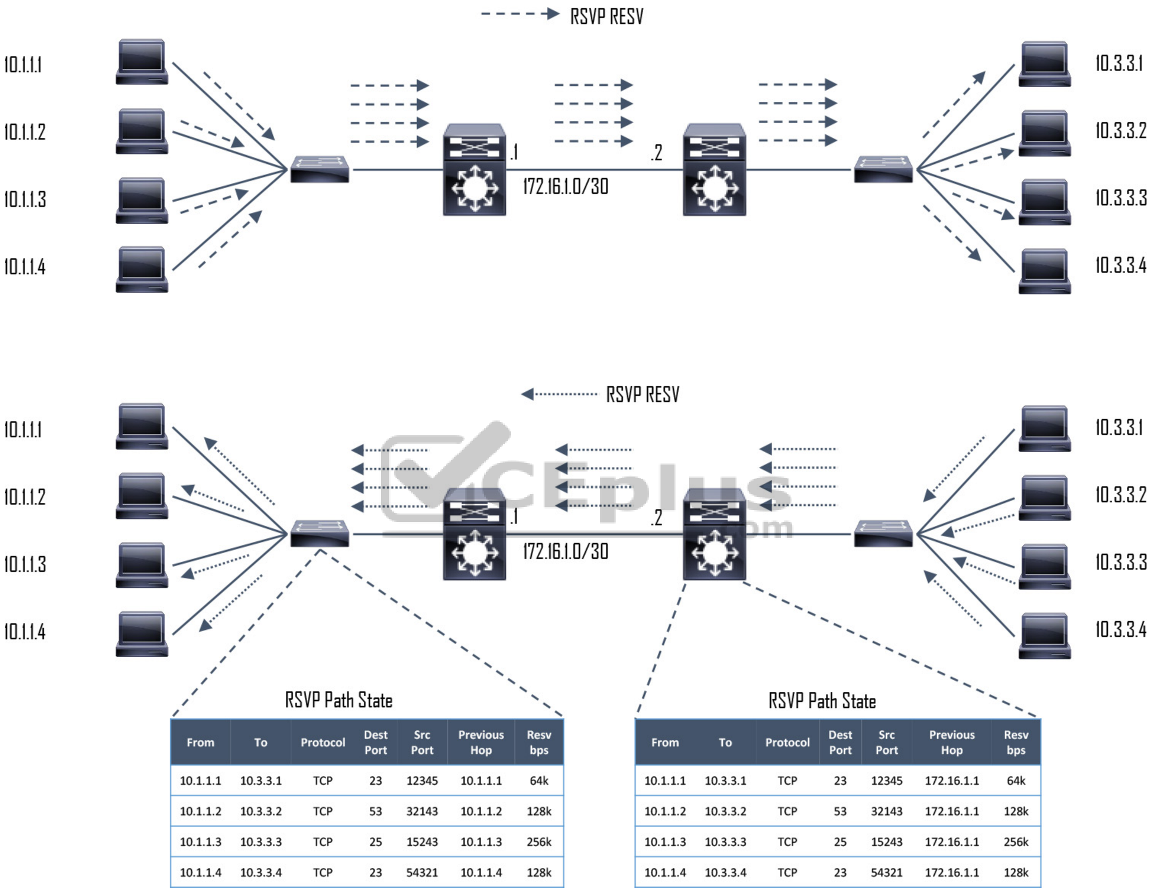
- **Differentiated Services (DiffServ):** The network identifies classes that require special QoS treatment.

Key Topic

The IntServ model was created for real-time applications such as voice and video that require bandwidth, delay, and packet-loss guarantees to ensure both predictable and guaranteed service levels. In this model, applications signal their requirements to the network to reserve the end-to-end resources (such as bandwidth) they require to provide an acceptable user experience. IntServ uses Resource Reservation Protocol (RSVP) to reserve resources throughout a network for a specific application and to provide call admission control (CAC) to guarantee that no other IP traffic can use the reserved bandwidth. The bandwidth reserved by an application that is not being used is wasted.

To be able to provide end-to-end QoS, all nodes, including the endpoints running the applications, need to support, build, and maintain RSVP path state for every single flow. This is the biggest drawback of IntServ because it means it cannot scale well on large networks that might have thousands or millions of flows due to the large number of RSVP flows that would need to be maintained.

Figure 14-1 illustrates how RSVP hosts issue bandwidth reservations.



RSVP Path State

From	To	Protocol	Dest Port	Src Port	Previous Hop	Resv bps
10.1.1.1	10.3.3.1	TCP	23	12345	10.1.1.1	64k
10.1.1.2	10.3.3.2	TCP	53	32143	10.1.1.2	128k
10.1.1.3	10.3.3.3	TCP	25	15243	10.1.1.3	256k
10.1.1.4	10.3.3.4	TCP	23	54321	10.1.1.4	128k

RSVP Path State

From	To	Protocol	Dest Port	Src Port	Previous Hop	Resv bps
10.1.1.1	10.3.3.1	TCP	23	12345	172.16.1.1	64k
10.1.1.2	10.3.3.2	TCP	53	32143	172.16.1.1	128k
10.1.1.3	10.3.3.3	TCP	25	15243	172.16.1.1	256k
10.1.1.4	10.3.3.4	TCP	23	54321	172.16.1.1	128k

Figure 14-1 RSVP Reservation Establishment

In Figure 14-1, each of the hosts on the left side (senders) are attempting to establish a one-to-one bandwidth reservation to each of the hosts on the right side (receivers). The senders start by sending RSVP PATH messages to the receivers along the same path used by regular data packets. RSVP PATH messages carry the receiver source address, the destination address, and the bandwidth they wish to reserve. This information is stored in the RSVP path state of each node. Once the RSVP PATH messages reach the receivers, each receiver sends RSVP reservation request (RESV) messages in the reverse path of the data flow toward the receivers, hop-by-hop. At each hop, the IP destination address of a RESV message is the IP address of the previous-hop node, obtained from the RSVP path state of each node. As RSVP RESV messages cross each hop, they reserve bandwidth on each of the links for the traffic flowing from the receiver hosts to the sender hosts. If bandwidth reservations are required from the hosts on the right side to the hosts on the left side, the hosts on the right side need to follow the same procedure of sending RSVP PATH messages, which doubles the RSVP state on each networking device in the data path. This demonstrates how RSVP state can increase quickly as more hosts reserve bandwidth. Apart from the scalability issues, long distances between hosts could also trigger long bandwidth reservation delays.

Key Topic

DiffServ was designed to address the limitations of the best-effort and IntServ models. With this model, there is no need for a signaling protocol, and there is no RSVP flow state to maintain on every single node, which makes it highly scalable; QoS characteristics (such as bandwidth and delay) are managed on a hop-by-hop basis with QoS policies that are defined independently at each device in the network. DiffServ is not considered an end-to-end QoS solution because end-to-end QoS guarantees cannot be enforced.

DiffServ divides IP traffic into classes and marks it based on business requirements so that each of the classes can be assigned a different level of service. As IP traffic traverses a network, each of the network devices identifies the packet class by its marking and services the packets according to this class. Many levels of service can be chosen with DiffServ. For example, IP phone voice traffic is very sensitive to latency and jitter, so it should always be given preferential treatment over all other application traffic. Email, on the other hand, can withstand a great deal of delay and could be given best-effort service and non-business, non-critical scavenger traffic (such as from YouTube) can either be heavily rate limited or blocked entirely. The DiffServ model is the most popular and most widely deployed QoS model and is covered in detail in this chapter.

CLASSIFICATION AND MARKING

Before any QoS mechanism can be applied, IP traffic must first be identified and categorized into different classes, based on business requirements. Network devices use classification to identify IP traffic as belonging to a specific class. After the IP traffic is classified, marking can be used to mark or color individual packets so that other network devices can apply QoS mechanisms to those packets as they traverse the network. This section introduces the concepts of classification and marking, explains the different marking options that are available for Layer 2 frames and Layer 3 packets, and explains where classification and marking tools should be used in a network.

Key Topic

Classification

Packet classification is a QoS mechanism responsible for distinguishing between different traffic streams. It uses traffic descriptors to categorize an IP packet within a specific class. Packet classification should take place at the network edge, as close to the source of the traffic as possible. Once an IP packet is classified, packets can then be marked/re-marked, queued, policed, shaped, or any combination of these and other actions.

Key Topic

The following traffic descriptors are typically used for classification:

- **Internal:** QoS groups (locally significant to a router)
- **Layer 1:** Physical interface, subinterface, or port
- **Layer 2:** MAC address and 802.1Q/p Class of Service (CoS) bits
- **Layer 2.5:** MPLS Experimental (EXP) bits

- **Layer 3:** Differentiated Services Code Points (DSCP), IP Precedence (IPP), and source/destination IP address

- **Layer 4:** TCP or UDP ports

- **Layer 7:** Next Generation Network-Based Application Recognition (NBAR2)

For enterprise networks, the most commonly used traffic descriptors used for classification include the Layer 2, Layer 3, Layer 4, and Layer 7 traffic descriptors listed here. The following section explores the Layer 7 traffic descriptor NBAR2.

Layer 7 Classification

Key Topic

NBAR2 is a deep packet inspection engine that can classify and identify a wide variety of protocols and applications using Layer 3 to Layer 7 data, including difficult-to-classify applications that dynamically assign Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers.

NBAR2 can recognize more than 1000 applications, and monthly protocol packs are provided for recognition of new and emerging applications, without requiring an IOS upgrade or router reload.

NBAR2 has two modes of operation:

- **Protocol Discovery:** Protocol Discovery enables NBAR2 to discover and get real-time statistics on applications currently running in the network. These statistics from the Protocol Discovery mode can be used to define QoS classes and policies using MQC configuration

- **Modular QoS CLI (MQC):** Using MQC, network traffic matching a specific network protocol such as Cisco Webex can be placed into one traffic class, while traffic that matches a different network protocol such as YouTube can be placed into another traffic class. After traffic has been classified in this way, different QoS policies can be applied to the different classes of traffic.

Key Topic

Marking

Packet *marking* is a QoS mechanism that colors a packet by changing a field within a packet or a frame header with a traffic descriptor so it is distinguished from other packets during the application of other QoS mechanisms (such as re-marking, policing, queuing, or congestion avoidance).

Key Topic

The following traffic descriptors are used for marking traffic:

- **Internal:** QoS groups
- **Layer 2:** 802.1Q/p Class of Service (CoS) bits
- **Layer 2.5:** MPLS Experimental (EXP) bits
- **Layer 3:** Differentiated Services Code Points (DSCP) and IP Precedence (IPP)

Note

QoS groups are used to mark packets as they are received and processed internally within the router and are automatically removed when packets egress the router. They are used only in special cases in which traffic descriptors marked or received on an ingress interface would not be visible for packet classification on egress interfaces due to encapsulation or de-encapsulation.

For enterprise networks, the most commonly used traffic descriptors for marking traffic include the Layer 2 and Layer 3 traffic descriptors mentioned in the previous list. Both of them are described in the following sections.

Layer 2 Marking

Key Topic

The 802.1Q standard is an IEEE specification for implementing VLANs in Layer 2 switched networks. The 802.1Q specification defines two 2-byte fields Tag Protocol Identifier (TPID) and Tag Control Information (TCI), which are inserted within an Ethernet frame following the Source Address field, as illustrated in Figure 14-2.

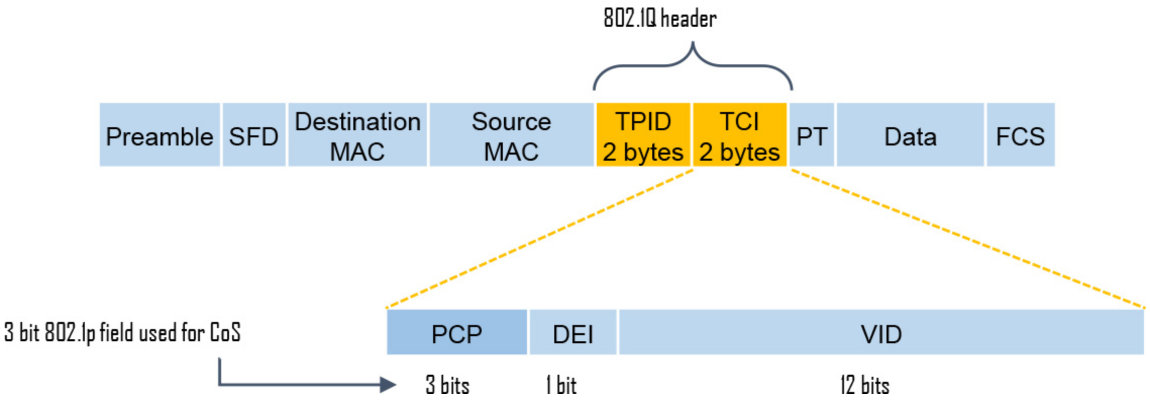


Figure 14-2 802.1Q Layer 2 QoS Using 802.1p CoS

The TPID value is a 16-bit field assigned the value 0x8100 that identifies it as an 802.1Q tagged frame.

Key Topic

The TCI field is a 16-bit field composed of the following three fields:

- Priority Code Point (PCP) field (3 bits)

- Drop Eligible Indicator (DEI) field (1 bit)

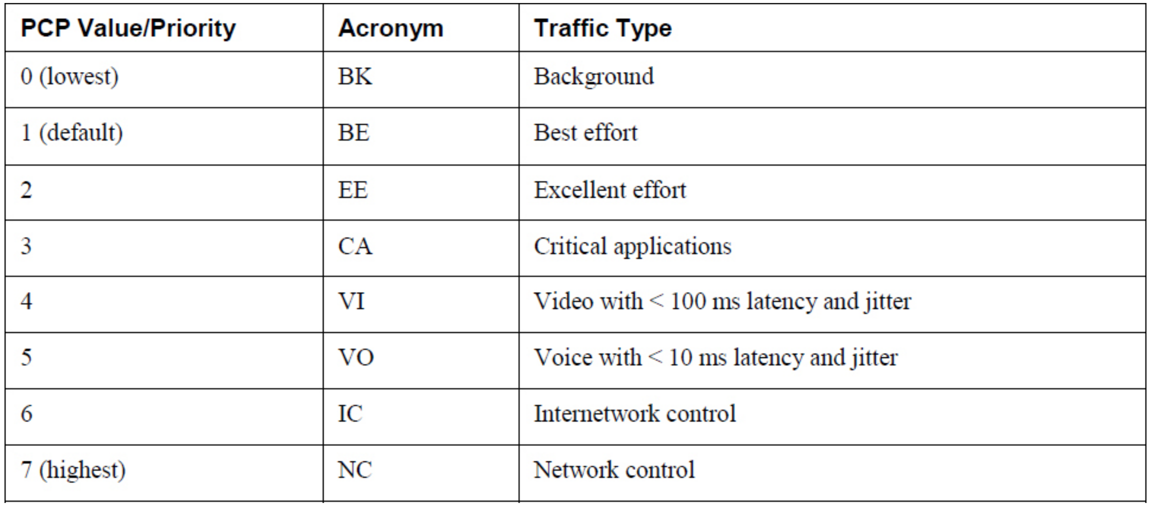
- VLAN Identifier (VLAN ID) field (12 bits)

Key Topic

Priority Code Point (PCP)

The specifications of the 3-bit PCP field are defined by the IEEE 802.1p specification. This field is used to mark packets as belonging to a specific CoS. The CoS marking allows a Layer 2 Ethernet frame to be marked with eight different levels of priority values, 0 to 7, where 0 is the lowest priority and 7 is the highest. Table 14-2 includes the IEEE 802.1p specification standard definition for each CoS.

Table 14-2 IEEE 802.1p CoS Definitions



PCP Value/Priority	Acronym	Traffic Type
0 (lowest)	BK	Background
1 (default)	BE	Best effort
2	EE	Excellent effort
3	CA	Critical applications
4	VI	Video with < 100 ms latency and jitter
5	VO	Voice with < 10 ms latency and jitter
6	IC	Internetwork control
7 (highest)	NC	Network control

One drawback of using CoS markings is that frames lose their CoS markings when traversing a non-802.1Q link or a Layer 3 network. For this reason, packets should be marked with other higher-layer markings whenever possible so the marking values can be preserved end-to-end. This is typically accomplished by mapping a CoS marking into another marking. For example, the CoS priority levels correspond directly to IPv4's IP Precedence Type of Service (ToS) values so they can be mapped directly to each other.

Drop Eligible Indicator (DEI)

The DEI field is a 1-bit field that can be used independently or in conjunction with PCP to indicate frames that are eligible to be dropped during times of congestion. The default value for this field is 0, and it indicates that this frame is not drop eligible; it can be set to 1 to indicate that the frame is drop eligible.

VLAN Identifier (VLAN ID)

The VLAN ID field is a 12-bit field that defines the VLAN used by 802.1Q. Since this field is 12 bits, it restricts the number of VLANs supported by 802.1Q to 4096, which may not be sufficient for large enterprise or service provider networks.

Layer 3 Marking

As a packet travels from its source to its destination, it might traverse non-802.1q trunked, or non-Ethernet links that do not support the CoS field. Using marking at Layer 3 provides a more persistent marker that is preserved end-to-end.

Figure 14-3 illustrates the ToS/DiffServ field within an IPv4 header.

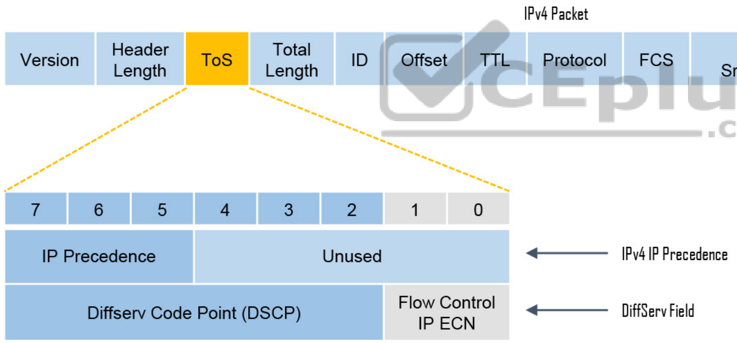


Figure 14-3 IPv4 ToS/DiffServ Field

Key Topic

The ToS field is an 8-bit field where only the first 3 bits of the ToS field, referred to as *IP Precedence (IPP)*, are used for marking, and the rest of the bits are unused. IPP values, which range from 0 to 7, allow the traffic to be partitioned in up to six usable classes of service; IPP 6 and 7 are reserved for internal network use.

Newer standards have redefined the IPv4 ToS and the IPv6 Traffic Class fields as an 8-bit Differentiated Services (DiffServ) field. The DiffServ field uses the same 8 bits that were previously used for the IPv4 ToS and the IPv6 Traffic Class fields, and this allows it to be backward compatible with IP Precedence. The DiffServ field is composed of a 6-bit Differentiated Services Code Point (DSCP) field that allows for classification of up to 64 values (0 to 63) and a 2-bit Explicit Congestion Notification (ECN) field.

DSCP Per-Hop Behaviors

Packets are classified and marked to receive a particular per-hop forwarding behavior (that is, expedited, delayed, or dropped) on network nodes along their path to the destination. The DiffServ field is used to mark packets according to their classification into DiffServ Behavior Aggregates (BAs). A DiffServ BA is a collection of packets with the same DiffServ value crossing a link in a particular direction. *Per-hop behavior (PHB)* is the externally observable forwarding behavior (forwarding treatment) applied at a DiffServ-compliant node to a collection of packets with the same DiffServ value crossing a link in a particular direction (DiffServ BA).

In other words, PHB is expediting, delaying, or dropping a collection of packets by one or multiple QoS mechanisms on a per-hop basis, based on the DSCP value. A DiffServ BA could be multiple applications—for example, SSH, Telnet, and SNMP all aggregated together and marked with the same DSCP value. This way, the core of the network performs only simple PHB, based on DiffServ BAs, while the network edge performs classification, marking, policing, and shaping operations. This makes the DiffServ QoS model very scalable.

Key Topic

Four PHBs have been defined and characterized for general use:

- **Class Selector (CS) PHB:** The first 3 bits of the DSCP field are used as CS bits. The CS bits make DSCP backward compatible with IP Precedence because IP Precedence uses the same 3 bits to determine class.

- **Default Forwarding (DF) PHB:** Used for best-effort service.

- **Assured Forwarding (AF) PHB:** Used for guaranteed bandwidth service.

- **Expedited Forwarding (EF) PHB:** Used for low-delay service.

Class Selector (CS) PHB

RFC 2474 made the ToS field obsolete by introducing the DiffServ field, and the Class Selector (CS) PHB was defined to provide backward compatibility for DSCP with IP Precedence. Figure 14-4 illustrates the CS PHB.

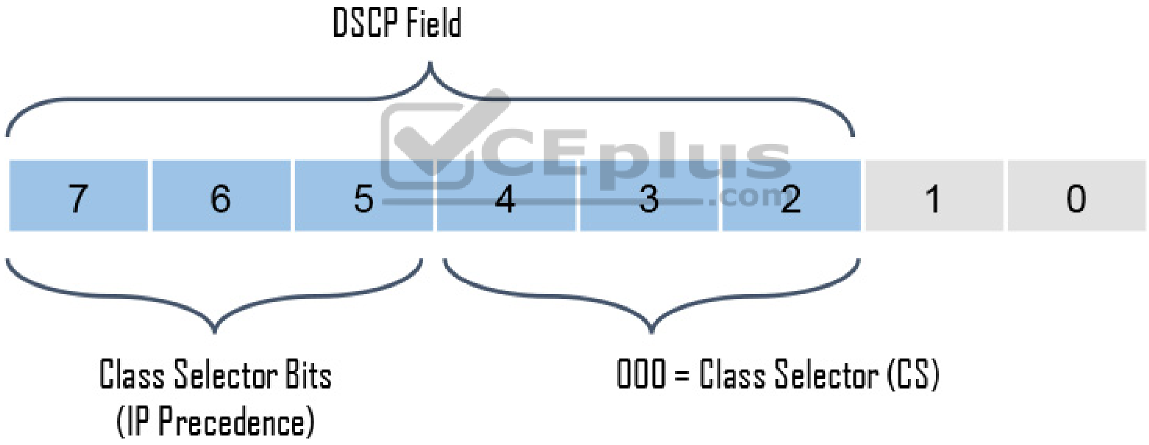


Figure 14-4 Class Selector (CS) PHB

Packets with higher IP Precedence should be forwarded in less time than packets with lower IP Precedence.

The last 3 bits of the DSCP (bits 2 to 4), when set to 0, identify a Class Selector PHB, but the Class Selector bits 5 to 7 are the ones where IP Precedence is set.

Bits 2 to 4 are ignored by non-DiffServ-compliant devices performing classification based on IP Precedence.

There are eight CS classes, ranging from CS0 to CS7, that correspond directly with the eight IP Precedence values.

Default Forwarding (DF) PHB

Default Forwarding (DF) and Class Selector 0 (CS0) provide best-effort behavior and use the DS value 000000. Figure 14-5 illustrates the DF PHB.

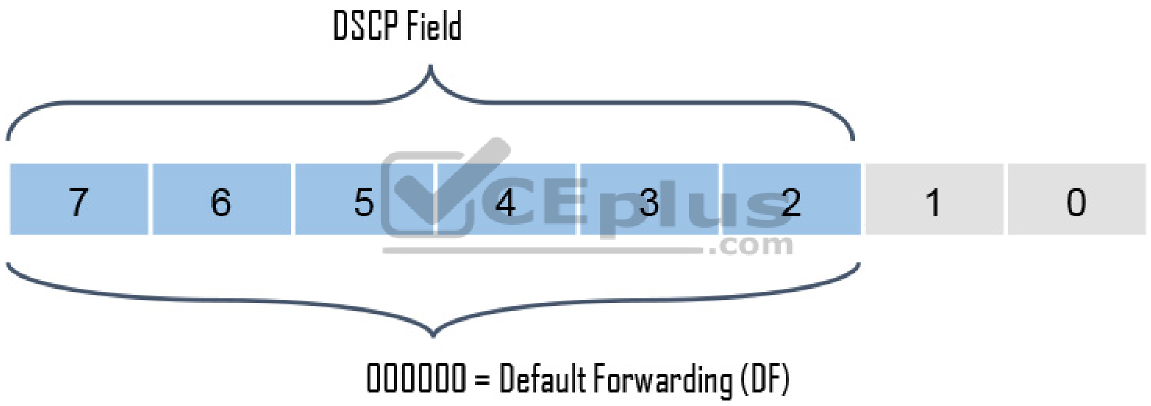
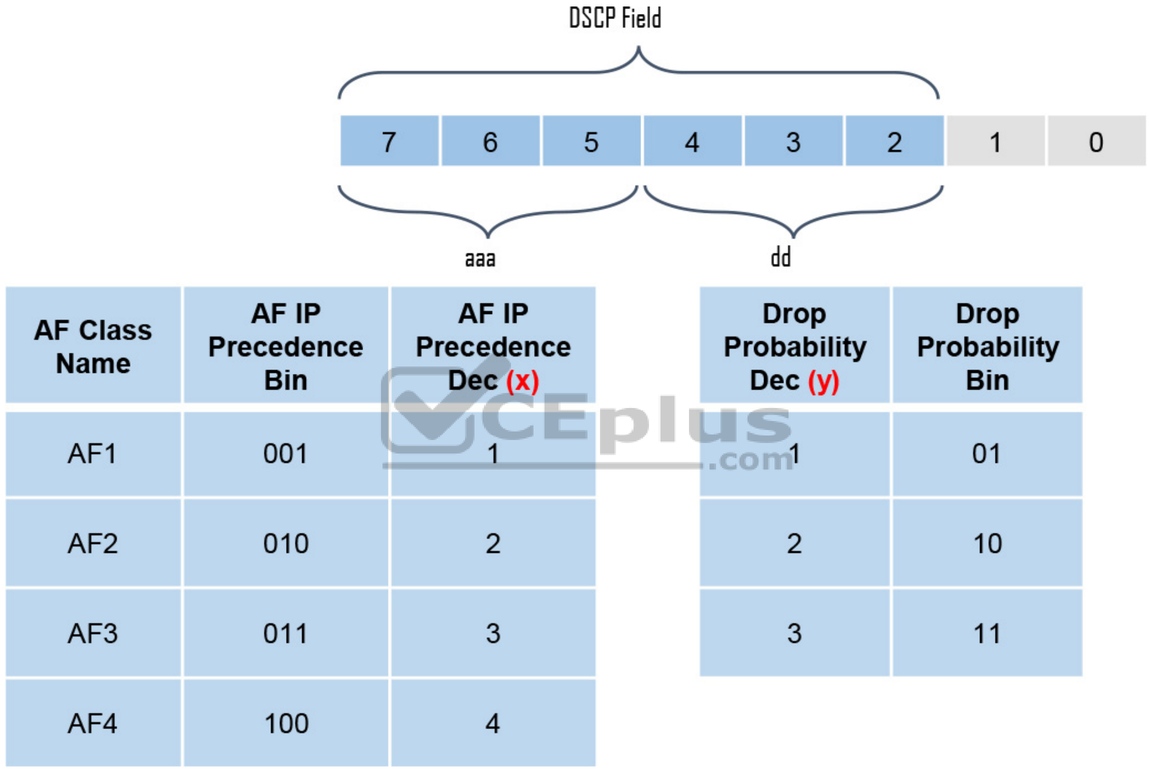


Figure 14-5 Default Forwarding (DF) PHB

Default best-effort forwarding is also applied to packets that cannot be classified by a QoS mechanism such as queueing, shaping, or policing. This usually happens when a QoS policy on the node is incomplete or when DSCP values are outside the ones that have been defined for the CS, AF, and EF PHBs.

Assured Forwarding (AF) PHB

The AF PHB guarantees a certain amount of bandwidth to an AF class and allows access to extra bandwidth, if available. Packets requiring AF PHB should be marked with DSCP value aaadd0, where aaa is the binary value of the AF class (bits 5 to 7), and dd (bits 2 to 4) is the drop probability and bit 2 is unused and always set to 0. Figure 14-6 illustrates the AF PHB.



AF Class Name	AF IP Precedence Bin	AF IP Precedence Dec (x)
AF1	001	1
AF2	010	2
AF3	011	3
AF4	100	4

Drop Probability Dec (y)	Drop Probability Bin
1	01
2	10
3	11

Figure 14-6 Assured Forwarding (AF) PHB

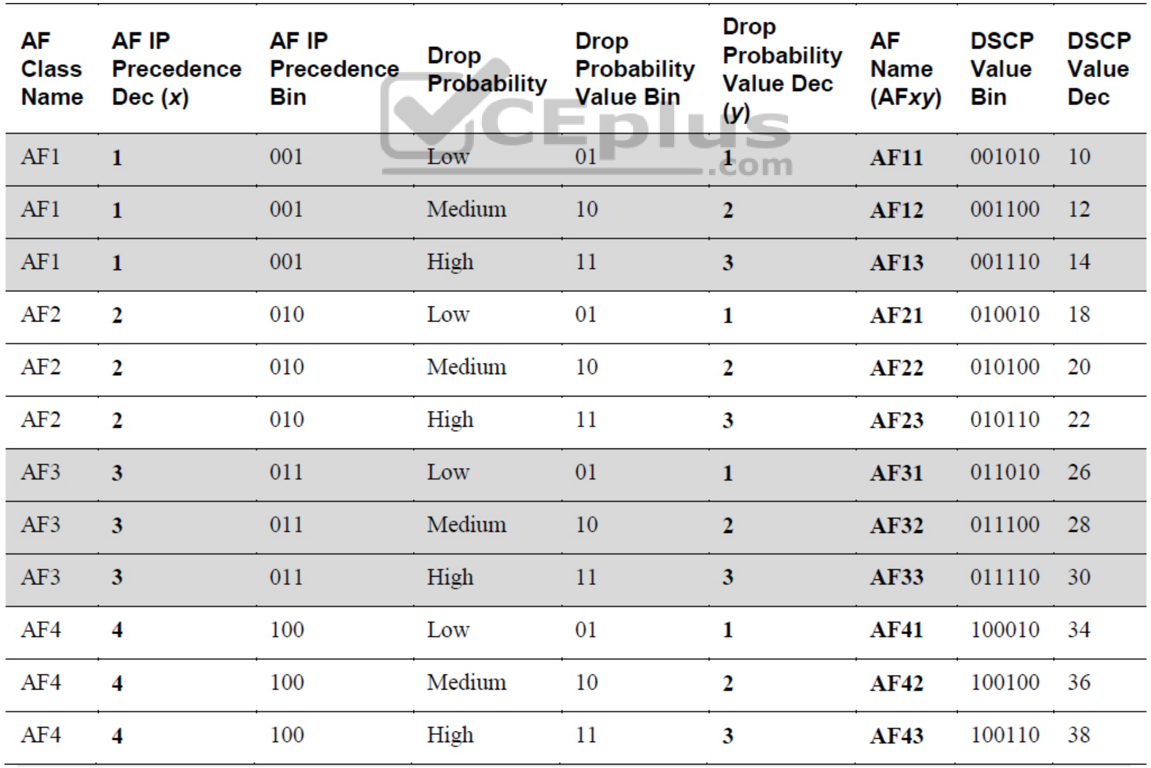
There are four standard-defined AF classes: AF1, AF2, AF3, and AF4. The AF class number does not represent precedence; for example, AF4 does not get any preferential treatment over AF1. Each class should be treated independently and placed into different queues.

Table 14-3 illustrates how each AF class is assigned an IP Precedence (under AF Class Value Bin) and has three drop probabilities low, medium, and high.

The AF Name (AF*xy*) is composed of the AF IP Precedence value in decimal (*x*) and the Drop Probability value in decimal (*y*). For example, AF41 is a combination of IP Precedence 4 and Drop Probability 1.

To quickly convert the AF Name into a DSCP value in decimal, use the formula 8*x* + 2*y*. For example, the DSCP value for AF41 is 8(4) + 2(1) = 34.

Table 14-3 AF PHBs with Decimal and Binary Equivalents



AF Class Name	AF IP Precedence Dec (*x*)	AF IP Precedence Bin	Drop Probability	Drop Probability Value Bin	Drop Probability Value Dec (*y*)	AF Name (AF*xy*)	DSCP Value Bin	DSCP Value Dec
AF1	**1**	001	Low	01	**1**	**AF11**	001010	10
AF1	**1**	001	Medium	10	**2**	**AF12**	001100	12
AF1	**1**	001	High	11	**3**	**AF13**	001110	14
AF2	**2**	010	Low	01	**1**	**AF21**	010010	18
AF2	**2**	010	Medium	10	**2**	**AF22**	010100	20
AF2	**2**	010	High	11	**3**	**AF23**	010110	22
AF3	**3**	011	Low	01	**1**	**AF31**	011010	26
AF3	**3**	011	Medium	10	**2**	**AF32**	011100	28
AF3	**3**	011	High	11	**3**	**AF33**	011110	30
AF4	**4**	100	Low	01	**1**	**AF41**	100010	34
AF4	**4**	100	Medium	10	**2**	**AF42**	100100	36
AF4	**4**	100	High	11	**3**	**AF43**	100110	38

Note

In RFC 2597, *drop probability* is referred to as *drop precedence*.

An AF implementation must detect and respond to long-term congestion within each class by dropping packets using a congestion-avoidance algorithm such as weighted random early detection (WRED). WRED uses the AF Drop Probability value within each class—where 1 is the lowest possible value, and 3 is the highest possible—to determine which packets should be dropped first during periods of congestion. It should also be able to handle short-term congestion resulting from bursts if each class is placed in a separate queue, using a queueing algorithm such as class-based weighted fair queueing (CBWFQ). The AF specification does not define the use of any particular algorithms to use for queueing and congestions avoidance, but it does specify the requirements and properties of such algorithms.

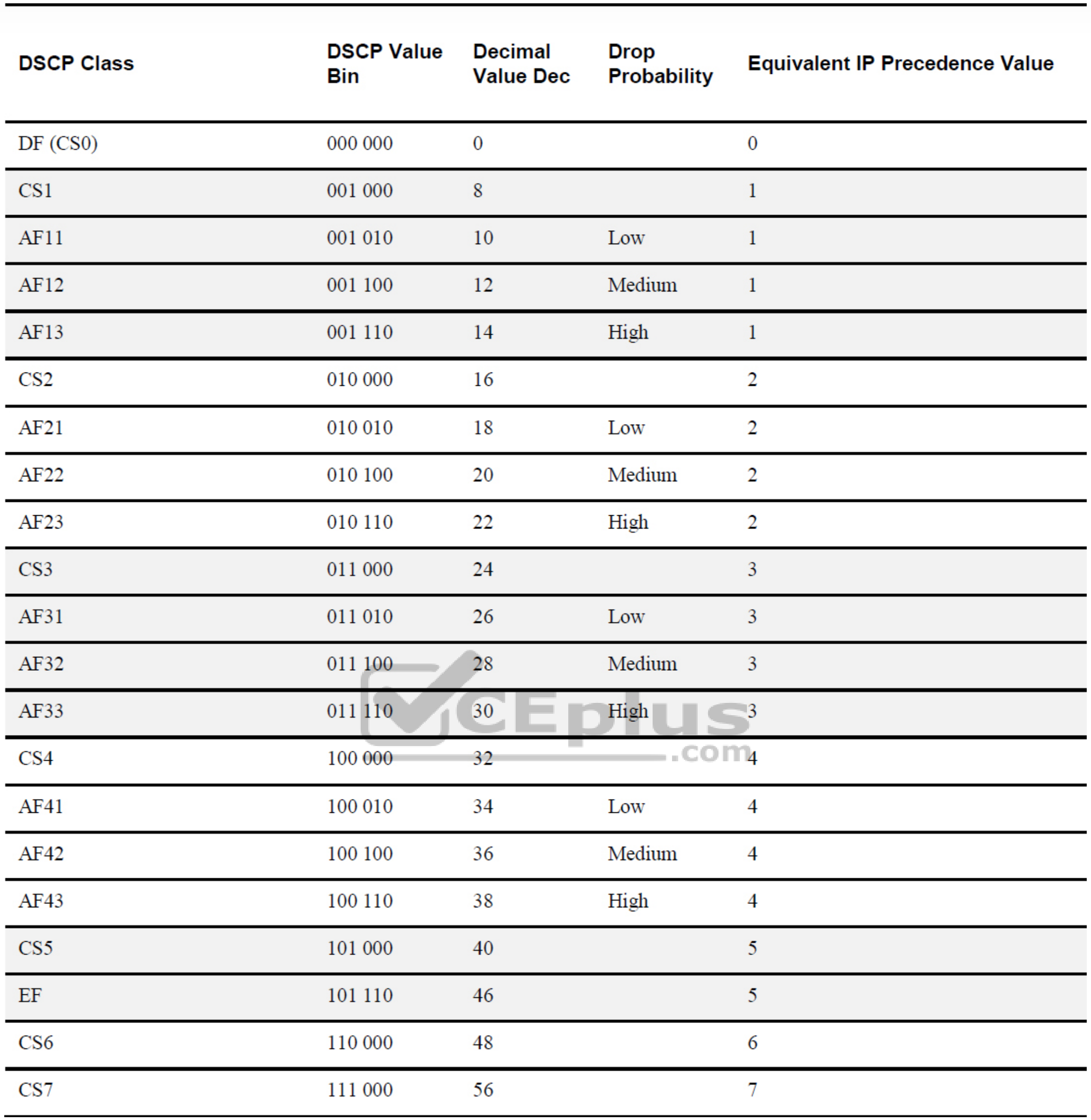
Expedited Forwarding (EF) PHB

The EF PHB can be used to build a low-loss, low-latency, low-jitter, assured bandwidth, end-to-end service. The EF PHB guarantees bandwidth by ensuring a minimum departure rate and provides the lowest possible delay to delay-sensitive applications by implementing low-latency queueing. It also prevents starvation of other applications or classes that are not using the EF PHB by policing EF traffic when congestion occurs.

Packets requiring EF should be marked with DSCP binary value 101110 (46 in decimal). Bits 5 to 7 (101) of the EF DSCP value map directly to IP Precedence 5 for backward compatibility with non-DiffServ-compliant devices. IP Precedence 5 is the highest user-definable IP Precedence value and is used for real-time delay-sensitive traffic (such as VoIP).

Table 14-4 includes all the DSCP PHBs (DF, CS, AF, and EF) with their decimal and binary equivalents. This table can also be used to see which IP Precedence value corresponds to each PHB.

Table 14-4 DSCP PHBs with Decimal and Binary Equivalents and IPP



DSCP Class	DSCP Value Bin	Decimal Value Dec	Drop Probability	Equivalent IP Precedence Value
DF (CS0)	000 000	0		0
CS1	001 000	8		1
AF11	001 010	10	Low	1
AF12	001 100	12	Medium	1
AF13	001 110	14	High	1
CS2	010 000	16		2
AF21	010 010	18	Low	2
AF22	010 100	20	Medium	2
AF23	010 110	22	High	2
CS3	011 000	24		3
AF31	011 010	26	Low	3
AF32	011 100	28	Medium	3
AF33	011 110	30	High	3
CS4	100 000	32		4
AF41	100 010	34	Low	4
AF42	100 100	36	Medium	4
AF43	100 110	38	High	4
CS5	101 000	40		5
EF	101 110	46		5
CS6	110 000	48		6
CS7	111 000	56		7

Scavenger Class

The scavenger class is intended to provide less than best-effort services. Applications assigned to the scavenger class have little or no contribution to the business objectives of an organization and are typically entertainment-related applications. These include peer-to-peer applications (such as Torrent), gaming applications (for example, Minecraft, Fortnite), and entertainment video applications (for example, YouTube, Vimeo, Netflix). These types of applications are usually heavily rate limited or blocked entirely.

Something very peculiar about the scavenger class is that it is intended to be lower in priority than a best-effort service. Best-effort traffic uses a DF PHB with a DCSP value of 000000 (CS0). Since there are no negative DSCP values, it was decided to use CS1 as the marking for scavenger traffic. This is defined in RFC 4594.

Key Topic

Trust Boundary

To provide and end-to-end and scalable QoS experience, packets should be marked by the endpoint or as close to the endpoint as possible. When an endpoint marks a frame or a packet with a CoS or DSCP value, the switch port it is attached to can be configured to accept or reject the CoS or DCSP values. If the switch accepts the values, it means it trusts the endpoint and does not need to do any packet reclassification and re-marking for the received endpoint’s packets. If the switch does not trust the endpoint, it rejects the markings and reclassifies and re-marks the received packets with the appropriate CoS or DSCP value.

For example, consider a campus network with IP telephony and host endpoints; the IP phones by default mark voice traffic with a CoS value of 5 and a DSCP value of 46 (EF), while incoming traffic from an endpoint (such as a PC) attached to the IP phone's switch port is re-marked to a CoS value of 0 and a DSCP value of 0. Even if the endpoint is sending tagged frames with a specific CoS or DSCP value, the default behavior for Cisco IP phones is to not trust the endpoint and zero out the CoS and DCSP values before sending the frames to the switch. When the IP phone sends voice and data traffic to the switch, the switch can classify voice traffic as higher priority than the data traffic, thanks to the high-priority CoS and DSCP markings for voice traffic.

For scalability, trust boundary classification should be done as close to the endpoint as possible. Figure 14-7 illustrates trust boundaries at different points in a campus network, where 1 and 2 are optimal, and 3 is acceptable only when the access switch is not capable of performing classification.

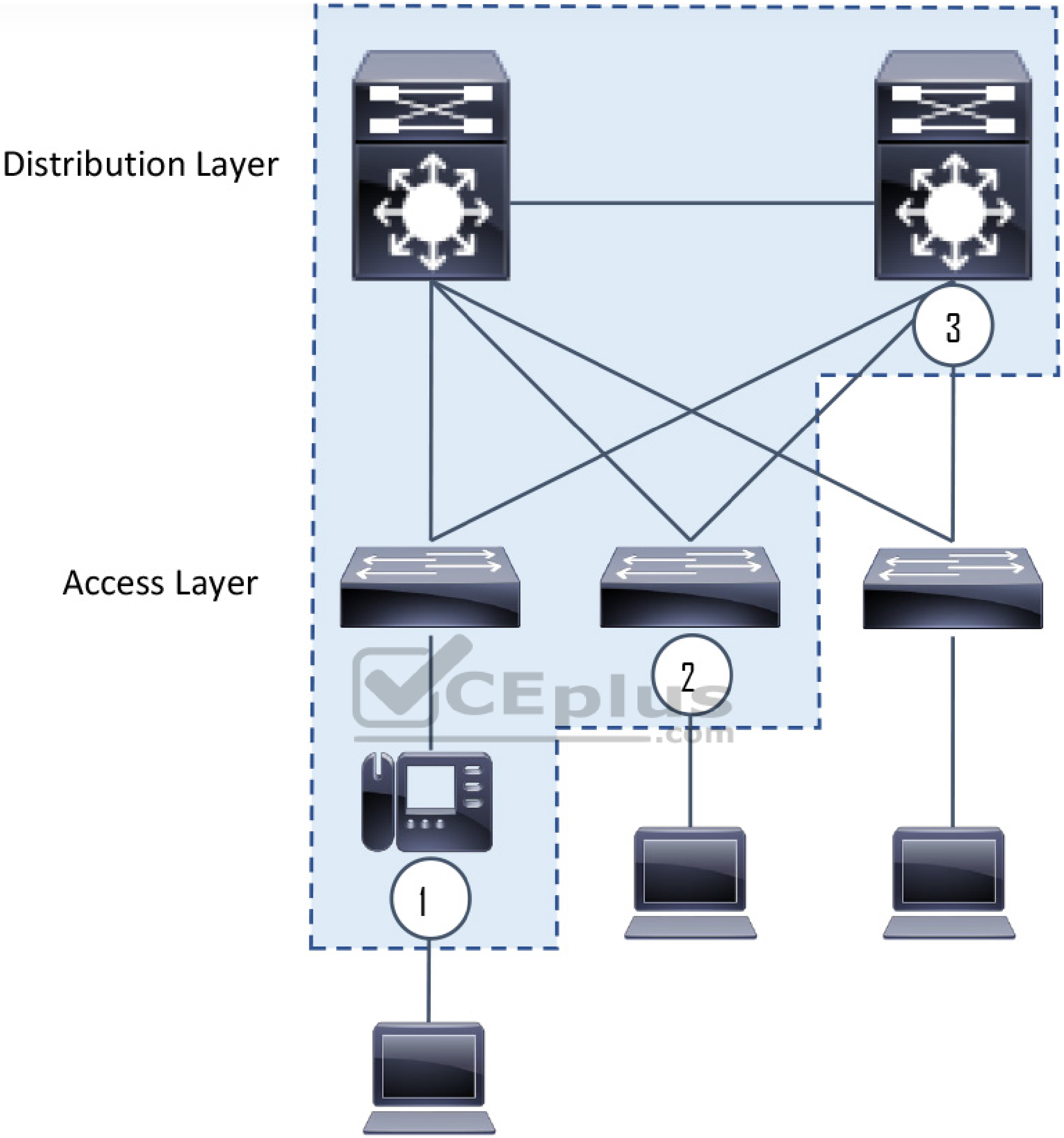


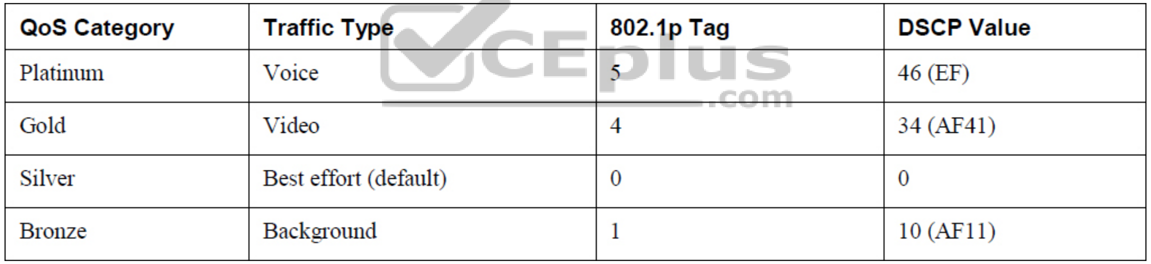
Figure 14-7 Trust Boundaries

A Practical Example: Wireless QoS

A wireless network can be configured to leverage the QoS mechanisms described in this chapter. For example, a wireless LAN controller (WLC) sits at the boundary between wireless and wired networks, so it becomes a natural location for a QoS trust boundary. Traffic entering and exiting the WLC can be classified and marked so that it can be handled appropriately as it is transmitted over the air and onto the wired network.

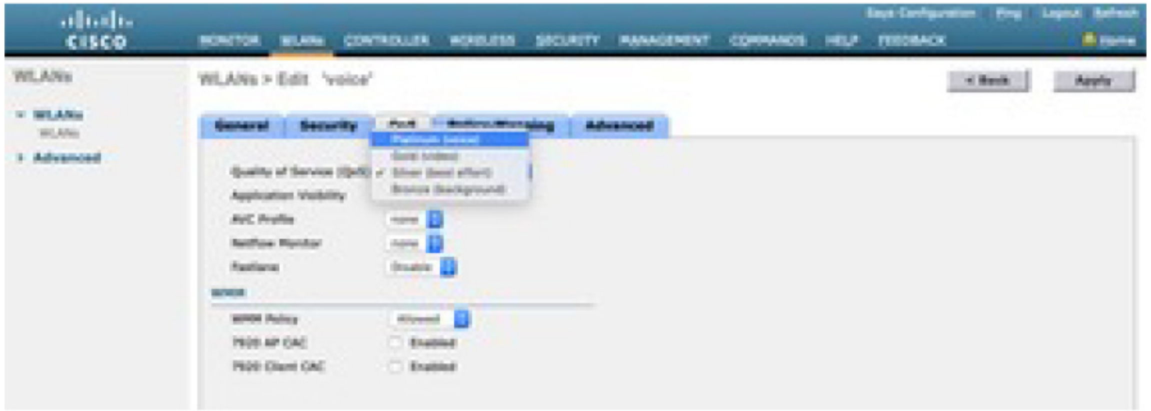
Wireless QoS can be uniquely defined on each wireless LAN (WLAN), using the four traffic categories listed in Table 14-5. Notice that the category names are human-readable words that translate to specific 802.1p and DSCP values.

Table 14-5 Wireless QoS Policy Categories and Markings



QoS Category	Traffic Type	802.1p Tag	DSCP Value
Platinum	Voice	5	46 (EF)
Gold	Video	4	34 (AF41)
Silver	Best effort (default)	0	0
Bronze	Background	1	10 (AF11)

When you create a new WLAN, its QoS policy defaults to Silver, or best-effort handling. In Figure 14-8, a WLAN named voice has been created to carry voice traffic, so its QoS policy has been set to Platinum. Wireless voice traffic will then be classified for low latency and low jitter and marked with an 802.1p CoS value of 5 and a DSCP value of 46 (EF).


cisco
MONITOR WLANs CONTROLLER WIRELESS SECURITY MANAGEMENT COMMANDS HELP FEEDBACK
Save Configuration | Ping | Logout | Refresh
Home
WLANs
WLANs
WLANs
Advanced
< Back
Apply
General
Security
QoS
Policy-Mapping
Advanced

WLANs > Edit 'voice'

Quality of Service (QoS)

Platinum (voice)
Gold (video)
✓ Silver (best effort)
Bronze (background)

Application Visibility

AVC Profile: none

Netflow Monitor: none

Fastlane: Disable

WMM

WMM Policy: Allowed

7920 AP CAC: □ Enabled

7920 Client CAC: □ Enabled

Figure 14-8 Setting the QoS Policy for a Wireless LAN

POLICING AND SHAPING

Key Topic

Traffic policers and shapers are traffic-conditioning QoS mechanisms used to classify traffic and enforce other QoS mechanisms such as rate limiting. They classify traffic in an identical manner but differ in their implementation:

- **Policers:** Drop or re-mark incoming or outgoing traffic that goes beyond a desired traffic rate.

- **Shapers:** Buffer and delay egress traffic rates that momentarily peak above the desired rate until the egress traffic rate drops below the defined traffic rate. If the egress traffic rate is below the desired rate, the traffic is sent immediately.

Figure 14-9 illustrates the difference between traffic policing and shaping. Policers drop or re-mark excess traffic, while shapers buffer and delay excess traffic.

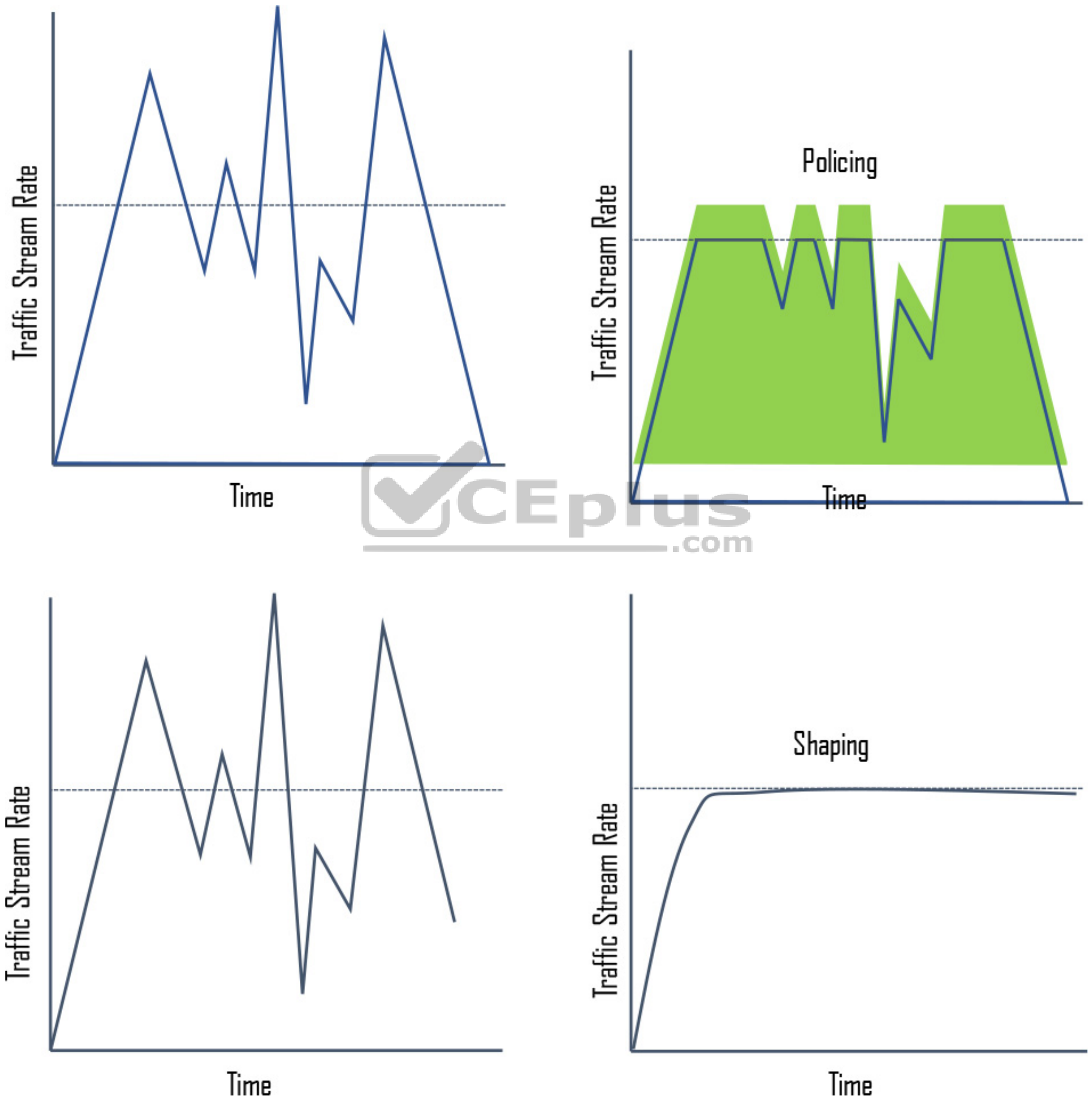


Figure 14-9 Policing Versus Shaping

Placing Policers and Shapers in the Network

Policers for incoming traffic are most optimally deployed at the edge of the network to keep traffic from wasting valuable bandwidth in the core of the network. Policers for outbound traffic are most optimally deployed at the edge of the network or core-facing interfaces on network edge devices. A downside of policing is that it causes TCP retransmissions when it drops traffic.

Shapers are used for egress traffic and typically deployed by enterprise networks on service provider (SP)–facing interfaces. Shaping is useful in cases where SPs are policing incoming traffic or when SPs are not policing traffic but do have a maximum traffic rate SLA, which, if violated, could incur monetary penalties. Shaping buffers and delays traffic rather than dropping it, and this causes fewer TCP retransmissions compared to policing.

Key Topic

Markdown

When a desired traffic rate is exceeded, a policer can take one of the following actions:

- Drop the traffic.

- Mark down the excess traffic with a lower priority.

Marking down excess traffic involves re-marking the packets with a lower-priority class value; for example, excess traffic marked with AFx1 should be marked down to AFx2 (or AFx3 if using two-rate policing). After marking down the traffic, congestion-avoidance mechanisms, such as DSCP-based weighted random early detection (WRED), should be configured throughout the network to drop AFx3 more aggressively than AFx2 and drop AFx2 more aggressively than AFx1.

Token Bucket Algorithms

Cisco IOS policers and shapers are based on token bucket algorithms. The following list includes definitions that are used to explain how token bucket algorithms operate:

- **Committed Information Rate (CIR):** The policed traffic rate, in bits per second (bps), defined in the traffic contract.

- **Committed Time Interval (Tc):** The time interval, in milliseconds (ms), over which the committed burst (Bc) is sent. Tc can be calculated with the formula Tc = (Bc [bits] / CIR [bps]) × 1000.

- **Committed Burst Size (Bc):** The maximum size of the CIR token bucket, measured in bytes, and the maximum amount of traffic that can be sent within a

Tc. Bc can be calculated with the formula Bc = CIR × (Tc / 1000).

- **Token:** A single token represents 1 byte or 8 bits.

- **Token bucket:** A bucket that accumulates tokens until a maximum predefined number of tokens is reached (such as the Bc when using a single token bucket); these tokens are added into the bucket at a fixed rate (the CIR). Each packet is checked for conformance to the defined rate and takes tokens from the bucket equal to its packet size; for example, if the packet size is 1500 bytes, it takes 12,000 bits (1500 × 8) from the bucket. If there are not enough tokens in the token bucket to send the packet, the traffic conditioning mechanism can take one of the following actions:

- Buffer the packets while waiting for enough tokens to accumulate in the token bucket (traffic shaping)

- Drop the packets (traffic policing)

- Mark down the packets (traffic policing)

It is recommended for the Bc value to be larger than or equal to the size of the largest possible IP packet in a traffic stream. Otherwise, there will never be enough tokens in the token bucket for larger packets, and they will always exceed the defined rate. If the bucket fills up to the maximum capacity, newly added tokens are discarded. Discarded tokens are not available for use to future packets.

Token bucket algorithms may use one or multiple token buckets. For single token bucket algorithms, the measured traffic rate can conform to or exceed the defined traffic rate. The measured traffic rate is conforming if there are enough tokens in the token bucket to transmit the traffic. The measured traffic rate is exceeding if there are not enough tokens in the token bucket to transmit the traffic.

Figure 14-10 illustrates the concept of the single token bucket algorithm.

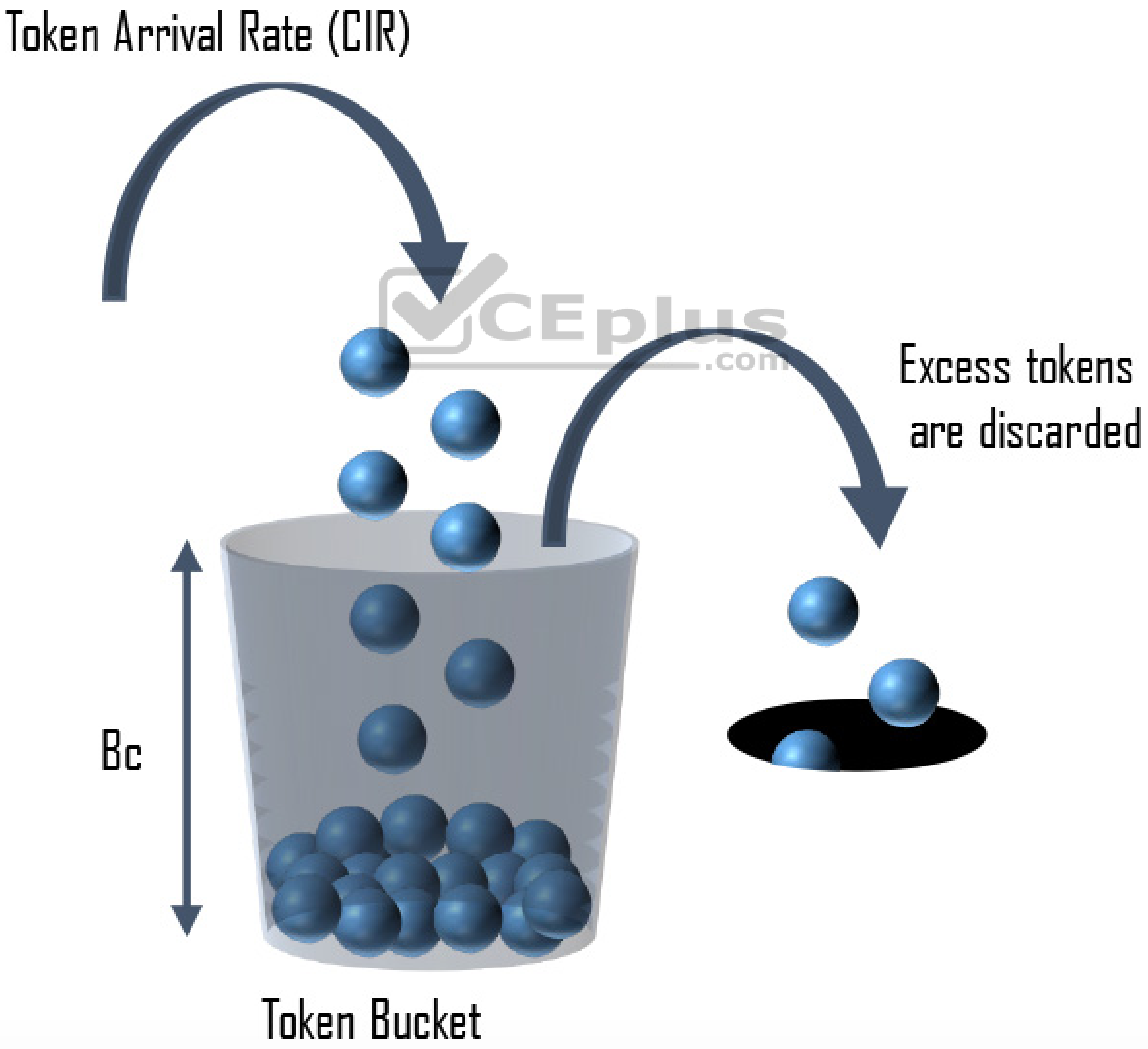


Figure 14-10 Single Token Bucket Algorithm

To understand how the single token bucket algorithms operate in more detail, assume that a 1 Gbps interface is configured with a policer defined with a CIR of 120 Mbps and a Bc of 12 Mb. The Tc value cannot be explicitly defined in IOS, but it can be calculated as follows:

Tc = (Bc [bits] / CIR [bps]) × 1000

Tc = (12 Mb / 120 Mbps) × 1000

Tc = (12,000,000 bits / 120,000,000 bps) × 1000 = 100 ms

Once the Tc value is known, the number of Tcs within a second can be calculated as follows:

Tcs per second = 1000 / Tc

Tcs per second = 1000 ms / 100 ms = 10 Tcs

If a continuous stream of 1500-byte (12,000-bit) packets is processed by the token algorithm, only a Bc of 12 Mb can be taken by the packets within each Tc (100 ms). The number of packets that conform to the traffic rate and are allowed to be transmitted can be calculated as follows:

Number of packets that conform within each Tc = Bc / packet size in bits (rounded down)

Number of packets that conform within each Tc = 12,000,000 bits / 12,000 bits = 1000 packets

Any additional packets beyond 1000 will either be dropped or marked down.

To figure out how many packets would be sent in one second, the following formula can be used:

Packets per second = Number of packets that conform within each Tc × Tcs per second

Packets per second = 1000 packets × 10 intervals = 10,000 packets

To calculate the CIR for the 10,000, the following formula can be used:

CIR = Packets per second × Packet size in bits

CIR = 10,000 packets per second × 12,000 bits = 120,000,000 bps = 120 Mbps

To calculate the time interval it would take for the 1000 packets to be sent at interface line rate, the following formula can be used:

Time interval at line rate = (Bc [bits] / Interface speed [bps]) × 1000

Time interval at line rate = (12 Mb / 1 Gbps) × 1000

Time interval at line rate = (12,000,000 bits / 1000,000,000 bps) × 1000 = 12 ms

Figure 14-11 illustrates how the Bc (1000 packets at 1500 bytes each, or 12Mb) is sent every Tc interval. After the Bc is sent, there is an interpacket delay of 113 ms (125 ms minus 12 ms) within the Tc where there is no data transmitted.

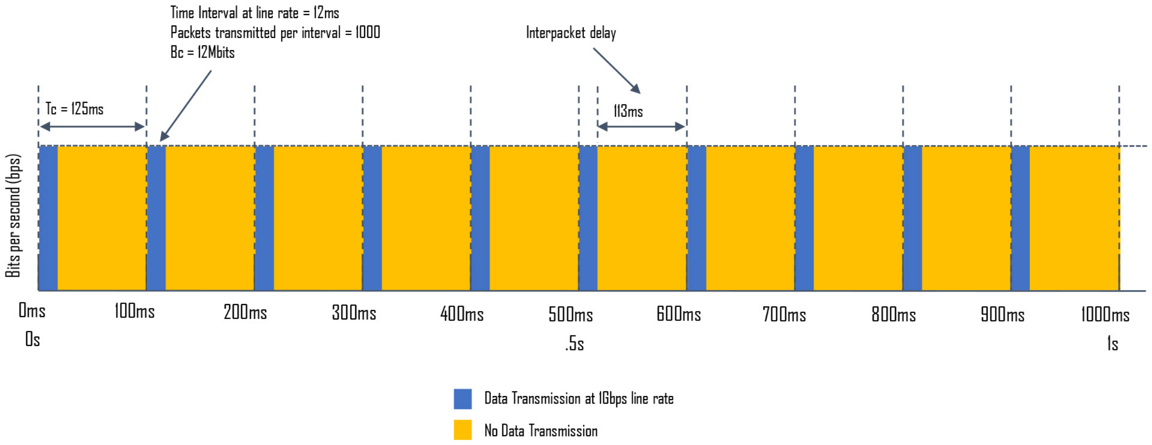


Figure 14-11 Token Bucket Operation

The recommended values for Tc range from 8 ms to 125 ms. Shorter Tcs, such as 8 ms to 10 ms, are necessary to reduce interpacket delay for real-time traffic such as voice. Tcs longer than 125 ms are not recommended for most networks because the interpacket delay becomes too large.

Types of Policers

Key Topic

There are different policing algorithms, including the following:

- Single-rate two-color marker/policer
- Single-rate three-color marker/policer (srTCM)
- Two-rate three-color marker/policer (trTCM)

Single-Rate Two-Color Markers/Policers

The first policers implemented use a single-rate, two-color model based on the single token bucket algorithm. For this type of policer, traffic can be either conforming to or exceeding the CIR. Marking down or dropping actions can be performed for each of the two states.

Figure 14-12 illustrates different actions that the single-rate two-color policer can take. The section above the dotted line on the left side of the figure represents traffic that exceeded the CIR and was marked down. The section above the dotted line on the right side of the figure represents traffic that exceeded the CIR and was dropped.

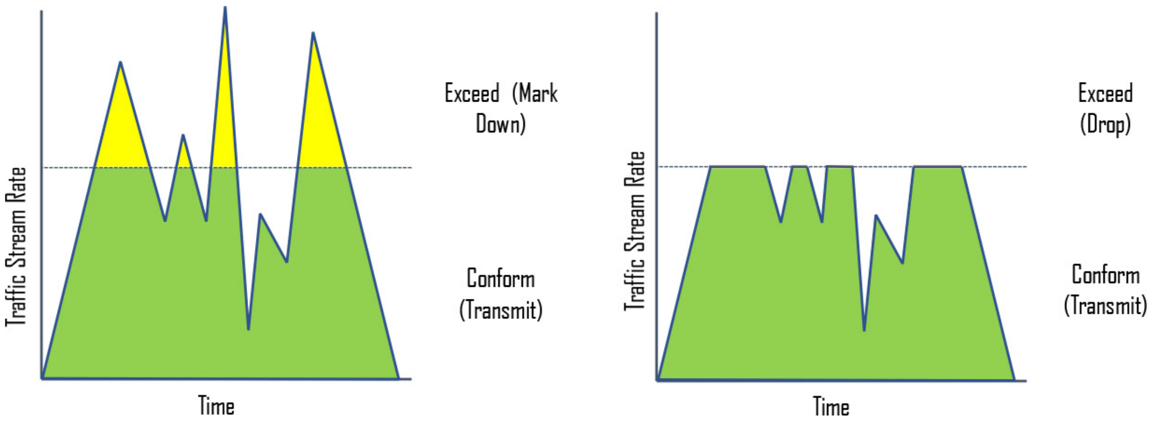


Figure 14-12 Single-Rate Two-Color Marker/Policer

Single-Rate Three-Color Markers/Policers (srTCM)

Single-rate three-color policer algorithms are based on RFC 2697. This type of policer uses two token buckets, and the traffic can be classified as either conforming to, exceeding, or violating the CIR. Marking down or dropping actions are performed for each of the three states of traffic.

The first token bucket operates very similarly to the single-rate two-color system; the difference is that if there are any tokens left over in the bucket after each time period due to low or no activity, instead of discarding the excess tokens (overflow), the algorithm places them in a second bucket to be used later for temporary bursts that might exceed the CIR. Tokens placed in this second bucket are referred to as the *excess burst (Be)*, and Be is the maximum number of bits that can exceed the Bc burst size.

With the two token-bucket mechanism, traffic can be classified in three colors or states, as follows:

- **Conform:** Traffic under Bc is classified as conforming and green. Conforming traffic is usually transmitted and can be optionally re-marked.
- **Exceed:** Traffic over Bc but under Be is classified as exceeding and yellow. Exceeding traffic can be dropped or marked down and transmitted.
- **Violate:** Traffic over Be is classified as violating and red. This type of traffic is usually dropped but can be optionally marked down and transmitted.

Figure 14-13 illustrates different actions that a single-rate three-color policer can take. The section below the straight dotted line on the left side of the figure represents the traffic that conformed to the CIR, the section right above the straight dotted line represents the exceeding traffic that was marked down, and the top section represents the violating traffic that was also marked down. The exceeding and violating traffic rates vary because they rely on random tokens spilling over from the Bc bucket into the Be. The section right above the straight dotted line on the right side of the figure represents traffic that exceeded the CIR and was marked down and the top section represents traffic that violated the CIR and was dropped.

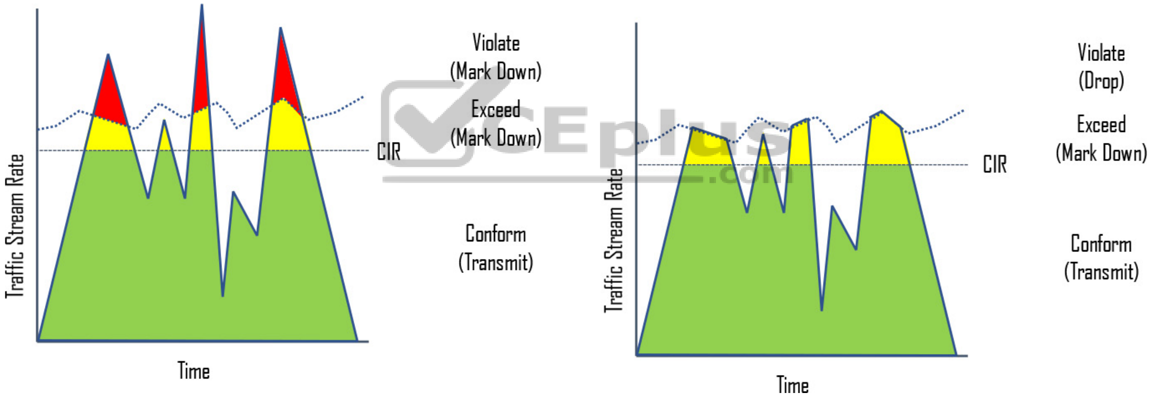


Figure 14-13 Single-Rate Three-Color Marker/Policer

The single-rate three-color marker/policer uses the following parameters to meter the traffic stream:

- **Committed Information Rate (CIR):** The policed rate.

- **Committed Burst Size (Bc):** The maximum size of the CIR token bucket, measured in bytes. Referred to as *Committed Burst Size (CBS)* in RFC 2697.

- **Excess Burst Size (Be):** The maximum size of the excess token bucket, measured in bytes. Referred to as *Excess Burst Size (EBS)* in RFC 2697.

- **Bc Bucket Token Count (Tc):** The number of tokens in the Bc bucket. Not to be confused with the committed time interval Tc.

- **Be Bucket Token Count (Te):** The number of tokens in the Be bucket.

- **Incoming Packet Length (B):** The packet length of the incoming packet, in bits.

Figure 14-14 illustrates the logical flow of the single-rate three-color marker/policer two-token-bucket algorithm.

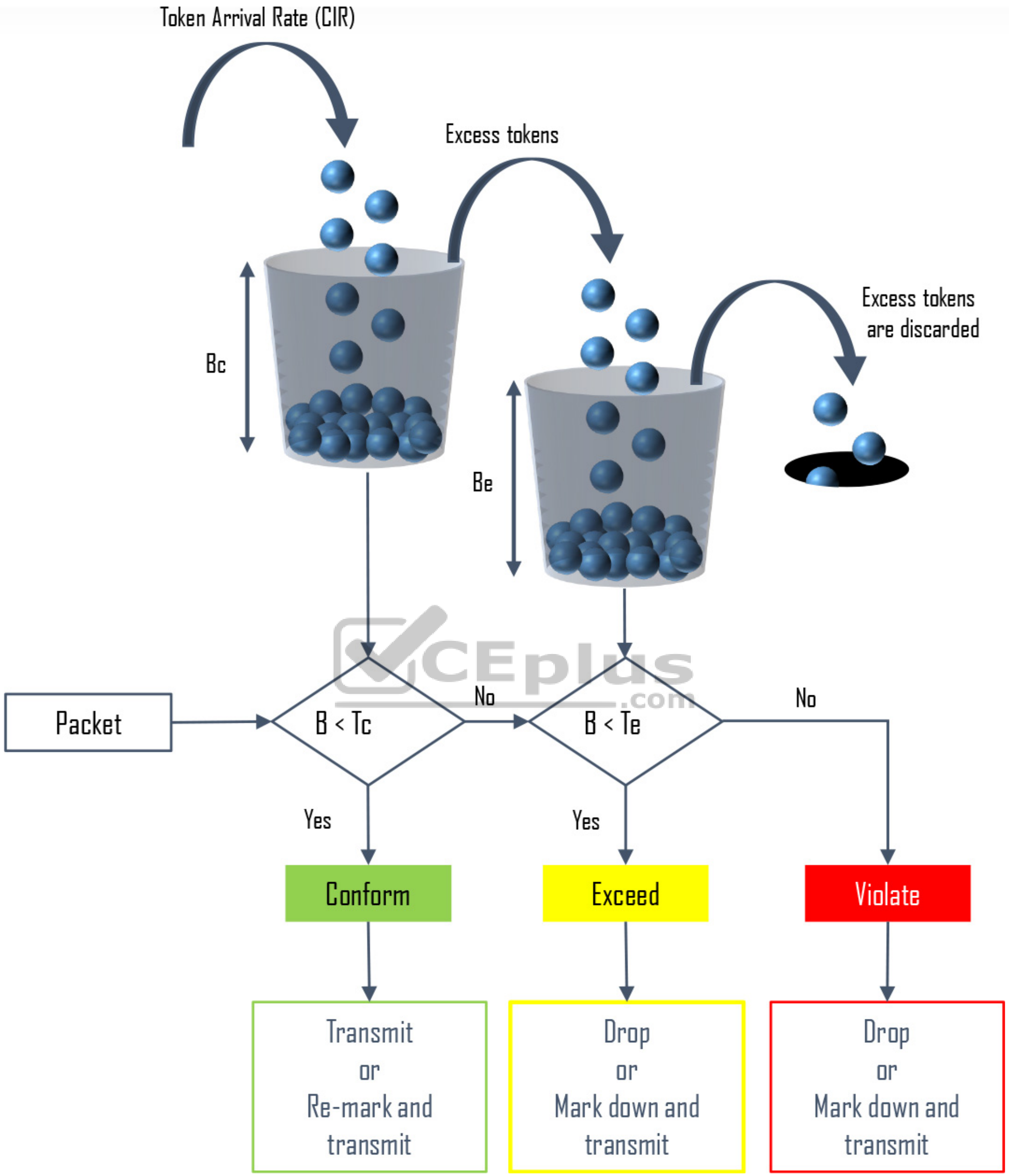

Token Arrival Rate (CIR)
Excess tokens
Excess tokens are discarded
Bc
Be
Packet
B < Tc
No
B < Te
No
Yes
Yes
Conform
Exceed
Violate
Transmit or Re-mark and transmit
Drop or Mark down and transmit
Drop or Mark down and transmit

Figure 14-14 Single-Rate Three-Color Marker/Policer Token Bucket Algorithm

The single-rate three-color policer's two bucket algorithm causes fewer TCP retransmissions and is more efficient for bandwidth utilization. It is the perfect policer to be used with AF classes (AFx1, AFx2, and AFx3). Using a three-color policer makes sense only if the actions taken for each color differ. If the actions for two or more colors are the same, for example, conform and exceed both transmit without re-marking, the single-rate two-color policer is recommended to keep things simpler.

Two-Rate Three-Color Markers/Policers (trTCM)

The two-rate three-color marker/policer is based on RFC 2698 and is similar to the single-rate three-color policer. The difference is that single-rate three-color policers rely on excess tokens from the Bc bucket, which introduces a certain level of variability and unpredictability in traffic flows; the two-rate three-color marker/policers address this issue by using two distinct rates, the CIR and the Peak Information Rate (PIR). The two-rate three-color marker/policer allows for a sustained excess rate based on the PIR that allows for different actions for the traffic exceeding the different burst values; for example, violating traffic can be dropped at a defined rate, and this is something that is not possible with the single-rate three-color policer. Figure 14-15 illustrates how violating traffic that exceeds the PIR can either be marked down (on the left side of the figure) or dropped (on the right side of the figure). Compare Figure 14-15 to Figure 14-14 to see the difference between the two-rate three-color policer and the single-rate three-color policer.

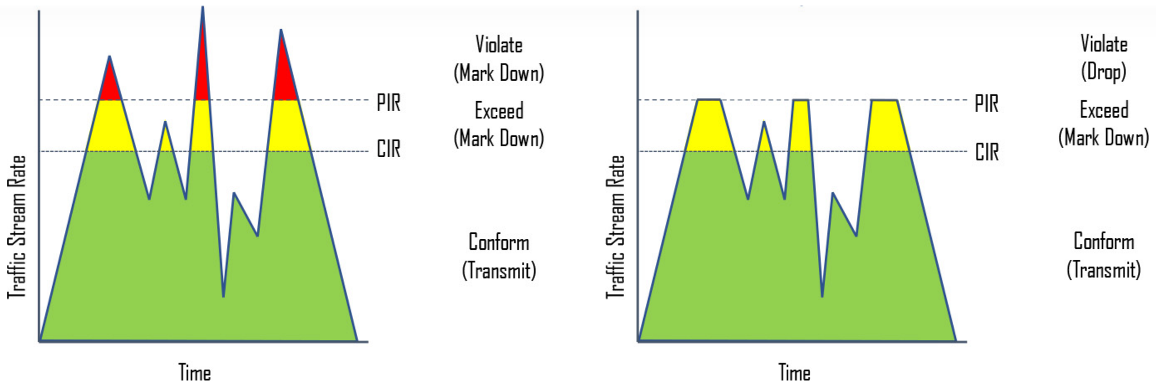


Figure 14-15 Two-Rate Three-Color Marker/Policer Token Bucket Algorithm

The two-rate three-color marker/policer uses the following parameters to meter the traffic stream:

- **Committed Information Rate (CIR):** The policed rate.

- **Peak Information Rate (PIR):** The maximum rate of traffic allowed. PIR should be equal or greater than the CIR.

- **Committed Burst Size (Bc):** The maximum size of the second token bucket, measured in bytes. Referred to as *Committed Burst Size (CBS)* in RFC 2698.

- **Peak Burst Size (Be):** The maximum size of the PIR token bucket, measured in bytes. Referred to as *Peak Burst Size (PBS)* in RFC 2698. Be should be equal to or greater than Bc.

- **Bc Bucket Token Count (Tc):** The number of tokens in the Bc bucket. Not to be confused with the committed time interval Tc.

- **Bp Bucket Token Count (Tp):** The number of tokens in the Bp bucket.

- **Incoming Packet Length (B):** The packet length of the incoming packet, in bits.

The two-rate three-color policer also uses two token buckets, but the logic varies from that of the single-rate three-color policer. Instead of transferring unused tokens from the Bc bucket to the Be bucket, this policer has two separate buckets that are filled with two separate token rates. The Be bucket is filled with the PIR tokens, and the Bc bucket is filled with the CIR tokens. In this model, the Be represents the peak limit of traffic that can be sent during a subsecond interval.

The logic varies further in that the initial check is to see whether the traffic is within the PIR. Only then is the traffic compared against the CIR. In other words, a violate condition is checked first, then an exceed condition, and finally a conform condition, which is the reverse of the logic of the single-rate three-color policer. Figure 14-16 illustrates the token bucket algorithm for the two-rate three-color marker/policer. Compare it to the token bucket algorithm of the single-rate three-color marker/policer in Figure 14-14 to see the differences between the two.

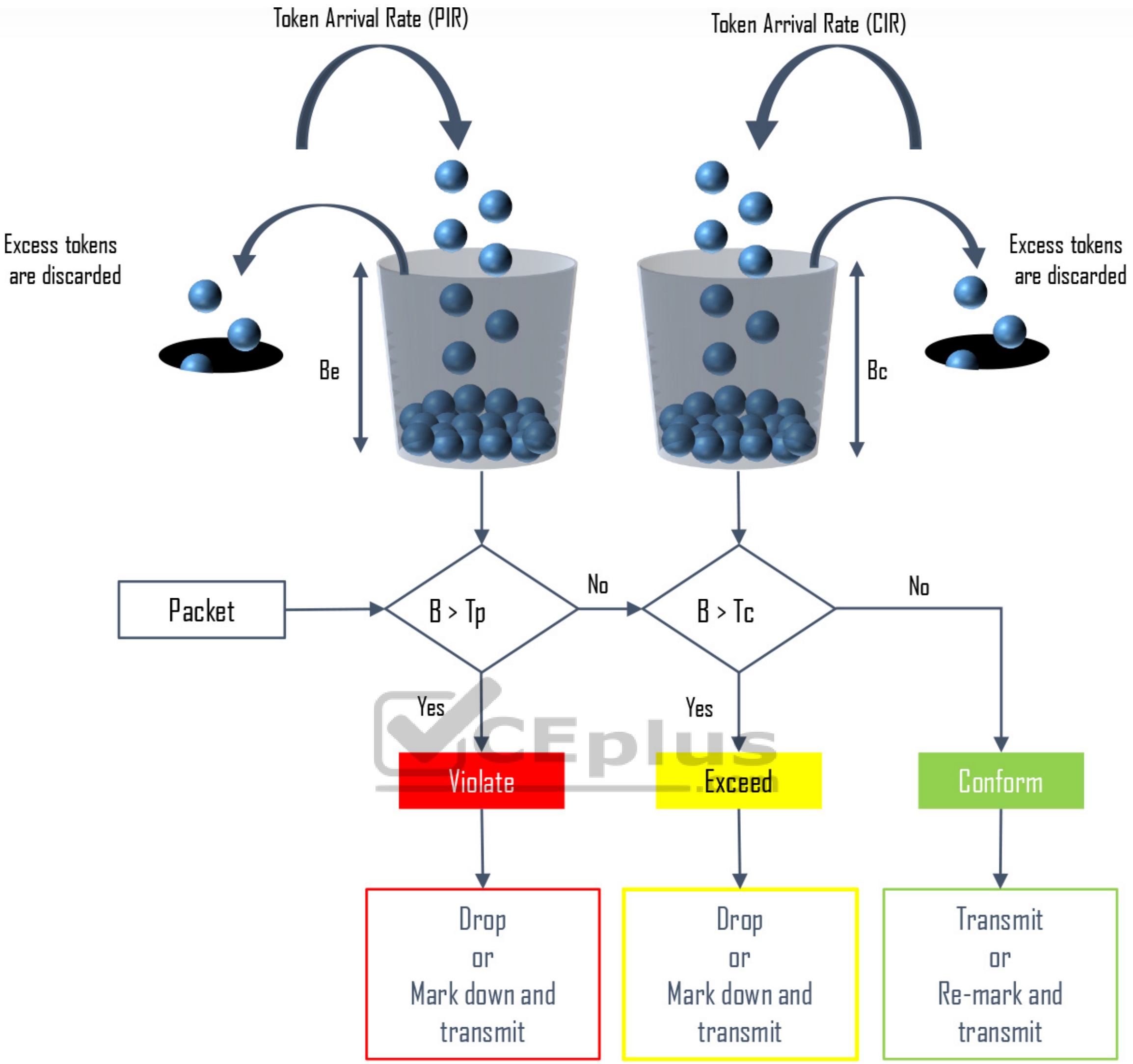

Token Arrival Rate (PIR)
Token Arrival Rate (CIR)
Excess tokens are discarded
Be
Bc
Excess tokens are discarded
Packet
B > Tp
No
B > Tc
No
Yes
Yes
Violate
Exceed
Conform
Drop or Mark down and transmit
Drop or Mark down and transmit
Transmit or Re-mark and transmit

Figure 14-16 Two-Rate Three-Color Marker/Policer Token Bucket Algorithm

CONGESTION MANAGEMENT AND AVOIDANCE

This section explores the queuing algorithms used for congestion management as well as packet drop techniques that can be used for congestion avoidance. These tools provide a way of managing excessive traffic during periods of congestion.

Congestion Management

Congestion management involves a combination of queuing and scheduling. Queuing (also known as *buffering*) is the temporary storage of excess packets. Queuing is activated when an output interface is experiencing congestion and deactivated when congestion clears. Congestion is detected by the queuing algorithm when a Layer 1 hardware queue present on physical interfaces, known as the *transmit ring (Tx-ring or TxQ)*, is full. When the Tx-ring is not full anymore, this indicates that there is no congestion on the interface, and queueing is deactivated. Congestion can occur for one of these two reasons:

- The input interface is faster than the output interface.

- The output interface is receiving packets from multiple input interfaces.

When congestion is taking place, the queues fill up, and packets can be reordered by some of the queuing algorithms so that higher-priority packets exit the output interface sooner than lower-priority ones. At this point, a scheduling algorithm decides which packet to transmit next. Scheduling is always active, regardless of whether the interface is experiencing congestion.

There are many queuing algorithms available, but most of them are not adequate for modern rich-media networks carrying voice and high-definition video traffic because they were designed before these traffic types came to be. The legacy queuing algorithms that predate the MQC architecture include the following:

Key Topic

- **First-in, first-out queuing (FIFO):** FIFO involves a single queue where the first packet to be placed on the output interface queue is the first packet to leave the interface (first come, first served). In FIFO queuing, all traffic belongs to the same class.

- **Round robin:** With round robin, queues are serviced in sequence one after the other, and each queue processes one packet only. No queues starve with round robin because every queue gets an opportunity to send one packet every round. No queue has priority over others, and if the packet sizes from all queues are about the same, the interface bandwidth is shared equally across the round robin queues. A limitation of round robin is it does not include a mechanism to prioritize traffic.

- **Weighted round robin (WRR):** WRR was developed to provide prioritization capabilities for round robin. It allows a weight to be assigned to each queue, and based on that weight, each queue effectively receives a portion of the interface bandwidth that is not necessarily equal to the other queues' portions.

- **Custom queuing (CQ):** CQ is a Cisco implementation of WRR that involves a set of 16 queues with a round-robin scheduler and FIFO queueing within each queue. Each queue can be customized with a portion of the link bandwidth for each selected traffic type. If a particular type of traffic is not using the bandwidth reserved for it, other traffic types may use the unused bandwidth. CQ causes long delays and also suffers from all the same problems as FIFO within each of the 16 queues that it uses for traffic classification.

- **Priority queuing (PQ):** With PQ, a set of four queues (high, medium, normal, and low) are served in strict-priority order, with FIFO queueing within each queue. The high-priority queue is always serviced first, and lower-priority queues are serviced only when all higher-priority queues are empty. For example, the medium queue is serviced only when the high-priority queue is empty. The normal queue is serviced only when the high and medium queues are empty; finally, the low queue is serviced only when all the other queues are empty. At any point in time, if a packet arrives for a higher queue, the packet from the higher queue is processed before any packets in lower-level queues. For this reason, if the higher-priority queues are continuously being serviced, the lower-priority queues are starved.

- **Weighted fair queuing (WFQ):** The WFQ algorithm automatically divides the interface bandwidth by the number of flows (weighted by IP Precedence) to allocate bandwidth fairly among all flows. This method provides better service for high-priority real-time flows but can't provide a fixed-bandwidth guarantee for any particular flow.

The current queuing algorithms recommended for rich-media networks (and supported by MQC) combine the best features of the legacy algorithms. These algorithms provide real-time, delay-sensitive traffic bandwidth and delay guarantees while not starving other types of traffic. The recommended queuing algorithms include the following:

Key Topic

- **Class-based weighted fair queuing (CBWFQ):** CBWFQ enables the creation of up to 256 queues, serving up to 256 traffic classes. Each queue is serviced based on the bandwidth assigned to that class. It extends WFQ functionality to provide support for user-defined traffic classes. With CBWFQ, packet classification is done based on traffic descriptors such as QoS markings, protocols, ACLs, and input interfaces. After a packet is classified as belonging to a specific class, it is possible to assign bandwidth, weight, queue limit, and maximum packet limit to it. The bandwidth assigned to a class is the minimum bandwidth delivered to the class during congestion. The queue limit for that class is the maximum number of packets allowed to be buffered in the class queue. After a queue has reached the configured queue limit, excess packets are dropped. CBWFQ by itself does not provide a latency guarantee and is only suitable for non-real-time data traffic.

- **Low-latency queuing (LLQ):** LLQ is CBWFQ combined with priority queueing (PQ) and it was developed to meet the requirements of real-time traffic, such as voice. Traffic assigned to the strict-priority queue is serviced up to its assigned bandwidth before other CBWFQ queues are serviced. All real-time traffic should be configured to be serviced by the priority queue. Multiple classes of real-time traffic can be defined, and separate bandwidth guarantees can be given to each, but a single priority queue schedules all the combined traffic. If a traffic class is not using the bandwidth assigned to it, it is shared among the other classes. This algorithm is suitable for combinations of real-time and non-real-time traffic. It provides both latency and bandwidth guarantees to high-priority real-time traffic. In the event of congestion, real-time traffic that goes beyond the assigned bandwidth guarantee is policed by a congestion-aware policer to ensure that the non-priority traffic is not starved.

Figure 14-17 illustrates the architecture of CBWFQ in combination with LLQ.

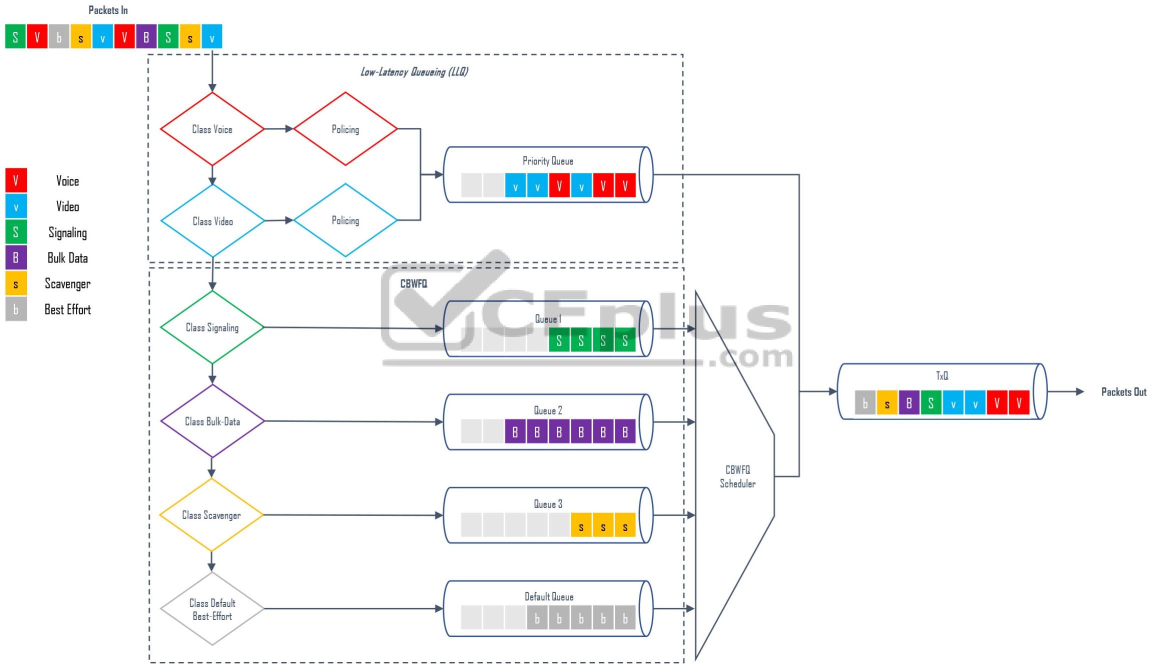


Figure 14-17 CBWFQ with LLQ

CBWFQ in combination with LLQ create queues into which traffic classes are classified. The CBWFQ queues are scheduled with a CBWFQ scheduler that guarantees bandwidth to each class. LLQ creates a high-priority queue that is always serviced first. During times of congestion, LLQ priority classes are policed to prevent the PQ from starving the CBWFQ non-priority classes (as legacy PQ does). When LLQ is configured, the policing rate must be specified as either a fixed amount of bandwidth or as a percentage of the interface bandwidth.

LLQ allows for two different traffic classes to be assigned to it so that different policing rates can be applied to different types of high-priority traffic. For example, voice traffic could be policed during times of congestion to 10 Mbps, while video could be policed to 100 Mbps. This would not be possible with only one traffic class and a single policer.

Congestion-Avoidance Tools

Congestion-avoidance techniques monitor network traffic loads to anticipate and avoid congestion by dropping packets. The default packet dropping mechanism is tail drop. Tail drop treats all traffic equally and does not differentiate between classes of service. With tail drop, when the output queue buffers are full, all packets trying to enter the queue are dropped, regardless of their priority, until congestion clears up and the queue is no longer full. Tail drop should be avoided for TCP traffic because it can cause TCP global synchronization, which results in significant link underutilization.

A better approach is to use a mechanism known as *random early detection (RED)*. RED provides congestion avoidance by randomly dropping packets before the queue buffers are full. Randomly dropping packets instead of dropping them all at once, as with tail drop, avoids global synchronization of TCP streams. RED monitors the buffer depth and performs early drops on random packets when the minimum defined queue threshold is exceeded.

Key Topic

The Cisco implementation of RED is known as weighted RED (WRED). The difference between RED and WRED is that the randomness of packet drops can be manipulated by traffic weights denoted by either IP Precedence (IPP) or DSCP. Packets with a lower IPP value are dropped more aggressively than are higher IPP values; for example, IPP 3 would be dropped more aggressively than IPP 5 or DSCP, AFx3 would be dropped more aggressively than AFx2, and AFx2 would be dropped more aggressively than AFx1.

WRED can also be used to set the IP Explicit Congestion Notification (ECN) bits to indicate that congestion was experienced in transit. ECN is an extension to WRED that allows for signaling to be sent to ECN-enabled endpoints, instructing them to reduce their packet transmission rates.

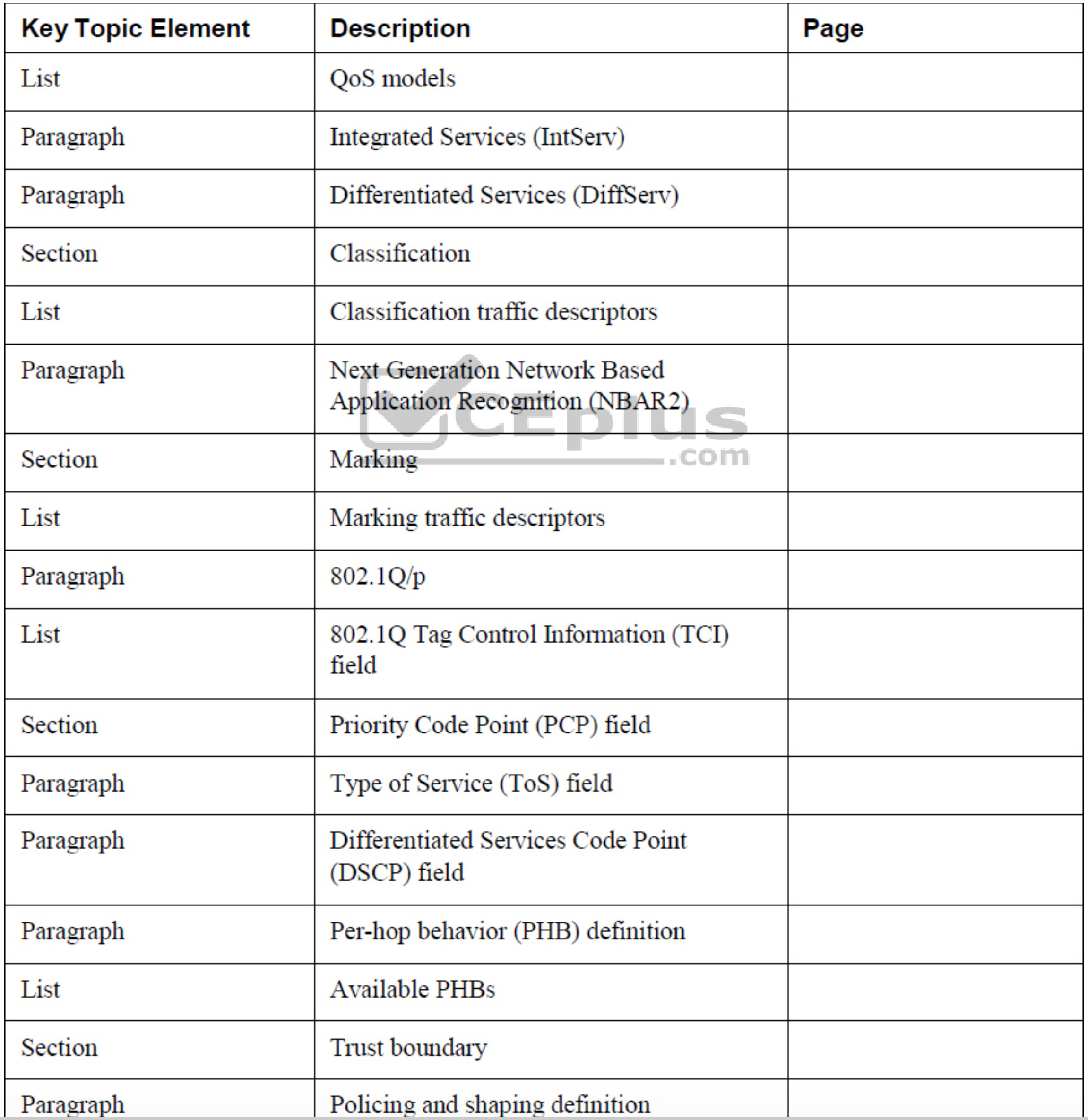
EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

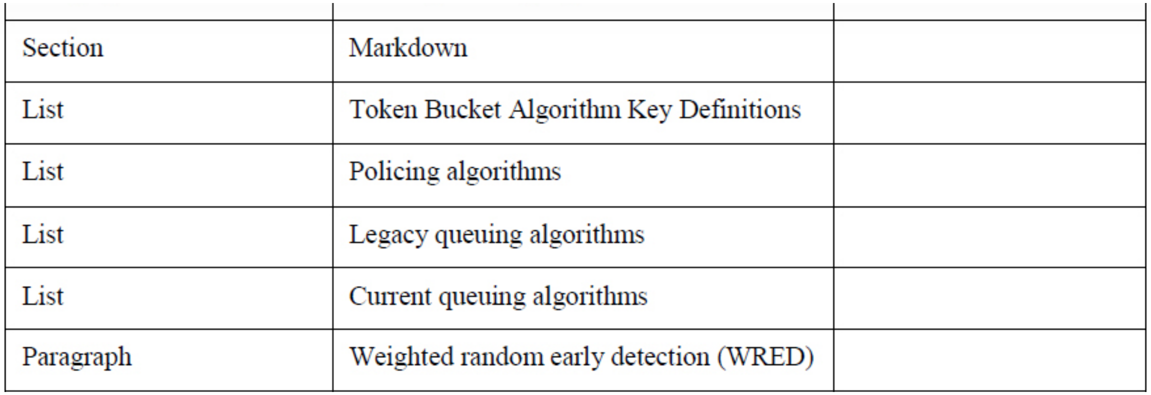
REVIEW ALL KEY TOPICS

Review the most important topics in the chapter, noted with the key topics icon in the outer margin of the page. Table 14-6 lists these key topics and the page number on which each is found.

Table 14-6 Key Topics for Chapter 14



Key Topic Element	Description	Page
List	QoS models	
Paragraph	Integrated Services (IntServ)	
Paragraph	Differentiated Services (DiffServ)	
Section	Classification	
List	Classification traffic descriptors	
Paragraph	Next Generation Network Based Application Recognition (NBAR2)	
Section	Marking	
List	Marking traffic descriptors	
Paragraph	802.1Q/p	
List	802.1Q Tag Control Information (TCI) field	
Section	Priority Code Point (PCP) field	
Paragraph	Type of Service (ToS) field	
Paragraph	Differentiated Services Code Point (DSCP) field	
Paragraph	Per-hop behavior (PHB) definition	
List	Available PHBs	
Section	Trust boundary	
Paragraph	Policing and shaping definition	



Section	Markdown	
List	Token Bucket Algorithm Key Definitions	
List	Policing algorithms	
List	Legacy queuing algorithms	
List	Current queuing algorithms	
Paragraph	Weighted random early detection (WRED)	

COMPLETE TABLES AND LISTS FROM MEMORY

Print a copy of Appendix B, "Memory Tables" (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. Appendix C, "Memory Tables Answer Key," also on the companion website, includes completed tables and lists you can use to check your work.

DEFINE KEY TERMS

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

802.1Q

802.1p

Differentiated Services (DiffServ)

Differentiated Services Code Point (DSCP)

per-hop behavior (PHB)

Type of Service (TOS)

REFERENCES IN THIS CHAPTER

RFC 1633, *Integrated Services in the Internet Architecture: an Overview*, R. Braden, D. Clark, S. Shenker, https://tools.ietf.org/html/rfc1633, June 1994

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, K. Nichols, S. Blake, F. Baker, D. Black, https://tools.ietf.org/html/rfc2474, December 1998

RFC 2475, *An Architecture for Differentiated Services*, S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, https://tools.ietf.org/html/rfc2475, December 1998

RFC 2597, *Assured Forwarding PHB Group*, J. Heinanen, Telia Finland, F. Baker, W. Weiss, J. Wroclawski, https://tools.ietf.org/html/rfc2597, June 1999

RFC 2697, *A Single Rate Three Color Marker*, J. Heinanen, Telia Finland, R. Guerin, IETF, https://tools.ietf.org/html/rfc2697, September 1999

RFC 2698, *A Two Rate Three Color Marker*, J. Heinanen, Telia Finland, R. Guerin, IETF, https://tools.ietf.org/html/rfc2698, September 1999

RFC 3140, *Per Hop Behavior Identification Codes*, D. Black, S. Brim, B. Carpenter, F. Le Faucheur, IETF, https://tools.ietf.org/html/rfc3140, June 2001

RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*, B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, https://tools.ietf.org/html/rfc3246, March 2002

RFC 3260, *New Terminology and Clarifications for Diffserv*, D. Grossman, IETF, https://tools.ietf.org/html/rfc3260, April 2002

RFC 3594, *Configuration Guidelines for DiffServ Service Classes*, J. Babiarz, K. Chan, F. Baker, IETF, https://tools.ietf.org/html/rfc4594, August 2006

draft-suznjevic-tsvwg-delay-limits-00, *Delay Limits for Real-Time Services*, M. Suznjevic, J. Saldana, IETF, https://tools.ietf.org/html/draft-suznjevic-tsvwg-delay-limits-00, June 2016

Chapter 15. IP Services

This chapter covers the following subjects:

- **Time Synchronization:** This section describes the need for synchronizing time in an environment and covers Network Time Protocol and its operations to keep time consistent across devices.

- **First-Hop Redundancy Protocol:** This section gives details on how multiple routers can provide resilient gateway functionality to hosts at the Layer 2/Layer 3 boundaries.

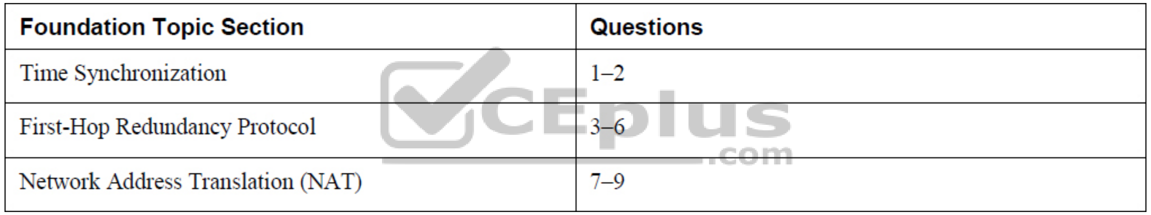
- **Network Address Translation (NAT):** This section explains how a router can translate IP addresses from one network realm to another.

In addition to routing and switching network packets, a router can perform additional functions to enhance a network. This chapter covers time synchronization, virtual gateway technologies, and Network Address Translation.

"DO I KNOW THIS ALREADY?" QUIZ

The “Do I Know This Already?” quiz allows you to assess whether you should read the entire chapter. If you miss no more than one of these self-assessment questions, you might want to move ahead to the “Exam Preparation Tasks” section. Table 15-1 lists the major headings in this chapter and the “Do I Know This Already?” quiz questions covering the material in those headings so you can assess your knowledge of these specific areas. The answers to the “Do I Know This Already?” quiz appear in Appendix A, “Answers to the ‘Do I Know This Already?’ Quiz Questions.”

Table 15-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping



Foundation Topic Section	Questions
Time Synchronization	1–2
First-Hop Redundancy Protocol	3–6
Network Address Translation (NAT)	7–9

1. NTP uses the concept of _________ to calculate the accuracy of the time source.

a. administrative distance

b. stratum

c. atomic half-life

d. deviation time

2. True or false: An NTP client can be configured with multiple NTP servers and can synchronize its local clock with all the servers.

a. True

b. False

3. In a resilient network topology, first-hop redundancy protocols (FHRP) overcome the limitations of which of the following? (Choose two.)

a. Static default routes

b. Link-state routing protocols

c. Vector based routing protocols

d. A computer with only one default gateway

4. Which of the following FHRPs are considered Cisco proprietary? (Choose two.)

a. VRRP

b. HSRP

c. GLBP

d. ODR

5. Which of the following commands defines the HSRP instance 1 VIP gateway instance 10.1.1.1?

a. standby 1 ip 10.1.1.1

b. hsrp 1 ip 10.1.1.1

c. hsrp 1 vip 10.1.1.1

d. hsrp 1 10.1.1.1

6. Which of the following FHRPs supports load balancing?

a. ODR

b. VRRP

c. HSRP

d. GLBP

7. Which command displays the translation table on a router?

a. show ip translations

b. show ip xlate

c. show xlate

d. show ip nat translations

8. A router connects multiple private networks in the 10.0.0.0/8 network range to the Internet. A user’s IP address of 10.1.1.1 is considered the __________ IP address.

a. inside local

b. inside global

c. outside local

d. outside global

9. The IP translation table times out and clears dynamic TCP connection entries from the translation table after how long?

a. 1 hour

b. 4 hours

c. 12 hours

d. 24 hours

Answers to the “Do I Know This Already?” quiz:

1. B

2. B

3. A, D

4. B, C

5. A

6. D

7. D

8. A

9. D

FOUNDATION TOPICS

Time Synchronization

A device's system time is used to measure periods of idle state or computation. Ensuring that the time is consistent on a system is important because applications often use the system time to tune internal processes. From the perspective of managing a network, it is important that the time be synchronized between network devices for several reasons:

- Managing passwords that change at specific time intervals

- Encryption key exchanges
- Checking validity of certificates based on expiration date and time
- Correlation of security-based events across multiple devices (routers, switches, firewalls, network access control systems, and so on)
- Troubleshooting network devices and correlating events to identify the root cause of an event

The rate at which a device maintains time can deviate from device to device. Even if the time was accurately set on all the devices, the time intervals could be faster on one device than on another device. Eventually the times would start to drift away from each other. Some devices use only a software clock, which is reset when the power is reset. Other devices use a hardware clock, which can maintain time when the power is reset.

Key Topic

Network Time Protocol

RFC 958 introduced Network Time Protocol (NTP), which is used to synchronize a set of network clocks in a distributed client/server architecture. NTP is a UDP-based protocol that connects with servers on port 123. The client source port is dynamic.

NTP is based on a hierarchical concept of communication. At the top of the hierarchy are authoritative devices that operate as an NTP server with an atomic clock. The NTP client then queries the NTP server for its time and updates its time based on the response. Because NTP is considered an application, the query can occur over multiple hops, requiring NTP clients to identify the time accuracy based on messages with other routers.

The NTP synchronization process is not fast. In general, an NTP client can synchronize a large time discrepancy to within a couple seconds of accuracy with a few cycles of polling an NTP server. However, gaining accuracy of tens of milliseconds requires hours or days of comparisons. In some ways, the time of the NTP clients drifts toward the time of the NTP server.

Key Topic

NTP uses the concept of stratums to identify the accuracy of the time clock source. NTP servers that are directly attached to an authoritative time source are stratum 1 servers. An NTP client that queries a stratum 1 server is considered a stratum 2 client. The higher the stratum, the greater the chance of deviation in time from the authoritative time source due to the number of time drifts between the NTP stratums.

Figure 15-1 demonstrates the concept of stratums, with R1 attached to an atomic clock and considered a stratum 1 server. R2 is configured to query R1, so it is considered a stratum 2 client. R3 is configured to query R2, so it is considered a stratum 3 client. This could continue until stratum 15. Notice that R4 is configured to query R1 over multiple hops, and it is therefore considered a stratum 2 client.

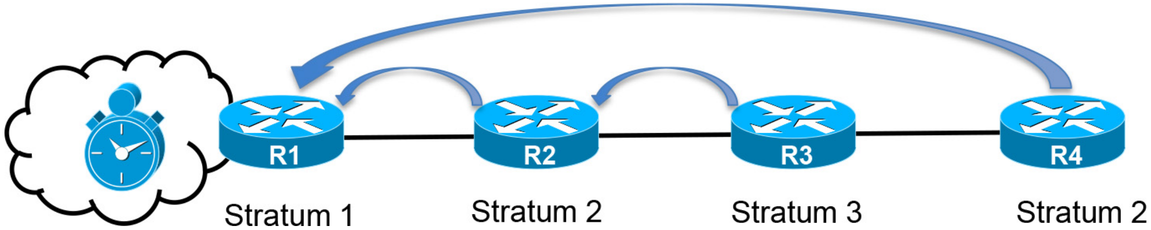


Figure 15-1 NTP Stratums

NTP Configuration

The configuration of an NTP client is pretty straightforward. The client configuration uses the global configuration command **ntp server** *ip-address* [**prefer**] [**source** *interface-id*]. The source interface, which is optional, is used to stipulate the source IP address for queries for that server. Multiple NTP servers can be configured for redundancy, and adding the optional **prefer** keyword indicates which NTP server time synchronization should come from.

Cisco devices can act as a server after they have been able to query an NTP server. For example, in Figure 15-1, once R2 has synchronized time with R1 (a stratum 1 time source), R2 can act as a server to R3. Configuration of external clocks is beyond the scope of this book. However, you should know that you can use the command **ntp master** *stratum-number* to statically set the stratum for a device when it acts as an NTP server.

Example 15-1 demonstrates the configuration of R1, R2, R3, and R4 from Figure 15-1.

Example 15-1 Simple Multi-Stratum NTP Configuration

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ntp master 1

R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ntp server 192.168.1.1

R3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# ntp server 192.168.2.2 source loopback 0

R4# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)# ntp server 192.168.1.1
```

To view the status of NTP service, use the command **show ntp status**, which has the following output in Example 15-2.

1. Whether the hardware clock is synchronized to the software clock (that is, whether the clock resets during power reset), the stratum reference of the local device, and the reference clock identifier (local or IP address)

2. The frequency and precision of the clock

3. The NTP uptime and granularity

4. The reference time

5. The clock offset and delay between the client and the lower-level stratum server

6. Root dispersion (that is, the calculated error of the actual clock attached to the atomic clock) and peer dispersion (that is, the root dispersion plus the estimated time to reach the root NTP server)

7. NTP loopfilter (which is beyond the scope of this book)

8. Polling interval and time since last update

Example 15-2 shows the output of the NTP status from R1, R2, and R3. Notice that the stratum has incremented, along with the reference clock.

Example 15-2 Viewing NTP Status

```
R1# show ntp status
Clock is synchronized, stratum 1, reference is .LOCL.
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precisio
ntp uptime is 2893800 (1/100 of seconds), resolution is 4000
reference time is E0E2D211.E353FA40 (07:48:17.888 EST Wed Jul 24
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 2.24 msec, peer dispersion is 1.20 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0
system poll interval is 16, last update was 4 sec ago.
```

```
R2# show ntp status
Clock is synchronized, stratum 2, reference is 192.168.1.1
nominal freq is 250.0000 Hz, actual freq is 249.8750 Hz, precisio
ntp uptime is 2890200 (1/100 of seconds), resolution is 4016
reference time is E0E2CD87.28B45C3E (07:28:55.159 EST Wed Jul 24
clock offset is 1192351.4980 msec, root delay is 1.00 msec
root dispersion is 1200293.33 msec, peer dispersion is 7938.47 ms
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 1 sec ago.

R3# show ntp status
Clock is synchronized, stratum 3, reference is 192.168.2.2
nominal freq is 250.0000 Hz, actual freq is 250.0030 Hz, precisio
ntp uptime is 28974300 (1/100 of seconds), resolution is 4000
reference time is E0E2CED8.E147B080 (07:34:32.880 EST Wed Jul 24
clock offset is 0.5000 msec, root delay is 2.90 msec
root dispersion is 4384.26 msec, peer dispersion is 3939.33 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0
system poll interval is 64, last update was 36 sec ago.
```

A streamlined version of the NTP server status and delay is provided with the command **show ntp associations**. The address 127.127.1.1 reflects to the local device when configured with the **ntp master** *stratum-number* command. Example 15-3 shows the NTP associations for R1, R2, and R3.

Example 15-3 Viewing the NTP Associations

```
R1# show ntp associations

  address         ref clock       st   when   poll reach  delay
```

```
*~127.127.1.1     .LOCL.  0      0     16   377  0.000   0.000
 * sys.peer, # selected, + candidate, - outlyer, x falseticker,

SW1# show ntp associations

  address         ref clock       st   when   poll reach  delay
*~192.168.1.1     127.127.1.1      1    115   1024     1  1.914
 * sys.peer, # selected, + candidate, - outlyer, x falseticker,

SW2# show ntp associations

  address         ref clock       st   when   poll reach  delay
*~192.168.2.2     192.168.1.1      2     24     64     1  1.000
 * sys.peer, # selected, + candidate, - outlyer, x falseticker,
```

Key Topic

Stratum Preference

An NTP client can be configured with multiple NTP servers. The device will use only the NTP server with the lowest stratum. The top portion of Figure 15-2 shows R4 with two NTP sessions: one session with R1 and another with R3.

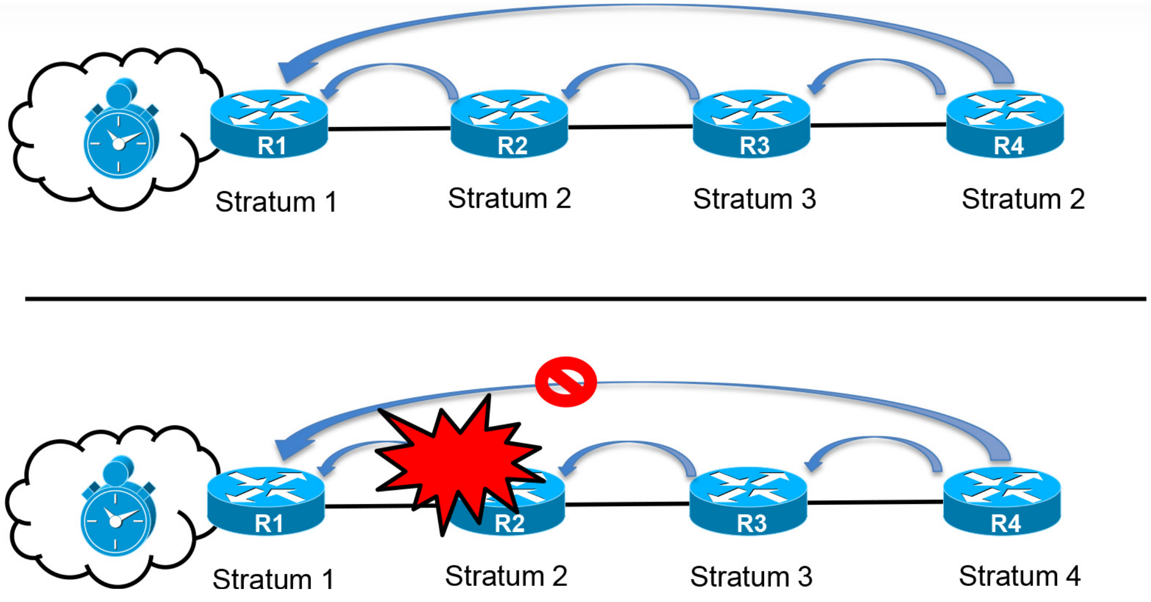


Figure 15-2 NTP Stratum Preferences

In the topology shown in Figure 15-2, R4 will always use R1 for synchronizing its time because it is a stratum 1 server. If R2 crashes, as shown at the bottom of Figure 15-2, preventing R4 from reaching R1, it synchronizes with R3's time (which may or may not be different due to time drift) and turns into a stratum 4 time device. When R2 recovers, R4 synchronizes with R1 and becomes a stratum 2 device again.

Key Topic

NTP Peers

Within the NTP client architecture, the NTP client changes its time to the time of the NTP server. The NTP server does not change its time to reflect the clients. Most enterprise organizations (such as universities, governments, and pool.ntp.org) use an external NTP servers. A common scenario is to designate two devices to query a different external NTP source and then to peer their local stratum 2 NTP devices.

NTP peers act as clients and servers to each other, in the sense that they try to blend their time to each other. The NTP peer model is intended for designs where other devices can act as backup devices for each other and use different primary reference sources.

Figure 15-3 shows a scenario where R1 is an NTP client to 100.64.1.1, and R2 is an NTP client to 100.64.2.2. R1 and R2 are NTP peers with each other, so they query each other and move their time toward each other.

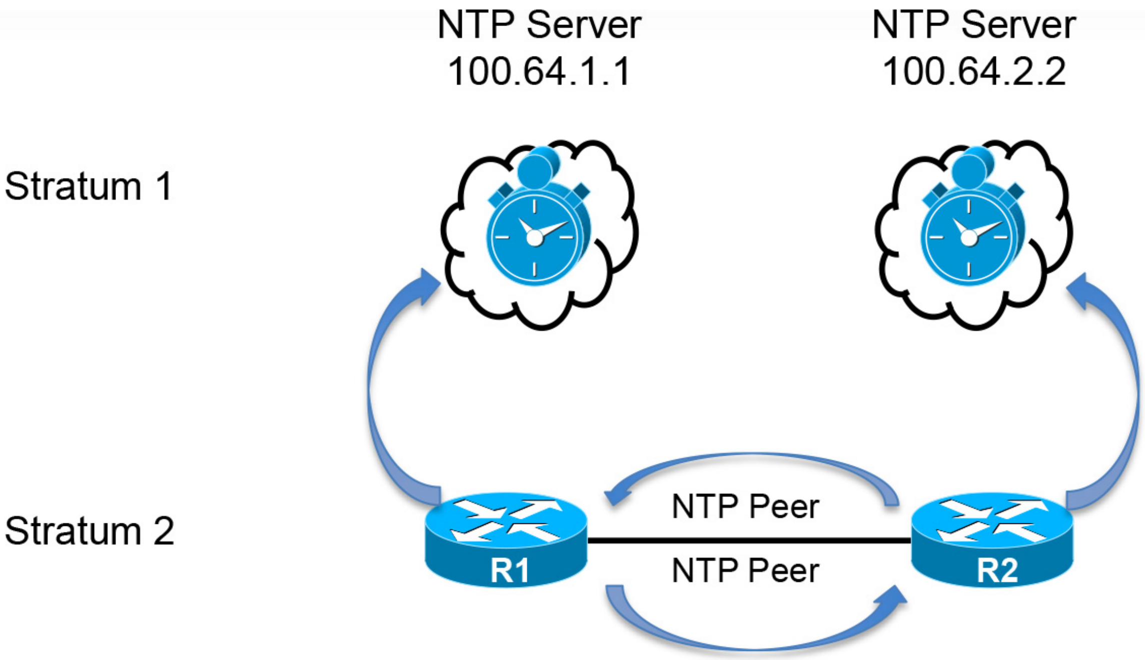


Figure 15-3 NTP Stratums

Note

An NTP peer that is configured with an authoritative time source treats its peer as an equal and shifts its clock to synchronize with the peer. The peers adjust at a maximum rate of two minutes per query, so large discrepancies take some time to correct.

NTP peers are configured with the command **ntp peer** *ip-address*. Example 15-4 shows the sample NTP peer configuration for R1 and R2 (refer to Figure 15-3) peering with their loopback interfaces.

Example 15-4 NTP Peer Configuration

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# ntp peer 192.168.2.2

R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# ntp peer 192.168.1.1
```

FIRST-HOP REDUNDANCY PROTOCOL

Network resiliency is a key component of network design. Resiliency with Layer 2 forwarding is accomplished by adding multiple Layer 2 switches into a topology. Resiliency with Layer 3 forwarding is accomplished by adding multiple Layer 3 paths or routers.

Figure 15-4 shows the concept of adding resiliency by using multiple Layer 2 switches and routers on the left or by adding resiliency with multiple multi-layer switches on the right. In both scenarios:

- Two devices (172.16.1.2 and 172.16.1.3) can be the PC’s gateway.

- There are two resilient Layer 2 links that connect SW6 to a switch that can connect the PC to either gateway.

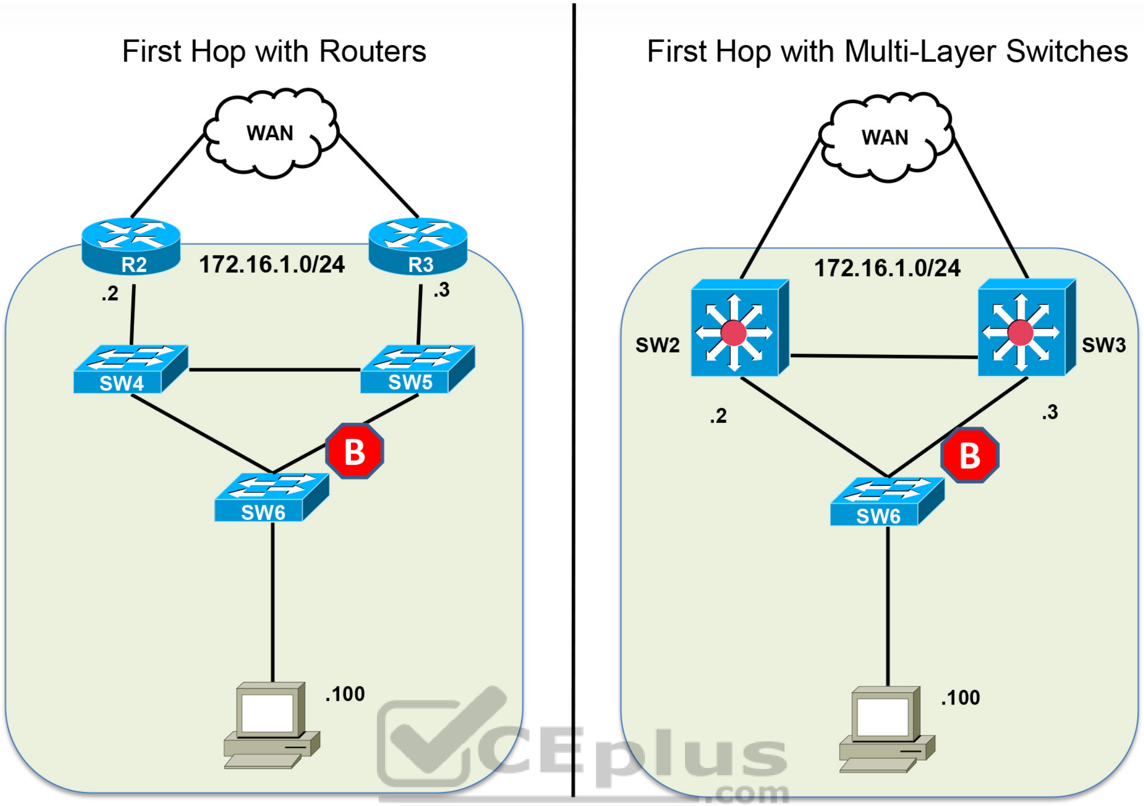


Figure 15-4 Resiliency with Redundancy with Layer 2 and Layer 3 Devices

Note

STP is blocking traffic between SW6 and SW5 on the left and between SW6 and SW3 on the right in Figure 15-4.

The PC could configure its gateway as 172.16.1.2, but what happens when that device fails? The same problem occurs if the other gateway was configured. How can a host be configured with more than one gateway? Some operating systems support the configuration of multiple gateways, and others do not. Providing gateway accessibility to all devices is very important.

Key Topic

The deployment of first-hop redundancy protocols (FHRP)s solves the problem of hosts configuring multiple gateways. FHRPs work by creating a virtual IP (VIP) gateway instance that is shared between the Layer 3 devices. This book covers the following FHRPs:

- Hot Standby Router Protocol (HSRP)

- Virtual Router Redundancy Protocol (VRRP)

- Gateway Load Balancing Protocol (GLBP)

Object Tracking

FHRPs are deployed in a network for reliability and high availability to ensure load balancing and failover capability in case of a router failover. To ensure optimal traffic flow when a WAN link goes down, it would be nice to be able to determine the availability of routes or the interface state to which FHRP route traffic is directed.

Object tracking offers a flexible and customizable mechanism for linking with FHRPs and other routing components (for example, conditional installation of a static route). With this feature, users can track specific objects in the network and take necessary action when any object's state change affects network traffic.

Figure 15-5 shows a simple topology with three routers exchanging routes with EIGRP and advertising their loopback interfaces to EIGRP.

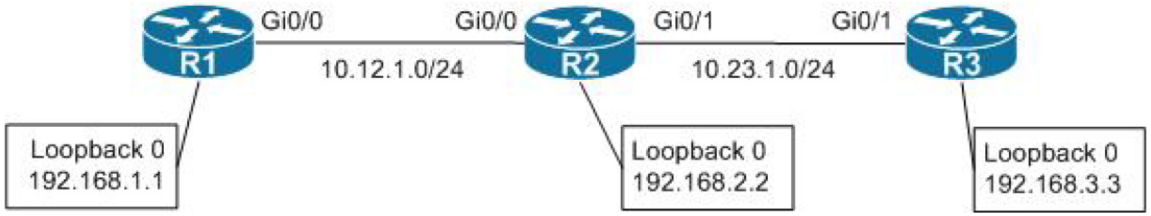


Figure 15-5 Object Tracking

Tracking of routes in the routing table is accomplished with the command **track** *object-number* **ip route** *route/prefix-length* **reachability**. The status object tracking can be viewed with the command **show track** [*object-number*].

Example 15-5 shows R1 being configured for tracking the route to R3's loopback interface. The route is installed in R1's RIB, and the tracked object state is up.

Example 15-5 Tracking R3's Loopback Interface

```
R1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# track 1 ip route 192.168.3.3/32 reachability
```

```
R1# show track
Track 1
  IP route 192.168.3.3 255.255.255.255 reachability
  Reachability is Up (EIGRP)
    1 change, last change 00:00:32
  First-hop interface is GigabitEthernGi0/0
```

Tracking of an interface’s line protocol state is accomplished with the command **track** *object-number* **interface** *interface-id* **line-protocol**.

Example 15-6 shows R2 being configured for tracking the Gi0/1 interface toward R3. The line protocol for the interface is up.

Example 15-6 Tracking R2’s Gi0/1 Interface Line Protocol State

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# track 2 interface GigabitEthernGi0/1 line-protocol

R2# show track
Track 2
  Interface GigabitEthernGi0/1 line-protocol
  Line protocol is Up
    1 change, last change 00:00:37
```

Shutting down R2’s Gi0/1 interface should change the tracked object state on R1 and R2 to a down state. Example 15-7 shows the shutdown of R2’s Gi0/1 interface. Notice that the tracked state for R2 and R1 changed shortly after the interface was shut down.

Example 15-7 Demonstrating a Change of Tracked State

```
R2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# interface GigabitEthernGi0/1
R2(config-if)# shutdown
*03:04:18.975: %TRACK-6-STATE: 2 interface Gi0/1 line-protocol Up
*03:04:18.980: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.23
03:04:20.976: %LINK-5-CHANGED: Interface GigabitEthernGi0/1, char
* 03:04:21.980: %LINEPROTO-5-UPDOWN: Line protocol on Interface G

R1#
03:04:24.007: %TRACK-6-STATE: 1 ip route 192.168.3.3/32 reachabil
```

Example 15-8 shows the current track state for R1 and R2. R1 no longer has the 192.168.3.3/32 network in the RIB, and R2's Gi0/1 interface is in shutdown state.

Example 15-8 Viewing the Track State After a Change

```
R1# show track
Track 1
  IP route 192.168.3.3 255.255.255.255 reachability
  Reachability is Down (no ip route)
    2 changes, last change 00:02:09
  First-hop interface is unknown
```

```
R2#

Track 2
  Interface GigabitEthernGi0/1 line-protocol
  Line protocol is Down ((hw admin-down))
    2 changes, last change 00:01:58
```

Object tracking works with protocols such as Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and Gateway Load Balancing Protocol (GLBP) so that they take action when the state of an object changes. FHRP commonly tracks the availability of the WAN interface or the existence of a route learned via that next hop.

Key Topic

Hot Standby Router Protocol

Hot Standby Routing Protocol (HSRP) is a Cisco proprietary protocol that provides transparent failover of the first-hop device, which typically acts as a gateway to the hosts.

HSRP provides routing redundancy for IP hosts on an Ethernet network configured with a default gateway IP address. A minimum of two devices are required to enable HSRP: One device acts as the active device and takes care of forwarding the packets, and the other acts as a standby that is ready to take over the role of active device in the event of a failure.

On a network segment, a virtual IP address is configured on each HSRP-enabled interface that belong to the same HSRP group. HSRP selects one of the interfaces to act as the HSRP active router. Along with the virtual IP address, a virtual MAC address is assigned for the group. The active router receives and routes the packets destined for the virtual MAC address of the group.

When the HSRP active router fails, the HSRP standby router assumes control of the virtual IP address and virtual MAC address of the group. The HSRP election selects the router with the highest priority (which defaults to 100). In the event of a tie in priority, the router with the highest IP address for the network segment is preferred.

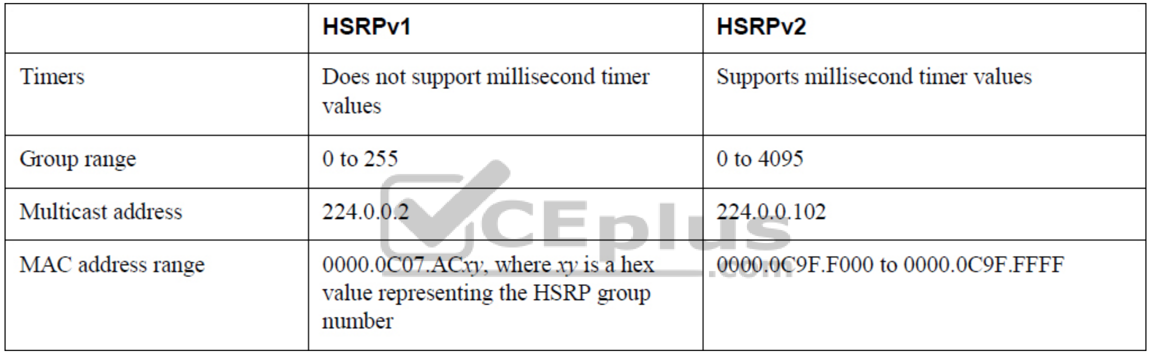
Note

HSRP does not support preemption by default, so when a router with lower priority becomes active, it does not automatically transfer its active status to a superior router.

HSRP-enabled interfaces send and receive multicast UDP-based hello messages to detect any failure and designate active and standby routers. If a standby device does not receive a hello message or the active device fails to send a hello message, the standby device with the second highest priority becomes HSRP active. The transition of HSRP active between the devices is transparent to all hosts on the segment because the MAC address moves with the virtual IP address.

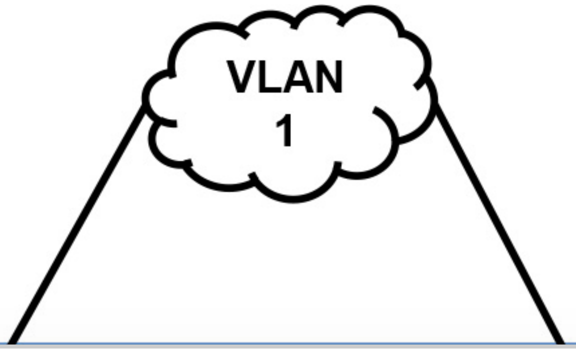
HSRP has two versions: Version 1 and Version 2. Table 15-2 shows some of the differences between HSRPv1 and HSRPv2:

Table 15-2 HSRP Versions



	HSRPv1	HSRPv2
Timers	Does not support millisecond timer values	Supports millisecond timer values
Group range	0 to 255	0 to 4095
Multicast address	224.0.0.2	224.0.0.102
MAC address range	0000.0C07.ACxy, where xy is a hex value representing the HSRP group number	0000.0C9F.F000 to 0000.0C9F.FFFF

Figure 15-6 shows a sample topology where SW2 and SW3 are the current gateway devices for VLAN 10. VLAN 1 provides transit routing to the WAN routers.



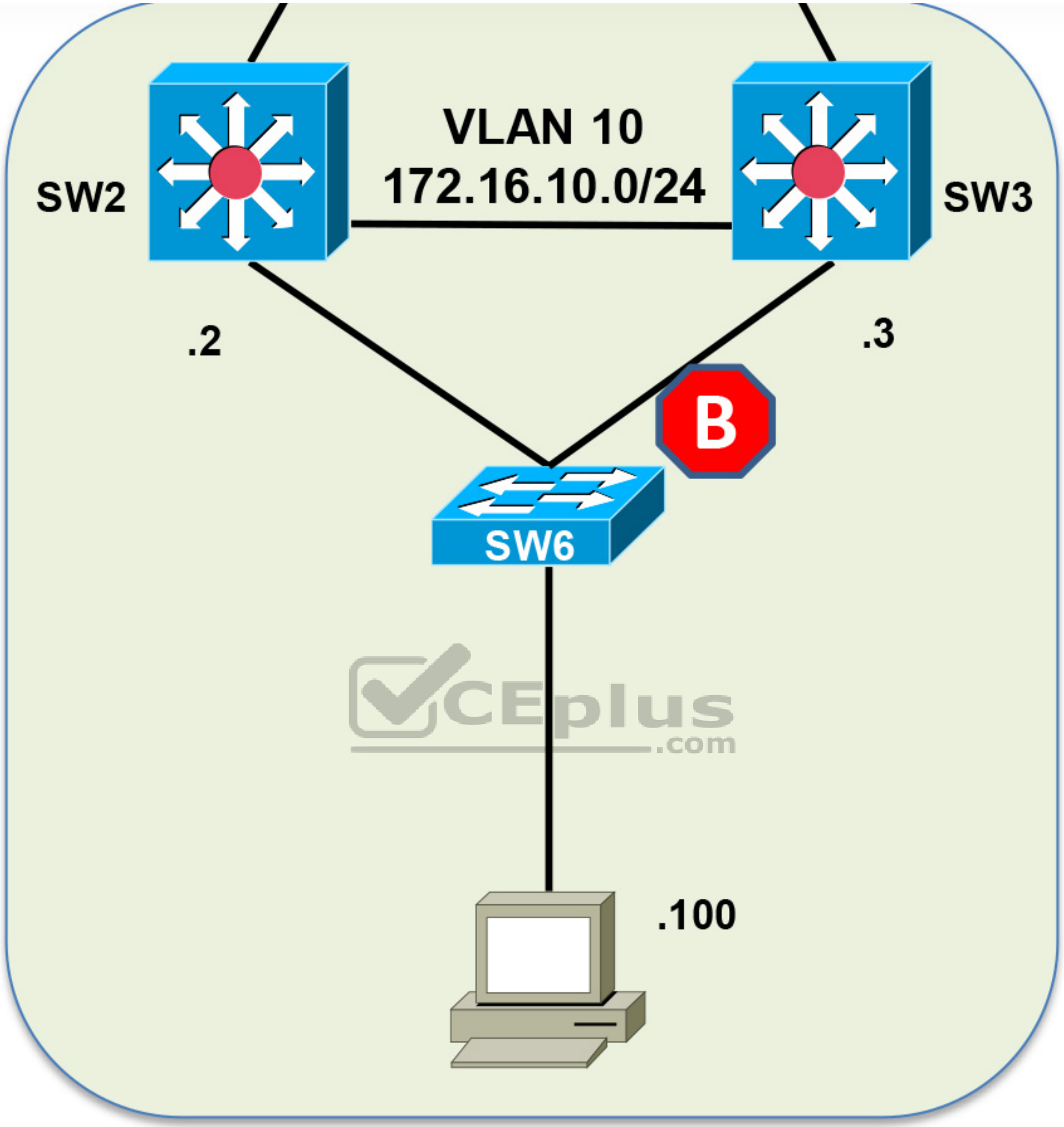


Figure 15-6 Sample HSRP Topology

The following steps show how to configure an HSRP virtual IP (VIP) gateway instance:

Key Topic

Step 1. Define the HSRP instance by using the command **standby** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Configure HSRP router pre-emption to allow a more preferred router to take the active router status from an inferior active HSRP router. Enable preemption with the command **standby** *instance-id* **preempt**.

Step 3. (Optional) Define the HSRP priority by using the command **standby** *instance-id* **priority** *priority*. The priority is a value between 0 and 255.

Step 4. Define the HSRP MAC Address (Optional)

The MAC address can be set with the command **standby** *instance-id* **mac-address** *mac-address*. Most organizations accept the automatically generated MAC address, but in some migration scenarios, the MAC address needs to be statically set to ease transitions when the hosts may have a different MAC address in their ARP table.

Step 5. (Optional) Define the HSRP timers by using the command **standby** *instance-id* **timers** {*seconds* | **msec** *milliseconds*}. HSRP can poll in intervals of 1 to 254 seconds or 15 to 999 milliseconds.

Step 6. (Optional) Establish HSRP authentication by using the command **standby** *instance-id* **authentication** {*text-password* | **text** *text-password* | **md5** {**key-chain** *key-chain* | **key-string** *key-string*}}.

> **Note**
>
> It is possible to create multiple HSRP instances for the same interface. Some network architects configure half of the hosts for one instance and the other half of the hosts for a second instance. Setting different priorities for each instance makes it possible to load balance the traffic across multiple routers.

Example 15-9 shows a basic HSRP configuration for VLAN 10 on SW1 and SW2, using the HSRP instance 10 and the VIP gateway instance 172.16.10.1. Notice that once preemption was enabled, that SW3 became the active speaker, and SW2 became the standby speaker.

Example 15-9 Simple HSRP Configuration

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface vlan 10
03:55:35.148: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vla
SW2(config-if)# ip address 172.16.10.2 255.255.255.0
SW2(config-if)# standby 10 ip 172.16.10.1
03:56:00.097: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> S
```

```
SW2(config-if)# standby 10 preempt

SW3(config)# interface vlan 10
03:56:04.478: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vla
SW3(config-if)# ip address 172.16.10.3 255.255.255.0
SW3(config-if)# standby 10 ip 172.16.10.1
SW1(config-if)# standby 10 preempt
03:58:22.113: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby ->
```

The HSRP status can be viewed with the command **show standby** [*interface-id*] [**brief**]. Specifying an interface restricts the output to a specific interface; this can be useful when troubleshooting large amounts of information.

Example 15-10 shows the command **show standby brief** being run on SW2, which includes the interfaces and the associated groups that are running HSRP. The output also includes the local interface's priority, whether preemption is enabled, the current state, the active speaker's address, the standby speaker's address, and the VIP gateway instance for that standby group.

Example 15-10 Viewing the Summarized HSRP State

```
SW2# show standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp  Pri P State   Active          Standby         Vi
Vl10        10   100 P Standby 172.16.10.3     local           17

SW3# show standby brief
```

```
                        P indicates configured to preempt.
                        |
Interface    Grp  Pri P State   Active          Standby         Vi
Vl10         10   100 P Active  local           172.16.10.2     17
```

The non-brief iteration of the **show standby** command also includes the number of state changes for the HSRP instance, along with the time since the last state change, the timers, and a group name, as shown in Example 15-11.

Example 15-11 Viewing the HSRP State

```
SW2# show standby
Vlan10 - Group 10
  State is Standby
    9 state changes, last state change 00:13:12
  Virtual IP address is 172.16.10.1
  Active virtual MAC address is 0000.0c07.ac0a (MAC Not In Use)
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.736 secs
  Preemption enabled
  Active router is 172.16.10.3, priority 100 (expires in 10.032 s
  Standby router is local
  Priority 100 (default 100)
  Group name is "hsrp-Vl10-10" (default)

SW3# show standby
Vlan10 - Group 10
  State is Active
    5 state changes, last state change 00:20:01
  Virtual IP address is 172.16.10.1
  Active virtual MAC address is 0000.0c07.ac0a (MAC In Use)
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.024 secs
  Preemption enabled
  Active router is local
  Standby router is 172.16.10.2, priority 100 (expires in 11.296
  Priority 100 (default 100)
  Group name is "hsrp-Vl10-10" (default)
```

Key Topic

HSRP provides the capability to link object tracking to priority. For example, assume that traffic should flow through SW2's WAN connection whenever feasible. Traffic can be routed by SW3 to SW2 and then on to SW2's WAN connection; however, making SW2 the VIP gateway streamlines the process. But when SW2 loses its link to the WAN, it should move the HSRP active speaker role to SW3.

This configuration is accomplished as follows:

- Configure a tracked object to SW2's WAN link (in this example, VLAN 1).

- Change SW2's priority to a value higher than SW3 (in this case, 110).

- Configure SW2 to lower the priority if the tracked object state changes to down. This is accomplished with the command **standby** *instance-id* **track** *object-id* **decrement** *decrement-value*. The decrement value should be high enough so that when it is removed from the priority; the value is lower than that of the other HSRP router.

Example 15-12 shows the configuration of SW2 where a tracked object is created against VLAN 1's interface line protocol, increasing the HSRP priority to 110, and linking HSRP to the tracked object so that the priority decrements by 20 if interface VLAN 1 goes down.

Example 15-12 Correlating HSRP to Tracked Objects

```
SW2(config)# track 1 interface vlan 1 line-protocol
SW2(config-track)# interface vlan 10
SW2(config-if)# standby 10 priority 110
04:44:16.973: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby ->
SW2(config-if)# standby 10 track 1 decrement 20
```

Example 15-13 shows that the HSRP group on VLAN 10 on SW2 correlates the status of the tracked object for the VLAN 1 interface.

Example 15-13 Verifying the Linkage of HSRP to Tracked Objects

```
SW2# show standby
! Output omitted for brevity
```

```
Vlan10 - Group 10
  State is Active
    10 state changes, last state change 00:06:12
  Virtual IP address is 172.16.10.1
..
  Preemption enabled
  Active router is local
  Standby router is 172.16.10.3, priority 100 (expires in 9.856 s
  Priority 110 (configured 110)
    Track object 1 state Up decrement 20
```

Example 15-14 verifies the anticipated behavior by shutting down the VLAN 1 interface on SW2. The syslog messages indicate that the object track state changed immediately after the interface was shut down, and shortly thereafter, the HSRP role changed to a standby state. The priority was modified to 90 because of the failure in object tracking, making SW2’s interface less preferred to SW3’s interface of 100.

Example 15-14 Verifying the Change of HSRP State with Object Tracking

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface vlan 1
SW2(config-if)# shut
 04:53:16.490: %TRACK-6-STATE: 1 interface Vl1 line-protocol Up -
 04:53:17.077: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Active ->
 04:53:18.486: %LINK-5-CHANGED: Interface Vlan1, changed state to
 04:53:19.488: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vl
 04:53:28.267: %HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak ->
```

```
SW2# show standby
! Output omitted for brevity
Vlan10 - Group 10
  State is Standby
    12 state changes, last state change 00:00:39
..
  Active router is 172.16.10.3, priority 100 (expires in 9.488 se
  Standby router is local
  Priority 90 (configured 110)
    Track object 1 state Down decrement 20
  Group name is "hsrp-Vl10-10" (default)
```

Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) is an industry standard and operates similarly to HSRP. The behavior of VRRP is so close to that of HSRP that the following differences should be noted:

- The preferred active router controlling the VIP gateway is called the *master router*. All other VRRP routers are known as *backup routers*.

- VRRP enables preemption by default.

- The MAC address of the VIP gateway uses the structure 0000.5e00.01*xx*, where *xx* reflects the group ID in hex.

- VRRP uses the multicast address 224.0.0.18 for communication.

There are currently two versions of VRRP:

- **VRRPv2:** Supports IPv4

- **VRRPv3:** Supports IPv4 and IPv6

The following sections review these versions.

Legacy VRRP Configuration

Early VRRP configuration supported only VRRPv2 and was non-hierarchical in its configuration. The following steps are used for configuring older software versions with VRRP:

Key Topic

Step 1. Define the VRRP instance by using the command **vrrp** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Define the VRRP priority by using the command **vrrp** *instance-id* **priority** *priority*. The priority is a value between 0 and 255.

Step 3. (Optional) Enable object tracking so that the priority is decremented when the object is false. Do so by using the command **vrrp** *instance-id* **track** *object-id* **decrement** *decrement-value*. The decrement value should be high enough so that when it is removed from the priority, the value is lower than that of the other VRRP router.

Step 4. (Optional) Establish VRRP authentication by using the command **vrrp** *instance-id* **authentication** {*text-password* | **text** *text-password* | **md5** {**key-chain** *key-chain* | **key-string** *key-string*}}.

R2 and R3 are two routes that share connect to a Layer 2 switch with their Gi0/0 interfaces, which both are on the 172.16.20.0/24 network. R2 and R3 use VRRP to create the VIP gateway 172.16.20.1.

Example 15-15 shows the configuration. Notice that after the VIP is assigned to R3, R3 preempts R2 and becomes the master.

Example 15-15 Legacy VRRP Configuration

```
R2# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# interface GigabitEthernet 0/0
R2(config-if)# ip address 172.16.20.2 255.255.2
R2(config-if)# vrrp 20 ip 172.16.20.1
 04:32:14.109: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Ba
 04:32:14.113: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Ba
 04:32:17.728: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Backup ->
 04:32:47.170: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Master ->
```

```
R3# configure term
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# interface GigabitEthernGi0/0
R3(config-if)# ip add 172.16.20.3 255.255.255.0
 04:32:43.550: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Ba
 04:32:43.554: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Init -> Ba
 04:32:47.170: %VRRP-6-STATECHANGE: Gi0/0 Grp 20 state Backup ->
```

The command **show vrrp** [**brief**] provides an update on the VRRP group, along with other relevant information for troubleshooting. Example 15-16 demonstrates the brief iteration of the command. All the output is very similar to output with HSRP.

Example 15-16 Viewing the Summarized VRRP State

```
R2# show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr
Gi0/0              20  100 3609       Y  Backup  172.16.20.3

R3# show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr
Gi0/0              20  100 3609       Y  Master  172.16.20.3
```

Example 15-17 examines the detailed state of VRRP running on R2.

Example 15-17 Viewing the Detailed VRRP State

```
R2# show vrrp
EthernGi0/0 - Group 20
  State is Backup
  Virtual IP address is 172.16.20.1
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 172.16.20.3, priority is 100
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec (expires in 2.904 sec)
```

Hierarchical VRRP Configuration

The newer version of IOS XE software provides configuration of VRRP in a multi-address format that is hierarchical. The steps for configuring hierarchical VRRP are as follows:

Key Topic

Step 1. Enable VRRPv3 on the router by using the command **fhrp version vrrp v3**.

Step 2. Define the VRRP instance by using the command **vrrp** *instance-id* **address-family** {**ipv4** | **ipv6**}. This places the configuration prompt into the VRRP group for additional configuration.

Step 3. (Optional) Change VRRP to Version 2 by using the command **vrrpv2**. VRRPv2 and VRRPv3 are not compatible.

Step 4. Define the gateway VIP by using the command **address** *ip-address*.

Step 5. (Optional) Define the VRRP priority by using the command **priority** *priority*. The priority is a value between 0 and 255.

Step 6. (Optional) Enable object tracking so that the priority is decremented when the object is false. Do so by using the command **track** *object-id* **decrement** *decrement-value*. The decrement value should be high enough so that when it is removed from the priority, the value is lower than that of the other VRRP router.

Example 15-18 shows the VRRP configuration on a pair of switches running 16.9.2 for VLAN 22 (172.16.22.0/24). The configuration looks similar to the previous VRRP configuration except that it is hierarchical. Associating parameters like priority and tracking are nested under the VRRP instance.

Example 15-18 Configuring Hierarchical VRRP Configuration

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# fhrp version vrrp v3
SW2(config)# interface vlan 22
 19:45:37.385: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vl
 state to up
SW2(config-if)# ip address 172.16.22.2 255.255.255.0
SW2(config-if)# vrrp 22 address-family ipv4
SW2(config-if-vrrp)# address 172.16.22.1
```

```
SW2(config-if-vrrp)# track 1 decrement 20
SW2(config-if-vrrp)# priority 110
SW2(config-if-vrrp)# track 1 decrement 20
 19:48:00.338: %VRRP-6-STATE: Vlan22 IPv4 group 22 state INIT ->
 19:48:03.948: %VRRP-6-STATE: Vlan22 IPv4 group 22 state BACKUP

SW3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW3(config)# fhrp version vrrp v3
SW3(config)# interface vlan 22
 19:46:13.798: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vl
SW3(config-if)# ip address 172.16.22.3 255.255.255.0
SW3(config-if)# vrrp 22 address-family ipv4
SW3(config-if-vrrp)# address 172.16.22.1
 19:48:08.415: %VRRP-6-STATE: Vlan22 IPv4 group 22 state INIT ->
```

The status of the VRRP routers can be viewed with the command **show vrrp** [**brief**]. The output is identical to that of the legacy VRRP configuration, as shown in Example 15-19.

Example 15-19 Viewing Hierarchical VRRP State

```
SW2# show vrrp brief
  Interface          Grp  A-F Pri  Time Own Pre State   Master ad
  Vl22                22 IPv4 110     0  N   Y  MASTER  172.16.22

SW2# show vrrp

Vlan22 - Group 22 - Address-Family IPv4
  State is MASTER
```

```
     State duration 51.640 secs
     Virtual IP address is 172.16.22.1
     Virtual MAC address is 0000.5E00.0116
     Advertisement interval is 1000 msec
     Preemption enabled
     Priority is 110
       Track object 1 state UP decrement 20
     Master Router is 172.16.22.2 (local), priority is 110
     Master Advertisement interval is 1000 msec (expires in 564 mse
     Master Down interval is unknown
     FLAGS: 1/1
```

Key Topic

Global Load Balancing Protocol

As the name suggests, Gateway Load Balancing Protocol (GLBP) provides gateway redundancy and load-balancing capability to a network segment. It provides redundancy with an active/standby gateway, and it provides load-balancing capability by ensuring that each member of the GLBP group takes care of forwarding the traffic to the appropriate gateway.

The GLBP contains two roles:

- **Active virtual gateway (AVG):** The participating routers elect one AVG per GLBP group to respond to initial ARP requests for the VIP. For example, when a local PC sends an ARP request for the VIP, the AVG is responsible for replying to the ARP request with the virtual MAC address of the AVF.

- **Active virtual forwarder (AVF):** The AVF routes traffic received from assigned hosts. A unique virtual MAC address is created and assigned by the AVG to the AVFs. The AVF is assigned to a host when the AVG replies to the ARP request with the assigned AVF's virtual MAC address. ARP replies are unicast and are not heard by other hosts on that broadcast segment. When a host sends traffic to the virtual AVF MAC, the current router is responsible for routing it to the appropriate network. The AVFs are also recognized as *Fwd* instances on the routers.

GLBP supports four active AVFs and one AVG per GLBP group. A router can be an AVG and an AVF at the same time. In the event of a failure of the AVG, there is not a disruption of traffic due to the AVG role transferring to a standby AVG device. In the event of a failure of an AVF, another router takes over the forwarding responsibilities for that AVF, which includes the virtual MAC address for that instance.

Key Topic

The following steps detail how to configure a GLBP:

Step 1. Define the GLBP instance by using the command **glbp** *instance-id* **ip** *vip-address*.

Step 2. (Optional) Configure GLBP preemption to allow for a more preferred router to take the active virtual gateway status from an inferior active GLBP router. Preemption is enabled with the command **glbp** *instance-id* **preempt**.

Step 3. (Optional) Define the GLBP priority by using the command **glbp** *instance-id* **priority** *priority*. The priority is a value between 0 and 255.

Step 4. (Optional) Define the GLBP timers by using the command **glbp** *instance-id* **timers** {*hello-seconds* | **msec** *hello-milliseconds*} {*hold-seconds* | **msec** *hold-milliseconds*}.

Step 5. (Optional) Establish GLBP authentication by using the command **glbp** *instance-id* **authentication** {**text** *text-password* | **md5** {**key-chain** *key-chain* | **key-string** *key-string*}}.

SW2 and SW3 configure GLBP for VLAN 30 (172.16.30.0/24), with 172.16.30.1 as the VIP gateway. Example 15-20 demonstrates the configuration of both switches. Notice that the first syslog message on SW2 is for the AVG, and the second syslog message is for the first AVF (Fwd 1) for the GLBP pair. The first syslog message on SW3 is the second AVF (Fwd 2) for the GLBP pair.

Example 15-20 Basic GLBP Configuration

```
SW2# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)# interface vlan 30
SW2(config-if)# ip address 172.16.30.2 255.255.255.0
SW2(config-if)# glbp 30 ip 172.16.30.1
  05:41:15.802: %GLBP-6-STATECHANGE: Vlan30 Grp 30 state Speak ->
SW2(config-if)#
  05:41:25.938: %GLBP-6-FWDSTATECHANGE: Vlan30 Grp 30 Fwd 1 state
SW2(config-if)# glbp 30 preempt

SW3# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW3(config)# interface vlan 30
SW3(config-if)# ip address 172.16.30.3 255.255.255.0
SW3(config-if)# glbp 30 ip 172.16.30.1
 05:41:32.239: %GLBP-6-FWDSTATECHANGE: Vlan30 Grp 30 Fwd 2 state
SW3(config-if)# glbp 30 preempt
```

The command **show glbp brief** shows high-level details of the GLBP group, including the interface, group, active AVG, standby AVG, and statuses of the AVFs.

Example 15-21 demonstrates the commands run on SW2 and SW3. The first entry contains a **-** for the Fwd state, which means that it is the entry for the AVG. The following two entries are for the AVF instances; they identify which device is active for each AVF.

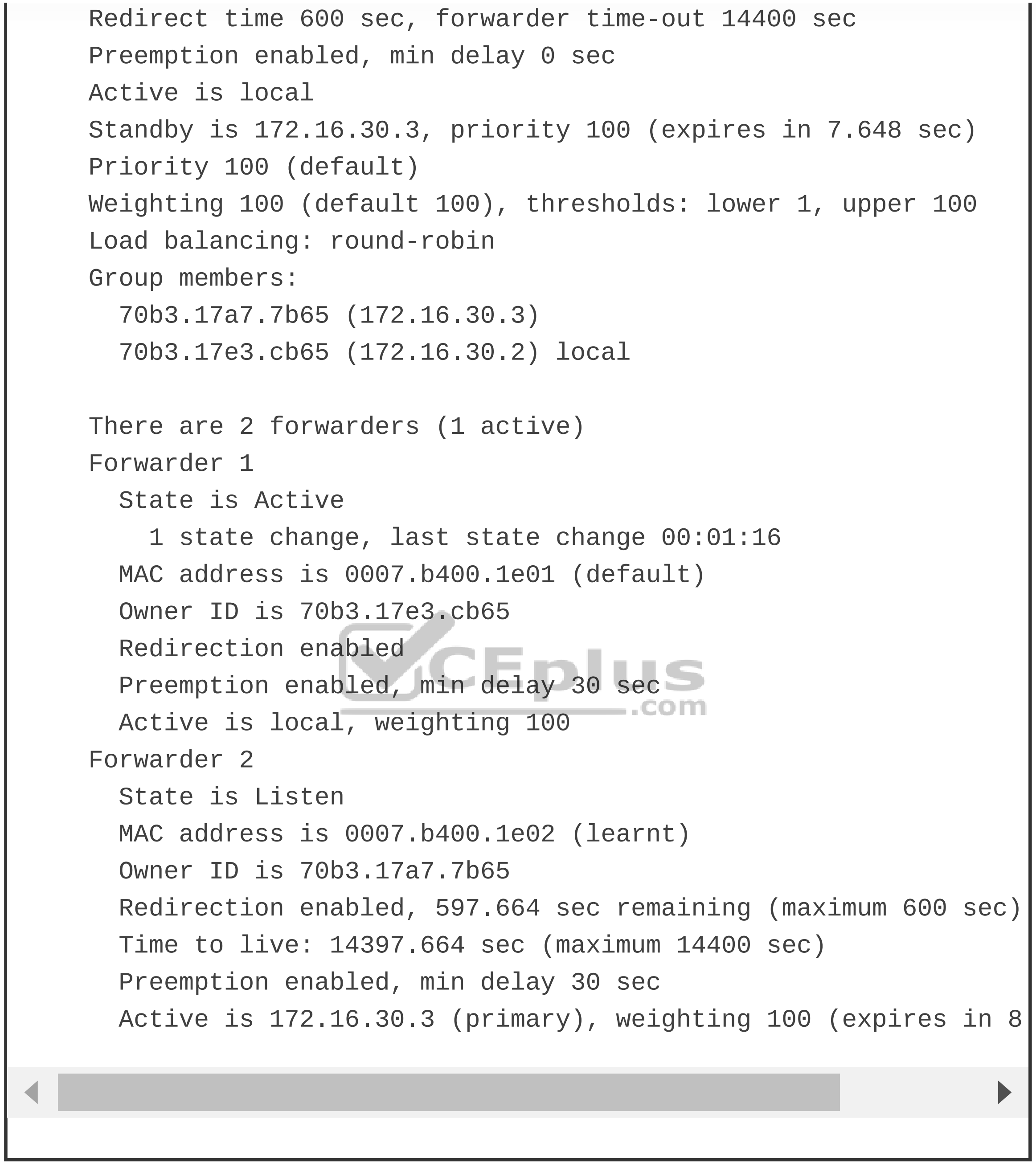
Example 15-21 Viewing the Brief GLBP Status

```
SW2# show glbp brief
Interface    Grp  Fwd Pri State    Address          Active router
Vl30         30   -   100 Active   172.16.30.1      local
Vl30         30   1   -   Active   0007.b400.1e01   local
Vl30         30   2   -   Listen   0007.b400.1e02   172.16.30.3

SW3# show glbp brief
Interface    Grp  Fwd Pri State    Address          Active router
Vl30         30   -   100 Standby  172.16.30.1      172.16.30.2
Vl30         30   1   -   Listen   0007.b400.1e01   172.16.30.2
Vl30         30   2   -   Active   0007.b400.1e02   local
```

The command **show glbp** displays additional information, including the timers, preemption settings, and statuses for the AVG and AVFs for the GLBP group. Example 15-22 shows the command **show glbp** run on SW2. Notice that the MAC addresses and interface IP addresses are listed under the group members, which can be used to correlate MAC address identities in other portions of the output.

Example 15-22 Viewing the Detailed GLBP Status

```
SW2# show glbp
Vlan30 - Group 30
  State is Active
    1 state change, last state change 00:01:26
  Virtual IP address is 172.16.30.1
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.664 secs
```



```
Redirect time 600 sec, forwarder time-out 14400 sec
Preemption enabled, min delay 0 sec
Active is local
Standby is 172.16.30.3, priority 100 (expires in 7.648 sec)
Priority 100 (default)
Weighting 100 (default 100), thresholds: lower 1, upper 100
Load balancing: round-robin
Group members:
  70b3.17a7.7b65 (172.16.30.3)
  70b3.17e3.cb65 (172.16.30.2) local

There are 2 forwarders (1 active)
Forwarder 1
  State is Active
    1 state change, last state change 00:01:16
  MAC address is 0007.b400.1e01 (default)
  Owner ID is 70b3.17e3.cb65
  Redirection enabled
  Preemption enabled, min delay 30 sec
  Active is local, weighting 100
Forwarder 2
  State is Listen
  MAC address is 0007.b400.1e02 (learnt)
  Owner ID is 70b3.17a7.7b65
  Redirection enabled, 597.664 sec remaining (maximum 600 sec)
  Time to live: 14397.664 sec (maximum 14400 sec)
  Preemption enabled, min delay 30 sec
  Active is 172.16.30.3 (primary), weighting 100 (expires in 8
```

By default, GLBP balances the load of traffic in a round-robin fashion, as highlighted in Example 15-22. However, GLBP supports three methods of load balancing traffic:

- **Round robin:** Uses each virtual forwarder MAC address to sequentially reply for the virtual IP address.

- **Weighted:** Defines weights to each device in the GLBP group to define the ratio of load balancing between the devices. This allows for a larger weight to be assigned to bigger routers that can handle more traffic.

- **Host dependent:** Uses the host MAC address to decide to which virtual forwarder MAC to redirect the packet. This method ensures that the host uses the same virtual MAC address as long as the number of virtual forwarders does not change within the group.

The load-balancing method can be changed with the command **glbp** *instance-id* **load-balancing** {**host-dependent** | **round-robin** | **weighted**}. The weighted load-balancing method has the AVG direct traffic to the AVFs based on the percentage of weight a router has over the total weight of all GLBP routers. Increasing the weight on more capable, bigger routers allows them to take more traffic than smaller devices. The weight can be set for a router with the command **glbp** *instance-id* **weighting** *weight*.

Example 15-23 shows how to change the load balancing to weighted and setting the weight to 20 on SW2 and 80 on SW3 so that SW2 receives 20% of the traffic and SW3 receives 80% of the traffic.

Example 15-23 Changing the GLBP Load Balancing to Weighted

```
SW2(config)# interface vlan 30
SW2(config-if)# glbp 30 load-balancing weighted
SW2(config-if)# glbp 30 weighting 20

SW3(config)# interface vlan 30
SW3(config-if)# glbp 30 load-balancing weighted
SW3(config-if)# glbp 30 weighting 80
```

Example 15-24 shows that the load-balancing method has been changed to weighted and that the appropriate weight has been set for each AVF.

Example 15-24 Verifying GLBP Weighted Load Balancing

```
SW2# show glbp
Vlan30 - Group 30
  State is Active
    1 state change, last state change 00:04:55
  Virtual IP address is 172.16.30.1
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.160 secs
```

```
     Redirect time 600 sec, forwarder time-out 14400 sec
     Preemption enabled, min delay 0 sec
     Active is local
     Standby is 172.16.30.3, priority 100 (expires in 9.216 sec)
     Priority 100 (default)
     Weighting 20 (configured 20), thresholds: lower 1, upper 20
     Load balancing: weighted
     Group members:
       70b3.17a7.7b65 (172.16.30.3)
       70b3.17e3.cb65 (172.16.30.2) local
     There are 2 forwarders (1 active)
     Forwarder 1
       State is Active
         1 state change, last state change 00:04:44
       MAC address is 0007.b400.1e01 (default)
       Owner ID is 70b3.17e3.cb65
       Redirection enabled
       Preemption enabled, min delay 30 sec
       Active is local, weighting 20
     Forwarder 2
       State is Listen
       MAC address is 0007.b400.1e02 (learnt)
       Owner ID is 70b3.17a7.7b65
       Redirection enabled, 599.232 sec remaining (maximum 600 sec)
       Time to live: 14399.232 sec (maximum 14400 sec)
       Preemption enabled, min delay 30 sec
       Active is 172.16.30.3 (primary), weighting 80 (expires in 9.4
```

NETWORK ADDRESS TRANSLATION

In the early stages of the Internet, large network blocks were assigned to organizations (for example, universities, companies). Network engineers started to realize that as more people connected to the Internet, the IP address space would become exhausted. RFC 1918 established common network blocks that should never be seen on the Internet (that is, they are non-globally routed networks):

- 10.0.0.0/8 accommodates 16,777,216 hosts.
- 172.16.0.0/24 accommodates 1,048,576 hosts.
- 192.168.0.0/16 accommodates 65,536 hosts.

These address blocks provide large private network blocks for companies to connect their devices together, but how can devices with private network addressing reach servers that are on the public Internet? If a packet is sourced from a 192.168.1.1 IP address and reaches the server with a 100.64.1.1 IP address, the server will not have a route back to the 192.168.1.1 network—because it does not exist on the Internet.

Key Topic

Connectivity is established with Network Address Translation (NAT). Basically, NAT enables the internal IP network to appear as a publicly routed external network. A NAT device (typically a router or firewall) modifies the source or destination IP addresses in a packet’s header as the packet is received on the outside or inside interface.

NAT can be used in use cases other than just providing Internet connectivity to private networks. It can also be used to provide connectivity when a company buys another company, and the two companies have overlapping networks (that is, the same network ranges are in use).

> **Note**
>
> Most routers and switches perform NAT translation only with the IP header addressing and do not translate IP addresses within the payload (for example, DNS requests). Some firewalls have the ability to perform NAT within the payload for certain types of traffic.

Key Topic

Four important terms are related to NAT:

- **Inside local:** The actual private IP address assigned to a device on the inside network(s).

- **Inside global:** The public IP address that represents one or more inside local IP addresses to the outside.

- **Outside local:** The IP address of an outside host as it appears to the inside network. The IP address does not have to be reachable by the outside but is considered private and must be reachable by the inside network.

- **Outside global:** The public IP address assigned to a host on the outside network. This IP address must be reachable by the outside network.

Key Topic

Three types of NAT are commonly used today:

- **Static NAT:** Provides a static one-to-one mapping of a local IP address to a global IP address.

- **Pooled NAT:** Provides a dynamic one-to-one mapping of a local IP address to a global IP address. The global IP address is temporarily assigned to a local IP address. After a certain amount of idle NAT time, the global IP address is returned to the pool.

- **Port Address Translation (PAT):** Provides a dynamic many-to-one mapping of many local IP addresses to one global IP address. The NAT device needs a mechanism to identify the specific private IP address for the return network traffic. The NAT device translates the private IP address and port to a different global IP address and port. The port is unique from any other ports, which enables the NAT device to track the global IP address to local IP addresses based on the unique port mapping.

The following sections explain these types of NAT.

NAT Topology

Figure 15-7 is used throughout this section to illustrate NAT. R5 performs the translation; its Gi0/0 interface (10.45.1.5) is the outside interface, and its Gi0/1 (10.56.1.5) interface is the inside interface. R1, R2, R3, R7, R8, and R9 all act as either clients or servers to demonstrate how NAT functions.

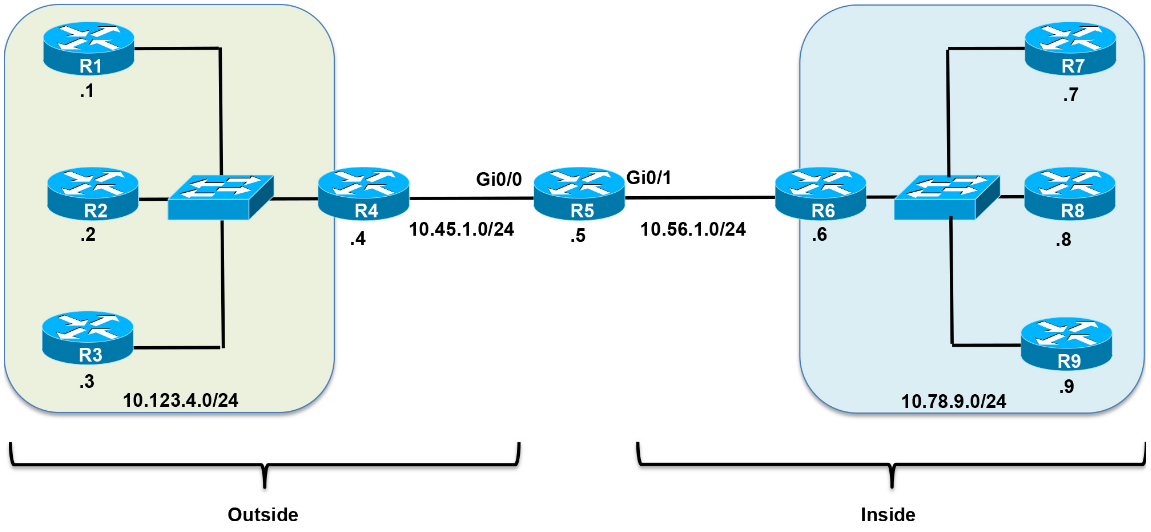


Figure 15-7 NAT Topology

R1, R2, and R3 all have a static default route toward R4, and R4 has a static default route toward R5. R7, R8, and R9 all have a static default route toward R6, and R6 has a static default route to R5. R5 contains a static route to the 10.123.4.0/24 network through R4, and a second static route to the 10.78.9.0/24 network through R6. Example 15-25 shows the routing tables of R1, R5, and R7.

Example 15-25 Routing Tables of R1, R5, and R7

```
R1# show ip route | begin Gateway
Gateway of last resort is 10.123.4.4 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.123.4.4
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.123.4.0/24 is directly connected, GigabitEthernGi0/0

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

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.45.1.0/24 is directly connected, GigabitEthernGi0/0
C        10.56.1.0/24 is directly connected, GigabitEthernGi0/1
S        10.78.9.0/24 [1/0] via 10.56.1.6
S        10.123.4.0/24 [1/0] via 10.45.1.4

R7# show ip route  | begin Gateway
Gateway of last resort is 10.78.9.6 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.78.9.6
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.78.9.0/24 is directly connected, GigabitEthernGi0/0
```

The topology provides full connectivity between the outside hosts (R1, R2, and R3) and the inside hosts (R7, R8, and R9). Example 15-26 shows a traceroute from R1 to R7.

Example 15-26 Traceroute from R1 to R7

```
R1# traceroute 10.78.9.7
Type escape sequence to abort.
Tracing the route to 10.78.9.7
VRF info: (vrf in name/id, vrf out name/id)
  1 10.123.4.4 1 msec 0 msec 0 msec
  2 10.45.1.5 1 msec 0 msec 0 msec
  3 10.56.1.6 1 msec 0 msec 0 msec
  4 10.78.9.7 1 msec *  1 msec
```

Using an IOS XE router for hosts (R1, R2, R3, R7, R8, and R9) enables you to log in using Telnet and identify the source and destination IP addresses by examining the TCP session details. In Example 15-27, R7 (10.78.9.7) initiates a Telnet connection to R1 (10.123.4.1). When you are logged in, the command **show tcp brief** displays the source IP address and port, along with the destination IP address and port.

The local IP address reflects R1 (10.123.4.1), and the remote address is R7 (10.78.9.7). These IP addresses match expectations, and therefore no NAT has occurred on R5 for this Telnet session.

Example 15-27 Viewing the Source IP Address

```
R7# telnet 10.123.4.1
Trying 10.123.4.1 ... Open
*************************************************************
* You have remotely connected to R1 on line 2
*************************************************************
User Access Verification
Password:

R1# show tcp brief
TCB       Local Address               Foreign Address
F69CE570  10.123.4.1.23              10.78.9.7.49024
```

Static NAT

Static NAT involves the translation of a global IP address to a local IP address, based on a static mapping of the global IP address to the local IP address. There are two types of static NAT, as described in the following sections:

- Inside static NAT

- Outside static NAT

Inside Static NAT

Inside static NAT involves the mapping of an inside local (private) IP address to an inside global (public) IP address. In this scenario, the private IP addresses are being hidden from the outside hosts.

The steps for configuring inside static NAT are as follows:

Step 1. Configure the outside interfaces by using the command **ip nat outside**.

Step 2. Configure the inside interface with the command **ip nat inside**.

Step 3. Configure the inside static NAT by using the command **ip nat inside source static** *inside-local-ip inside-global-ip*.

Example 15-28 shows the inside static NAT configuration on R5, where packets sourced from R7 (10.78.9.7) appear as if they came from 10.45.1.7.

Example 15-28 Configuring Inside Static NAT

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat inside source static 10.78.9.7 10.45.1.7
```

Note

Most network engineers assume that the *inside-global-ip* must reside on the outside network. In this scenario, that would be an IP address on the 10.45.1.0/24 network. First, the *inside-global-ip* address should not be associated with the outside interface. Second, the *inside-global-ip* address could be an address for a network that does not exist on the NAT router (for example, 10.77.77.77). However, all outside routers must have a route for forwarding packets toward the router performing the NAT for that IP address (that is, 10.77.77.77).

Now that the NAT has been configured on R5, R7 initiates a Telnet session with R1, as demonstrated in Example 15-29. Upon viewing the TCP session on R1, the local address remains 10.123.4.1 as expected, but the remote address now reflects 10.45.1.7. This is a different source IP address than the baseline example in Example 15-27, where the remote address is 10.78.9.7.

Example 15-29 Identification of the Source with Inside Static NAT

```
R7# telnet 10.123.4.1
Trying 10.123.4.1 ... Open
*************************************************************
```

```
* You have remotely connected to R1 on line 3
*************************************************************
User Access Verification
Password:

R1# show tcp brief
TCB       Local Address              Foreign Address
F6D25D08  10.123.4.1.23             10.45.1.7.56708
```

Key Topic

The NAT translation table consists of static and dynamic entries. The NAT translation table is displayed with the command **show ip nat translations**. Example 15-30 shows R5's NAT translation table after R7 initiated a Telnet session to R1. There are two entries:

- The first entry is the dynamic entry correlating to the Telnet session. The inside global, inside local, outside local, and outside global fields all contain values. Notice that the ports in this entry correlate with the ports in Example 15-29.

- The second entry is the inside static NAT entry that was configured.

Example 15-30 NAT Translation Table for Inside Static NAT

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outs
tcp 10.45.1.7:56708    10.78.9.7:56708    10.123.4.1:23      10.1
--- 10.45.1.7          10.78.9.7          ---                ---
```

Key Topic

Figure 15-8 displays the current topology with R5’s translation table. The NAT translation follows these steps:

1. As traffic enters on R5’s Gi0/1 interface, R5 performs a route lookup for the destination IP address, which points out of its Gi0/0 interface. R1 is aware that the Gi0/0 interface is an outside NAT interface and that the Gi0/1 interface is an inside NAT interface and therefore checks the NAT table for an entry.

2. Only the inside static NAT entry exists, so R5 creates a dynamic inside NAT entry with the packet’s destination (10.123.4.1) for the outside local and outside global address.

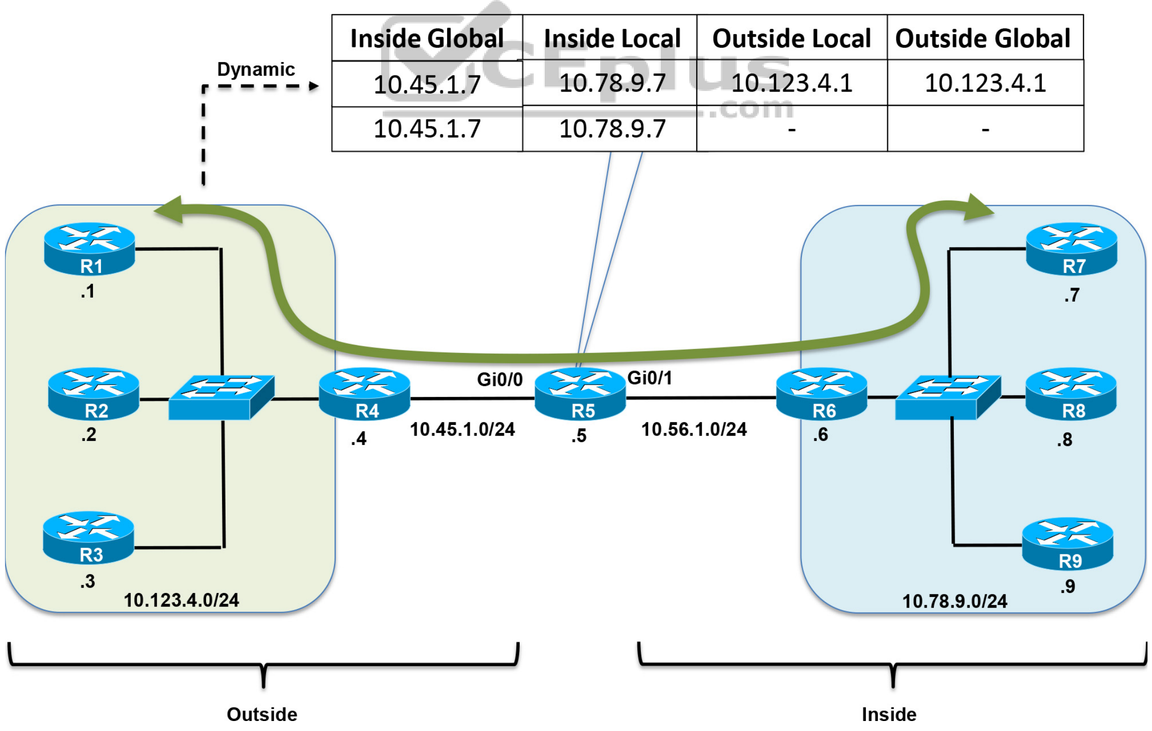
3. R5 translates (that is, changes) the packet’s source IP address from 10.78.9.7 to 10.45.1.7.

4. R1 registers the session as coming from 10.45.1.7 and then transmits a return packet. The packet is forwarded to R4 using the static default route, and R4 forwards the packet using the static default route.

5. As the packet enters on R5’s Gi0/0 interface, R5 is aware that the Gi0/0 interface is an outside NAT interface and checks the NAT table for an entry.

6. R5 correlates the packet’s source and destination ports with the first NAT entry, as shown in Example 15-30, and knows to modify the packet’s destination IP address from 10.45.1.7 to 10.78.9.7.

7. R5 routes the packet out the Gi0/1 interface toward R6.



Inside Global	Inside Local	Outside Local	Outside Global
10.45.1.7	10.78.9.7	10.123.4.1	10.123.4.1
10.45.1.7	10.78.9.7	-	-

Figure 15-8 Inside Static NAT Topology for R7 as 10.45.1.7

Remember that a static NAT entry is a one-to-one mapping between the inside global and the inside local address. As long as the outside devices can route traffic to the inside global IP address, they can use it to reach the inside local device as well.

In Example 15-31, R2, with no sessions to any device in the topology, establishes a Telnet session with R7, using the inside global IP address 10.45.1.7. R5 simply creates a second dynamic entry for this new session. From R7’s perspective, it has connected with R2 (10.123.4.2).

Example 15-31 Connectivity from External Devices to the Inside Global IP Address

```
R2# telnet 10.45.1.7
Trying 10.45.1.7 ... Open
*************************************************************
* You have remotely connected to R7 on line 2
*************************************************************
User Access Verification
Password:

R7# show tcp brief
TCB       Local Address               Foreign Address
F6561AE0  10.78.9.7.23               10.123.4.2.63149
F65613E0  10.78.9.7.33579            10.123.4.1.23

R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outs
tcp 10.45.1.7:56708    10.78.9.7:56708    10.123.4.1:23      10.1
tcp 10.45.1.7:23       10.78.9.7:23       10.123.4.2:63149   10.1
--- 10.45.1.7          10.78.9.7          ---                ---
```

Outside Static NAT

Outside static NAT involves the mapping of an outside global (public) IP address to an outside local (private) IP address. In this scenario, the real external IP addresses are being hidden from the inside hosts.

Key Topic

The steps for configuring outside static NAT are as follows:

Step 1. Configure the outside interfaces by using the command **ip nat outside**.

Step 2. Configure the inside interface by using the command **ip nat inside**.

Step 3. Configure the outside static NAT entry by using the command **ip nat outside source static** *outside-global-ip outside-local-ip* [**add-route**]. The router performs a route lookup first for the *outside-local-ip* address, and a route must exist for that network to forward packets out of the outside interface before NAT occurs. The optional **add-route** keyword adds the appropriate static route entry automatically.

Example 15-32 shows the outside static NAT configuration on R5, where packets sent from R6, R7, R8, or R9 to 10.123.4.222 will be sent to R2 (10.123.4.2). R5 already has a static route to the 10.123.4.0/24 network, so the **add-route** keyword is not necessary.

Example 15-32 Configuring Outside Static NAT

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat outside source static 10.123.4.2 10.123.4.222
```

R6, R7, R8, or R9 could initiate a Telnet session directly with R2’s IP address (10.123.4.2), and no NAT translation would occur. The same routers could initiate a Telnet session with the R2’s outside local IP address 10.123.4.222; or R2 could initiate a session with any of the inside hosts (R6, R7, R8, or R9) to demonstrate the outside static NAT entry.

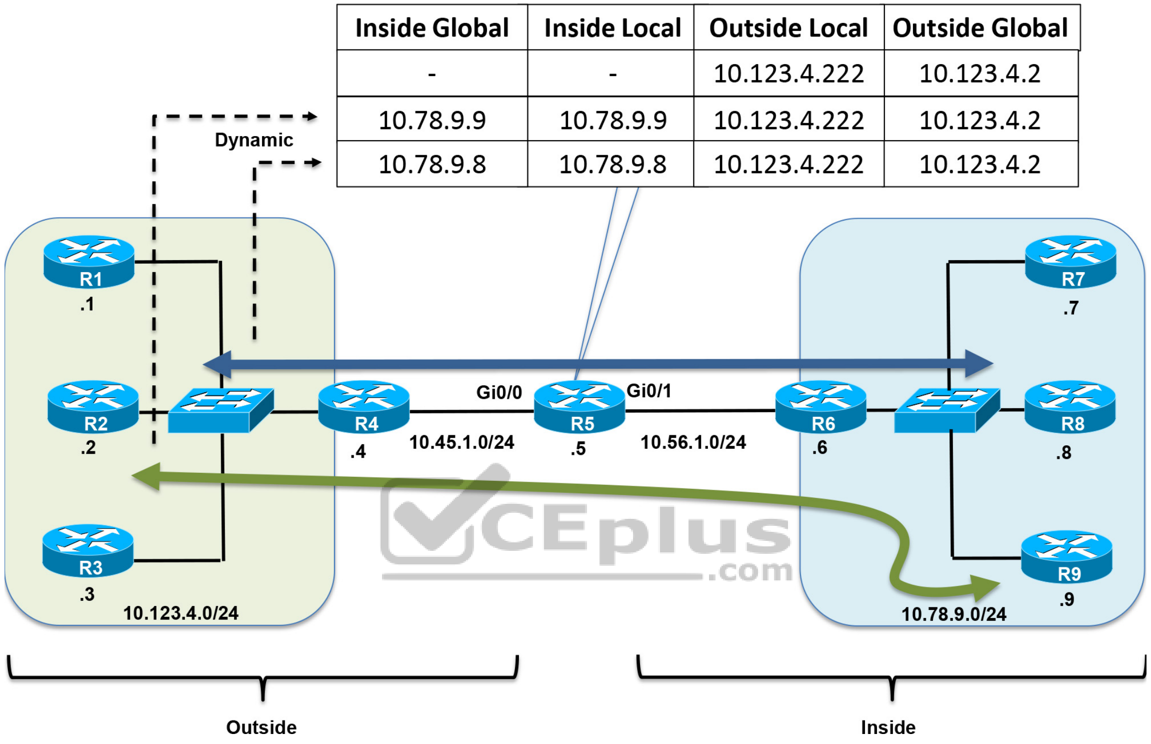
Example 15-33 shows R2 establishing a Telnet session with R9 (10.78.9.9). From R9’s perspective, the connection came from 10.123.4.222. At the same time, R8 initiated a Telnet session with the outside static NAT outside local IP address

(10.123.4.222), but from R2’s perspective, the source address is R8’s 10.78.9.8 IP address.

Example 15-33 Generating Network Traffic with Outside Static NAT

```
R2# telnet 10.78.9.9
Trying 10.78.9.9 ... Open
*************************************************************
* You have remotely connected to R9 on line 2
*************************************************************
User Access Verification
Password:

R9#show tcp brief
TCB       Local Address              Foreign Address
F6A23AF0  10.78.9.9.23              10.123.4.222.57126

R8# telnet 10.123.4.222
Trying 10.123.4.222 ... Open
*************************************************************
* You have remotely connected to R2 on line 2
*************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB       Local Address              Foreign Address
F64C9460  10.123.4.2.57126          10.78.9.9.23
F64C9B60  10.123.4.2.23             10.78.9.8.11339
```

Figure 15-9 shows R5’s translation table for R2’s outside static NAT entry for 10.123.4.222. Notice that there is a static mapping, and there are two dynamic entries for the two sessions on R2.



Inside Global	Inside Local	Outside Local	Outside Global
-	-	10.123.4.222	10.123.4.2
10.78.9.9	10.78.9.9	10.123.4.222	10.123.4.2
10.78.9.8	10.78.9.8	10.123.4.222	10.123.4.2

Figure 15-9 Outside Static NAT Topology for R2 as 10.123.4.222

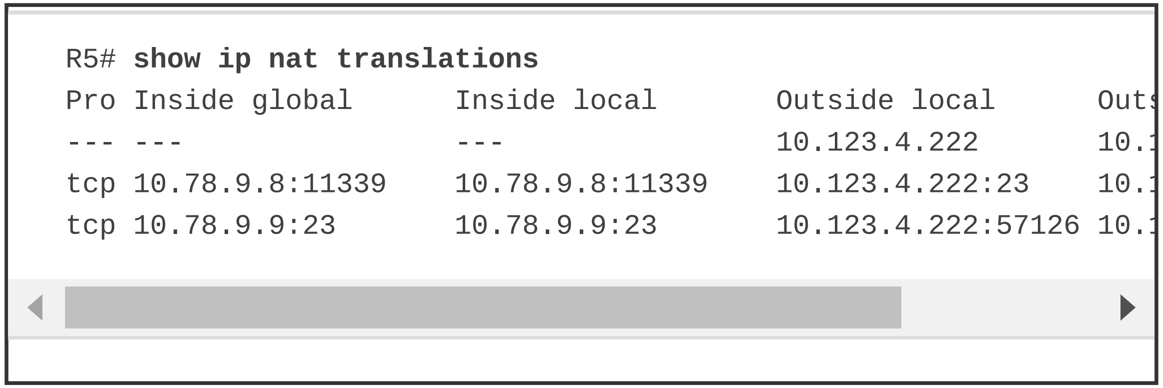
Example 15-34 shows R5’s NAT translation table. There are three entries:

- The first entry is the outside static NAT entry that was configured.

- The second entry is the Telnet session launched from R8 to the 10.123.4.222 IP address.

- The third entry is the Telnet session launched from R2 to R9’s IP address (10.78.9.9).

Example 15-34 NAT Translation Table for Outside Static NAT



```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Out
--- ---                ---                10.123.4.222       10.
tcp 10.78.9.8:11339    10.78.9.8:11339    10.123.4.222:23    10.
tcp 10.78.9.9:23       10.78.9.9:23       10.123.4.222:57126 10.
```

Note

Outside static NAT configuration is not very common and is typically used to overcome the problems caused by duplicate IP/network addresses in a network.

Pooled NAT

Static NAT provides a simple method of translating addresses. A major downfall to the use of static NAT is the number of configuration entries that must be created on the NAT device; in addition, the number of global IP addresses must match the number of local IP addresses.

Pooled NAT provides a more dynamic method of providing a one-to-one IP address mapping—but on a dynamic, as-needed basis. The dynamic NAT translation stays in the translation table until traffic flow from the local address to the global address has stopped and the timeout period (24 hours by default) has expired. The unused global IP address is then returned to the pool to be used again.

Pooled NAT can operate as inside NAT or outside NAT. In this section, we focus on inside pooled NAT. The steps for configuring inside pooled NAT are as follows:

Key Topic

Step 1. Configure the outside interfaces by using the command **ip nat outside**.

Step 2. Configure the inside interface by using the command **ip nat inside**.

Step 3. Specify which by using a standard or extended ACL referenced by number or name. Using a user-friendly name may be simplest from an operational support perspective.

Step 4. Define the global pool of IP addresses by using the command **ip nat pool** *nat-pool-name starting-ip ending-ip* **prefix-length** *prefix-length*.

Step 5. Configure the inside pooled NAT by using the command **ip nat inside source list** *acl* **pool** *nat-pool-name*.

Example 15-35 shows a sample configuration for inside pooled NAT. This example uses a NAT pool with the IP addresses 10.45.1.10 and 10.45.1.11. A named ACL, ACL-NAT-CAPABLE, allows only packets sourced from the 10.78.9.0/24 network to be eligible for pooled NAT.

Example 15-35 Configuring Inside Pooled NAT

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# ip access-list standard ACL-NAT-CAPABLE
R5(config-std-nacl)# permit 10.78.9.0 0.0.0.255
R5(config-std-nacl)# exit
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config-if)# exit
R5(config)# ip nat pool R5-OUTSIDE-POOL 10.45.1.10 10.45.1.11 pre
R5(config)# ip nat inside source list ACL-NAT-CAPABLE pool R5-OUT
```

To quickly generate some traffic and build the dynamic inside NAT translations, R7 (10.78.9.7) and R8 (10.78.9.8) ping R1 (10.123.4.1), as demonstrated in Example 15-36. This could easily be another type of traffic (such as Telnet).

Example 15-36 Initial Traffic for Pooled NAT

```
R7# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 second
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1

R8# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 second
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1
```

In this case, the pings should have created a dynamic inside NAT translation and removed the 10.45.1.10 and 10.45.1.11 binding. Example 15-37 confirms this assumption. There are a total of four translations in R5’s translation table. Two of them are for the full flow and specify the protocol, inside global, inside local, outside local, and outside global IP addresses.

Example 15-37 Viewing the Pooled NAT Table for R5

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outs
icmp 10.45.1.10:0      10.78.9.7:0        10.123.4.1:0       10.1
--- 10.45.1.10         10.78.9.7          ---                ---
icmp 10.45.1.11:0      10.78.9.8:0        10.123.4.1:0       10.1
--- 10.45.1.11         10.78.9.8          ---                ---
```

The other two translations are dynamic one-to-one mappings that could be used as R7 or R8 create additional dynamic flows and maintain the existing global IP address. Based on the mapping before the flow, the additional flows from R8 (10.78.9.8) should be mapped to the global IP address 10.45.1.11.

In Example 15-38, R8 establishes a Telnet session with R2. R2 detects that the remote IP address of the session is 10.45.1.11. A second method of confirmation is to examine the NAT translation on R5, there is a second dynamic translation entry for the full Telnet session.

Example 15-38 Using the Dynamic One-to-One Mappings for Address Consistency

```
R8# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
*************************************************************
* You have remotely connected to R2 on line 2
*************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB       Local Address              Foreign Address
F3B64440  10.123.4.2.23             10.45.1.11.34115

R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outs
```

```
icmp 10.45.1.10:1      10.78.9.7:1        10.123.4.1:1       10.1
--- 10.45.1.10         10.78.9.7          ---                ---
icmp 10.45.1.11:1      10.78.9.8:1        10.123.4.1:1       10.1
tcp 10.45.1.11:34115   10.78.9.8:34115    10.123.4.2:23      10.1
--- 10.45.1.11         10.78.9.8          ---                ---
```

A downfall to using pooled NAT is that when the pool is exhausted, no additional translation can occur until the global IP address is returned to the pool. To demonstrate this concept, R5 has enabled debugging for NAT, and R9 tries to establish a Telnet session with R1. Example 15-39 demonstrates the concept, with the NAT translation failing on R5 and the packet being dropped.

Example 15-39 Failed NAT Pool Allocation

```
R9# telnet 10.123.4.1
Trying 10.123.4.1 ...
% Destination unreachable; gateway or host down

R5# debug ip nat detailed
IP NAT detailed debugging is on
R5#
 02:22:58.685: NAT: failed to allocate address for 10.78.9.9, lis
 02:22:58.685:  mapping pointer available mapping:0
 02:22:58.685: NAT*: Can't create new inside entry - forced_punt_
 02:22:58.685: NAT: failed to allocate address for 10.78.9.9, lis
 02:22:58.685:  mapping pointer available mapping:0
 02:22:58.685: NAT: translation failed (A), dropping packet s=10
```

The default timeout for NAT translations is 24 hours, but this can be changed with the command **ip nat translation timeout** *seconds*. The dynamic NAT translations can be cleared out with the command **clear ip nat translation** {*ip-address* | *****}, which removes all existing translations and could interrupt traffic flow on active sessions as they might be assigned new global IP addresses.

Example 15-40 demonstrates the reset of the NAT translations on R5 for all IP addresses and then on R9, which is successfully able to gain access to R1 through the newly allocated (reset) global IP address.

Example 15-40 Clearing NAT Translation to Reset the NAT Pool

```
R5# clear ip nat translation *

R9# telnet 10.123.4.1
Trying 10.123.4.1 ... Open
***************************************************************
* You have remotely connected to R1 on line 2
***************************************************************
User Access Verification
Password:

R1#
```

Port Address Translation

Pooled NAT translation simplifies the management of maintaining the one-to-one mapping for NAT (compared to static NAT). But pooled NAT translation still faces the limitation of ensuring that the number of global IP addresses is adequate to meet the needs of the local IP addresses.

Key Topic

Port Address Translation (PAT) is an iteration of NAT that allows for a mapping of many local IP addresses to one global IP address. The NAT device maintains the state of translations by dynamically changing the source ports as a packet leaves the outside interface. Another term for PAT is *NAT overload*.

Configuring PAT involves the following steps:

Key Topic

Step 1. Configure the outside interface by using the command **ip nat outside**.

Step 2. Configure the inside interface by using the command **ip nat inside**.

Step 3. Specify which traffic can be translated by using a standard or extended ACL referenced by number or name. Using a user-friendly name may name may be simplest from an operational support perspective.

Step 4. Configure Port Address Translation by using the command the command **ip nat inside source list** *acl* {**interface** *interface-id* | **pool** *nat-pool-name*} **overload**. Specifying an interface involves using the primary IP address assigned to that interface. Specifying a NAT pool requires the creation of the NAT pool, as demonstrated earlier, and involves using those IP addresses as the global address.

Example 15-41 demonstrates R5's PAT configuration, which allows network traffic sourced from the 10.78.9.0/24 network to be translated to R5's Gi0/0 interface (10.45.1.5) IP address.

Example 15-41 Configuring PAT on R5

```
R5# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)# ip access-list standard ACL-NAT-CAPABLE
R5(config-std-nacl)# permit 10.78.9.0 0.0.0.255
R5(config-std-nacl)# exit
R5(config)# interface GigabitEthernGi0/0
R5(config-if)# ip nat outside
R5(config-if)# interface GigabitEthernGi0/1
R5(config-if)# ip nat inside
R5(config)# ip nat source list ACL-NAT-CAPABLE interface Gigabit
```

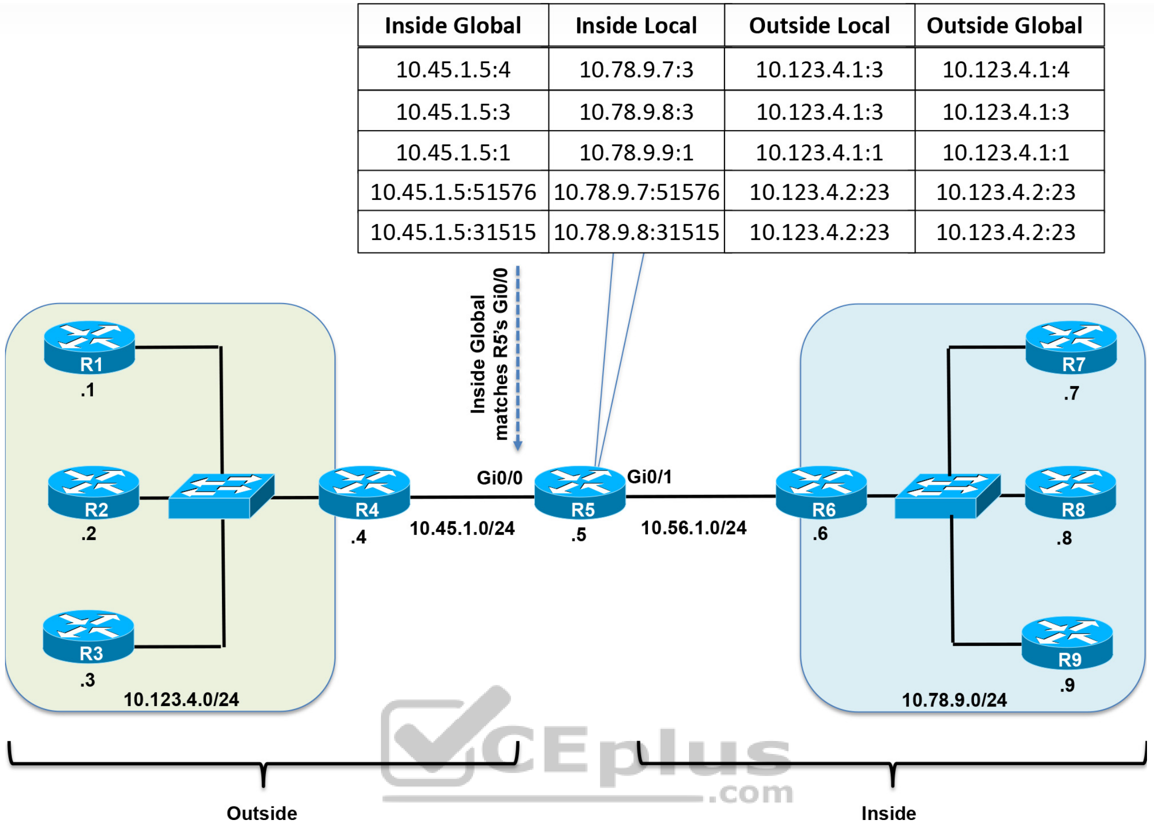
Now that PAT has been configured on R5, traffic can be generated for testing. R7, R8, and R9 ping R1 (10.123.4.1), and R7 and R8 establish a Telnet session. Based on the TCP sessions in Example 15-42, you can see that both Telnet sessions are coming from R5's Gi0/0 (10.45.1.5) IP address. R7 has a remote port of 51,576, while R8 has a remote port of 31,515.

Example 15-42 Generating Network Traffic for PAT

```
R7# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 secon
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1

R8# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 secon
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1

R9# ping 10.123.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.123.4.1, timeout is 2 secon
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1

R7# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
*************************************************************
* You have remotely connected to R2 on line 2
*************************************************************
User Access Verification
```

```
Password:

R2# show tcp brief
TCB       Local Address                 Foreign Address
F3B64440  10.123.4.2.23                10.45.1.5.51576

R8# telnet 10.123.4.2
Trying 10.123.4.2 ... Open
*************************************************************
* You have remotely connected to R2 on line 3
*************************************************************
User Access Verification
Password:

R2# show tcp brief
TCB       Local Address                 Foreign Address
F3B64440  10.123.4.2.23                10.45.1.5.51576
F3B65560  10.123.4.2.23                10.45.1.5.31515
```

Figure 15-10 shows R5’s translation table after all the various flows have established. Notice that the inside global IP address is R5’s Gi0/0 (10.45.1.5) IP address, while the inside local IP addresses are different. In addition, notice that the ports for the inside global entries are all unique—especially for the first two entries, which have an outside local entry for 10.123.4.1:3. PAT must make the inside global ports unique to maintain the one-to-many mapping for any return traffic.



Inside Global	Inside Local	Outside Local	Outside Global
10.45.1.5:4	10.78.9.7:3	10.123.4.1:3	10.123.4.1:4
10.45.1.5:3	10.78.9.8:3	10.123.4.1:3	10.123.4.1:3
10.45.1.5:1	10.78.9.9:1	10.123.4.1:1	10.123.4.1:1
10.45.1.5:51576	10.78.9.7:51576	10.123.4.2:23	10.123.4.2:23
10.45.1.5:31515	10.78.9.8:31515	10.123.4.2:23	10.123.4.2:23

Figure 15-10 R5's Translation Table for PAT

Example 15-43 shows R5's NAT translation table. By taking the ports from the TCP brief sessions on R2 and correlating them to R5's NAT translation table, you can identify which TCP session belongs to R7 or R8.

Example 15-43 R5's NAT Translation Table with PAT

```
R5# show ip nat translations
Pro Inside global      Inside local       Outside local      Outs
icmp 10.45.1.5:4       10.78.9.7:3        10.123.4.1:3       10.1
icmp 10.45.1.5:3       10.78.9.8:3        10.123.4.1:3       10.1
icmp 10.45.1.5:1       10.78.9.9:1        10.123.4.1:1       10.1
tcp 10.45.1.5:51576    10.78.9.7:51576    10.123.4.2:23      10.1
tcp 10.45.1.5:31515    10.78.9.8:31515    10.123.4.2:23      10.1
```

EXAM PREPARATION TASKS

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 30, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

REVIEW ALL KEY TOPICS

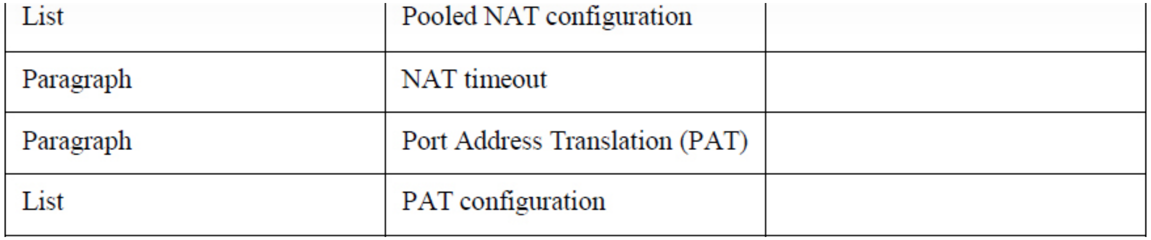
Review the most important topics in the chapter, noted with the key topics icon in the outer margin of the page. Table 15-3 lists these key topics and the page number on which each is found.

Key Topic

Table 15-3 Key Topics for Chapter 15

Key Topic Element	**Description**	**Page**
Section	Network Time Protocol	

Paragraph	NTP stratums	
Section	NTP stratum preference	
Section	NTP peers	
Paragraph	First-hop redundancy protocol (FHRP)	
Section	Hot Standby Router Protocol (HSRP)	
List	HSRP configuration	
Paragraph	HSRP object tracking	
Section	Virtual Router Redundancy Protocol (VRRP)	
List	Legacy VRRP configuration	
List	Hierarchical VRRP configuration	
Section	Global Load Balancing Protocol (GLBP)	
List	GLBP configuration	
List	GLBP load-balancing options	
Paragraph	Network Address Translation (NAT)	
List	NAT terms	
List	Common NAT types	
List	Inside static NAT configuration	
Paragraph	Viewing the NAT translation table	
Paragraph	NAT processing	
List	Outside static NAT configuration	



List	Pooled NAT configuration	
Paragraph	NAT timeout	
Paragraph	Port Address Translation (PAT)	
List	PAT configuration	

COMPLETE TABLES AND LISTS FROM MEMORY

There are no memory tables in this chapter.

DEFINE KEY TERMS

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

first-hop redundancy protocol

inside global

inside local

Network Address Translation (NAT)

NTP client

NTP peer

NTP server

outside local

outside global

pooled NAT

Port Address Translation (PAT)

static NAT

stratum

USE THE COMMAND REFERENCE TO CHECK YOUR MEMORY

Table 15-4 lists the important commands from this chapter. To test your memory, cover the right side of the table with a piece of paper, read the description on the left side, and see how much of the command you can remember.

Table 15-4 Command Reference

Task	Command Syntax
Configure a device as an NTP client with the IP address of the NTP server	**ntp server** *ip-address* [**prefer**] [**source** *interface-id*]
Configure a device so that it can respond authoritatively to NTP requests when it does not have access to an atomic clock or an upstream NTP server	**ntp master** *stratum-number*
Configure the peering with another device with NTP	**ntp peer** *ip-address*
Configure the tracking of an interface's line protocol state	**track** *object-number* **interface** *interface-id* **line-protocol**
Configure a device to track the installation of a route in the routing table	**track** *object-number* **ip route** *route/prefix-length* **reachability**
Configure the VIP for the HSRP instance	**standby** *instance-id* **ip** *vip-address*
Enable preemption for the HSRP instance	**standby** *instance-id* **preempt**
Specify the MAC address for the HSRP VIP	**standby** *instance-id* **mac-address** *mac-address*
Configure the HSRP timers for neighbor health checks	**standby** *instance-id* **timers** {**seconds** | **msec milliseconds**}
Link object tracking to a decrease in priority upon failure of the HSRP	**standby** *instance-id* **track** *object-id* **decrement** *decrement-value*
Configure the VIP gateway for the VRRP instance	**vrrp** *instance-id* **ip** *vip-address*
Configure the priority for the VRRP instance	**vrrp** *instance-id* **priority** *priority*
Link object tracking to a decrease in priority upon failure with VRRP	**vrrp** *instance-id* **track** *object-id* **decrement** *decrement-value*
Configure the VIP gateway for a GLBP instance	**glbp** *instance-id* **ip** *vip-address*
Enable preemption for a GLBP instance Configure the priority for a GLBP instance	**glbp** *instance-id* **preempt**
Configure the priority for a GLBP instance	**glbp** *instance-id* **priority** *priority*
Configure GLBP timers for neighbor health checks	**glbp** *instance-id* **timers** {*hello-seconds* | **msec** *hello-milliseconds*} {*hold-seconds* | **msec** *hold-milliseconds*}
Configure the GLBP load-balancing algorithm	**glbp** *instance-id* **load-balancing** {**host-dependent** | **round-robin** | **weighted**}.

Configure the devices GLBP weight for traffic load balancing	**glbp** *instance-id* **weighting** *weight*
Configure an interface as an outside interface for NAT	**ip nat outside**
Configure an interface as an inside interface for NAT	**ip nat inside**
Configure static inside NAT	**ip nat inside source static** *inside-local-ip inside-global-ip*
Configure static outside NAT	**ip nat outside source static** *outside-global-ip outside-local-ip* [**add-route**]
Configure pooled NAT	**ip nat pool** *nat-pool-name starting-ip ending-ip* **prefix-length** *prefix-length*
Define the NAT pool for global IP addresses	**ip nat inside source list** *acl* **pool** *nat-pool-name*
Configure a device for PAT	**ip nat inside source list** *acl* {**interface** *interface-id* | **pool** *nat-pool-name*} **overload**
Modify the NAT timeout period	**ip nat translation timeout** *seconds*
Clear a dynamic NAT entry	**clear ip nat translation** {*ip-address* | *}
Display the status of the NTP service, hardware clock synchronization status, reference time, and time since last polling cycle	**show ntp status**
Display the list of configured NTP servers and peers and their time offset from the local device	**show ntp associations**
Display the status of a tracked object	**show track** [*object-number*]
Display the status of an HSRP VIP	**show standby** [*interface-id*] [**brief**]
Display the status of a VRRP VIP	**show vrrp** [**brief**]
Display the status of a GLBP VIP	**show glbp** [**brief**]
Display the translation table on a NAT device	**show ip nat translations**