

*"This is the course I wish had been available
when I was studying for my CCNA. Great Job Paul!"
Luke Winslow CCIE 10788*

Cisco CCNA Simplified

4th Edition

Your Complete Guide to Passing the Cisco CCNA
Routing and Switching Exam

By Paul Browning CCNP, Farai Tafa CCIE, Daniel Gheorghe CCIE



Exam tips, labs and real world advice to ensure that you not only pass the CCNA exam but have the confidence to apply your skills in the real world.

study - practice - pass



Table of Contents

FOREWORD

ACKNOWLEDGMENTS

BIOGRAPHIES

About the Technical Reviewer

INTRODUCTION

The Problem with CCNA Study Guides

Cisco CCNA Simplified versus Cisco CCNA in 60 Days

Free Bonus Material

Cisco CCNA Simplified Video Course

Reviews

Also from Reality Press Ltd.

How to Read Cisco CCNA Simplified

The Cisco CCNA Exam Format

How to Do the Labs

PART 1 — ICND1

CHAPTER 1 — OPERATION OF IP DATA NETWORKS

Overview of Networking Equipment

Hub

Switch

Router

The Open Systems Interconnection Model

Encapsulation

Application Layer

Presentation Layer

Session Layer

Transport Layer

- Network Layer
- Data Link Layer
- Physical Layer
- Summary—The OSI Model

- The TCP/IP Model

- TCP/IP Application Layer
- TCP/IP Transport/Host-to-Host Layer
- TCP/IP Internet/Network Layer
- TCP/IP Network Access Layer

- TCP/IP Services

- File Transfer Protocol
- Trivial File Transfer Protocol
- Simple Mail Transfer Protocol
- HyperText Transfer Protocol
- Telnet
- Internet Control Message Protocol
- Traceroute
- Address Resolution Protocol
- Mini-lab – Checking the ARP Cache
- Proxy ARP
- Mini-lab – Discovering Proxy ARP
- Reverse Address Resolution Protocol
- Gratuitous ARP
- Simple Network Management Protocol
- Domain Name System
- Mini-lab – Pinging Hostnames
- Cisco Discovery Protocol
- Mini-lab – Checking for CDP Neighbors
- Ethernet Concepts

[CSMA/CD](#)

[Duplex Settings](#)

[Mini-lab – Configuring Ethernet Speed and Duplex Settings](#)

[Ethernet Frames](#)

[LAN Traffic](#)

[IEEE Standards](#)

[The Cisco Hierarchical Networking Model](#)

[Cabling the Network](#)

[LAN Cabling](#)

[IEEE and Cabling Standards](#)

[WAN Cabling](#)

[Router Interfaces and Connectors](#)

[RJ-45](#)

[Aux Connectors](#)

[Console Connectors](#)

[WAN Connectors](#)

[Router Interfaces and Slots](#)

[Connecting to a Router](#)

[USB Console Connection](#)

[Router Modes](#)

[User Mode](#)

[Privileged Mode](#)

[Global Configuration Mode](#)

[Interface Configuration Mode](#)

[Line Configuration Mode](#)

[Router Configuration Mode](#)

[Reloading the Router](#)

[Abbreviating the Commands](#)

[Configuring a Router](#)

Loopback Interfaces

Editing Commands

Mini-lab – Putting an IP Address on an Interface

Show Commands

Debug Commands

Pipes

The Configuration Register

Mini-lab – Changing the Configuration Register

End of Chapter Questions

Chapter 1 Labs

Lab 1: Basic Lab – Router Modes and Commands

Lab 2: ARP, CDP, Ping, and Telnet Lab

Lab 3: Traceroute from Router A to Router B

Further Reading

CHAPTER 2 — LAN SWITCHING TECHNOLOGIES

Layer 2 Switching Functions

Learning MAC Addresses

Mini-lab – Checking the MAC Address Table

Filtering and Forwarding Frames

Preventing Loops in the Network

Switching Methods

Cut-through

Store-and-Forward

Fragment-free (Modified Cut-through/Runt-free)

Virtual Local Area Networks (VLANs)

VLAN Membership

VLAN Numbers

VLAN Links

Dynamic Trunking Protocol

Mini-lab – Configuring VLANs and Trunk Links

IEEE 802.1Q Native VLAN

InterVLAN Routing

Mini-lab – InterVLAN Routing Using Physical Router Interfaces

Mini-lab – InterVLAN Routing Using Router Subinterfaces

Mini-lab – InterVLAN Routing Using Switched Virtual Interfaces

VLAN Trunking Protocol

VTP Modes

Mini-lab – Configuring VTP

VTP Pruning

Configuring a Cisco Switch

Mini-lab – IP Default Gateway on a Switch

End of Chapter Questions

Chapter 2 Labs

Lab 1: VLANs on an IOS Switch

Lab 2: Trunking across IOS Switches

Lab 3: InterVLAN Routing

CHAPTER 3 — IP ADDRESSING

How Binary Works

How Hexadecimal Works

Have a Try

IP Version 4

Powers of Two

IP Addressing

Private IPv4 Addresses

Classless Inter-Domain Routing

Subnetting

Address Depletion

How to Subnet

- How to Write Subnet Masks
- How Many Subnets and How Many Hosts?
- Shortcut Method
- Working out How Many Hosts and How Many Subnets
- Secondary IP Address
- Route Summarization
 - Working Out Summary Routes
 - Route Summarization Prerequisites
 - Breaking the Subnet Boundary
 - VLSM
 - VLSM Practice for the CCNA Exam
 - Chopping Down
 - In Summary
- End of Chapter Questions
 - Mini-lab – Troubleshooting IP Addressing
- Chapter 3 Labs
 - Lab 1: IP Addressing
- CHAPTER 4 — IPV6 ADDRESSING**
 - Why Do We Need IPv6?
 - Anatomy of an IP Packet
 - IPv6 Address Representation
 - The Different IPv6 Address Types
 - Migrating from IPv4 to IPv6
 - Tunneling
 - Manually Configured Dual Stack
 - Automatic 6to4
 - ISATAP Tunnels
 - NAT-PT
 - IPv6 Functionality Protocols

DHCP for IPv6

ICMPv6

Neighbor Discovery Protocol

Router Discovery

Duplicate Address Detection

Neighbor Address Resolution

Mini-lab – Neighbor Discovery Protocol in Action

Mini-lab – Configuring IPv6

End of Chapter Questions

Chapter 4 Labs

Lab 1: Simple IPv6

CHAPTER 5 — IP ROUTING TECHNOLOGIES

What Is Routing?

Prefix Matching

Building the IP Routing Table

Static Routing

Mini-lab – Configuring a Static Route

Mini-lab – Configuring an IPv6 Static Route

Gateway of Last Resort

IP Default-Gateway

IP Route 0.0.0.0 0.0.0.0

IP Default-Network

Administrative Distance

Classful/Classless

Mini-lab – Classful and Classless Routing Protocols

Dynamic Routing

Metrics

Distance Vector Protocols

Distance Vector Problems

Solving Distance Vector Problems

Autonomous Systems

Passive Interface

IP Unnumbered

Link State Protocols

Planes of Operation

Topology-based (CEF) Switching

Cisco Express Forwarding

Accelerated and Distributed CEF

Mini-lab – Configuring Cisco Express Forwarding

End of Chapter Questions

Chapter 5 Labs

Lab 1: Static Routes

CHAPTER 6 — OSPF

Overview of OSPF

OSPF Terminology

OSPF Router ID

OSPF Timers

OSPF Routes

OSPF Load Balancing

OSPF Network Types and Neighbors

Mini-lab – Configuring Single-area OSPF

Mini-lab – Configuring OSPF Interfaces

Mini-lab – OSPF Passive Interfaces

OSPFv3

Mini-lab – Configuring Single-area OSPFv3

Link State Problems

End of Chapter Questions

Chapter 6 Labs

Lab 1: Single-area OSPF

CHAPTER 7 — IP SERVICES

DHCP Functionality

Mini-lab – DHCP Configuration on Cisco IOS Routers

IP Helper Address

IP Forward Protocol

Access Control Lists

Access List Numbers

Standard IP Access Lists

Wildcard Masks

Access List Logging

Extended IP Access Lists

Port Numbers

Access Lists and Routing Protocols

Access List Rules

Configuring Access Lists

ACL Sequence Numbers

An Alternative to Access Lists

Network Address Translation

Static NAT

Dynamic NAT and Port Address Translation

Configuring and Verifying NAT

Troubleshooting NAT

Network Time Protocol

End of Chapter Questions

Chapter 7 Labs

Lab 1: Configuring a Router as a DHCP Server

Lab 2: Access Lists (Standard)

Lab 3: Access Lists (Extended)

Lab 4: Access Lists (Named)

Lab 5: Static NAT

Lab 6: NAT Pool

Lab 7: NAT Overload

CHAPTER 8 — NETWORK DEVICE SECURITY

Network Security Devices

Network Device Passwords

Enable Password

Enable Secret

Service Password Encryption

Auxiliary Password

Mini-lab – Adding a Telnet Password

Console Password

Configuring Local Usernames and User-Specific Passwords

Securing Network Devices

Privilege Levels

Login

Logging Router Access

Prevent Telnet Access

Enable SSH

Mini-lab – Enabling SSH Access

Disable HTTP

Disable CDP

Add a Banner Message

External Authentication Methods

Shut Down Unused Ports

Network Device Clock and NTP

Update the IOS

Disable Unused Services

Using ACLs to Limit Telnet and SSH Access

Restrict VLAN Information

Change the Native VLAN

Change the Management VLAN

SNMP

Securing VTP

Switch Port Security

Enabling Port Security

Who Can Connect?

How Many Can Connect?

Violation Action

Fine-tuning Port Security Configuration

Error Disable Recovery

End of Chapter Questions

Chapter 8 Labs

Lab 1: Basic Router Security

Lab 2: Switch Security

Lab 3: Switch Port Security

CHAPTER 9 — NETWORK TROUBLESHOOTING

Your Troubleshooting Plan

Network Debugging

Layer 1 Troubleshooting

Troubleshooting VLAN Issues

Troubleshooting Trunks

Troubleshooting VTP

Mini-Lab – Troubleshooting VTP, VLANS, and Trunking

Troubleshooting Host IP Addressing Issues

Troubleshooting Access Lists

End of Chapter Questions

PART 2 — ICND2

CHAPTER 10 — LAN SWITCHING TECHNOLOGIES

Spanning Tree Protocol

Port States in STP

STP Convergence

Mini-lab – STP Operations

STP Timers

Cisco's Enhancements to STP

Mini-lab – Configuring UplinkFast

STP Security

Mini-lab – Configuring BPDU Guard

Rapid Spanning Tree Protocol

RSTP Link Types

RSTP Port Roles

RSTP Port States

Per-VLAN STP and per-VLAN Rapid STP

Mini-lab – Configuring PVRST+

Load Balancing Using RSTP/STP

End of Chapter Questions

Chapter 10 Labs

Lab 1: Spanning Tree Protocol

CHAPTER 11 — UNDERSTANDING ETHERCHANNELS

EtherChannels

Link Aggregation Control Protocol

Port Aggregation Protocol

Configuring EtherChannels on Cisco IOS

Mini-lab – PAgP Configuration

Mini-lab – LACP Configuration

Layer 3 EtherChannel Configuration

Port Channel Mode On

End of Chapter Questions

Chapter 11 Labs

Lab 1: LACP EtherChannels

CHAPTER 12 — ROUTER ARCHITECTURE

Router Architecture

Router Memory

Router Boot-up Sequence

Managing the IOS

Booting Options

Cisco IOS Licensing

End of Chapter Questions

Chapter 12 Labs

Lab 1: Copy Startup Config Using TFTP

CHAPTER 13 — ADVANCED OSPF FEATURES

Advanced OSPF Concepts

Designated Router and Backup Designated Router

Establishing Adjacencies

OSPF Priority

OSPF Router Types

OSPF Link State Advertisements

OSPF Areas

Mini-lab – Configuring Multi-area OSPF

Mini-lab – Configuring and Verifying OSPFv3 Multi-area in Cisco IOS Software

End of Chapter Questions

Chapter 13 Labs

Lab 1: Multi-area OSPF

Lab 2: Multi-area OSPFv3

CHAPTER 14 — ENHANCED INTERIOR GATEWAY ROUTING

PROTOCOL (EIGRP)

EIGRP

EIGRP Terminology

EIGRP Composite Metric Calculation

EIGRP Neighbors

Reliable Transport Protocol

Understanding DUAL and Feasibility Condition

EIGRP Router ID

Mini-lab – EIGRP Passive Interfaces

EIGRP Load Balancing

Mini-lab – Configuring EIGRP

End of Chapter Questions

Chapter 14 Labs

Lab 1: EIGRP

CHAPTER 15 — ADVANCED IP SERVICES

HSRP

HSRP Interface Tracking

Configuring HSRP Interface Tracking

Mini-lab – HSRP Configuration

VRRP

Mini-lab – VRRP Configuration

Configuring VRRP Interface Tracking

GLBP

GLBP Configuration

Syslog

Mini-lab – Cisco IOS Syslog Configuration

SNMP

Cisco IOS SNMP Configuration

End of Chapter Questions

Chapter 15 Labs

Lab 1: HSRP

Lab 2: Syslog

Lab 3: SNMP

CHAPTER 16 — WAN TECHNOLOGIES

WAN Technologies

Common WAN Networking Terms

WAN Connection Types

Point-to-Point Protocols

High-Level Data Link Control

Point-to-Point Protocol

PPP Authentication

LCP Configuration Options

Mini-lab – Configuring PPP

PPPoE

Mini-lab – PPPoE Configuration

Frame Relay

Configuring Frame Relay

Troubleshooting Frame Relay

Metro Ethernet

MPLS

VSAT

Cellular Networks

T1/E1

ISDN

DSL

ADSL

HDSL

IDSL

RADSL

VDSL

Cable

VPN

End of Chapter Questions

Chapter 16 Labs

Lab 1: WAN Lab – Point-to-Point Protocol

Lab 2: Basic Frame Relay

Lab 3: Frame Relay Subinterfaces

CHAPTER 17 — ADVANCED NETWORK TROUBLESHOOTING

Troubleshooting Using NetFlow

Troubleshooting STP

 Incorrect Root Bridge

 Incorrect Root Port

 Incorrect Designated Port

Troubleshooting InterVLAN Routing Problems

Troubleshooting Routing Issues

Troubleshooting OSPF

 Mini-Lab – Troubleshooting OSPF

Troubleshooting EIGRP

Troubleshooting WAN Connectivity

Troubleshooting EtherChannels

End of Chapter Questions

CHAPTER 18 — ADVANCED LABS

This study guide and material is not sponsored by, endorsed by, or affiliated with Cisco Systems, Inc., Cisco™, Cisco Systems™, CCDA™, CCNA™, CCDP™, CCNPTM, CCIE™, or CCSI™. The Cisco Systems logo and the CCIE logo are trademarks or registered trademarks of Cisco Systems, Inc. in the United States and certain other countries. All other trademarks are trademarks of their respective owners.

Copyright Notice

Copyright © 2004–2016 Paul Browning, all rights reserved. No portion of this book may be reproduced mechanically, electronically, or by any other means, including photocopying, without written permission from the publisher.

ISBN-13: 978-1530353194

ISBN-10: 153035319X

4th Edition

Published by: Reality Press Ltd.

Midsummer Court

314 Midsummer Blvd.

Milton Keynes

MK9 2UB

help@reality-press.com

Legal Notice

The advice in this book is designed to help you achieve the standard of Cisco Certified Network Associate, which is Cisco's foundation internetworking examination. A CCNA can carry out basic router and switch installations and troubleshooting. Before you carry out more complex operations, it is advisable to seek the advice of experts or Cisco Systems, Inc.

The practical scenarios in this book are meant to illustrate a technical point only and should be used on your privately owned equipment, never on a live network. The output on our routers and switches may differ from yours due to IOS and platform model.

FOREWORD

If you are reading this book, you are embarking on the road toward becoming a Cisco-certified networking engineer. Or at the least, you are contemplating the journey and are curious to know what lies ahead.

It was not too many years ago that the only technical certification Cisco Systems offered—indeed, the only serious internetworking certification anyone offered—was the Cisco Certified Internetwork Expert (CCIE) certification. Acquiring the CCIE meant months of intensive study to tackle a hands-on lab, the details of which were mysterious to most. Those who passed the lab (few passed on their first or even second attempt) had lengthy hands-on experience in internetworking before even beginning their course of preparation. Only a very few intrepid individuals passed the CCIE lab without extensive prior experience.

Cisco Systems has, since that time, added to their program a number of intermediate certifications that can be used as stepping stones toward the coveted CCIE. You must still, in the end, prove your expertise in the dreaded hands-on lab, but these intermediate certifications are wonderful for getting you acclimatized to the rigors of testing your skills and knowledge without having to step into the most difficult of them cold.

The first milestone on the certification path is the Cisco Certified Network Associate (CCNA), the subject of Paul Browning's outstanding preparation guide. Within these pages Paul imparts to you the knowledge you need to pass the CCNA exam, step by step, using abundant illustrations, examples, and exercises. But beyond the coursework, in this book you will find the practical advice necessary for gaining hands-on experience and preparing for the certification exam that will benefit you not only for the CCNA but also through subsequent Cisco certifications to whatever objective you set for yourself.

Follow the advice and exercises Paul has provided for you here and you will have made an excellent start.

Jeff Doyle
CCIE #1919

ACKNOWLEDGMENTS

Thanks to Paul Bokor for helping with the technical edit.

BIOGRAPHIES

Paul Browning

Paul Browning (LLB[Hons], CCNP, MCSE, A+, and Network+) spent 12 years as a police officer in Coventry and Birmingham in the UK. He left for a career in IT in June 2000 and started out working at an IT help desk. After passing his MCSE and CCNA, he got a job working at the Cisco TAC north of London, which closed in March 2003. Paul then started his IT consultancy and training company, Networks, Inc. Ltd.

Since 2003, Paul has written several best-selling IT books and has trained thousands of students through his classroom and web-based IT courses.



Farai Tafa



Farai Tafa, CCIE #14811 RS/SP, is an internetwork engineer with over 10 years of experience in core IP routing, LAN and WAN switching, IP telephony, and wireless LAN implementation. He currently holds two Cisco CCIE certifications in the Routing and Switching and the Service Provider tracks. His other certifications include CCVP, JNCIA, JNCIS, and ITILv3 Foundation.

Farai lives in Dallas, Texas, with his wife and two daughters.

Daniel Gheorghe

Daniel Gheorghe is a CCIE in Routing and Switching. He is currently

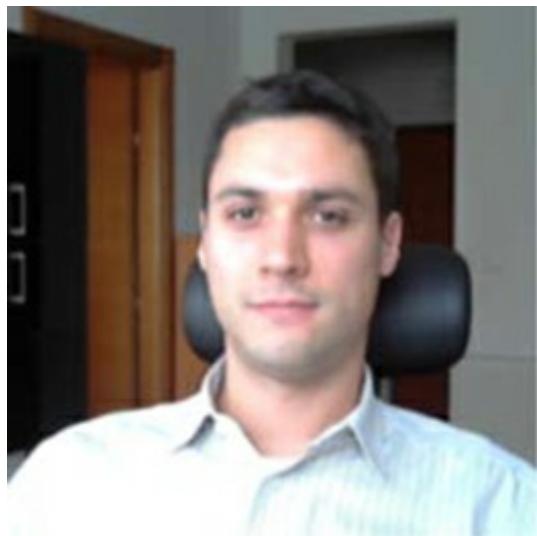


preparing for his second CCIE certification (in Security) and he is developing his skills in system penetration testing. He also holds numerous certifications in networking and security, from Cisco and other vendors, including CCNA, CCDA, CCNA Security, CCNP, CCDP, CCIP, FCNSA, FCNSP, and CEH. He took an interest in IT at an early age and soon developed a passion for computer networking, which made him study hard to reach the expert level.

Daniel has worked for different Cisco Partners and System Integrators in Romania in system design, implementation, and troubleshooting enterprise-level networks. He is also involved in several international freelance consulting projects in his areas of expertise. Daniel is a very dynamic person, and in his spare time he likes to travel and to participate in all kinds of sports.

About the Technical Reviewer

Dario Barinic



Dario Barinic is a network expert (Dual CCIE #25071 – Routing and Switching, and Service Provider) with a Master of Engineering degree and eight years of experience in the networking field. He also holds other certifications, such as Cisco CCNA and CCNP, HP AIS, ASE, and MASE, and various Cisco specializations.

Dario is specialized in the area of routing and switching (designing, implementing, troubleshooting, and operating service provider and large enterprise WAN and LAN networks). His major fields of interest are service provider/large enterprise networks (core routing and switching), network security, and passing on knowledge to enthusiastic individuals who are at the start of their networking career.

Dario works as a regional systems integrator for a Cisco Gold Partner in Zagreb, Croatia, where he lives. He is also involved in various international freelance consulting projects, primarily in the area of routing and switching.

INTRODUCTION

The Problem with CCNA Study Guides

I failed my first attempt at the Cisco CCNA exam. I studied four hours per day for around three months, but I was smashed in the exam and fell short of the required pass mark. With the benefit of having worked for Cisco Systems, running my IT consulting company, and then an IT training company specializing in Cisco networking, I now realize the problem.

First, most Cisco manuals are written by trainers who haven't worked on a live network for years. They moved into training years or decades ago and stayed there. As you know, IT changes quickly and if you don't use your skills on live networks, they get rusty fast. I experienced this myself when I started teaching, which is why I decided to keep freelancing on the side and hire three top Cisco consultants who design, install, and troubleshoot enterprise networks on a daily basis to help me write this book.

This inspired me to ask the technical editor, Dario, to add his comments whenever he felt he had something useful to add. It could be an important change to industry best practices, a change to default IOS commands, or even a configuration tip that you will benefit from in the real world. Keep your eyes peeled for a photo of Dario when he has something to add.



"Hi, from me. I'm a Dual CCIE working on enterprise-class networks that span the globe. Because I design, install, and troubleshoot very large networks, I'll be sharing some of the things I've learned with you to help you pass the CCNA exam and become a better network engineer."

Second, if around 60% of the CCNA exam's final score comes from hands-on labs and troubleshooting, then where are all the labs? At most, CCNA study guides will give you a few lines of configuration to copy but this falls well short of the mark for what you need to know for the exam. In the CCNA exam, you will be asked to configure and troubleshoot complex topics, including routing, security, and switching, using the Cisco configuration commands. All this will be timed, adding even more pressure.

For this reason, I've added 43 mini-labs that you can copy to help you understand how to configure the technologies. There are also 32 end-of-chapter full labs where you will

follow my configurations of more complex protocols and services, and you can also see the full router and switch configurations. There are also advanced labs that you will find particularly challenging. These advanced labs feature multiple layers of routing, security, and protocols that would test any CCNA-level engineer to his or her limits.

Cisco CCNA Simplified versus Cisco CCNA in 60 Days

I seem to be best known for my Cisco CCNA in 60 Days study system, but that was, in fact, the seventh Cisco study guide I'd written. The first study guide, Cisco CCNA Simplified, was published back in 2004. The first version of Cisco CCNA Simplified was written at my desk while I was working for Cisco Systems as a WAN support engineer. My entire team of 40 engineers had just been informed that our jobs had been outsourced to the Philippines, meaning we were being "let go."

I had no idea that when I printed the book, it would become an international best-seller. It was used by many thousands of students and Cisco trainers for years, but when Cisco added one of its three yearly updates to the exam, I decided to replace Cisco CCNA Simplified with another study guide called Cisco CCNA in 60 Days. I figured that breaking everything down into daily study tasks, exams, and hands-on labs was a far better solution, and the results were amazing. Every week I hear from students who tell me that they've finally passed the exam after wasting months or even years trying other methods.

But over time, I also heard from other students. Busy parents who were raising young families and people who were working long hours or had other time constraints simply couldn't put in the two hours per day required by the Cisco CCNA in 60 Days study system. I realized that there needed to be an equally effective but somewhat less intense study method. For this reason, I decided to do a complete rewrite of Cisco CCNA Simplified, not only to update it with all the latest exam topics but also to improve on the entire study experience based on what I've learned over the last 15 years in the IT industry that really works for students.

This is an entirely new and different book from Cisco CCNA in 60 Days, so you are best off using just one of these two study methods. If you've bought this book (as opposed to reading a sample or browsing through it at a book store), then stick with this book and find another CCNA manual to use for reference if you need one.

Free Bonus Material

I've learned so much since publishing my first Cisco study guide. One of the main lessons is that even with hundreds of proofreadings and technical edits, mistakes still creep in. Even the mighty Cisco Press is often heavily criticized for having a large number of errors in their manuals. For this reason, I've added free bonus website

access. On the URL below you will find:

- Errata
- End-of-chapter interactive exams
- Bonus labs (with solutions)
- Bonus exams
- Other study goodies

www.howtonetwork.com/ccnasimplified

This gives me an easy way to ensure that I plug any gaps in the learning that you get from the study guide, as well as save on the size of the manual, which would be over 1,000 pages if I added all the extra bonus goodies here.

I also have an older website for the old version of the CCNA exam. Around 70% of the content is still relevant for the current exam, but because I've updated it all I made it free access. Just bear in mind that it doesn't match the current exam syllabus but, nonetheless, it has a lot of useful videos, exams, and labs.

www.howtonetwork.net

Cisco CCNA Simplified Video Course

All you need to pass your CCNA exam is this book, the free online exams you get access to, and hands-on time with some form of Cisco rack. I will go into more detail on your rack options in the "How to Do the Labs" section below.

I also run an online IT certification website called www.howtonetwork.com, where I've added a video training course to accompany this study guide. Membership also includes access to over 30 other IT training courses and two live Cisco racks. For those of you who are just about to jump over to Amazon to kick me for "selling" stuff, I want to make it clear that you **DO NOT NEED** the video course. It's there for anyone who prefers lectures as a learning style.

Unfortunately, I have to make a small charge for access to cover the thousands of gigabytes of streaming video costs and the hosting and power costs of the live Cisco racks and web hosting, but it works out to be just 66 cents per day (\$20 monthly). Just to be clear once again, you don't need to use the video course to pass the exam, so if money is tight, you are all set with what you already have.

Reviews

Many people think that Reality Press Ltd. is a big publishing house, but it's really just me and some freelancers helping out with technical edits, proofreading, and formatting. I don't have the big promotion teams and massive marketing budgets of Cisco Press, so I

rely on people like you to help me out by telling others about the book and posting reviews.

I know that you are busy, but if you find time to post a review on Amazon, I'd like to say thanks with a 50-question CCNA exam. To deal with the strange people who think that I'm out to bribe them, you get the exam no matter what the review is, good or bad, one or five stars. If you think there is an issue with the book, please be a decent human being and get in touch so that I can add any updates or clarifications to my website, instead of kicking me on Amazon.

Send a screenshot of your review to ccnasimplified@howtonetwork.com and I'll send you your exam (no matter what your review says).

If you have any suggestions for changes or spot mistakes, then please visit <https://www.howtonetwork.com/ccnasimplified> and post a message and I'll add corrections and updates to the bonus page for the book.

Also from Reality Press Ltd.

Cisco CCNA in 60 Days

101 Labs for the Cisco CCNA Exams

101 Labs for the Cisco CCNP Exams

How to Read Cisco CCNA Simplified

Most students read study guides and by the time they have turned the page, they have forgotten what they just read. Reading a technical manual and expecting the information to become burned into your brain is a recipe for frustration and failure.

Since you have trusted me enough to buy this study guide, trust me a bit more and allow me to share my secret to learning the subject matter in record time. It involves using a variety of modalities and borrows some tried and tested speed reading techniques.

First, read through the entire book glancing at each page. Just cast your eyes over each page and look at any diagrams that may be present. Do a few pass-throughs, allowing just a few seconds per page. You don't need to be concerned about recalling anything at this stage.

Do this once per day and then each day focus on a specific chapter. Glance over it and then do another four pass-throughs, each time taking a bit longer. Note any important

commands that you see in Courier New font, but don't be concerned about recalling the information.

Finally, do a slower read-through but still don't worry about what sinks in or doesn't. You don't have to take the exam tomorrow, so why worry? Get a notepad and jot down any important learning points or configuration commands that you see. It's vital that you have access to a Cisco router, GNS3, or Packet Tracer so that when you see a command you can type it out.

If there is a follow-along lab, do the lab and then take the end-of-chapter exam. Again, if you score 10%, who cares? Just let the knowledge soak in at the rate that is right for you. You will have good days and bad days.

So, by the end of each session, you will have:

- Scanned the entire book
- Scanned and then slowly checked over that day's chapter
- Made some useful notes
- Typed out some important IOS commands
- Taken a practice exam

If that sounds like a perfect study session to you, then you are right. Compare that to staring at the page, trying to burn 20 pages of advanced OSPF concepts into your memory, and then getting so frustrated that you want to throw the book out the window. That's what everyone else will be doing.

The next day, review the previous day's lesson without regard to how much has sunk in. Then scan the entire book again (read it in five minutes) and start on the next chapter, doing what you did the day before (i.e., making notes, typing commands, doing labs, and taking an exam).

As the days go by your confidence will grow, you will find that concepts are beginning to look familiar, you will know what's coming on the next page, and you will know which commands do what. You will also be able to apply your knowledge to hands-on labs and exam questions.

On the free resources page, I've included some super-hard labs and super-hard exams. Once you can do these with ease, you will know that you are ready for the real exam. This might take you two months or it might take you six, depending on your dedication, previous level of knowledge, and how consistent you are. If you try studying the hard way, you will be doomed to failure.

It's worth mentioning that due to the way the syllabus is set out, I have to mention some topics in brief before actually explaining them in detail, such as VLSM and NBMA. It's

impossible to discuss routing protocols without mentioning NBMA, but Cisco has put WAN topics at the very end of the ICND2 syllabus. If you have scanned the book a few times, it will really help you to at least recognize what these terms are referring to.

I've listed the syllabus topics covered at the start of each chapter. They are in the ICND1 and ICND2 format so you will see that some numbers are duplicated. Please also bear in mind that Cisco regularly adds non-syllabus topics to the exam, so it's impossible to cover every conceivable topic.

The Cisco CCNA Exam Format

If you visit <http://www.cisco.com/go/ccna> and read up on the CCNA Routing and Switching (CCNA RS) exam, you will glean some very valuable information about the exam's format, syllabus, and options. You will also see that you can pass the CCNA exam by taking a one- or two-exam route.

It's worth noting the exam numbers because there are several types of CCNA exams, including Wireless, Security, Service Provider, etc., and you could easily book the wrong exam. Your exam options are:

200-120 CCNA Interconnecting Cisco Networking Devices: Accelerated (CCNAX)

OR

100-101 ICND1 Interconnecting Cisco Networking Devices Part 1 (ICND1) and then the

200-101 ICND2 Interconnecting Cisco Networking Devices Part 2 (ICND2)

There is no right or wrong answer to which route is best. If you take the one-exam route, you will save money but you will have to deal with every subject in the syllabus. If you take the two-exam route, you will focus on fewer subjects per exam; however, you will have to pay for two exams. Passing the ICND1 exam gets you the Cisco Certified Entry Networking Technician (CCENT) qualification, which is a minimum requirement for some CCNA specialization exams such as CCNA Security.

I've modeled this study guide on the two-exam format. If you want to take the CCNA RS exam in two parts, then focus on the first section for the ICND1/CCENT exam and then the second section for the ICND2 exam. If you want to do it all in one exam, then study the entire guide.

The CCNA exam is broken down into:

- Multiple-choice single answer
- Multiple-choice multiple answers
- Fill-in-the-blank questions

- Drag-and-drop questions
- Questions based on diagrams
- Questions based on access to routers or switches
- Hands-on configuration labs
- Hands-on troubleshooting labs

Make no mistake, the exam is very hard indeed. But if you follow all my advice, take the exams I provide, and do all the labs until you know all the theory and commands by heart, then you will pass.

Earlier versions of the Cisco CCNA exam required you to have a basic understanding of the Cisco IOS, IP addressing, subnetting, and TCP/IP. Things have changed dramatically since then; today, CCNA-level students are required to have a strong understanding of routing concepts, IPv6, troubleshooting, and hands-on configuration skills using routers and switches. In addition, many CCNP-level subjects such as FHRP, EtherChannels, and advanced OSPF have been added to the CCNA exam, which must be understood in order to carry out configurations and troubleshooting.

That being said, one can safely assume that the CCNA RS is not a suitable exam for those new to internetworking. I was surprised to receive some negative comments from readers of my other CCNA manuals about the fact that I don't explain some basic networking concepts. Cisco explains in its CCNA documentation that:

“CCNA Routing and Switching is for Network Specialists, Network Administrators, and Network Support Engineers with 1-3 years of experience.”

For this reason, you should already have a good grasp of subjects such as TCP/IP, Ethernet, and network cabling. I do recap some of these subjects in the book, but if you are a beginner, you should study a CompTIA Network+ book first. If I added all the basic networking material to this book, it would be at least 1,000 pages thick.

Cisco does provide a lot of very useful information on their website about the exam, as well as the testing format. At the moment, the exam is 90 minutes long and has between 50 and 60 questions. Because part of the exam is practical, you will never know how the marks are allocated, so do your best to answer all of the questions and complete all of the labs. The pass mark can vary, but you need to aim for at least 860 out of 1000 (or 86%); again, this can vary.

You need to prepare well to pass the exam, with a focus on these areas in particular:

1. Theoretical Questions
2. IP Network Design

3. Hands-on Labs
4. Troubleshooting (theory and labs)

1. Theoretical Questions

This area involves applying what you have read in your study guide to answer the questions. These could be multiple-choice questions with a single answer, multiple correct answers, or fill-in-the-blank answers. In this part of the exam, you will look at a network diagram or screenshot of router or switch output and answer questions based on the image, or you may have to give a best-match answer, such as seeing a certain error message and deciding what is its most likely cause.

To prepare for these questions, you need to read your study guide many times over, make notes, and take a lot of practice exams.

2. IP Network Design

By network design, I mean allocating addressing schemes or designing the best addressing system to solve the challenge that you will be presented with. For example, you could be shown a network design that requires you to provide an IP addressing scheme that will give the correct amount of networks and hosts per network. You could also be given a network addressing issue to solve. We will cover IP addressing and related topics in greater detail later in this book.

3. Hands-on Labs

This is possibly the most challenging part of becoming a Cisco engineer, as you must learn how to use the Cisco Internetwork Operating System (IOS) to both configure and troubleshoot routers and switches. In the exam you will be presented with a topology that you may not be familiar with, and because you are using a simulator rather than a live router or switch, not all of the commands or shortcuts you are used to will be available to you. You are also timed in the exam, whereas you are not in the real world.

Hands-on labs can be broken down into:

- Configuring devices according to instructions
- Troubleshooting a non- or partly functional network
- Using show commands to answer multiple questions

This guide and the bonus material is jam-packed with hands-on labs and configuration examples. Every time you see a command you are not familiar with, you should type it out on your router or switch. You must do this over and over again until they become second nature to you.

4. Troubleshooting

A big change to the latest CCNA exam is the introduction of troubleshooting. You need to understand troubleshooting methodologies (tactics) and be able to answer questions about the most likely cause of an issue given a set of facts. You may also be asked to log in to network equipment and use show commands to determine the cause of a fault, or you may have to remedy the fault with configuration commands.

As you progress through the book, you will be armed with all the skills you will need to tackle the troubleshooting sections of the exam.

How to Do the Labs

Success in the CCNA lab hinges on your hands-on Cisco skills. To this end, this guide has hands-on labs addressing all of the major areas you will be expected to know for the lab. This guide goes further than any other guide on the market I have seen by addressing troubleshooting skills and guiding you through every step of the configuration, so you really understand what you are doing.

Hands-on experience is crucial to your success in the exam and your credibility as a CCNA. Some vendors' certifications used to attract a huge amount of kudos, but as employers began to discover, many people were passing the exams by reading books and studying from brain dump sites. How would you like an operation to be performed on you by a surgeon who had learned how to operate only from books and exam cheat sources? Well, the same goes for network engineers.

You have four options in attaining the crucial hands-on exposure needed to pass the practical part of the CCNA exam:

1. Buy a home study rack
2. Rent online router time
3. Use a router simulator
4. Use a router emulator

1. Buy a home study rack

The first is the best option, by far. Use the real equipment, putting in the cables and doing the labs over and over and over again until you feel like throwing the routers out the window.

There are several companies on eBay that sell CCNA racks for exam preparation. You should invest in at least three routers and two switches. Check the Cisco.com website for the models you need because Cisco updates their models every year. At the time of writing this guide, the current switch was the 2960 model. Cisco does not specify the model of the router but the IOS version is 15.x; in my honest opinion, you can get away

with any router running 12.4. Just ask the seller which version is installed because many sellers have a contract with Cisco to add whichever version they need.

2. Rent online router time

Online rack time can be a little tricky. Most online Cisco rack rental companies cater to the CCIE exam and include 15 or so devices, which is overkill for the CCNA exam. You also have to rent large chunks of time because if you are taking the actual CCIE exam, it takes at least 18 months to study for it.

Cisco Systems offers a virtual rack solution that you can rent by the hour. I'm afraid I've never used it so I can't comment with any authority. However, there is a free CCNA/CCNP rack available to members of www.howtonetwork.com via the link below:

<https://www.howtonetwork.com/live-cisco-racks/>

3. Use a router simulator

These were all the rage when the CCNA exam was first developed. A router simulator doesn't run Cisco software. Instead, many of the commands available on Cisco devices are added to a list of options for you to enter into a simulator. As other options have become available, simulators have all but died out. The only exception is Packet Tracer, which was created by Cisco. At the moment, Packet Tracer is officially available only to Cisco Network Academy students.

If you can't get a hold of live devices or an online rack, Packet Tracer will be enough for you to study for the CCNA exam. Please bear in mind though that the commands and results of your configurations won't always match what you see on live equipment. This often causes confusion to students trying to complete labs configured on live equipment.

4. Use a router emulator

A router emulator allows you to run actual Cisco IOS code on a Linux machine or via a virtual machine on your PC. This has been developed into a robust solution for companies and individuals to study for exams or do testing. The most accessible solution is free and provided by www.gns3.com. It doesn't support switching in the way you will need to study for the CCNA exam, but all routing is supported.

I've created a free virtual version via the link below. Please note that I don't offer any support and you must have your own version of Cisco IOS to load.

<https://www.howtonetwork.com/vrack/>

Rack Topologies

I decided not to use a fixed topology for this guide because so many of you will be using

different resources (listed above), so feel free to swap whatever interfaces I'm using for what you have available. If I'm using Serial 0/0 but you have Serial 0/1/0, then just write out the topology on a piece of paper swapping my interfaces for yours.

If I used a fixed topology in this guide, there is the risk of you getting so used to this configuration that come exam day you will be thrown by the new interface numbers and connection types. This has happened to me and other IT colleagues so I wanted to prevent this from happening to you. For this reason, I strongly recommend mixing things up a bit once you get used to doing a few labs. For example, change interface names and, if possible, configure the same protocol over Ethernet instead of Serial connections.

Other Resources

There are many other study websites and, of course, study guides out there. I'd personally advise you to use one main study guide and use either another study guide or Cisco.com to clarify any areas you still aren't clear about. So many students buy several IT guides, website memberships, and other resources and then they become overwhelmed.

PART 1 — ICND1

Chapter 1 — Operation of IP Data Networks

What You Will Learn in This Chapter

Overview of Networking Equipment

Understanding the OSI Model

The TCP/IP Model

TCP/IP Services

Ethernet Concepts

IEEE Standards

Router Interfaces and Connectors

Connecting to a Router

Router Modes

Configuring a Router

The Configuration Register

Syllabus Topics Covered

1.0 Operation of IP Data Networks

- 1.1 Recognize the purpose and functions of various network devices, such as routers, switches, bridges, and hubs
- 1.2 Select the components required to meet a given network specification
- 1.3 Identify common applications and their impact on the network
- 1.4 Describe the purpose and basic operation of the protocols in the OSI and TCP/IP models
- 1.5 Predict the data flow between two hosts across a network
- 1.6 Identify the appropriate media, cables, ports, and connectors to connect Cisco network devices to other network devices and hosts in a LAN

2.0 LAN Switching Technologies

- 2.1 Determine the technology and media access control method for Ethernet networks
- 2.2 Identify basic switching concepts and the operation of Cisco switches
 - 2.2.a Collision domains
 - 2.2.b Broadcast domains
- 4.4 Verify router configuration and network connectivity using

4.4.a Ping

4.4.a (i) Extended ping

4.4.b Traceroute

4.4.c Telnet

Let's start our CCNA journey by looking at some internetworking basics. If you have studied for the CompTIA Network+ exam previously, which thoroughly covers many subjects that CCNA study guides can't cover in detail due to space, such as cabling standards, network security, cloud computing, virtualization, and much more, then this should already be familiar to you.

Overview of Networking Equipment

The earlier versions of the CCNA exam did not focus on basic networking concepts and theory, but this has changed now. Cisco expects you to have a good working knowledge of general networking, LAN and WAN topologies, and equipment (hence my suggestion to read a good Network+ study guide). We will cover some of the basics in this book, but I'll presume that this is just a recap for you.

A computer network can be as small as two computers connected by a single cable (or wirelessly) to the largest network in the world—the Internet. To connect a large number of PCs, specialized equipment and protocols have been designed to carry out tasks such as segmenting domains, preventing broadcast storms, and moving packets from one part of the network to the other as efficiently as possible.

Hub

The most rudimentary piece of networking equipment is a hub. Hubs are fairly rare nowadays, but we will refer to them later on because they explain why we needed to change the way traffic is sent across a Local Area Network (LAN).

A hub simply allows several networking devices to communicate. Each device plugs into a port on the hub using a network cable (more on these cables later). The simplest network you can build is a few PCs connected to a hub. Hubs have no memory or hard drive so they can never remember which device is plugged into which port. When a hub receives data on one port, it just forwards it to all the other ports. This causes a lot of unnecessary traffic to pass through the network (see Figure 1.1 below).

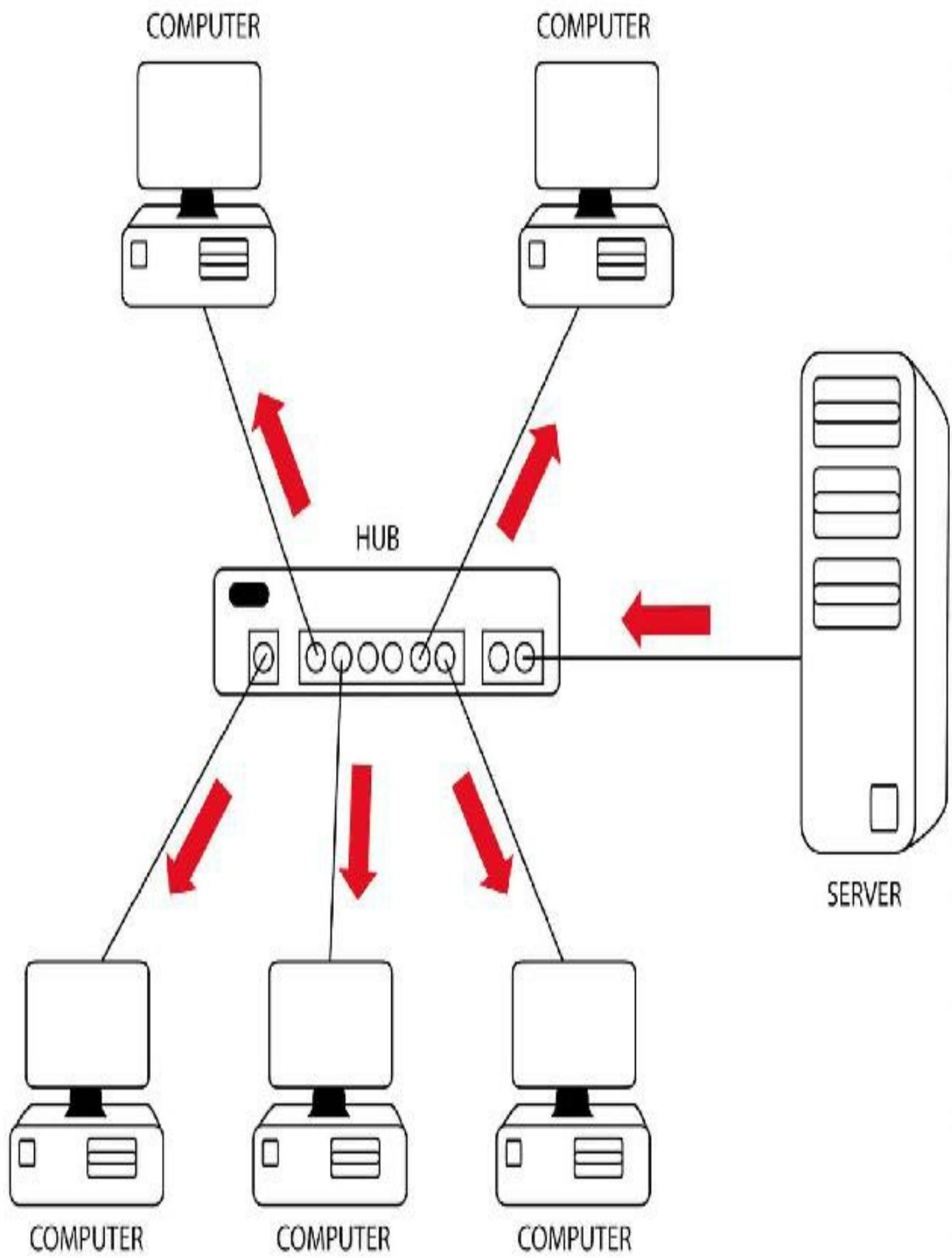


FIG 1.1 – Every frame is received by every device when a hub is used

Switch

One drawback of using hubs is that, because they have no memory, they can never keep a record of which PC is plugged into which port. For this reason, every time one PC wants to speak to another, every single PC plugged into the hub gets a copy of the information (contained within a frame) sent out on the wire. Every time a PC receives this information, it has to use processing power to determine whether it is the intended recipient. As you can imagine, this is very inefficient and can become a major problem as more and more devices are connected to the hub.

A data frame sent to every device on a network segment at the same time is referred to as a broadcast. Too many broadcasts in a network can cause delays and dramatically reduce performance. A high amount of broadcasts causes an enormous amount of traffic to traverse the network at any one time. A broadcast is usually sent when a data frame is trying to find a host in the network and doesn't know its current location.

Switches operate by building a list of which PCs are connected to which ports, allowing the available bandwidth to be used a lot more efficiently. If a PC wants to send data to another PC via a switch, the switch will forward the traffic only to the port the intended recipient is connected to. If it doesn't know the port, it will send out a broadcast to find out where in the network the PC is. Switches and hubs are designed to forward broadcast traffic as data frames addressed to every device in the network.

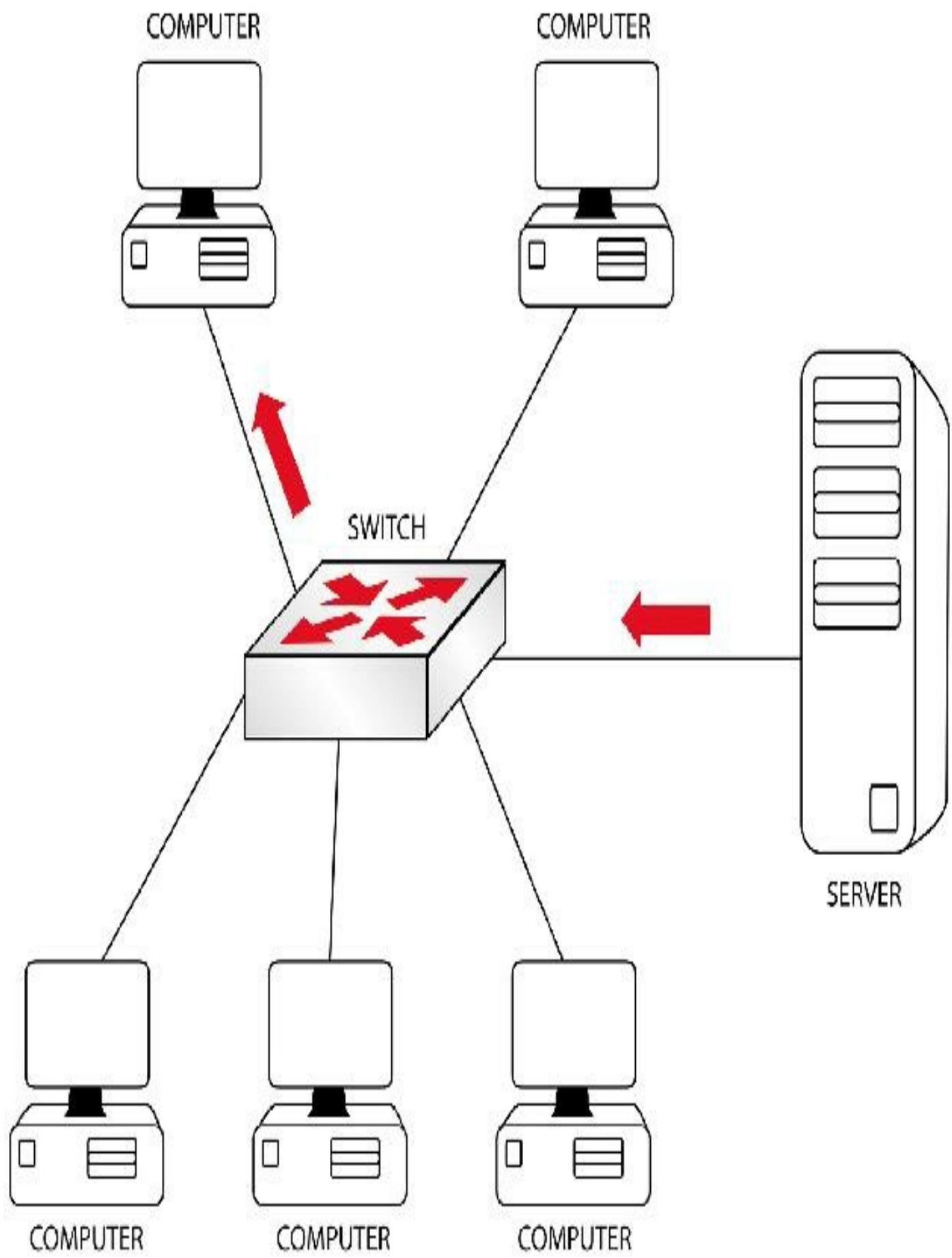


FIG 1.2 – Switches forward frames only to the relevant port

Because switches only forward broadcast information when the destination is unknown, they are used to create smaller collision domains. A collision domain consists of an area in the network that a data frame will reach if there is a collision. In earlier implementations of Ethernet, all the devices were in the same collision domain because they were on the same wire or they were all connected to a hub.

A collision occurs when a data frame, traveling along a network cable, collides with another frame. The collision causes the data inside the frame to become corrupted. This corrupted frame is received by every device within the collision domain. Smaller collision domains mean that traffic will move faster throughout the network.

It's very important to be able to recognize collision domains in the CCNA exam because you could be presented with a diagram showing routers, switches, and hubs and asked how many collision and broadcast domains have been created. Figure 1.3 below demonstrates a network hub that has created one collision domain (it's also one broadcast domain because there are no VLANs or routers present). If you ever see a hub in a network diagram referring to collision and broadcast domains, remember that the hub does not increase the number of collision domains or reduce the number of broadcast domains.

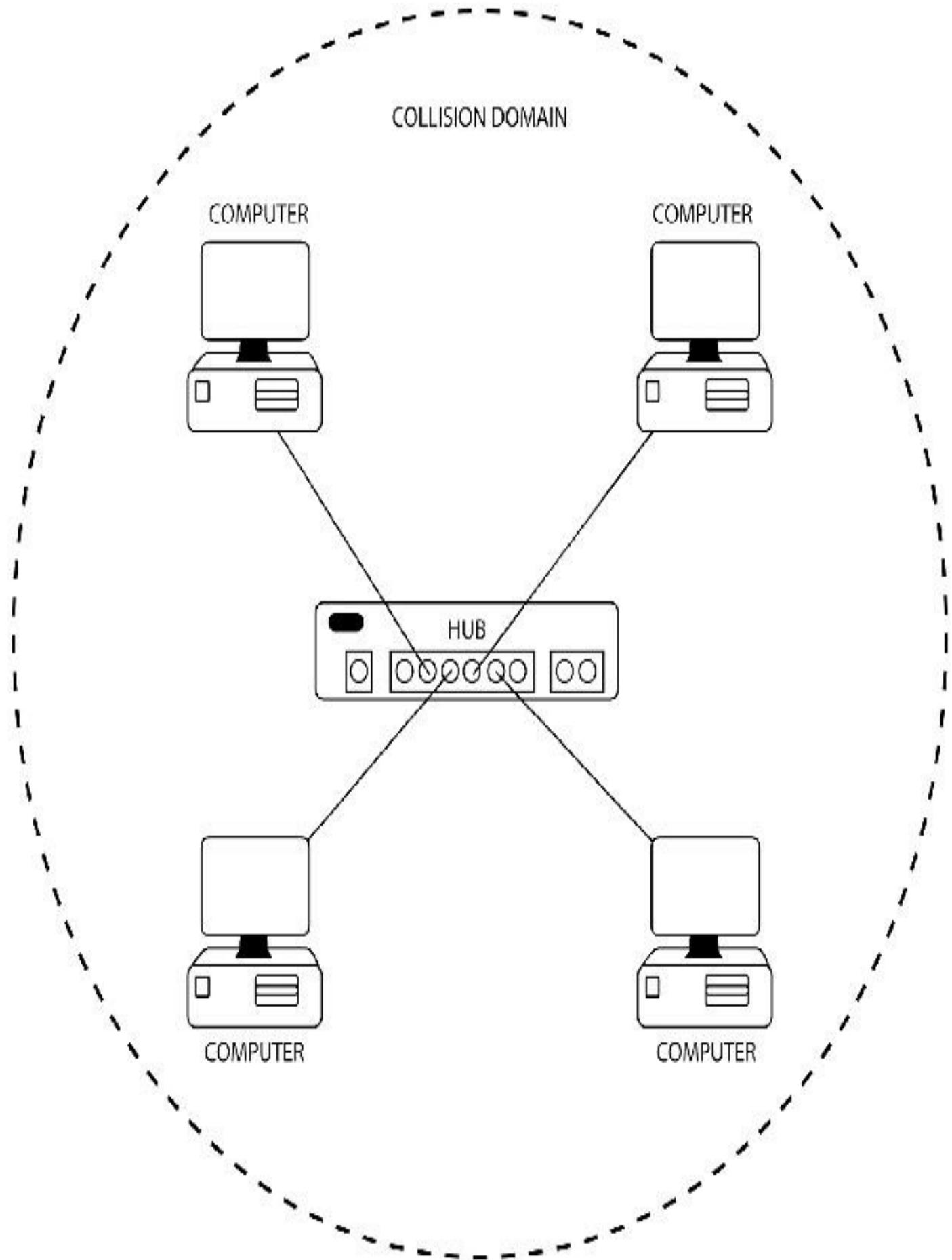


FIG 1.3 – Hubs are in one collision domain

If you swapped the hub for a switch you would have four collision domains (one per port used). All the devices would still be in the same broadcast domain though.

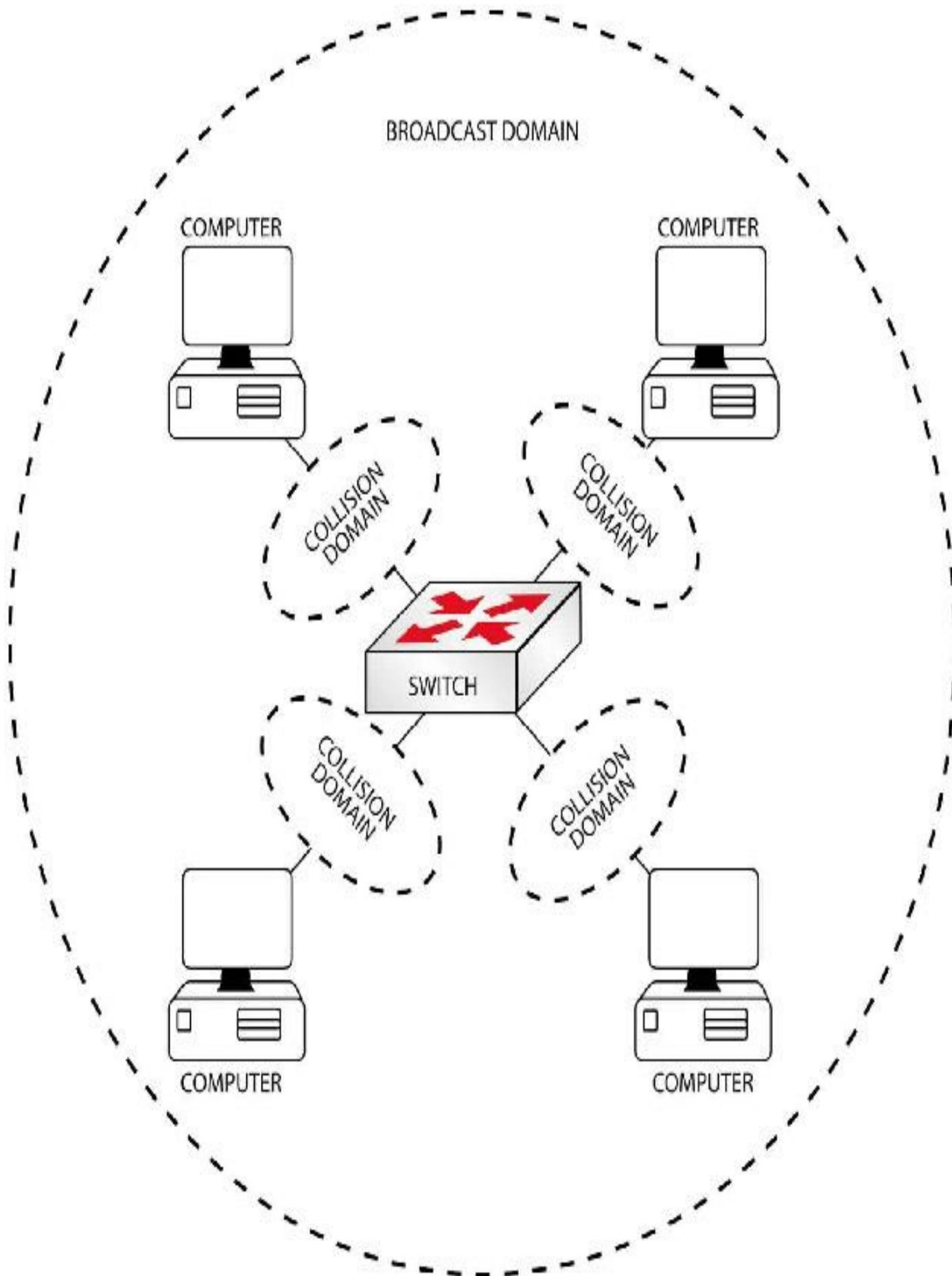


FIG 1.4 – Each switch port creates a collision domain

We will cover this in more detail later, but it's worth noting that if you created two VLANs on the switch you would have two broadcast domains. More VLANs equal more broadcast domains.

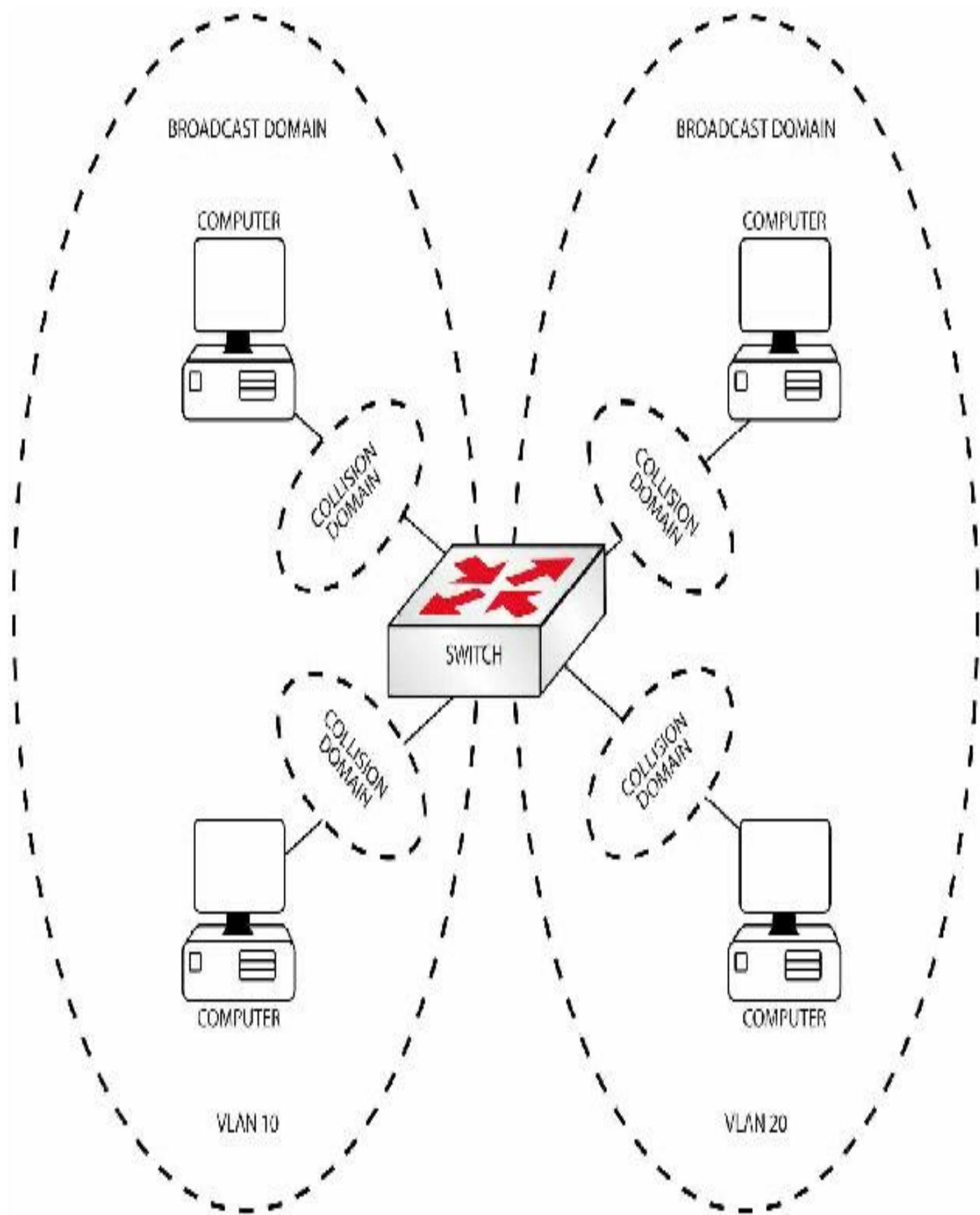


FIG 1.5 – VLANs create broadcast domains

We will cover collision domains in more detail in the next chapter.

Modern network standards have all but eliminated collisions in the network, but you still need to understand them in case you need to troubleshoot them on your network. Figures 1.6 and 1.7 illustrate the problem caused when frames collide on the wire.

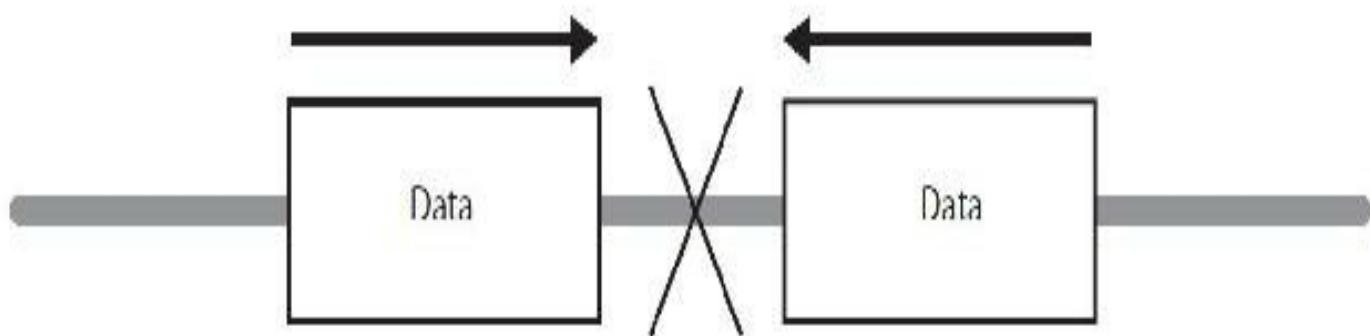


FIG 1.6 – Frames can collide on the wire

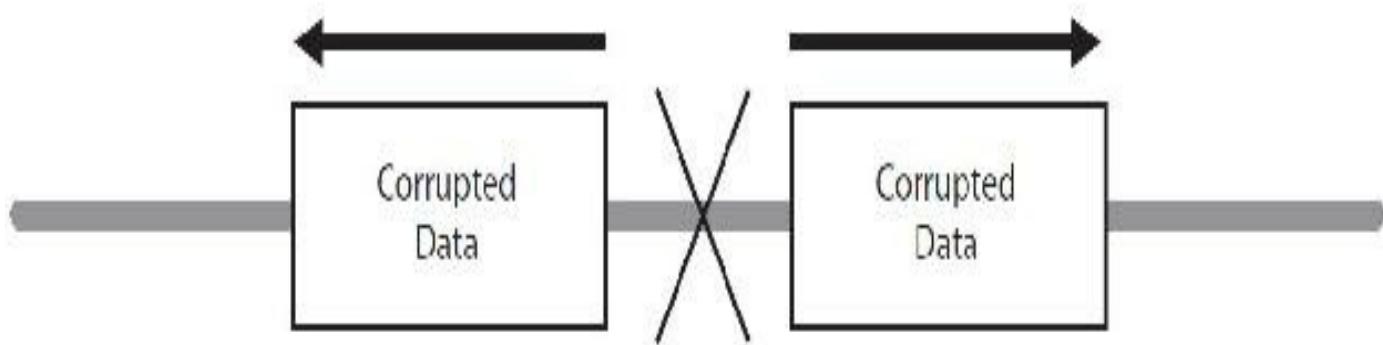


FIG 1.7 – A corrupted frame is heard by every device in a collision domain

You may have read about network bridges in older networking manuals. A bridge is similar to a switch; however, they usually have only two ports, whereas a switch has several ports. A switch is basically a multiport bridge.

Cisco Systems offers a large range of network switches to service small offices all the way up to large service providers. It would be well worth your time to visit Cisco.com and browse the available models and features. At the time of writing this guide, the switch used in the exam was the 2960 model.



FIG 1.8 – Cisco 2960 Switch

Using layer 2 switches to create fewer users per segment is known as

microsegmentation. Microsegmentation creates dedicated network segments (i.e., one user per segment). Each user receives instant access to the full bandwidth and therefore does not have to contend for available bandwidth with other users. The outcome is that collisions (commonplace when using hubs) no longer occur (provided you are using full-duplex).

Switches offer another advantage over hubs. Most can store frames in buffer memory, allowing them to be stored and then forwarded sequentially when the wire is clear.

Router

A router is designed to store a directory of networks. Rather than concerning itself with which PC is where, a router's job is to find out where different networks are. It then sends the traffic via the best path. This path could be the fastest, most reliable, or shortest, or a combination of these features, depending on how you want traffic to be sent as the network administrator. If the router does not know how to get traffic to its intended destination, it will either drop the packet or forward it to another router that should know how to get it there (we will cover default gateways later).

It is important to remember that by default, routers do not forward broadcasts. If they did, you would find that most networks, including the Internet, would be extremely slow because of all the broadcasts passing across them.



FIG 1.9 – Cisco 1900 Series Router

Because they do not forward broadcast information, routers are used to create broadcast domains. Broadcasts in the network will stop at the router (unless you configure it to forward them, which isn't recommended). We'll put all of this together when we revisit collision domains in the next chapter.

For the exam remember that switches segment collision domains. Every port on the switch is a separate collision domain, which means a collided frame won't travel past the port. Also remember that routers segment broadcast domains, and every port on the

router is a separate broadcast domain. Finally, all the ports in a hub are in ONE collision domain, while all the ports in a switch are in the same broadcast domain.

The Open Systems Interconnection Model

In the 1980s, there was a huge increase in the amount of companies producing networking equipment and protocols. It was very difficult to connect networks together and almost impossible to do so using different vendors' equipment. The job of standardizing networking fell on the International Organization for Standardization (ISO). The ISO created a model for every company to follow when designing networking hardware and software. This model was named the Open Systems Interconnection model, more commonly known as the OSI model.

The OSI model not only serves as a reference, there is also a practical value in using it. There must be some way to order things so we know which devices do which job. What if your company wants to buy switches from a different vendor than the one they buy their routers from? How can it be sure the equipment will work together?

Advantages of using the OSI model include the following:

- Allows different vendors' equipment to work together
- Allows different types of network hardware and software to communicate
- A change made in one layer does not affect any of the other layers

The OSI model consists of seven different layers and each layer is responsible for a specific function or set of functions. We always refer to the model in the order below, starting with layer 7:

- Application (layer 7)
- Presentation (layer 6)
- Session (layer 5)
- Transport (layer 4)
- Network (layer 3)
- Data link (layer 2)
- Physical (layer 1)

The application layer is also known as layer 7; it is never called layer 1. The physical layer is always known as layer 1 and so on.



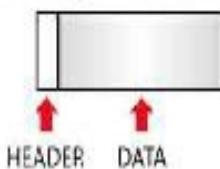
An easy way to remember the order of the layers is to use the acronym APSTNDP (All People Seem To Need Data Processing).

Encapsulation

As data passes down each OSI layer, a new header is added to it; this process is called encapsulation. The header contains information on how the data should be treated by the receiver. As data is encapsulated while moving down the layers, it will be known by a different name. This is necessary because each layer requires a different set of information and addressing to work properly. When the data is received at the destination, it is then de-encapsulated, a process that removes each header, and then the information is passed up to the next layer.

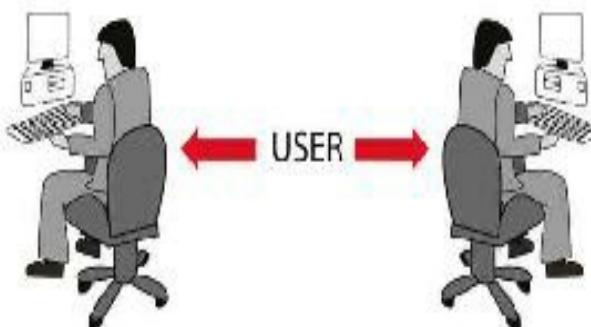
As shown in Figure 1.10 below, the order of data encapsulation is data, segment, packet, frame, and bit. An easy way to remember the order of data encapsulation is to use the acronym DSPFB (Don't Some People Fry Bacon).

PDU (Protocol Data Unit)
(units of data passed between layers)



TERM FOR
A UNIT OF
DATA AT
THIS LAYER

TRANSMIT



RECEIVE

TERM FOR
A UNIT OF
DATA AT
THIS LAYER

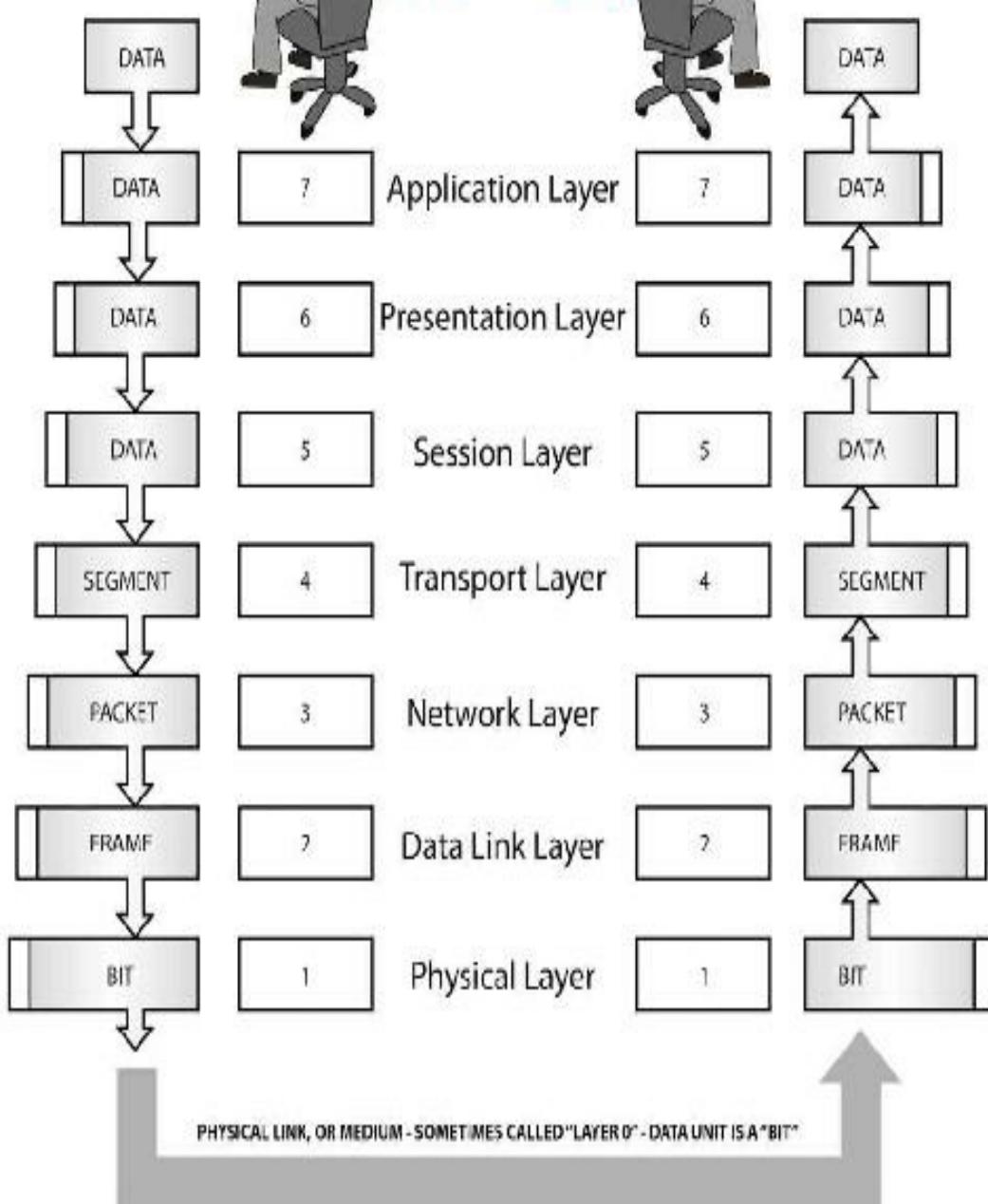


FIG 1.10 – Five steps of data encapsulation

Data turns into a segment at the transport layer, which turns into a packet at the network layer, which turns into a frame at the data link layer, which turns into a bit at the physical layer. Any exam question that refers to a data frame or packet and asks which OSI layer it applies to can be easily answered if you remember the acronym for the five steps of data encapsulation above.

An easy way to see the division of the various layers is by examining a packet capture from software such as Wireshark. We will be looking at packet captures throughout this guide, so download your own free copy at <https://www.wireshark.org/> (if you downloaded the GNS3 virtual machine, packet capture software is included).

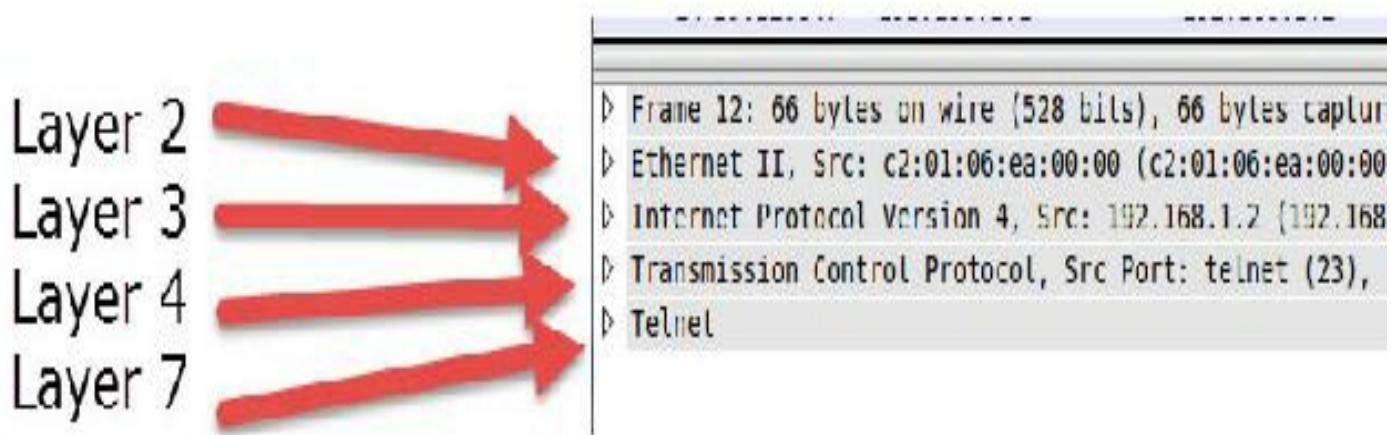


FIG 1.11 – OSI as seen inside Wireshark

If you are using Wireshark you can use another view via Protocol Hierarchy Statistics, which will display the OSI layers used as shown in Figure 1.12 below:

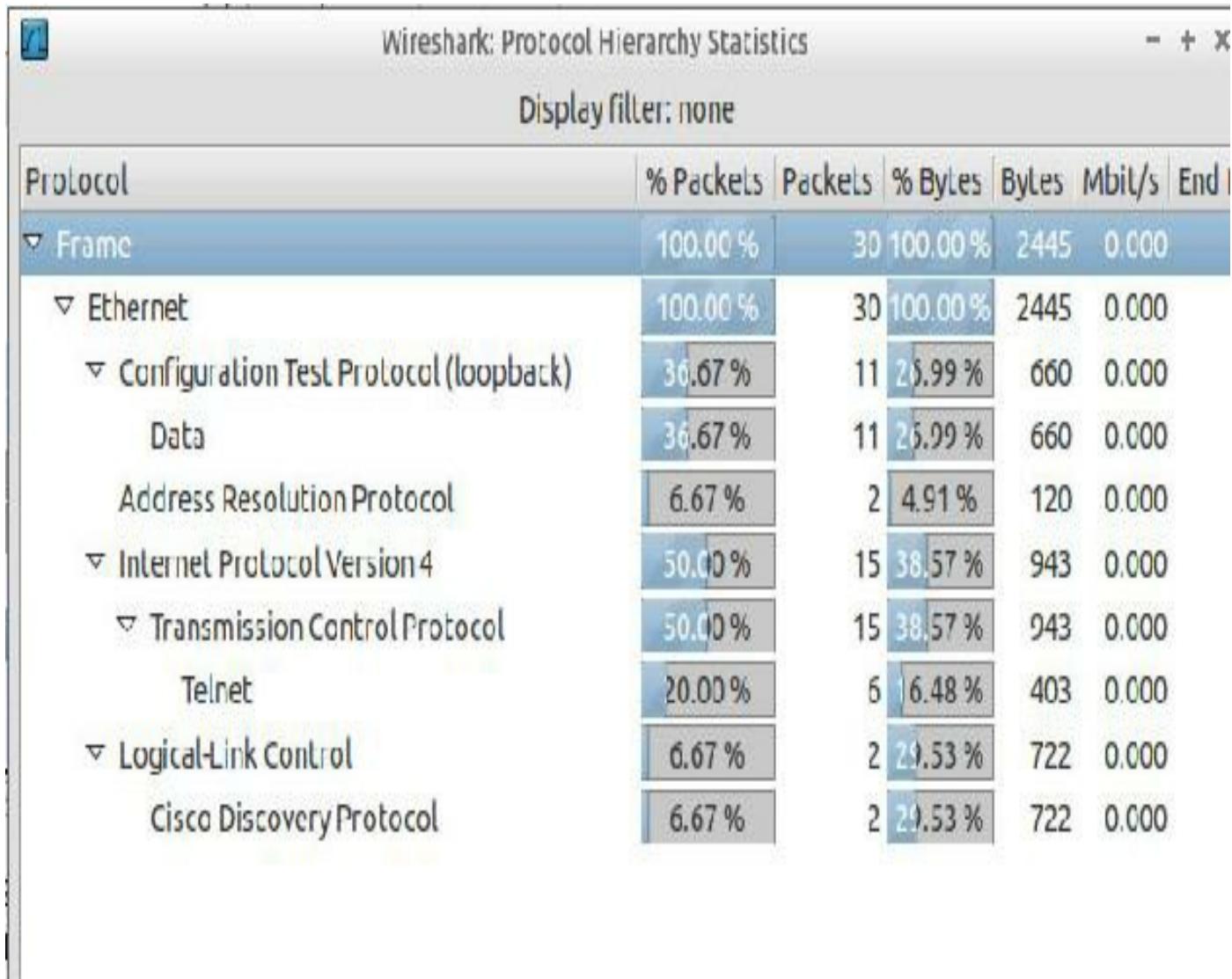


FIG 1.12 – Protocol hierarchy statistics inside Wireshark

Application Layer

The application layer is where most users interact with the network. It establishes whether the destination is available to communicate and determines whether sufficient resources are available to do so.

There are many services that operate at the application layer. Some of them include:

- **World Wide Web (WWW)** – connects millions of users to servers and provides multimedia functions such as text, graphics, and sound
- **E-mail (SMTP, POP)** – the standard used to send and receive e-mail all over the world
- **File Transfer Protocol (FTP)** – provides a means to upload and download large files over networks (imagine having to e-mail a colleague a 20 Mb file!)
- **Telnet** – used to connect to networking devices remotely (many network

engineers connect to their networking equipment many miles away from the actual physical location)

In the exam you might be expected to explain how you would test whether all seven OSI layers are working correctly on your network. The answer would be that you could telnet or FTP to another host.

Presentation Layer

The function of the presentation layer is to present data to the application layer. It converts coded data into a format the application layer can understand. It is also responsible for data encryption, data decryption, and, finally, data compression.

The presentation layer converts many multimedia functions for the application layer, including:

- JPEG (Joint Photographic Experts Group) – a widely used image format
- MPEG (Moving Pictures Experts Group) – the format used for video compression and coding
- QuickTime – manages audio and video for Macs or iPads
- ASCII (American Standard Code for Information Interchange) – the standard format for text and data

Aside from what is listed Graphic Image File (GIF), Bitmap (BMP), MP3, and EBCDIC or mainframe language, any protocol that will change the look of the data operates at the presentation layer.



Session Layer

In the session layer, sessions or dialogs between applications are set up, managed, and eventually terminated. A session is coordinated and synchronized to prevent different applications' data from becoming mixed up during transfer.

Some of the protocols that operate at the session layer include:

- **Network File System (NFS)** – developed by Sun/IBM for use with TCP/IP and UNIX to allow transparent remote access to resources
- **Structured Query Language (SQL)** – provides a simple means of accessing

- system information on local or remote systems
- **Remote Procedure Call (RPC)** – procedures created on a client and performed on a server

Transport Layer

In this layer, end-to-end data transport services are provided to the upper OSI layers. The transport layer takes data from the upper layers, breaks it into smaller units called segments, and adds logical transport information in the header.

Before communication can take place, an end-to-end logical connection called a virtual circuit has to be established. The transport layer includes several protocols, the most common being Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), which are part of the TCP/IP (Internet Protocol) suite of protocols. The TCP/IP suite is the standard suite on which most of the Internet operations are based.

TCP is considered a reliable connection-oriented protocol. It uses reliable mechanisms to initiate and terminate connections. It also has flow control and congestion avoidance mechanisms to ensure that data gets to its destination safely. Many application layer protocols use TCP as the transport protocol. Some of them include Telnet, HTTPS (HyperText Transfer Protocol Secure), and FTP (although they sit at the application layer, they do use TCP).

In TCP, a logical end-to-end connection is achieved by each end-system agreeing that a connection is about to be initiated. This process is known as a three-way handshake. The handshake can be seen if the packets on the wire are read and can be identified by fields in the packet marked as SYN, SYN ACK, and ACK.

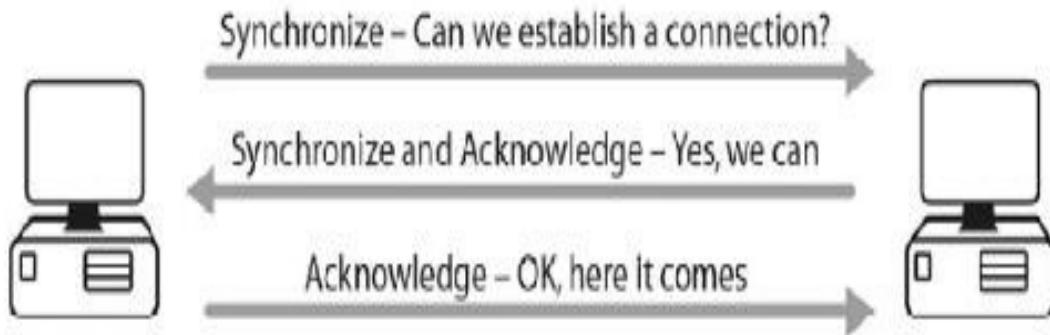


FIG 1.13 – The three-way handshake

You can see this process happening with a packet capture. We will also touch on this

process later on.

9 29.056074	192.168.1.1	192.168.1.2	TCP	60 41263 > telnet [SYN] Seq=0 Win=4128 Len=0
10 29.077245	192.168.1.2	192.168.1.1	TCP	60 telnet > 41263 [SYN, ACK] Seq=0 Ack=1 Win=
11 29.000067	192.168.1.1	192.168.1.2	TCP	60 41263 > telnet [ACK] Seq=1 Ack=1 Win=4120
12 29.099120	192.168.1.2	192.168.1.1	TELNET	66 Telnet Data ...
13 29.109332	192.168.1.2	192.168.1.1	TELNET	91 Telnet Data ...
14 29.120847	192.168.1.1	192.168.1.2	TELNET	63 Telnet Data ...

Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)

Transmission Control Protocol, Src Port: 41263 (41263), Dst Port: telnet (23), Seq: 0, Len: 0

Source port: 41263 (41263)
Destination port: telnet (23)
[Stream index: 0]
Sequence number: 0 (relative sequence number)
Header length: 24 bytes
> Flags: 0x002 (SYN)
Window size value: 4120
Calculated window size: 4120

FIG 1.14 – Packet capture of a three-way handshake

Data transfer using TCP as the transport protocol is considered to be reliable. This means that there is a guarantee that the data sent will reach the intended destination. This is accomplished by using three methods:

1. Flow control
2. Windowing
3. Acknowledgments

Flow Control

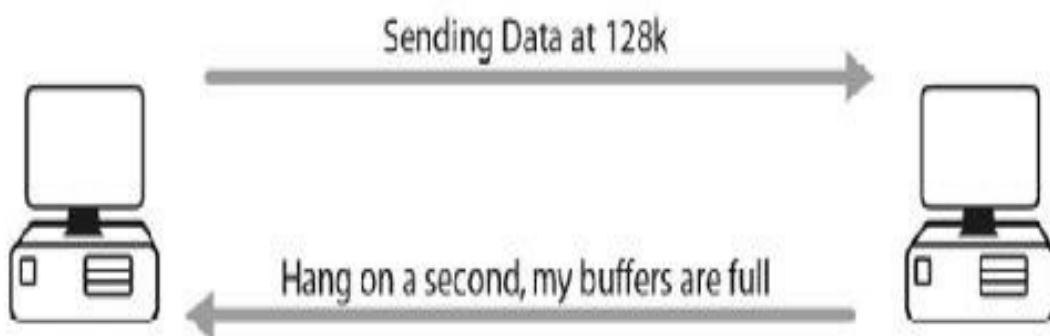


FIG 1.15 – Flow control

If the receiver is sent more information than it can process, it will ask the sender to stop for a short while. An example of when this can occur is when both sides are using different speeds (e.g., one side is using broadband while the other is using a dial-up modem). The packet sent to stop the sender is known as a source quench message.

Windowing

The TCP window is the amount of data that can be sent before an acknowledgment is required from the receiver. The sender and receiver agree on the window size and this can be scaled up and down as required.



FIG 1.16 – Windowing

Acknowledgments

Acknowledgments are messages indicating the successful receipt of TCP segments. If a sender does not receive acknowledgments for the segments sent after a certain period, then it knows there is something wrong.

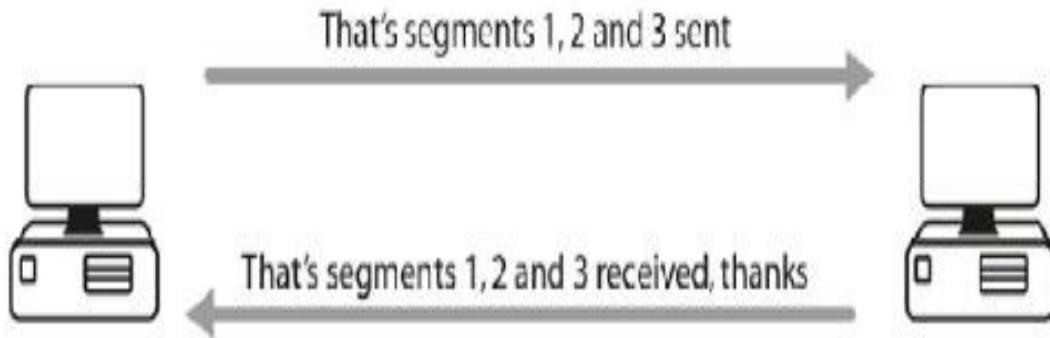


FIG 1.17 – Acknowledgments

UDP, on the other hand, is a connectionless protocol. In other words, it does not care about sequencing or acknowledgments, and it does not have all the fancy mechanisms that TCP uses to ensure that its segments reach their destination safely. This means that applications using UDP must be responsible for their own reliability.

Why is UDP used at all? Unlike TCP, UDP is lightweight. Since it does not have to initiate a connection using a three-way handshake, UDP can be used for applications where speed and bandwidth are a concern. In some cases, these issues are more important than the reliability that TCP provides. Protocols carried on UDP include SNMP (Simple Network Management Protocol) and TFTP (Trivial File Transfer Protocol).

Network Layer

The role of the network layer is to determine the best path or route for data to take from one network to another. Data from the session layer are assembled into packets at this layer, and this is where the end-to-end delivery of packets occurs.

Because networks need some way of identifying themselves, logical addressing also takes place at the network layer. The most popular form of network addressing today is IP addressing using IPv4 or IPv6.

Table 1-1: Router B best-path routing table

Destination Network	Next Hop	Number of Hops Away
Network 1	None	Directly connected
Network 2	None	Directly connected
Network 3	Router A	1
Network 3	Router C	1
Network 4	Router A	1
Network 4	Router C	2

The best path is decided at the network layer. Each router stores a table of which networks are directly connected and how to get to the networks that are not. You can see the routing table for Router B in Table 1-1 above.

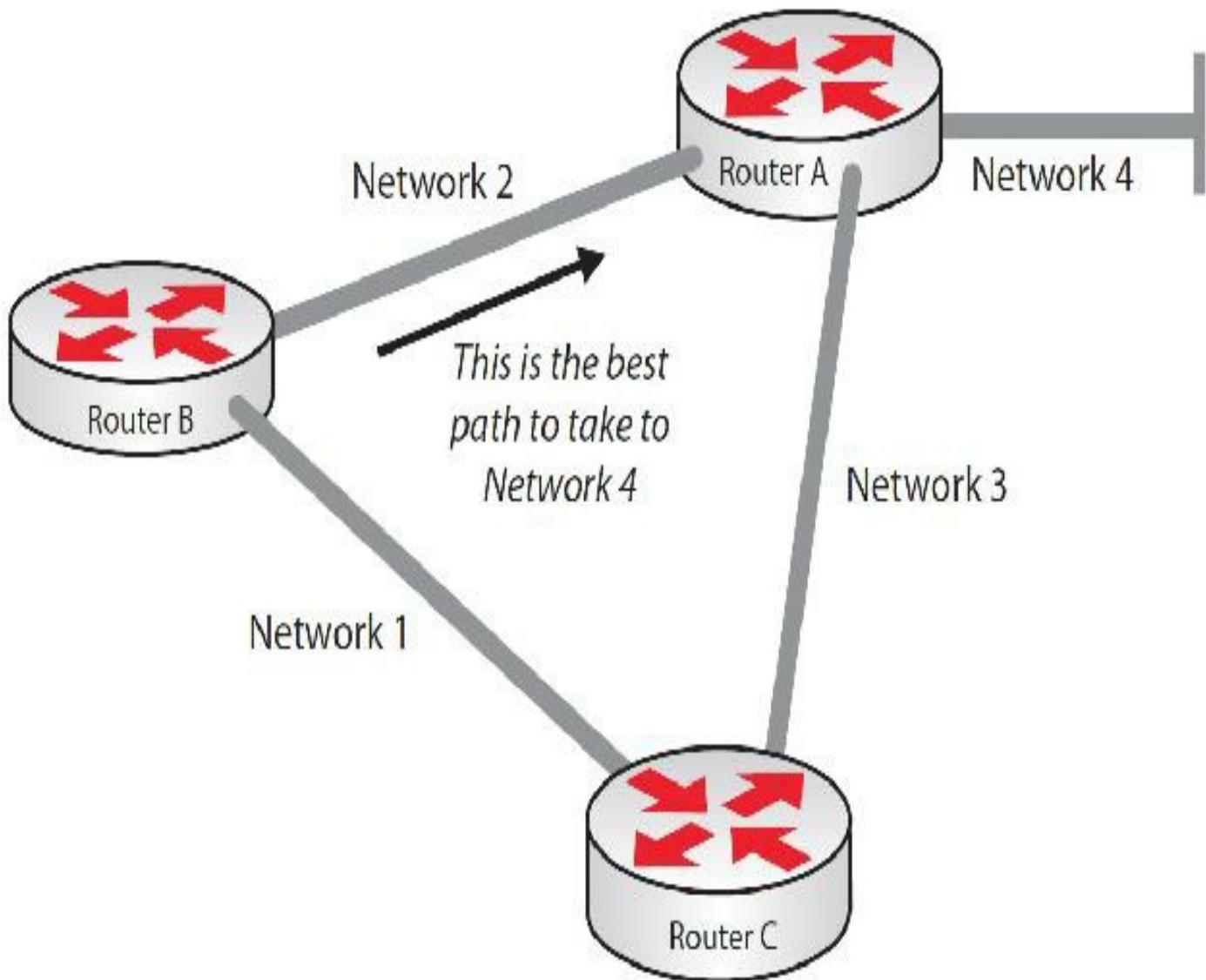


FIG 1.18 – Best path is decided at the network layer

Routers operate exclusively at the network layer of the OSI model. When a packet arrives at a router interface, the router looks at the destination network address and decides whether that network is directly connected. If it is not, the router looks at its routing table to see which interface it should leave by.

Network Layer Addressing

Logical addressing for TCP/IP uses 32 binary bits to make up a network address (if using IPv4 addressing). Binary is written out in decimal to make it easier to read and understand. An example of a logical address for TCP/IP is 192.168.2.3, where 192.168.2 identifies the network and 3 belongs to a host in that network. IP addressing is covered in more detail in Chapter 3 of this study guide.

Network 192.168.2.x

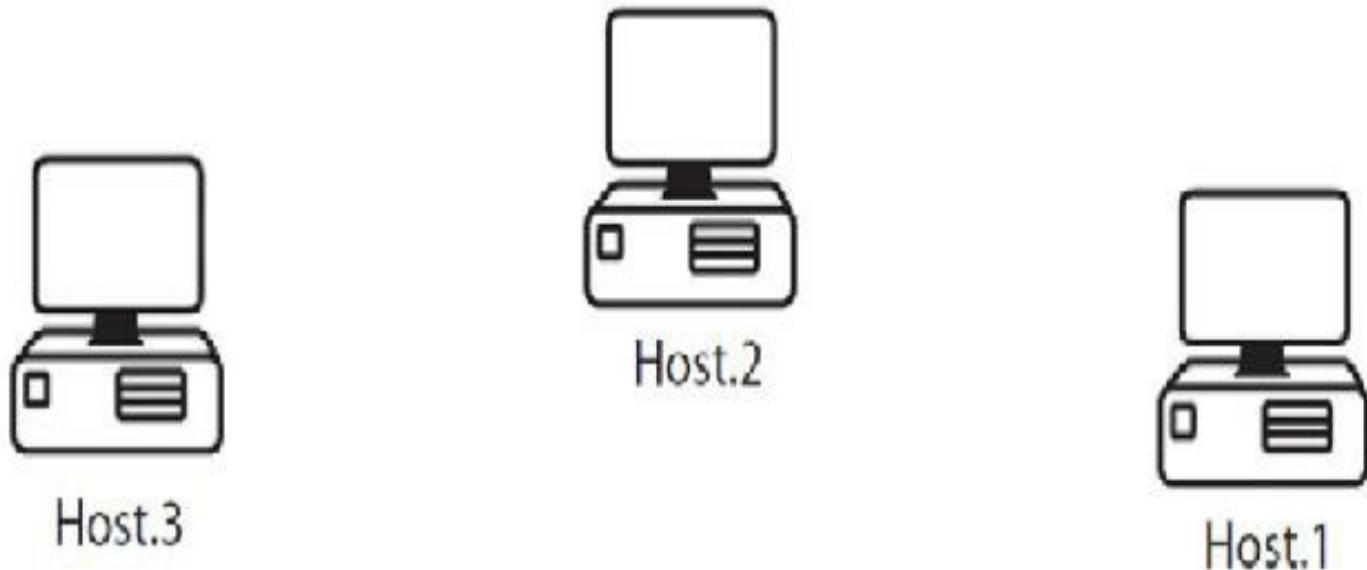


FIG 1.19 – Hosts in a network

Network layer protocols include IP, IPX, and AppleTalk, although the last two are now obsolete.

Data Link Layer

The data link layer is divided into two sublayers—LLC and MAC—as shown in Figure 1.20 below:

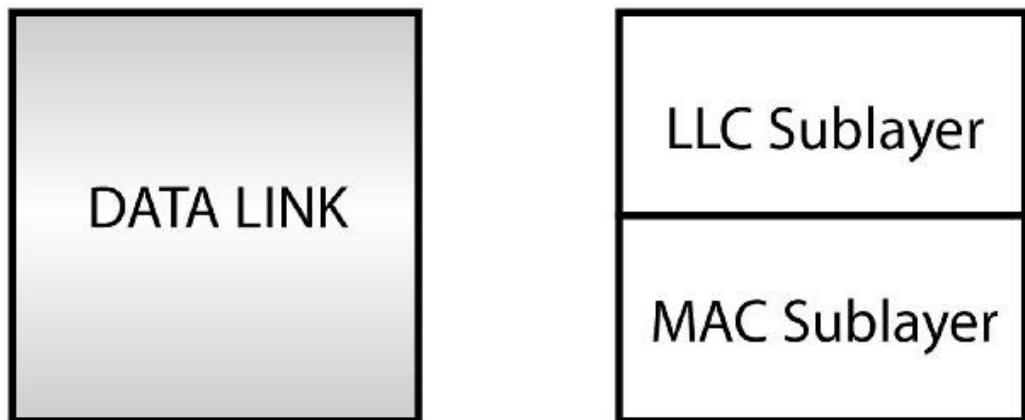


FIG 1.20 – The data link layer

The data link layer takes packets from the network layer and divides them into smaller units known as frames. Frames are then transported across a physical medium (i.e.,

wires). The data link layer has its own way of addressing known as hardware addressing. While the network layer determines where networks are located, the data link layer determines where hosts on a particular network are located.

Logical Link Control Sublayer (IEEE 802.2)

The LLC sublayer interfaces with the network layer and provides Service Access Points (SAPs); these allow the MAC sublayer to communicate with the upper layers of the OSI model.

Media Access Control Sublayer (IEEE 802.3)

The MAC layer directly interfaces with the physical layer. This is where the physical address of the interface or device is stored. A MAC address is a 48-bit address expressed as 12 hexadecimal digits. This address identifies both the manufacturer of the device and the specific host.

Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

```
C:\Users\owner>ipconfig /all
```

Windows IP Configuration

Host Name	: owner-PC
Primary Dns Suffix	
Node Type	: Hybrid
IP Routing Enabled	: No
WINS Proxy Enabled	: No
DNS Suffix Search List	: BigPond localdomain

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix	: BigPond
Description	: Realtek PCIe GBE Family Controller
Physical Address	: 60-A4-4C-41-33-77
DHCP Enabled	: Yes
Autoconfiguration Enabled	: Yes
IPv4 Address	: 10.0.0.32(Preferred)
Subnet Mask	: 255.255.255.0
Lease Obtained	: Monday, 10 August 2015 9:17:30 PM
Lease Expires	: Wednesday, 12 August 2015 9:17:29 AM

FIG 1.21 – MAC address on a PC: 60-A4-4C-41-33-77

The best example of MAC addressing is the address hard-coded onto the network interface card (NIC) in a PC or server. You can see the MAC address (shown as the physical address) of your network card by typing ipconfig /all after the command prompt on your windows PC. You can use /sbin/ifconfig on the terminal if you are using a MAC or a Linux machine.

In Figure 1.21 above, the first three bytes identify the vendor for the NIC (60-A4-4C).

This is known as the Organizationally Unique Identifier (OUI), which is assigned to manufacturers by the Institute of Electrical and Electronics Engineers (IEEE). The next three bytes (41-33-77) are assigned by the vendor and must be unique in order to prevent the same address from being used twice. Some shady manufacturers do not follow these rules, so always buy vendor-approved equipment to avoid problems.

The MAC address allows devices to have a unique layer 2 address and allows communication to take place at layer 2. Switches and bridges operate at the data link layer of the OSI model. A table of MAC addresses and which port they are connected to is maintained by both devices.

Data Link Protocols

There are many protocols operating at the data link layer. Protocols are an agreed format through which devices in a network communicate with one another. The reason why protocols operate at the data link layer is twofold. First, a connection has to take place before network layer communication can start. Second, data link layer communication is a lot faster than network layer communication because there is far less overhead involved in data link layer networking.

Data link protocols operate on both LANs and WANs and include Ethernet, PPP, Frame Relay, and many more. We will address those pertinent to the CCNA exam in the relevant chapters.

FOR THE EXAM: Because you are using packets at the network layer and frames at the data link layer, remember that you ROUTE PACKETS AND FORWARD FRAMES.

Physical Layer

The physical layer takes frames from the data link layer and converts them into bits. The physical layer has to use bits (binary digits) since data on a wire can be sent only as a pulse of electricity or light—that is, only as one of two values, either a 1 or a 0.

The physical layer deals with the physical characteristics of the medium, such as number of pins and their uses. Physical layer specifications include IEEE 802.3, FDDI, Ethernet, RJ-45, and many more.

Hubs operate at the physical layer of the OSI model. Hubs take the bits, strengthen the signal if it has been degraded, and send them out to every device connected to the ports.



FIG 1.22 – Hubs and repeaters strengthen the signal on the wire

Summary—The OSI Model

The OSI model can be summarized as shown in Table 1-2 below:

Table 1-2: The OSI model summarized

Layer	Encapsulation	Function	Services	Device
7 Application	Data	Establishes availability of resources	FTP, SMTP, Telnet, POP3	Hosts/Firewalls
6 Presentation	Data	Compression, encryption, and decryption	JPEG, GIF, MPEG, ASCII	Hosts/Firewalls
5 Session	Data	Establishes, maintains, and terminates sessions	NFS, SQL, RPC	Hosts/Firewalls
4 Transport	Segment	Establishes end-to-end connection; uses virtual circuits, buffering, windowing, and flow control	TCP, UDP	Hosts/Firewalls
3 Network	Packet	Determines best path for packets to take	IP	Router
2 Data Link (LLC, MAC)	Frame	Transports data across a physical connection; error detection	Frame Relay, PPP, HDLC	Switch/Bridge
1 Physical	Bits	Puts data onto the wire		Hub/Cables

FOR THE EXAM: A thorough knowledge of the OSI model is vital for the exam. Know each level and encapsulation formats and which device sits where.

The TCP/IP Model

There are different models of representation for internetworking. After the OSI model comes the TCP/IP model in terms of popularity. The TCP/IP model, which is loosely outlined in RFC 1122 as a four-layer model, does not map directly to the OSI model.

RFCs are Requests for Comments, which are documents proposing network protocols and services.

Table 1-3: The OSI and TCP/IP models

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport/Host-to-Host
Network	Internet
Data Link	Network Access/Network Interface
Physical	

Unfortunately, there are a variety of TCP/IP models that use different terms for the layers or have five layers instead of four as shown below. The lowest TCP layer can be referred to as the physical layer, the link layer, or the network interface.

Application
Transport
Network
Data Link
Physical

The Cisco Network Academy course book refers to the TCP/IP model as having four layers (see Mark A. Dye, Rick McDonald, and Antoon W. Ruij. Network Fundamentals: CCNA Exploration Companion Guide. 2007. ISBN 1-58713-208-7), while Douglas E. Comer's highly regarded textbook Internetworking with TCP/IP: Principles, Protocols and Architecture (2005. Pearson Prentice Hall. ISBN 0-13-187671-6) refers to the TCP/IP model as having five layers.

TCP/IP Application Layer

The application layer in the TCP/IP model covers the functionality of the session, presentation, and application layers in the OSI model. Various protocols can be used in this layer, including:

- SMTP, POP3 – used to provide e-mail services

- HTTP – World Wide Web browser content delivery protocol
- FTP – used in file transfer
- DNS – used in domain name translation
- SNMP – network management protocol
- DHCP – used to automatically assign IP addresses to network devices
- Telnet – used to remotely manage and control network devices

The TCP/IP application layer does not provide the actual services; it does, however, define the services the applications require. An example would be your web browser requesting HTTP services from the network. The application layer would provide the interface for this to take place.

Confusingly for many students, some routing protocols such as Border Gateway Protocol (BGP) and Routing Information Protocol (RIP) reside at the TCP/IP application layer; for example, BGP uses TCP to transport its messages, while RIP uses UDP. However, for other protocols such as Open Shortest Path First (OSPF) this isn't the case because they encapsulate messages within IP packets.

TCP/IP Transport/Host-to-Host Layer

The protocols that operate at and control the transport/host-to-host layer are specified in that layer. The TCP/IP transport layer controls the end-to-end logical connection between two devices. Both the TCP/IP transport and Internet layer demonstrate considerable differences compared with the corresponding OSI layers. The transport layer is based on two protocols:

- **TCP** – **This** provides connection-oriented communication. This means the path that data travels on in the network is reliable, as the endpoints establish a synchronized connection before sending the data. Every data packet is acknowledged by the receiving host and includes a Checksum field to check for error detection. FTP is an example of a protocol that uses TCP.
- **UDP** – **This** provides unreliable, connectionless communication between hosts. Unlike TCP, UDP does not check the segments that arrive at the destination to ensure that they are valid and in the proper order. This means that the integrity verifications and the error connection process will occur in the application layer. In fact, unlike TCP, UDP doesn't set up a connection between the sender and recipient. On the other hand, UDP has a smaller overhead than TCP because the UDP header is much smaller. TFTP is an example of a protocol that uses UDP.

The TCP and UDP protocol data units are segments. Each segment contains a number of fields that carry different information about the data as shown in Figure 1.23 below (we

will cover some of the individual fields later in this manual):

Source Port Number	Destination Port Number
Length	Checksum
Data	

UDP Segment

Source Port Number	Destination Port Number		
Sequence Number			
Acknowledgment Number			
Header Length	Reserved	Code Bits	Window Size
Checksum		Urgent	
Option			
Data			

TCP Segment

FIG 1.23 – UDP and TCP segment fields

The TCP header is larger than the UDP header because of all the extra fields needed to ensure a reliable connection.

Applications that are dependent on TCP and UDP use specific port numbers to operate. Port numbers can take values up to 65535. Most of the common applications are assigned well-known port numbers, which are numbers up to 1023. Port numbers 1024 through 49151 are registered port numbers and the range 49152 through 65535 defines dynamic port numbers (automatically assigned by network devices). Port numbers are used to distinguish between applications running on the same device. Examples of well-known port numbers include:

- HTTP – TCP port 80
- FTP – TCP port 20 (data) and 21 (control)
- TFTP – UDP port 69
- POP3 – TCP port 110
- SMTP – TCP port 25
- DNS – TCP and UDP port 53

- SNMP – UDP port 161/162
- Telnet – TCP port 23

You've already seen that when a TCP connection is established, it follows a process called a three-way handshake. This process uses SYN and ACK bits in the code bits in the TCP segment's Sequence and Acknowledgment Number fields. Figure 1.24 below illustrates an example of this process during a TCP operation:

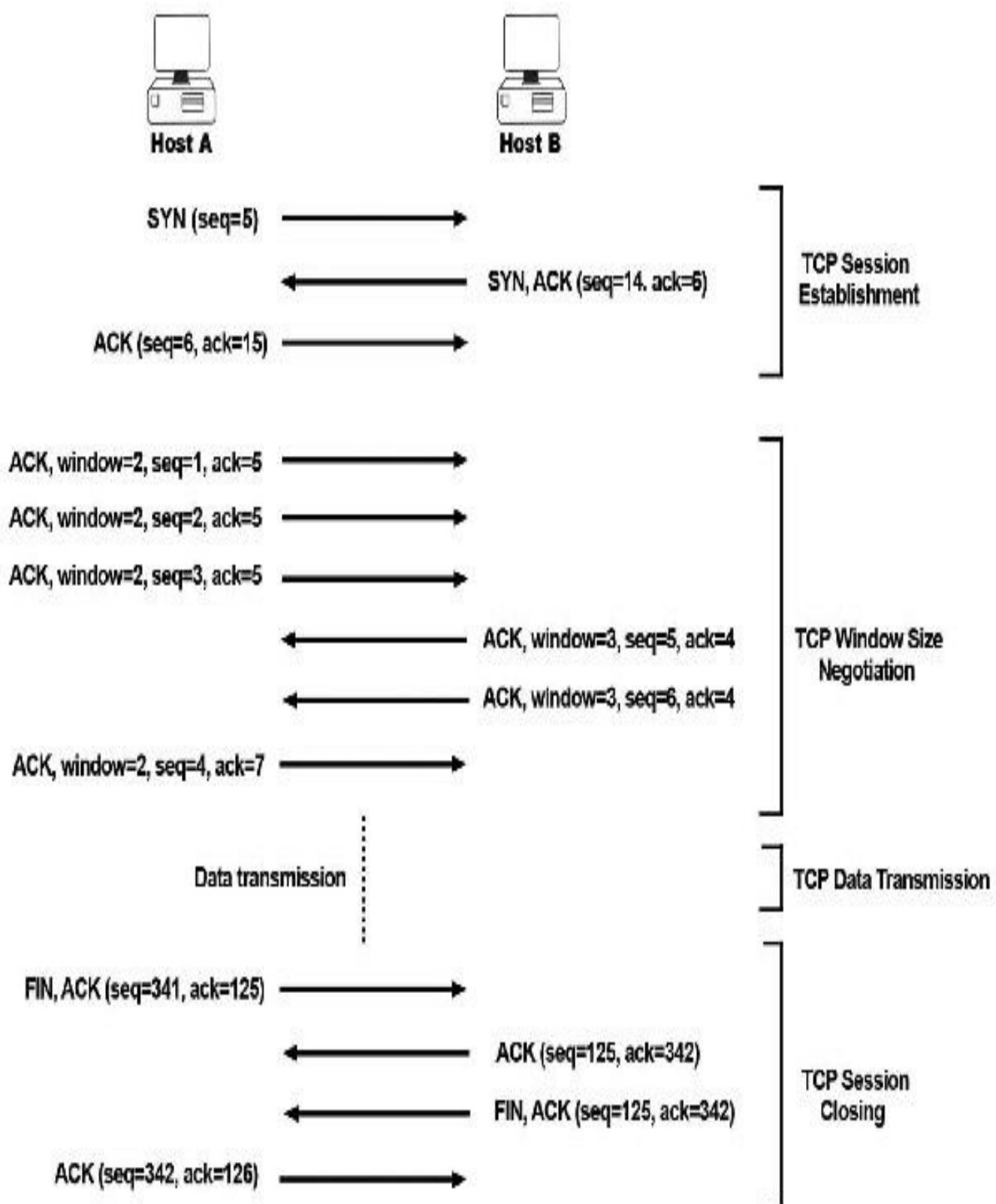


FIG 1.24 – TCP operation

In the example depicted above, Host A tries to establish a TCP connection with Host B. Host A sends a segment with the SYN bit set, letting the other device know that it wants to synchronize. The segment includes the initial sequence number 5 that Host A is using. Host B accepts, establishes a session, and sends back a segment with the SYN bit set. Host B also sets the ACK bit to acknowledge that it has received the initial segment sent by Host A.

The acknowledgment number represents the next segment it expects to receive, 6 in this example (this is also called an expectational acknowledgment). The new segment includes the initial sequence number from Host B, 14 in this example. Host A replies with an ACK segment that contains sequence number 6, because this is what Host B is expecting, and acknowledgment number 15, informing Host B that it can send the next segment. This concludes the TCP session establishment phase.

The window size informs the remote host about the number of bytes a device will accept before it must send an acknowledgment. The window sizes may not match on the two endpoints. Host A has a window size of 2 and Host B has a window size of 3. When Host A sends data, it can send 3 bytes before waiting for an acknowledgment; however, Host B can send only 2 bytes before receiving an ACK.

NOTE: The window size specifies the number of bytes (octets) a device will accept, not the number of segments.

After all the data is sent between the two hosts, the session can be closed. Host A sends a segment with the FIN bit set, letting Host B know that it wants to end the TCP session. The segment includes the sequence number Host B is using at that specific moment (341 in this example). Host B acknowledges the request and sets the ACK bit to acknowledgment number 342 to confirm that it received number 341.

The segment also includes the current sequence number of Host B (125 in this example). Host B sends a new segment with the FIN bit set, announcing that the application it is running also requests to close the session. The last step before the session is closed is Host A sending an ACK segment number 126 to confirm that it received number 125 from Host B.

TCP/IP Internet/Network Layer

The Internet layer in the TCP/IP model corresponds to OSI layer 3 (network layer). This layer is responsible for routing data, including addressing and packet format, and uses the following protocols:

- IP (Internet Protocol) – This connectionless protocol offers best-effort delivery of packets in the network. It relies on transport layer protocols like TCP to

ensure a reliable connection. IP addresses are assigned to each network device or interface in the network. IP comes in two flavors: IPv4 and IPv6. These aspects will be covered in detail later in this book.

- ICMP (Internet Control Message Protocol) – This protocol sends messages and error reports through the network. The most-used application that relies on ICMP is ping, which sends an ICMP echo message to the destination and expects an ICMP echo reply to ensure that the destination can be reached and to give information about the delay between the two endpoints. (We will look at ICMP in more detail shortly.)

As mentioned, IP packets can be either in IPv4 format or in IPv6 format. An IPv4 packet, as defined in RFC 791, contains the following fields:

Version	Header Length	Type of Service	Total Length			
Identification		Flags		Fragment Offset		
Time To Live	Protocol		Header Checksum			
Source IP Address						
Destination IP Address						
IP Options			Padding			
Data						

FIG 1.25 – IPv4 packet fields

If you want to read up on the individual fields in detail, then please refer to the Requests for Comments (RFCs) on www.ietf.org/rfc.html. We will mention some of these fields below and later in this guide as they become relevant, but it's unlikely that you would be tested on the individual fields at CCNA level. Having said this, I strongly recommend that you take a Wireshark Certified Network Analyst (WCNA) course at some point if you want to be a successful network engineer. The course will give you a very thorough understanding of TCP and the mechanics of internetworking, as well as make you much more confident in your day-to-day role.

The Version field in the packet identifies the IP version; for IPv4 the value would be 0100, which is 4 in binary, and for IPv6 it would be 0110, which is 6 in binary. We will cover binary math in the IP addressing chapter.

The Time to Live (TTL) field is populated with a number when the packet is generated. Pings, for example, start at 255 and decrement down. As the packet traverses each router, the number will decrement by one. If the number reaches zero the packet will be discarded to prevent packets from endlessly circulating the network. Note that this facility is unavailable for Ethernet frames, but other mechanisms have been created to address this problem. We will revisit TTL later.

The Protocol field is populated with the protocol number. There are a number of protocols available, such as EIGRP (88), OSPF (89), UDP (17), and TCP (6), and we will discuss some of them as we progress through this guide.

An IPv6 packet contains the following fields:

Version	Traffic Class	Flow Label
Payload Length	Next Header	Hop Limit
Source Address		
Destination Address		
Data		

FIG 1.26 – IPv6 packet fields

The size of an IPv6 address is four times the size of an IPv4 address (i.e., an IPv4 address is 32 bits, while an IPv6 address is 128 bits). More details will be provided on this subject in the IPv6 addressing chapter.

TCP/IP Network Access Layer

The network access layer maps to the OSI data link layer and physical layer, and it has the same functionality as those layers.

A common protocol used at the network access layer is ARP (Address Resolution Protocol), which requests the MAC addresses of a host with a known IP address. This works by sending a broadcast message to all the hosts on a subnet and asking for the MAC address of the host that has the IP address. The host with that IP address responds with its MAC address and the sender caches this in its memory for a period of time. Once the MAC address is known, it is used as a destination address in the frames sent in that specific direction.

TCP/IP Services

If you have had any interest in computer networking, you may have already heard of some of the services to follow. Most novice users don't know (or care) that they all come under the umbrella of TCP/IP. We'll be covering many of these in greater detail throughout the book but for now, here is a brief overview.

File Transfer Protocol

FTP is an application layer protocol that is used to reliably transfer files from a source to a destination. To ensure reliability, FTP uses TCP as its transport protocol.

On Cisco devices the `debug ip ftp` command can be used to debug FTP traffic that is destined to the device. Debugs are Cisco troubleshooting tools that display information you can use to troubleshoot various protocols and services. We will discuss debugs later in this guide because they will be an important part of your troubleshooting tool bag, but they must be used with caution.

FTP uses TCP ports 20 and 21. A control connection is made from the client to the FTP server on port 21. A second data connection is then made from either the FTP server on port 20 (in active FTP) or a random port on the client to port 20 on the FTP server (in passive FTP).

Trivial File Transfer Protocol

TFTP provides an alternative file transfer method that is faster but less reliable than FTP. TFTP uses UDP as its transport protocol and operates on port 69. When using TFTP, you need to specify the exact directory the file is located in since you cannot list the directories.

TFTP can be used to backup and copy router configurations. You will need a TFTP client (which can be the router) and a TFTP service (which can be either another router that has the files stored locally or a server in the network) to accomplish this task. If you are using a server, you need to have TFTP server software installed on the client.

We will demonstrate TFTP in the ICND2 section.

Simple Mail Transfer Protocol

SMTP defines how e-mails are sent to the e-mail server from the client and it uses TCP as its transport protocol. E-mails can be retrieved from an SMTP server in different ways depending on the e-mail client. One of the protocols for e-mail retrieval is POP3, while another popular e-mail retrieval protocol is IMAP.

HyperText Transfer Protocol

HTTP is found in the application layer and is used to communicate between web servers and clients on TCP port 443. For secure communication, HTTP can be encrypted using Secure Sockets Layer (SSL) or Transport Layer Security (TLS). This secure HTTP (HTTPS) uses TCP protocol 443.

A Cisco router or switch can be connected to via a web browser. You can debug HTTP traffic destined for a Cisco device with the debug ip http command.

Telnet

Telnet is used for remote connections to network devices and it operates on TCP port 23. Traffic sent using Telnet is in clear text so it is insecure. A more secure method to remotely access devices is Secure Shell (SSH).

Troubleshooting Tip: Telnet is a good utility to validate that the seven layers of the OSI model are working correctly.

To set up Telnet on a Cisco router, a password should be set on virtual terminal (VTY) lines. To disconnect from a Telnet session, simply type exit or disconnect. To break out of a Telnet session, you can press Ctrl+Shift+6 together, release, and then press x to quit.

On Cisco devices, you can debug Telnet using the debug telnet command. We will use the telnet command several times throughout this guide; however, here is some output

from the debug telnet command for a router receiving an incoming Telnet request:

```
R2#debug telnet
```

Incoming Telnet debugging is on

```
R2#
```

```
*Mar 1 00:58:58.035: Telnet98: 1 1 251 1
```

```
*Mar 1 00:58:58.039: TCP98: Telnet sent WILL ECHO (1)
```

```
*Mar 1 00:58:58.039: Telnet98: 2 2 251 3
```

```
*Mar 1 00:58:58.039: TCP98: Telnet sent WILL SUPPRESS-GA (3)
```

Secure Shell

SSH is a cryptographic network protocol used for secure remote command-line login. SSH is often used for remote command executions, such as configuring routers and switches. Your first introduction to SSH may well be using a Telnet/SSH client program such as PuTTY, which enables you to remotely connect to network devices.

We will be looking at remotely connecting to network devices using SSH and Telnet (via PuTTY) later in this book.

SSH File Transfer Protocol

SFTP was devised by the Internet Engineering Task Force (IETF). SFTP provides the same service as FTP (i.e., file access, transfer, and management) but does so securely. The underlying security features and the fact that it can work with an SSH connection make it preferable in secure environments. SFTP is packaged with SSH but is usually integrated into graphical FTP tools such as FileZilla.

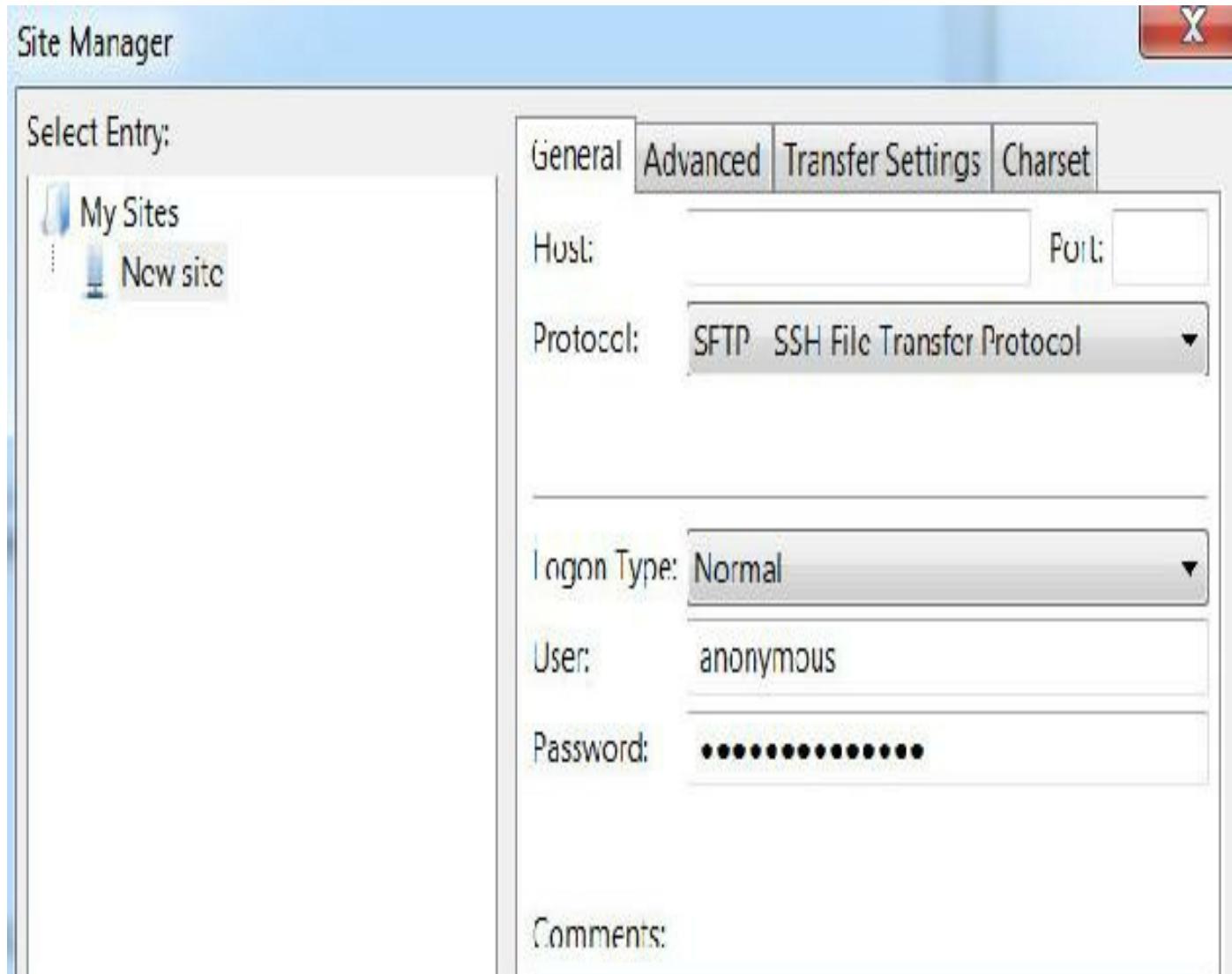


FIG 1.27 – SFTP connection manager

SFTP presumes that it is being run over a secure channel (such as SSH), that the client has been authenticated by the server, and that the identity of the client user is available to the protocol. It uses port 22 (as does SSH).

Secure Sockets Layer and Transport Layer Security

Both SSL and its replacement, TLS, are application layer cryptographic protocols created to provide secure communication over a computer network. SSL was created by Netscape to provide HTTPS access for its Navigator browser.

TLS is an IETF standard protocol based on SSL and currently version 1.3 is in draft mode. TLS ensures privacy between application users (such as e-mails) by preventing tampering or eavesdropping.

The server and client authenticate each other using the TLS Handshake Protocol, which then allows them to negotiate an encryption algorithm and cryptographic keys before the

exchange of data. Typically, the client remains unauthenticated and only the server is authenticated and its identity ensured.

Internet Control Message Protocol

As mentioned earlier, the ping utility uses ICMP to test network connectivity by sending ICMP messages. In some cases, it has actually been used by hackers to attack networks by sending lots of ICMP traffic and exhausting network resources. This is called a denial-of-service (DoS) attack.

ICMP is defined in RFC 792 and it is used to detect and report problems in an IP network. It is also used for diagnostic and control purposes. When there is an error in an IP message in the network, the IP packet is dropped and an ICMP error message is sent back to the sender.

The most common use of ICMP is the ping command to test connectivity between two devices. When a ping command is issued, an ICMP echo request is sent to the destination. If the packet makes it successfully to the destination, the destination replies with an ICMP echo reply. Other information can be derived from a ping, such as the round-trip time (the amount of time it takes for information to get to the device and back). An example of a ping command and output on a Windows machine is shown below:

```
C:\] ping cisco.com
```

Pinging cisco.com [72.163.4.161] with 32 bytes of data:

```
Reply from 72.163.4.161: bytes=32 time=147ms TTL=240
```

```
Reply from 72.163.4.161: bytes=32 time=153ms TTL=240
```

```
Reply from 72.163.4.161: bytes=32 time=148ms TTL=240
```

```
Reply from 72.163.4.161: bytes=32 time=151ms TTL=240
```

Ping statistics for 72.163.4.161:

 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

 Approximate round-trip times in milli-seconds:

 Minimum = 147ms, Maximum = 153ms, Average = 149ms

Let's examine the ping output in more detail. You can see that the reply is 32 bytes long, it takes about 150 ms, and the Time to Live is 240. TTL is the value that tells you how many hops (or devices) are between the source and destination devices. When a packet

is sent, it has a TTL of 255 and this number is decreased by 1 as the packet goes from one device to the other. If the TTL gets to 0, then the packet is dropped and an ICMP error message is sent. In this case, the TTL is 240, which tells you that you are 15 hops away from cisco.com.



The TTL value varies according to the destination OS:

Windows = 128
Linux, iOS = 64
Cisco = 255
Solaris = 255

Cisco routers have an extended ping feature that allows you to specify some ICMP and IP parameters in the ping. You can access this feature by typing ping and pressing Enter as shown below:

Router#ping i **Press Enter here**

Protocol [ip]: i **Press Enter here**

Target IP address: 172.16.1.1

Repeat count [5]:

Datagram size [100]: 1200

Timeout in seconds [2]:

Extended commands [n]: yes

Source address:

Type of service [0]:

Set DF bit in IP header? [no]: yes

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Type escape sequence to abort.

Sending 5, 1000-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

UUUUU

Success rate is 0% percent, round-trip min/avg/max = 4/6/12 ms

Cisco uses several notations to represent the response the ping packet receives:

- ! – Successful reply
- . – Request timed out
- U – Destination unreachable

- **N** – Network unreachable
- **P** – Protocol unreachable
- **Q** – Source quench message
- **M** – Could not fragment
- **?** – Unknown packet type

You can interrupt a ping session by pressing the Ctrl+Shift+6 keys together.

ICMP packet types are defined in RFC 1700. Learning all of the code numbers and names is outside the scope of the CCNA syllabus (check before you take your exam though).

You can debug ICMP traffic sourced from or destined to a Cisco router with the debug ip icmp command.

Traceroute

Traceroute is another widely utilized ICMP utility that is used to determine the path that a packet will take to reach its destination. Traceroute works by sending UDP packets (or ICMP packets) with a TTL of 1, and then increasing the TTL for the next packet until the entire path from source to destination is determined. At each hop, the TTL is decremented to 0, and an ICMP error (time exceeded) message is sent back to the sender. The sender of these time-exceeded packets is displayed to the user (the hops taken to reach the destination).

Cisco routers (and all other UNIX-based devices) use the traceroute command, whereas Windows PCs use tracert. Please remember this difference for the exam.

C:\>tracert cisco.com

Tracing route to cisco.com [72.163.4.161]

over a maximum of 30 hops:

```

1  [1 ms  [1 ms  [1 ms 192.168.10.1
2  59 ms   21 ms   9 ms cm-80.111.156.001.ntlworld.ie [80.111.156.1]
3  11 ms   8 ms   10 ms cm-80.111.156.001.ntlworld.ie [80.111.156.1]
4  8 ms    8 ms   11 ms 188-141-127-1.dynamic.upc.ie [188.141.127.1]
5  87 ms   86 ms   86 ms 84.116.238.54
6  89 ms   85 ms   89 ms 84.116.137.74
7  148 ms  132 ms 136 ms 84.116.137.34

```

```
8 90 ms 87 ms 85 ms 84.116.135.98
9 86 ms 91 ms 85 ms xe-0-0-0 [204.148.20.177]
10 149 ms 152 ms 148 ms 0.ae2.XT3.DFW9.ALTER.NET [140.222.225.55]
11 153 ms 149 ms 152 ms TenGigE0-4-0-0.GW15.DFW9.ALTER.NET
[152.63.98.10]
12 149 ms 147 ms 149 ms cisco-gw.customer.alter.net [157.130.134.190]
13 * * * Request timed out.
14 149 ms 148 ms 150 ms rcdn9-cd2-dmzdcc-gw2-por1.cisco.com [72.163.0.182]
15 149 ms 149 ms 148 ms rcdn9-16b-dcz05n-gw2-por2.cisco.com [72.163.2.110]
16 151 ms 148 ms 149 ms www1.cisco.com [72.163.4.161]
```

Trace complete.

The fields in the traceroute output are the same as those for the ping responses earlier.

Address Resolution Protocol

Different addressing formats are used to identify network hosts at various layers of the OSI model, including URLs, NICs, and device interfaces. At layer 3, the IP address is used to identify hosts. To communicate with hosts in the same network, the IP address needs to be mapped to the layer 2 address of the hosts. On Ethernet networks, the layer 2 addresses are known as MAC addresses. The protocol used to determine MAC addresses from IP addresses is the Address Resolution Protocol (ARP).

When a host needs to communicate with another host, it sends an ARP request for the MAC address of the host. This request is a broadcast and all the hosts in the network segment receive it. The host with the relevant IP address responds with its MAC address and layer 2 communication can then begin.

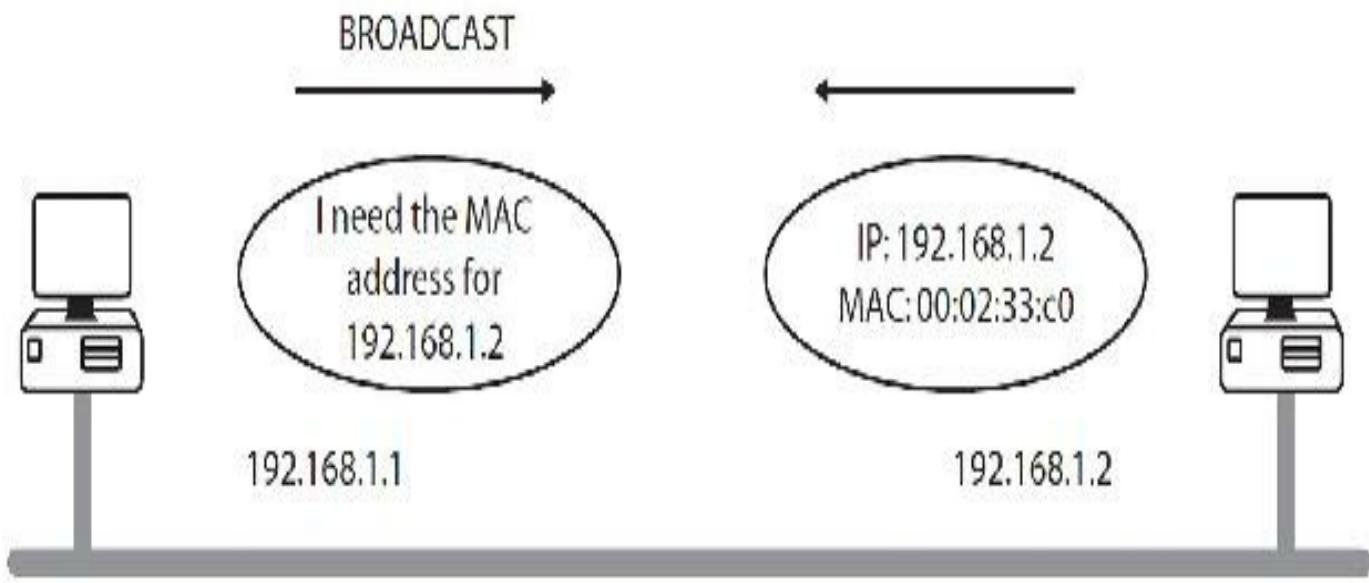


FIG 1.28 – A host broadcasts for another host's MAC address

You can debug ARP with the debug arp command.

Mini-lab – Checking the ARP Cache

If you have no experience configuring Cisco routers, then feel free to come back to this lab later because we have to add interface addressing. For a simple check of the ARP process, you can connect two routers together with a switch or directly with a crossover cable as per Figure 1.29 below. Add the IP addresses (of course, your MAC addresses will differ from mine so issue a show interface X command to see what your MAC is as shown below).

Here is the configuration I added to Router 1. On Router 2, change the IP address to 192.168.1.2 and configure the hostname R2.

```
Router#conf t
```

```
Router(config)#hostname R1
```

```
R1(config)#int f0/0
```

```
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
R1(config-if)#no shut
```

On Router 1, I can issue a show arp command to check the ARP cache. (Note that the MAC address starts with c201 for R1 and c202 for R2.) I actually used GNS3 for this lab because it's much easier to access and it works really well with Wireshark, which is free. A device will store its own ARP entry for a connected interface but in the Age column there will be a “–,” which indicates that it will never time out.

```
R1#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
----------	---------	-----------	---------------	------	-----------

Internet	192.168.1.1	-	c201.07f6.0000	ARPA	Fa0/0
----------	-------------	---	-----------------------	------	-------

Router 1 Fast Ethernet 0/0 interface has MAC address c201.07f6.0000, which can be seen above and below. I have added 192.168.1.1 as an IP address.

R1#show int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is **c201.07f6.0000** (bia c201.07f6.0000)

Internet address is 192.168.1.1/24

Router 2 has the MAC address c202.07f6.0000 and the IP address 192.168.1.2 as shown below:

R2#show int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is **c202.07f6.0000** (bia c202.07f6.0000)

Internet address is 192.168.1.2/24

If R1 wants to ping R2, it must establish the layer 2 address (MAC) to encapsulate the packet correctly. When I issue the ping 192.168.1.2 command, R1 will ARP for the device that is configured with this IP address. The ARP packet will be sent BEFORE the ping, which is why you will often see the first ping fail. Bear in mind that several devices could be on this segment of the network.

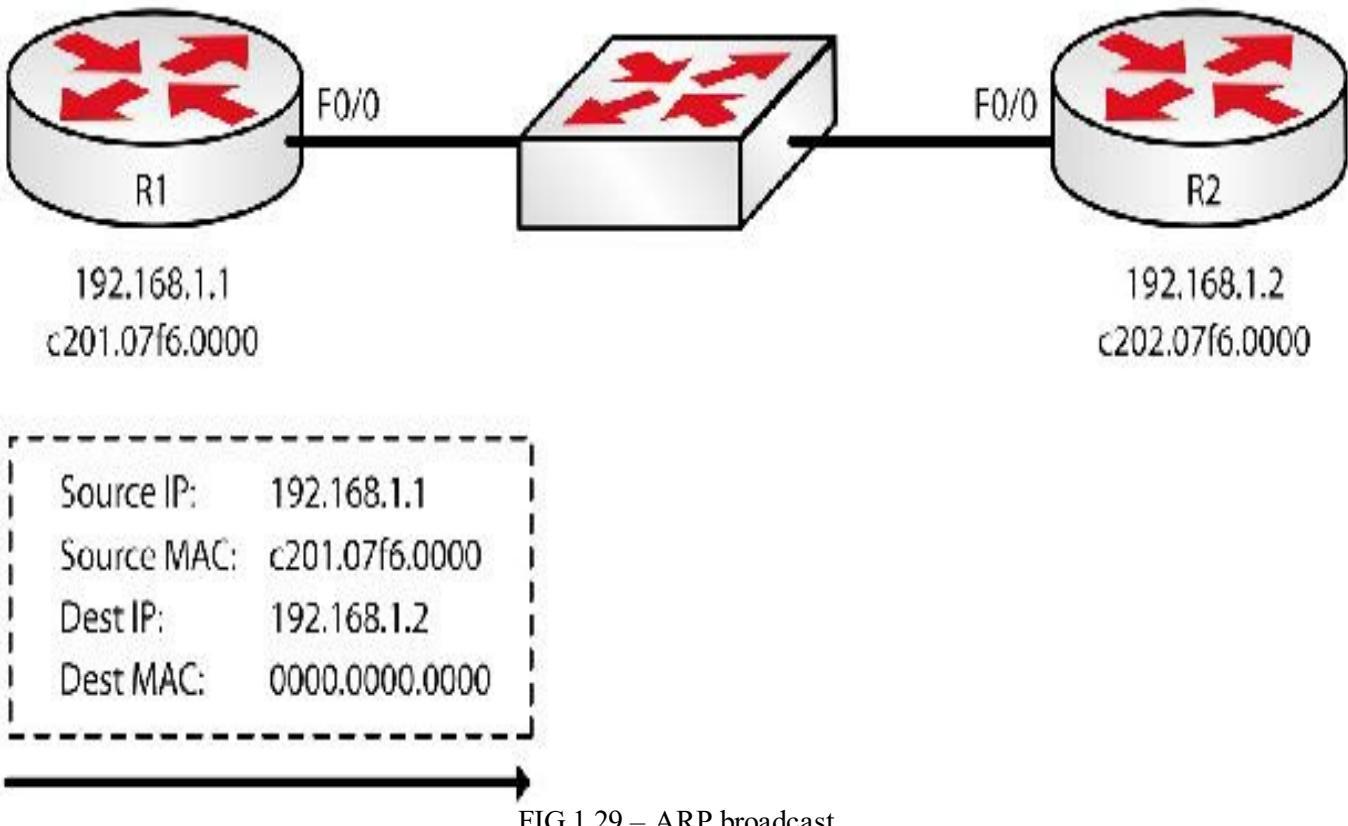


FIG 1.29 – ARP broadcast

Figure 1.30 below shows a packet capture of the ARP broadcast using Wireshark. You will see that the R1 MAC address starting with source c201 is set as the source MAC. The Destination field is unknown so it is set to be a broadcast, which in hexadecimal is all Fs.

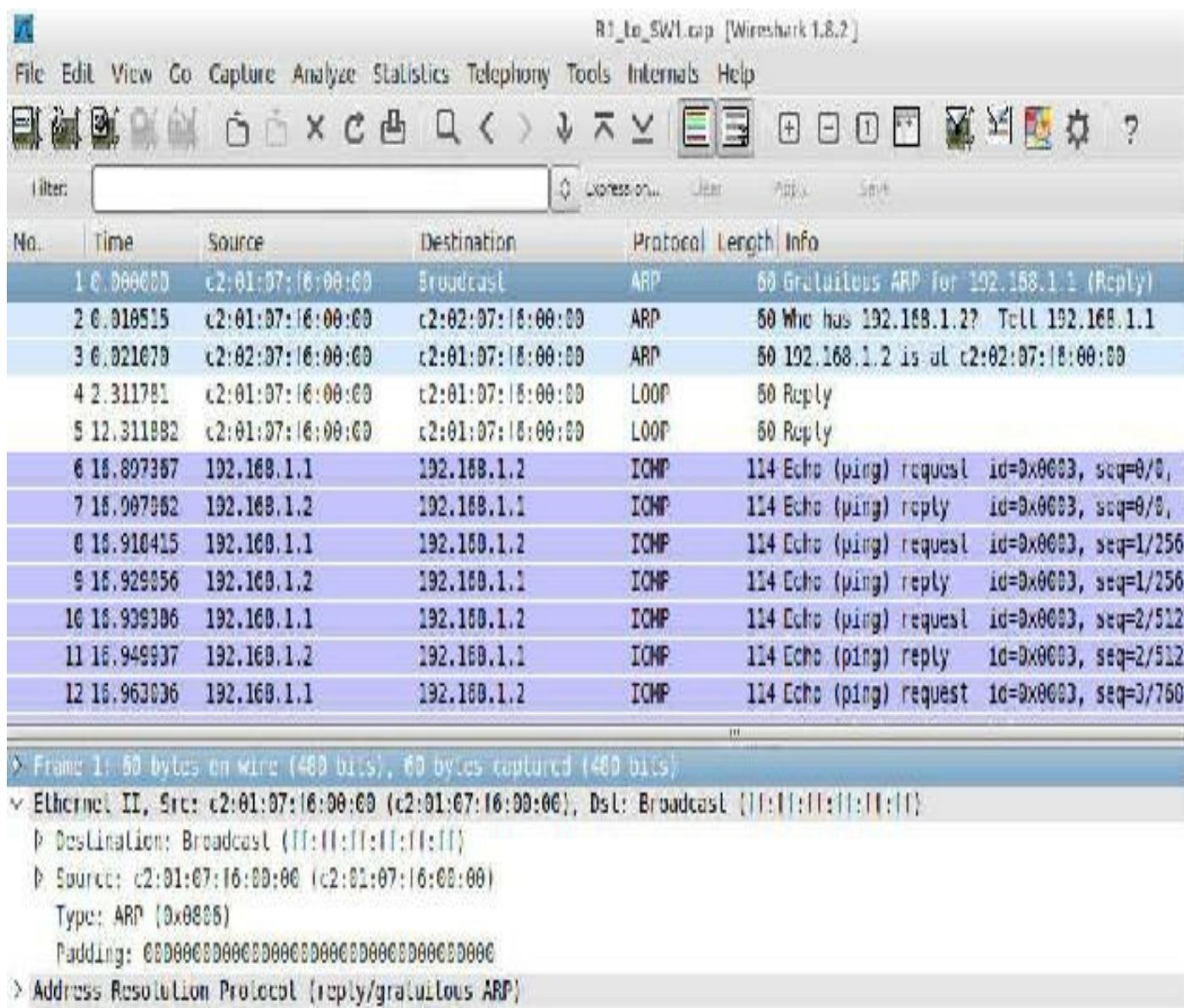


FIG 1.30 – ARP packet capture

If you examine the ARP packet you will see the following:

▽ Address Resolution Protocol (reply/gratuitous ARP)

Hardware type: Ethernet (1)

Protocol type: IP (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

[Is gratuitous: True]

Sender MAC address: c2:01:07:f6:00:00 (c2:01:07:f6:00:00)

Sender IP address: 192.168.1.1 (192.168.1.1)

Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)

Target IP address: 192.168.1.1 (192.168.1.1)

FIG 1.31 – ARP packet with destination field as a broadcast

The target MAC is a broadcast but the sending host knows the target IP address (because I typed it in at the command prompt), so the reply is unicast.

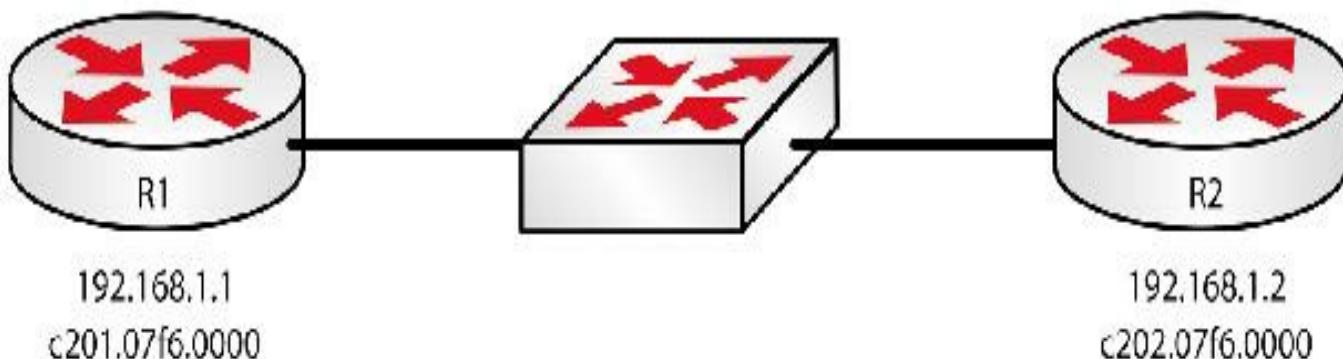


FIG 1.32 – ARP response

You can also see host 192.168.1.2 reply that this IP address is using MAC c202.07f6.0000:

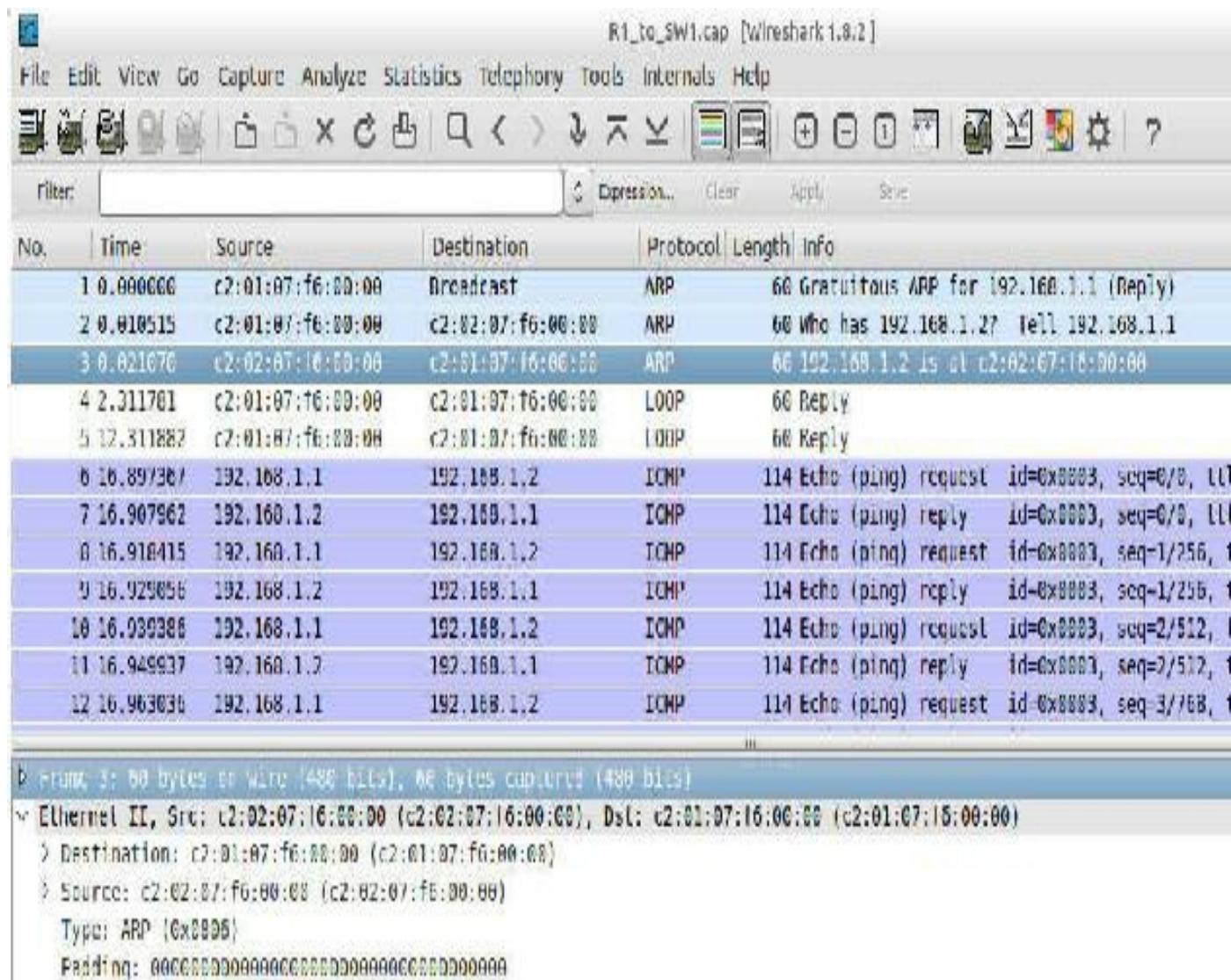


FIG 1.33 – ARP response packet capture

And in particular the ARP fields:

▽ Address Resolution Protocol (reply)

Hardware type: Ethernet (1)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: c2:02:07:f6:00:00 (c2:02:07:f6:00:00)
Sender IP address: 192.168.1.2 (192.168.1.2)
Target MAC address: c2:01:07:f6:00:00 (c2:01:07:f6:00:00)
Target IP address: 192.168.1.1 (192.168.1.1)

FIG 1.34 – Sender MAC address field is populated

Finally, you can see that the ARP table on R1 is now populated. If you need to send packets to R2 again, there will be no need to ARP for the MAC Address. The ARP table will eventually clear if no traffic is sent to R2.

R1#show arp

Protocol Address	Age(min)	Hardware Addr	Type	Interface
Internet 192.168.1.1	-	c201.07f6.0000	ARPA	FastEthernet0/0
Internet 192.168.1.2	15	c202.07f6.0000	ARPA	FastEthernet0/0

It's worth noting the Age field, which indicates how long the entry has been known. The MAC address for 192.168.1.2 was learned 15 minutes ago according to the output above. If the entry was learned less than one minute ago, the entry would be 0. The dash after 192.168.1.1 indicates a directly attached entry so it won't be flushed. These seemingly insignificant details are very important to note for the exam.

If for any reason you wanted to flush the ARP table on the router, you would issue the clear arp command. If you want to examine the ARP cache on a Windows PC, the command is arp -a. Bear in mind that network hosts will usually store the IP address and MAC address for the default gateway, which it will use to reach any device on another segment or network.

[END OF MINI-LAB]

Proxy ARP

Proxy ARP is a mechanism for a router to send a reply (on behalf of another device but containing its own MAC address). Since routers do not forward broadcasts, an ARP request from one segment to another segment cannot reach the intended recipient. To facilitate communication, the router responds with its own MAC address and then forwards the packet to the correct destination when it receives it. This mechanism is defined in RFC 1027.

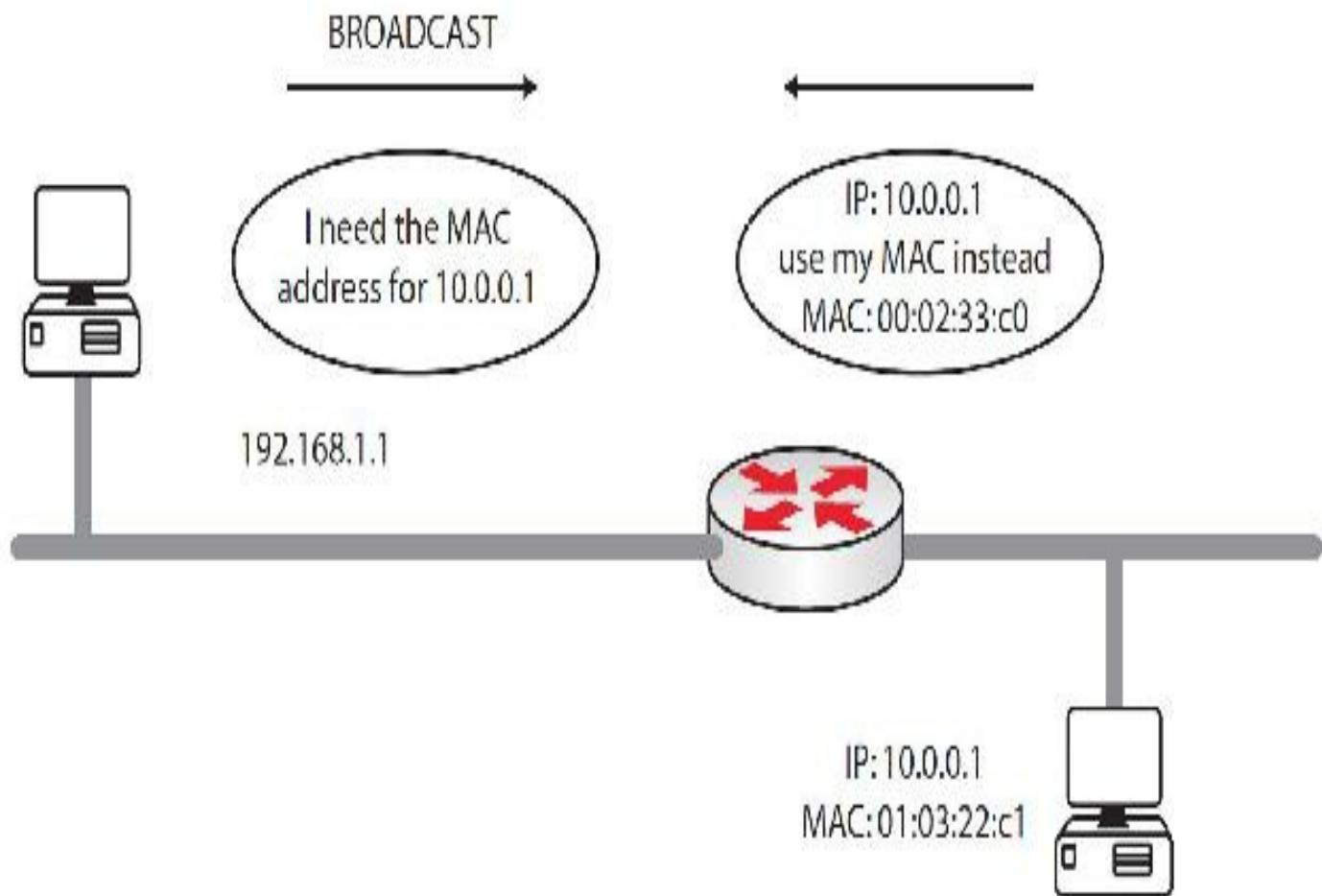


FIG 1.35 – Proxy ARP in action

You may hear Proxy ARP referred to as promiscuous ARP in some circles.

Proxy ARP is turned on in Cisco routers by default. It can be turned on/off using the [no] ip proxy arp command. The function of Proxy ARP is summarized above in Figure 1.35. The router responds to the ARP request with the MAC address nearest to the requesting host. This allows the sending host to encapsulate the layer 2 Destination field in the packet, and the router will then correctly address the packet before sending it to the destination host. Note that the sending host's ARP cache shows the router MAC address mapping to the destination host's IP address 10.0.0.1.

Proxy ARP is a hot exam topic, so let's go into some more detail.

Mini-lab – Discovering Proxy ARP

Figure 1.36 below shows a network with two hosts connecting to R2. Because I've used GNS3 for this example, it was quicker in fact to use two other routers as hosts (PCs). You can see that each interface has its own IP address and MAC address.

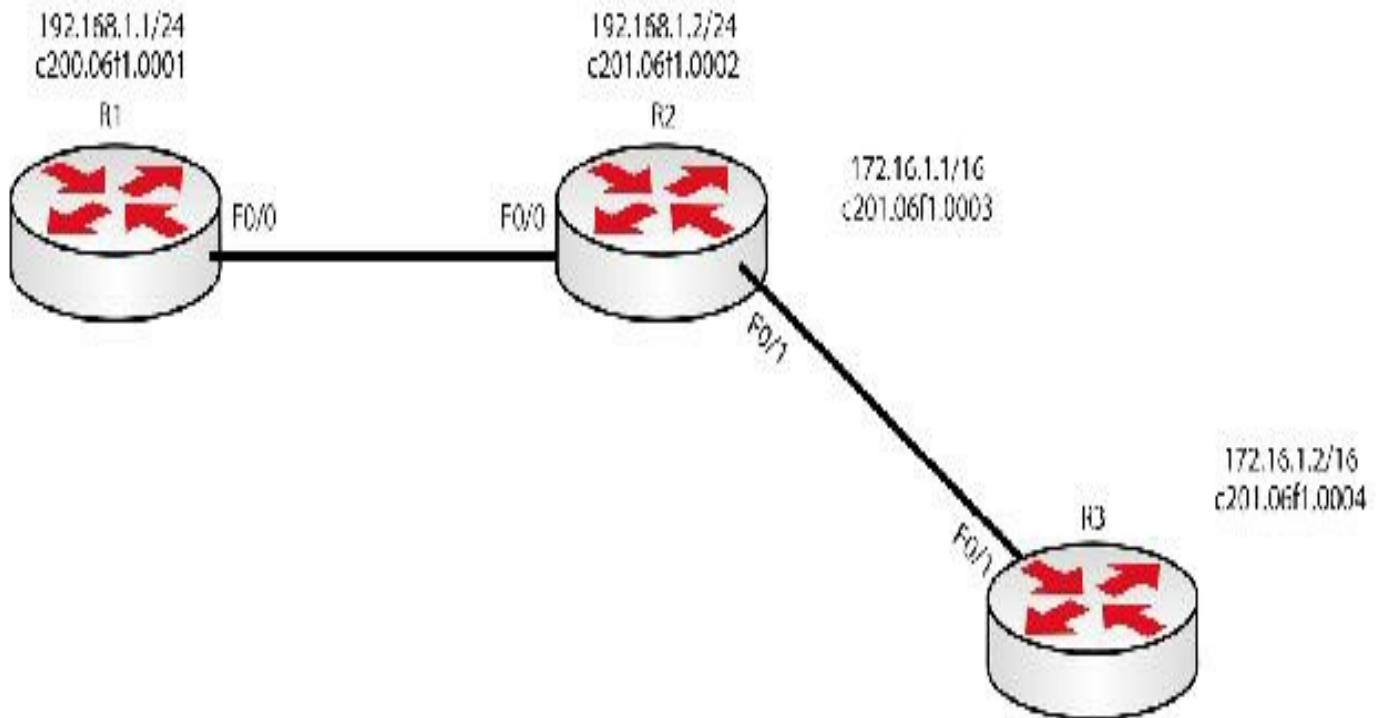


FIG 1.36 – Mini-lab: Proxy ARP

There are two networks here, 192.168.1.0/24 and 172.16.1.0/16. When R1 wants to send a packet to R3, it needs to ARP for the correct MAC address for R3 so it can encapsulate it correctly. The layer 3 address is known but not the layer 2 address. The layer 3 address can never change as the packet traverses the network, but the layer 2 address will change from host to host.

The only time you will see the layer 3 address change is if NAT is in place. You will learn how NAT works in a later chapter.



Please add all the IP addresses as per Figure 1.36, but you will also need to add some static routes. We don't cover these until later so feel free to come back to this lab after

reading up on static routes and IP addressing.

After adding the IP addresses, the static routes below need to be added to both R1 and R3 (you already know how to change the hostname from the previous configuration). You don't need to add a static route to R2 because both the 192 and 172 networks are attached (so R2 already knows where they both are).

```
R1(config)#interface f0/0
```

```
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
R1(config-if)#no shut
```

```
R1(config-if)#exit
```

```
R1(config)#ip route 0.0.0.0 0.0.0.0 f0/0
```

```
R2(config)#int fast0/0
```

```
R2(config-if)#ip add 192.168.1.2 255.255.255.0
```

```
R2(config-if)#no shut
```

```
R2(config-if)#int fast 0/1
```

```
R2(config-if)#ip add 172.16.1.1 255.255.0.0
```

```
R2(config-if)#no shut
```

```
R3(config)#int f0/0
```

```
R3(config-if)#ip add 172.16.1.2 255.255.0.0
```

```
R3(config-if)#no shut
```

```
R3(config-if)#ip route 0.0.0.0 0.0.0.0 f0/1
```

R1 has no ARP entry for 172.16.1.2 so it will need to broadcast for it:

```
R1#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
----------	---------	-----------	---------------	------	-----------

Internet	192.168.1.1 -	c201.06f1.0001	ARPA	FastEthernet0/0
----------	---------------	----------------	------	-----------------

If you issue a ping from R1 to reach 172.16.1.2 there will be a short delay due to the ARP broadcast and then a response. The delay causes the first few ping packets to fail due to a timeout indicated by the . below. If you didn't have the static route, R1 would have dropped the packet because there is no route to it and routers don't send broadcasts (by default).

```
R1#ping 172.16.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:

....!!

Success rate is 40 percent (2/5), round-trip min/avg/max = 40/46/52 ms

R2 uses Proxy ARP and sends its own MAC address as the destination for R3. This allows R1 to address the packet and send it. R2 will then swap the destination MAC address for the correct one attached to the R3 interface.

You can see the ARP request from R1 in the packet capture below. Item 9 is asking “Who has 172.16.1.2?” Enable Wireshark on GNS3 if you are using it or on your home lab.

No.	Time	Source	Destination	Protocol	Length	Info
1	8.000000	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	60	60 Reply
2	8.784888	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	60	60 Reply
3	9.991534	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	60	60 Reply
4	16.185948	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	60	60 Reply
5	20.000320	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	60	60 Reply
6	26.203568	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	60	60 Reply
7	29.833365	c2:01:06:f1:00:02	CDP/VTLP/STP/PAgP/LDLD CDP	370	Device ID: R2.lab.local Port ID: FastEthernet0/8	
8	29.995520	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	60	60 Reply
9	32.826860	c2:01:06:f1:00:01	Broadcast	ARP	60	60 Who has 172.16.1.2? Tell 192.168.1.1
10	32.937908	c2:01:06:f1:00:02	c2:01:06:f1:00:01	ARP	60	60 172.16.1.2 is at c2:01:06:f1:00:01
11	34.817507	192.168.1.1	172.16.1.2	ICMP	114	Echo (ping) request id=0x8060, seq=1/256, ttl=255
12	36.218194	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	60	60 Reply

Frame 9: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: c2:01:06:f1:00:01 (c2:01:06:f1:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 6
Opcode: request (1)
Sender MAC address: c2:01:06:f1:00:01 (c2:01:06:f1:00:01)
Sender IP address: 192.168.1.1 (192.168.1.1)
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 172.16.1.2 (172.16.1.2)

FIG 1.37 – ARP request

The Proxy ARP reply comes from R2 and you can see that the MAC address ends in 02, which belongs to the Fast Ethernet interface connecting R2 to R1.

No.	Time	Source	Destination	Protocol	Length	Info
1	8.000000	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	68	Reply
2	8.284600	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	68	Reply
3	9.991534	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	68	Reply
4	16.185946	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	68	Reply
5	20.898328	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	68	Reply
6	26.293580	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	68	Reply
7	29.833365	c2:01:06:f1:00:02	c2:01:06:f1:00:02	CDP/VTP/DTP/PnGp/LDLD CDP	370	Device ID: R2.lab.local Port ID: FastEthernet0/0
8	29.995528	c2:01:06:f1:00:02	c2:01:06:f1:00:02	LOOP	68	Reply
9	32.826888	c2:01:06:f1:00:01	Broadcast	ARP	68	Who has 172.16.1.2 Tell 192.168.1.1
10	32.837686	c2:01:06:f1:00:02	c2:01:06:f1:00:01	ARP	68	172.16.1.2 is at c2:01:06:f1:00:02
11	34.817987	192.168.1.1	172.16.1.2	ICMP	114	Echo (ping) request id=0x0000, seq=1/256, ttl=255
12	36.218194	c2:01:06:f1:00:01	c2:01:06:f1:00:01	LOOP	68	Reply

▷ Frame 10: 68 bytes on wire (480 bits), 68 bytes captured (480 bits)
 ▷ Ethernet II, Src: R2 (c2:01:06:f1:00:02) (c2:01:06:f1:00:02), Dst: c2:01:06:f1:00:01 (c2:01:06:f1:00:01)
 ▷ Address Resolution Protocol (reply)

Hardware type: Ethernet [1]
 Protocol type: IP (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: reply [2]
 Sender MAC address: c2:01:06:f1:00:02 (c2:01:06:f1:00:02)
 Sender IP address: 172.16.1.2 (172.16.1.2)
 Target MAC address: c2:01:06:f1:00:01 (c2:01:06:f1:00:01)
 Target IP address: 192.168.1.1 (192.168.1.1)

FIG 1.38 – Proxy ARP response

You can see that the ARP cache on R1 now has an entry for 172.16.1.2, but the MAC address is for the interface connected on R2. The fact that R2 will swap this for the correct MAC address is transparent to R1.

R1#show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	172.16.1.2	11	c201.06f1.0002	ARPA	Fa0/0
Internet	192.168.1.1	-	c201.06f1.0001	ARPA	Fa0/0

One final point is that if you do this lab using GNS3, you might have to manually change the MAC addresses because GNS3 sometimes duplicates the same one. Here is how I did it on R3:

R3(config)#int f0/1

R3(config-if)#mac-address c201.06f1.0004

[END OF MINI-LAB]

As a packet traverses the network, the source and destination IP addresses will never change in the packet (unless NAT is in use as Dario stated). The MAC address will have to change from hop to hop though. You can see an example of this in Figure 1.39 below:

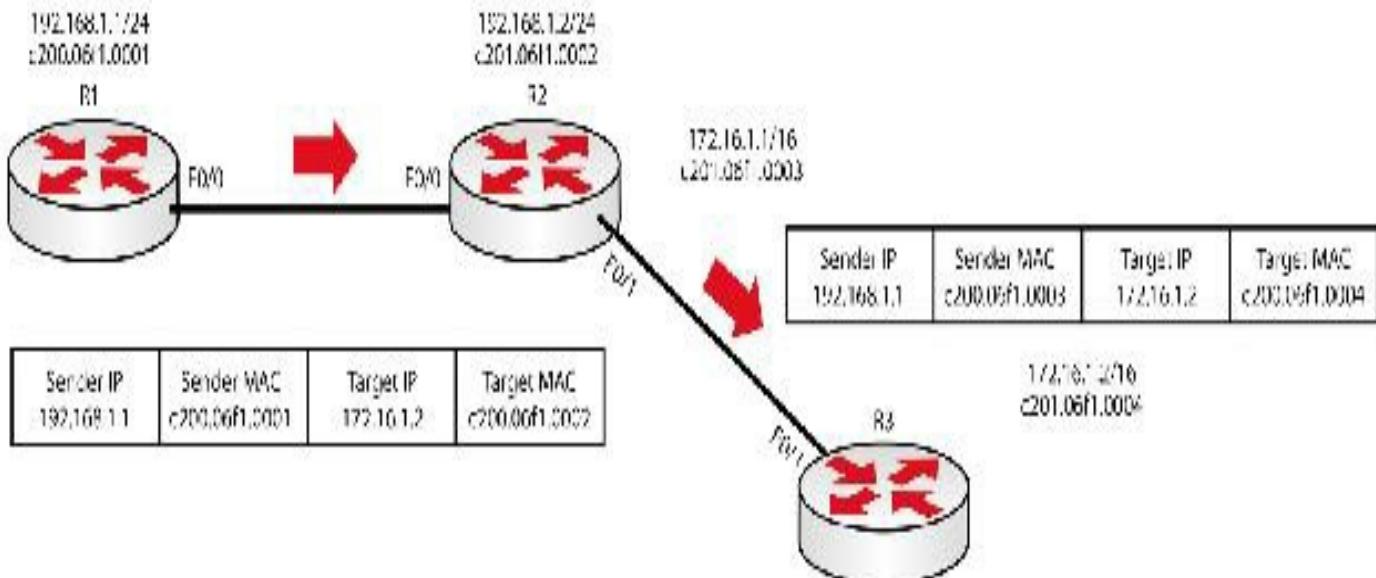


FIG 1.39 – MAC address never changes

This entire process is transparent to the hosts who view the connection, as shown in Figure 1.40 below:



FIG 1.40 – How the connection appears to hosts

If you see multiple IP addresses mapping to the same MAC address in your ARP cache, this indicates that Proxy ARP is in use. Please also bear in mind that you issue a show arp command on a router but on Windows PCs the command is arp -a.

If you found that the configurations above were a little hard to follow, come back to them later when you have a bit more confidence and knowledge.

Reverse Address Resolution Protocol

Reverse ARP (RARP) is the opposite of ARP in that it maps a known MAC address to an IP address. A typical use for this protocol is thin clients obtaining their IP addresses from servers when they are booting up. RARP has been replaced by Dynamic Host Configuration Protocol (DHCP), which can supply more than just an IP address (DHCP will be covered in detail later in this book).

Gratuitous ARP

On occasion (e.g., after the link goes up or the interface gets enabled), a device might issue an ARP request with its own IP address as the target address. This type of ARP request is known as **Gratuitous ARP (GARP)**.

GARP is used if a device wants to check whether a certain IP address is already in use in the network. If a response is received, then the IP address is in use. GARP is also used by HSRP (which will be covered later) when a router on another subnet takes over as an active router.

You can see a GARP packet capture below in Figure 1.41 (the image is copyrighted by Wireshark.org):

FIG 1.41 – GARP packet capture

Simple Network Management Protocol

SNMP is a management protocol that allows a management station to read and write specific values for different parameters of a network device. A network device can also send messages (called traps) to a management station to inform it of events such as high CPU usage, an interface fault, or other issues. You can debug SNMP traffic with the `debug snmp` command.

SNMP is an important CCNA exam topic that will be covered in more detail later in this manual.

Domain Name System

You've already learned that ARP resolves MAC addresses mapped to IP addresses. The Domain Name System (DNS) protocol uses UDP to resolve hostnames mapped to IP addresses. This allows you to enter www.howtonetwork.com, rather than an IP address, in a Web browser. Every Internet Service Provider (ISP) has its own DNS servers that list the most common hostnames and their corresponding IP addresses. If the IP address

of the hostname you want is not there, then the ISP will pass the request on to a DNS root server.



DNS uses UDP only for the first DNS Query sent by the client. If it doesn't receive a response from a DNS server it must retransmit the DNS Query using TCP. Furthermore, communication between DNS servers (Zone Transfer) uses TCP because it is necessary to maintain a consistent DNS database between them.

If you are using DHCP on your network, you are usually allocated an IP address for your nearest DNS server. If you are using a home network, the IP address will often be that of your router, which then passes the DNS Query on to your hosting company.

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of the "ipconfig /all" command. The output details the configuration for the "Ethernet adapter Local Area Connection".

```
C:\Users\owner>ipconfig /all

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : BigPond
Description . . . . . : Realtek PCIe GBE Family Controller
Physical Address . . . . . : 60-A4-4C-41-33-77
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv4 Address. . . . . : 10.0.0.16(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Wednesday, 22 July 2015 9:18:14 PM
Lease Expires . . . . . : Friday, 24 July 2015 9:18:14 AM
Default Gateway . . . . . : 10.0.0.138
DHCP Server . . . . . : 10.0.0.138
DNS Servers . . . . . : 10.0.0.138
NetBIOS over Tcpip. . . . . : Enabled
```

FIG 1.42 – ipconfig /all output

You can configure a hostname mapped to an IP address on a router with the command `ip host [name] [ip address]`. See the mini-lab below for more information.

Mini-lab – Pinging Hostnames

As hostnames are much easier to remember than a long list of IP addresses, mapping a hostname to an IP address can be accomplished by using the hostnames facility. Please add the IP addresses as per Figure 1.43 below (you learned how to do this in the earlier mini-labs). In this case, you won't need any static routes because the network is directly

connected.

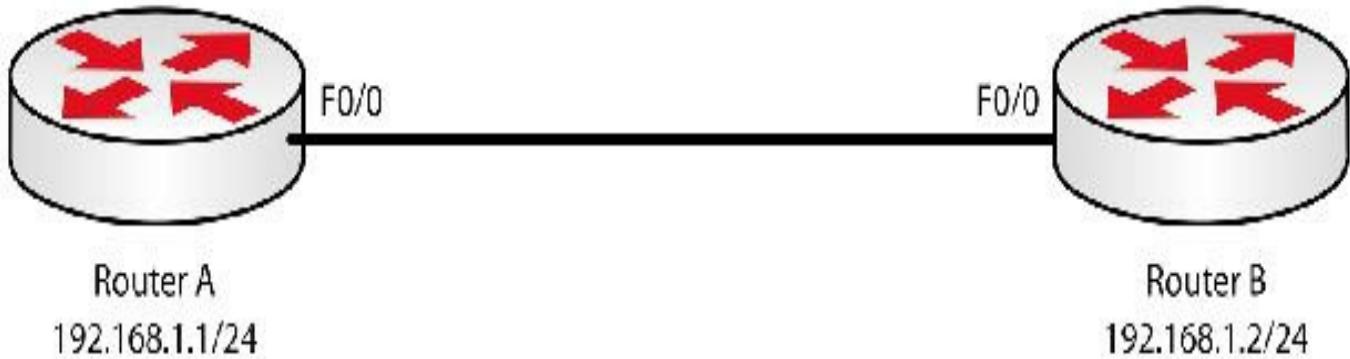


FIG 1.43 – Mini-lab: Pinging Hostnames

Add a hostname to the IP address mapping on Router A:

```
RouterA(config)#ip host RouterB 192.168.1.2
```

You can then ping the hostname:

```
RouterA#ping routerb i Not case sensitive
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

[END OF MINI-LAB]

You can specify one or more servers to act as DNS servers to resolve IP addresses mapped to hostnames. To do this you need to use the ip name-server [server-address 1] [server-address 2] command. Type the command below on Router A:

```
RouterA(config)#ip name-server 192.168.1.1 172.16.1.1
```

One problem you will no doubt encounter is the fact that routers automatically try to resolve an entry to a hostname if it is not an IOS command. If the router does not know what the hostname is, it tries to translate it. Type out the following on your router:

```
RouterA#tggt
```

Translating tggt...domain server (255.255.255.255) i **Broadcast packet**

Translating tggt...domain server (255.255.255.255)

% Unknown command or computer name, or unable to find computer address

```
RouterA#
```

Your output may differ depending on your IOS or whether you are using Packet Tracer or GNS3. This can be very frustrating because you have to wait for several seconds

while the router tries to resolve the hostname. You can disable name resolution using the no ip domain-lookup command:

```
RouterA#config terminal  
RouterA(config)#no ip domain-lookup  
RouterA(config)#^z  
RouterA#tgg
```

Translating tgg

```
% Unknown command or computer name, or unable to find computer address  
RouterA#'
```

Your IOS release may already have the no ip domain-lookup command configured by default. The output of a show run command on my router shows which commands are on by default:

```
RouterA#sh run  
Building configuration...  
  
Current configuration : 951 bytes  
  
!  
version 15.1  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname R1  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
ip cef  
!
```

no ip domain lookup

Router as a DNS Client

From the router command prompt, you can enter DNS servers and use fully qualified domain names (FQDN). Let's add a public DNS server to the IOS and ping a website by name. You will need access to a public DNS server to get this to work, but please do type the commands out regardless.

```
Router(config)#ip name-server 4.2.2.2
Router(config)#exit
Router#ping www.cisco.com
Translating www.cisco.com;
% Unrecognized host or address, or protocol not running
```

Next, tell the router to use DNS (remember, you disabled it a few outputs ago) by enabling domain-lookups:

```
Router(config)#ip domain-lookup
Router(config)#^z
Router#ping www.cisco.com
Translating www.cisco.com;...domain server (4.2.2.2) [OK]
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 198.133.219.25, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5)

If you look closer at the output above, you can see that the router queried the DNS server at 4.2.2.2 and it responded back with the IP address 198.133.219.25. You could, of course, enter your private DNS servers and you would be able to ping your internal names from your network equipment. This is for internal network management only (the management plane); in the next section, we'll look at using IOS for hosting DNS solutions.

Router as a DNS Proxy

There are occasions where your router provides DHCP services, and it would make life easier if that same router could forward DNS services too. Well, it can! Let's take a look at how to do that. Just like above, you have to tell the router where to find a DNS server and enable the router to provide a DNS service:

```
Router(config)#ip name-server 4.2.2.2
```

```
Router(config)#^Z  
Router#ping www.howtonetwork.com  
Translating www.howtonetwork.com  
% Unrecognized host or address, or protocol not running.
```

```
Router(config)#ip domain-lookup
```

Next, enable DNS on the router with the following command:

```
Router(config)#ip dns server
```

That is it! Now clients can use the router as a DNS server. Figure 1.44 below helps explain this process. But what if you need the router to resolve an internal address? You can now add DNS records directly on the router:

```
Router(config)#ip host server1.mydomain.com 10.10.10.5
```

From a host, let's set the DNS server on the router to look up the server1 DNS name:

```
F:\]nslookup
```

```
Default Server: dns-p1.mydomain.com  
Address: 10.10.10.11
```

```
> server 10.10.10.254 i Set the system to use the DNS server
```

```
Default Server: [10.10.10.254]
```

```
Address: 10.10.10.254
```

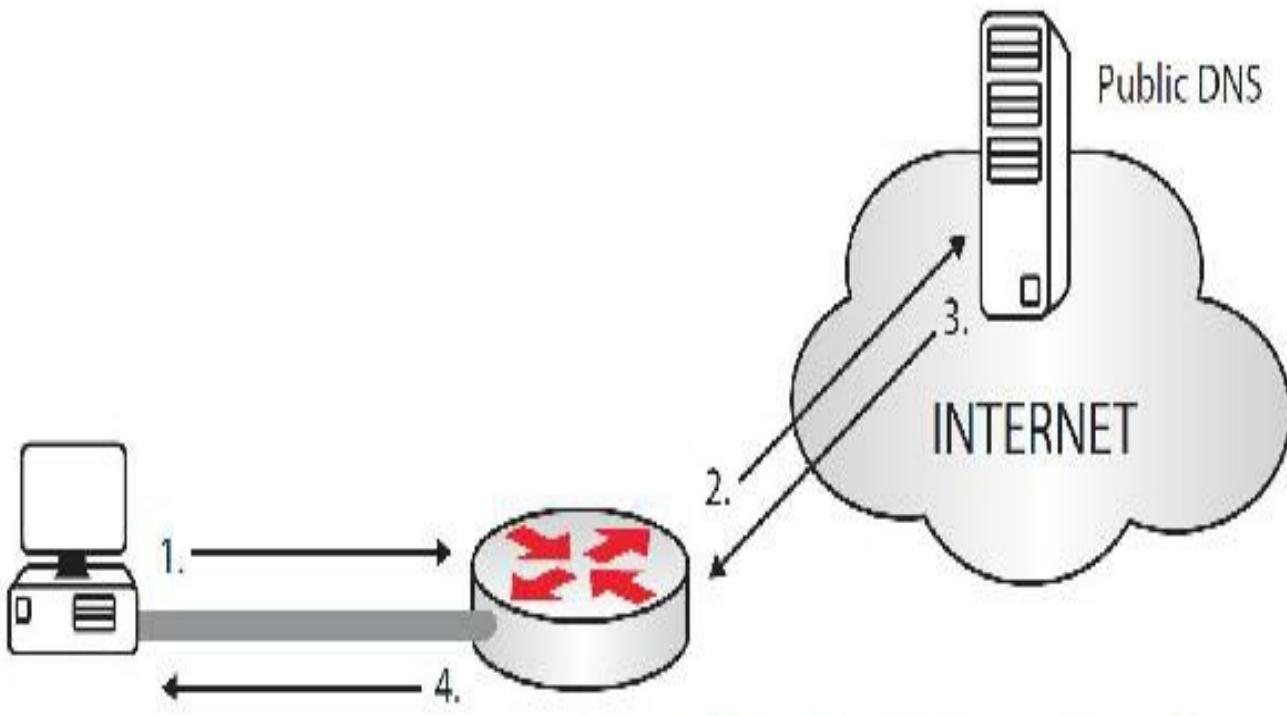
```
> server1.mydomain.com i Query for the name server1.mydomain.com
```

```
Server: 10.10.10.254 i The FQDN for the DNS server
```

```
Address: 10.10.10.254
```

```
Name: server1.mydomain.com i The response from the router running DNS
```

```
Address: 10.10.10.5 i The IP address of server1.mydomain.com
```



1. Client queries its configured DNS server, the router in our example
2. The router on behalf of the client queries the public DNS server
3. The public DNS server responds with the name resolution
4. The router forwards the name resolution back to the client

FIG 1.44 – Public DNS server

It is recommended that you do not use a router as a DNS server. A router is meant to route packets and that is what it does best. The previous scenario was for demonstration purposes and should be avoided when possible.

You can debug DNS traffic with the `debug domain` command.

It's worth noting that DNS for IPv4 and IPv6 are very similar in that they resolve hostnames mapped to IP addresses. One difference, however, is the name used for IPv4 and IPv6 DNS records. For IPv4 they are known as A records (made from 32 bits), whereas for IPv6 they are known as AAAA records (made from 128 bits). “A” simply stands for address.

There is a large amount of information to learn about DNS server hierarchy and zone transfers; however, this is more typically expected for a Network+ exam, not for the CCNA exam.

Cisco Discovery Protocol

CDP is a Cisco proprietary protocol designed to collect information about neighboring network devices. CDP is on by default on Cisco devices.

Because of its always-on feature, it presents a security risk to the network, but it is a very useful troubleshooting tool. For these reasons, we will cover CDP again in more detail later in this guide. CDP has always been an exam favorite.

It's worth noting that CDP only runs on Cisco devices (as mentioned) but it doesn't run on hubs, so even if one is connected, you can't use CDP to find it. This is why you might see MAC addresses recorded on your switch but no switch connected via CDP.

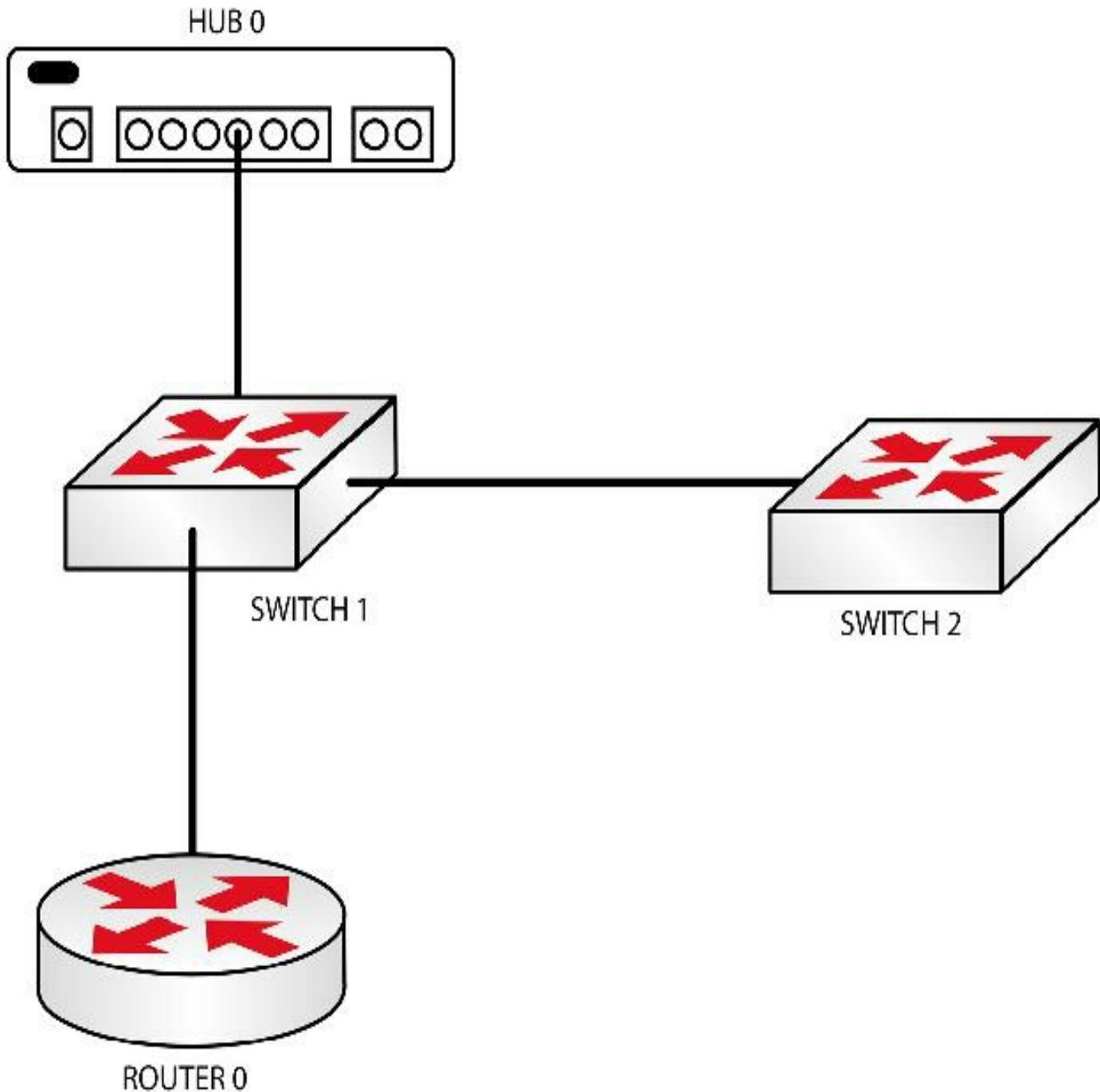


FIG 1.45 – CDP in operation

```
Switch1#show cdp neighbor
```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge, S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID	Local Interface	Holdtime	Capability	Platform	Port ID
-----------	-----------------	----------	------------	----------	---------

Switch	Fas0/1	136	S	2960	Fas0/1
--------	--------	-----	---	------	--------

Router	Gig1/1	139	R	C1900	Gig0/0
--------	--------	-----	---	-------	--------

Mini-lab – Checking for CDP Neighbors

In Figure 1.46 below, we have a simple Ethernet connection between R1 and R2. Please add the IP addresses to the relevant interfaces.

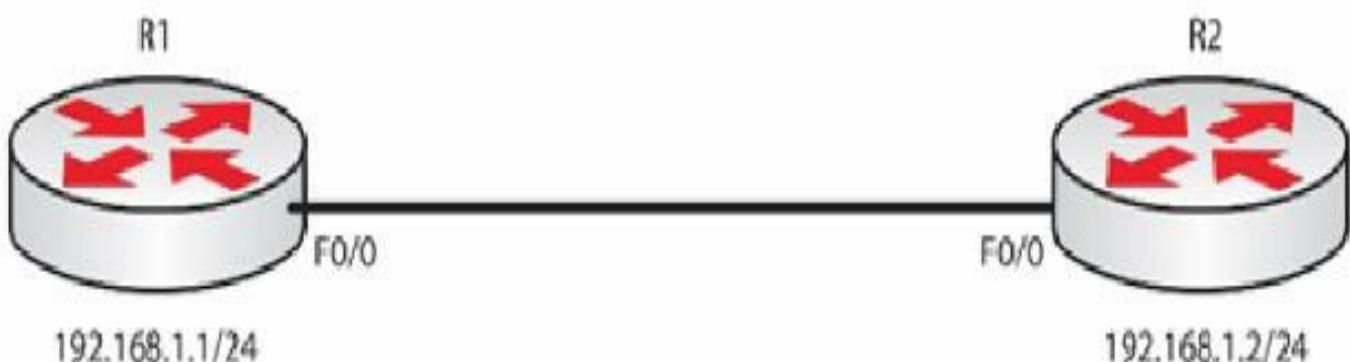


FIG 1.46 – Mini-lab: Checking for CDP Neighbors

The show cdp command will display basic CDP protocol information for your device. Of course, your output may differ if you have different router models and IOS releases because Cisco changes things over time.

```
R1#show cdp
```

Global CDP information:

- Sending CDP packets every 60 seconds

- Sending a holdtime value of 180 seconds

- Sending CDPv2 advertisements is enabled

You can see a basic CDP output with the show cdp neighbor command (note the U.S. spelling). I've shortened the command slightly because this is how you will use the commands in the real world.

```
R1#show cdp nei
```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge, S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Interface	Holdtime	Capability	Platform	Port ID
-----------	-----------------	----------	------------	----------	---------

R2 Fas0/0 170 R S I 3725 Fas0/0

Next, you will see how powerful a tool CDP can be for troubleshooting and how it can present a security vulnerability when you add the detail tag to the command to reveal far more information:

R1#show cdp neighbors detail

Device ID: R2.lab.local

Entry address(es):

IP address: 192.168.1.2

Platform: Cisco 3725, Capabilities: Router Switch IGMP

Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/0

Holdtime : 161 sec

Version :

Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 15.1(15)T7, RELEASE SOFTWARE (fc3)

advertisement version: 2

VTP Management Domain: “ ”

Duplex: half

R1 can see the remote IP address of R2 (192.168.1.2), the IOS release (15.1(15)T7), the platform (C3725), and the interface on which R1 learned this information (FastEthernet0/0). I've often used the commands above when called in to troubleshoot an unfamiliar network. With these commands you can quickly work out the physical topology, as well as where to try to telnet to and ping.

[END OF MINI-LAB]

Ethernet Concepts

Ethernet has become the de facto standard in modern networking.

You already know that Ethernet operates at layer 2 of the OSI model and consists of the two sublayers LLC and MAC. Moreover, you should already be familiar with Ethernet from previous studies, such as CompTIA Network+ (because, again, the CCNA is not a suitable exam for internetworking beginners).

Ethernet networking is actually a group or family of networking technologies. It was originally developed by Bob Metcalfe in 1972 while he was working at Xerox. It

became commercially available in the early 1980s with the IEEE forming the 802.3 subcommittee. The 802.3 standard gradually replaced vendor-centric technologies such as Token Ring and FDDI. Ethernet is known as a contention-based access method, meaning all of the hosts in the network segment are fighting to be heard. This brings us to CSMA/CD.

CSMA/CD

In earlier versions of Ethernet networks, there were many workstations sharing a single wire. It was quickly discovered that frames would often collide on the wire, causing errors in the data and lost frames. In order to control all the traffic, an algorithm known as carrier sense multiple access with collision detection (CSMA/CD) was created. This ensures that when a network interface card (NIC) wants to transmit data, it first listens to the wire to determine whether any other host is transmitting. It works in much the same way as a formal board meeting, where only one host can talk at a time (and the others need to remain quiet until their turn has come).

The very first implementation of Ethernet networking used coaxial cables, which are only capable of sending or receiving a signal, but not at the same time. This is why CSMA/CD was required.

Ethernet using CSMA/CD transmits frames in this order:

- Carrier sense – checks the cable to ensure that it is not in use (signal on the wire)
- Transmit if quiet – sends the frame if the line is clear
- Collision – both frames are destroyed if they collide on the wire; the collision is measured by the amount of voltage on the cable
- Jam signal – if the sending device detects the collision, it stops transmitting and sends a jam signal to destroy any fragmented frames
- Backoff algorithm – a random timer to prevent all stations from retransmitting at the same time

The entire CSMA/CD process can be explained through several steps:

1. The device wanting to transmit a frame listens for a carrier signal on the wire. No signal means that it is clear to send the frame.
2. The frame is put onto the wire.
3. The sending device listens to discover whether a collision has occurred.
4. If there was a collision, all sending devices send a jamming signal to announce the fact that there was a collision.
5. A random timer runs and no frames can be sent by the original devices until it

expires.

6. The device starts back at step 1.

Figure 1.47 below gives you a visual representation of the process:

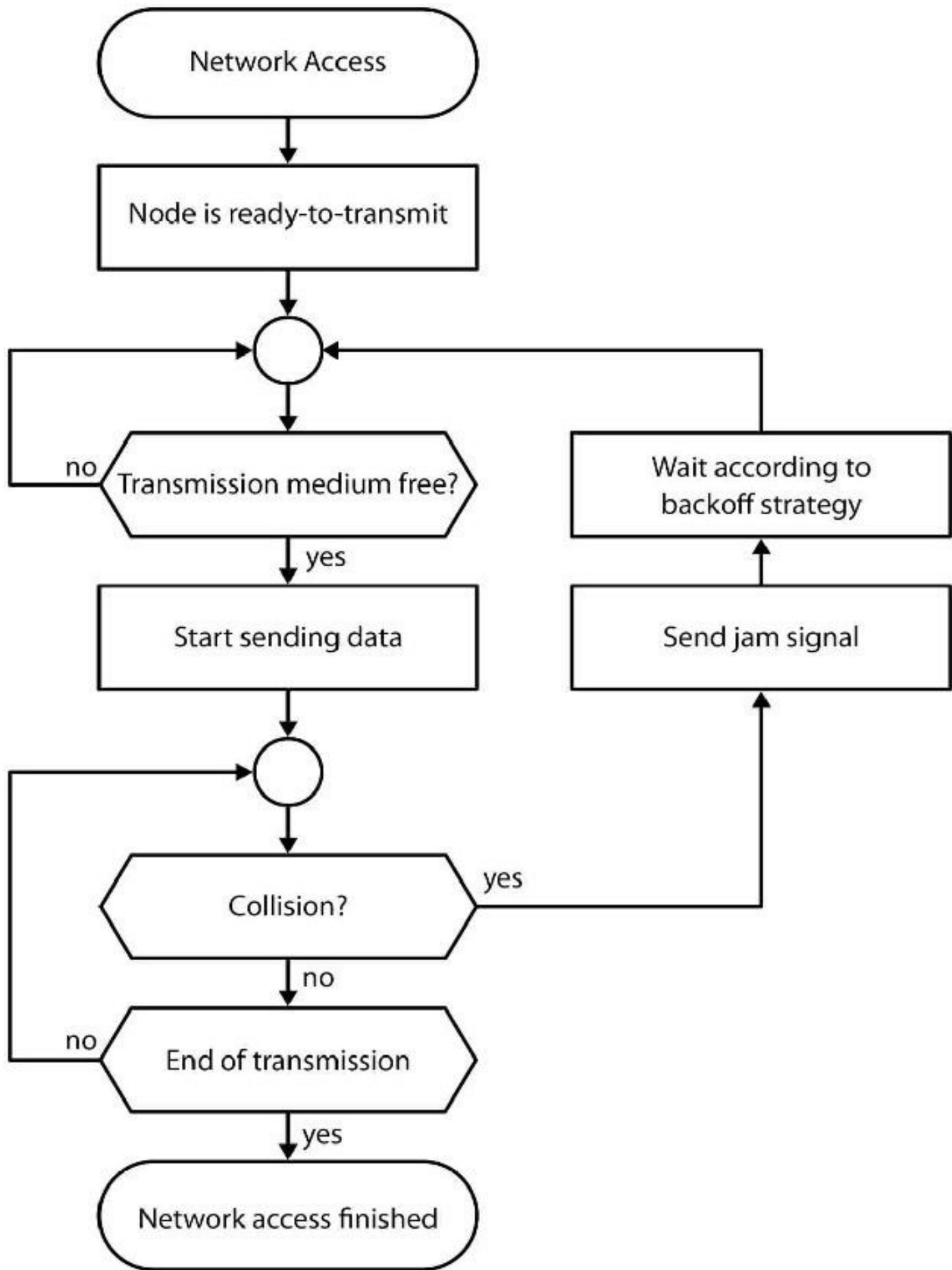


FIG 1.47 – CSMA/CD

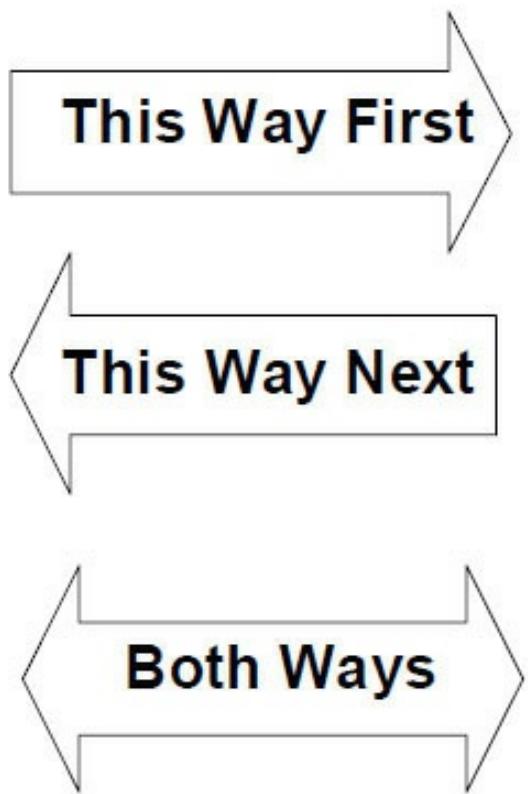
Modern implementations of Ethernet no longer require CSMA/CD because different wires are used to send and receive, so there are no collisions. Despite this fact, Cisco wants you to understand this concept for the exam.

Duplex Settings

The traditional method for Ethernet to operate was to use half-duplex mode. This works in the same way as walkie-talkies: a host can either transmit or receive—it can do either but not both at the same time. Half-duplex is a technology that originally ran on coaxial cables. This method is still used today if there are hubs in the network.

Full-duplex mode allows the host to send and receive frames at the same time. This is achieved by using UTP cables (which we will cover later), allowing a dedicated connection between devices on a LAN and providing separate paths for the data to be transmitted on. Because a wire is not needed to listen for collisions, it is free to be used to send or receive frames. Full-duplex is equivalent to Ethernet minus CSMA/CD.

IN THE REAL WORLD: A very large amount of late collisions showing up on your Ethernet interface is a sure sign of duplex mismatch settings on your Ethernet ports. We will cover this situation in the troubleshooting section.



**HALF DUPLEX -
ONE WAY AT A TIME**

**FULL DUPLEX -
BOTH DIRECTIONS
AT THE SAME TIME**

FIG 1.48 – Duplex operations

Bear in mind that a network running half-duplex at 100 Mbps won't work at 200 Mbps when you enable full-duplex. Instead, you will be able to transmit 100 Mbps both upstream and downstream at the same time. These are theoretical speeds that are dependent on other factors, like the limitations of the hardware used in the network.

Duplex and CSMA/CD issues are generally a thing of the past because most devices have a dedicated connection to the network switch and different wires are used to send and receive traffic, so there will not be any collisions on the wire. Moreover, there is no need to check whether the wire is free to transmit or to check for collisions, and there will be no retransmissions, which are all good news for software drivers and CPU cycles.

Autonegotiation

Network engineers originally had to determine which devices could transmit at 10 or 100 Mbps, as well as those working at full- and half-duplex. This made configuring devices, switches, and routers very time intensive, as well as increased the likelihood of configuration errors.

Autonegotiation uses fast link pulses (FLPs) to allow the devices to agree to send data at the highest available rate. As the network administrator, you can let devices autonegotiate or you can use commands to set the speed and duplex to be used. You are bound to encounter autonegotiation issues in the real world at some point and possibly in

the CCNA exam.

Mini-lab – Configuring Ethernet Speed and Duplex Settings

You should configure speed and duplex settings on a router Fast Ethernet interface rather than let it autonegotiate. Your default settings may differ from mine and your interface name may also differ, so use the show ip interface brief command to see which interface types you have available. You don't need to add an IP address, but I already have one on my router.

R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.1	YES	manual	up	up
FastEthernet0/1	unassigned	YES	unset	administratively down	down

R1#show interfaces FastEthernet0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c200.06f4.0000 (bia c200.06f4.0000)

Internet address is 192.168.1.1/24

MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

Keepalive set (10 sec)

Half-duplex, 10Mb/s, 100BaseTX/FX

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#interface f0/0

R1(config-if)#speed ?

10 Force 10 Mbps operation

100 Force 100 Mbps operation

auto Enable AUTO speed configuration

R1(config-if)#speed 100

```
R1(config-if)#duplex ?
  auto  Enable AUTO duplex configuration
  full  Force full-duplex operation
  half  Force half-duplex operation
```

```
R1(config-if)#duplex full
```

```
R1(config-if)#end
```

```
R1#show int f0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
  Hardware is Gt96k FE, address is c200.06f4.0000 (bia c200.06f4.0000)
```

```
  Internet address is 192.168.1.1/24
```

```
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
```

```
    reliability 255/255, txload 1/255, rxload 1/255
```

```
  Encapsulation ARPA, Loopback not set
```

```
  Keepalive set (10 sec)
```

Full-duplex, 100Mb/s, 100BaseTX/FX

[END OF MINI-LAB]

In my installation experiences over the years, best practice has been to always hard set the interfaces to 100 or 1000, if available, with full-duplex, but check with your network manager for your policy. If you had the interface operating at half-duplex, then CSMA/CD would be in effect, regardless of the speed.

Ethernet Frames

When LAN devices transmit information to each other, they must use an agreed method to format the data so that all the devices understand what the fields signify. LAN data encapsulated at layer 2 is referred to as frames. If you inspected the data it would be encapsulated with a layer 2 header and trailer. If the data was encapsulated with layer 3 information, such as an IP address, it would be referred to as a packet.

Ethernet has four different frame types available. They have been improved on over the years but some have become obsolete.

1. Ethernet 802.2 SAP
2. Ethernet 802.2 SNAP
3. Ethernet 802.3
4. Ethernet II

The 802.2 standard is not used today, while 802.3 was eventually superseded by Ethernet II, which is also known as Ethernet 2 or DIX Ethernet (named after DEC, Intel, and Xerox, who were the original designers).

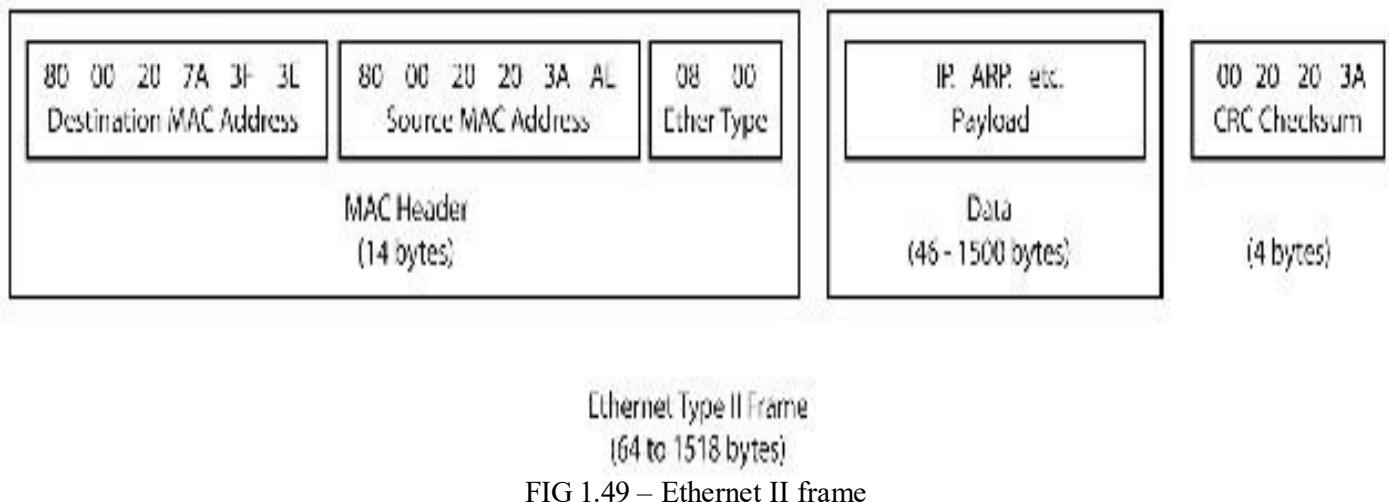


FIG 1.49 – Ethernet II frame

The IEEE Ethernet II frame consists of specific fields that have been determined by the IEEE committee:

- **Destination Address** – destination MAC address; can be unicast, broadcast, or multicast
- **Source Address** – MAC address of the sending host
- **Ether Type** – identifies upper layer protocols such as IPv4 (0x0800) or IPv6 (0x86DD)
- **Data** – payload in the frame; the data being transferred

Not shown in Figure 1.49 above are two fields: the Preamble, which starts the frame, and the Frame Check Sequence (FCS), which ends it.

- **Preamble** – allows devices to synchronize their receiver clocks
- **Frame Check Sequence (FCS)** – provides a cyclic-redundancy check (CRC) on all data in the frame to check for corruption

LAN Traffic

Ethernet addresses can be one of three types (IPv6 differs but we will cover this later):

1. Unicast (one-to-one) – a single LAN interface is the recipient
2. Broadcast (one-to-all) – all devices on the LAN are the recipients
3. Multicast (one-to-many) – a subset of all devices are the recipients

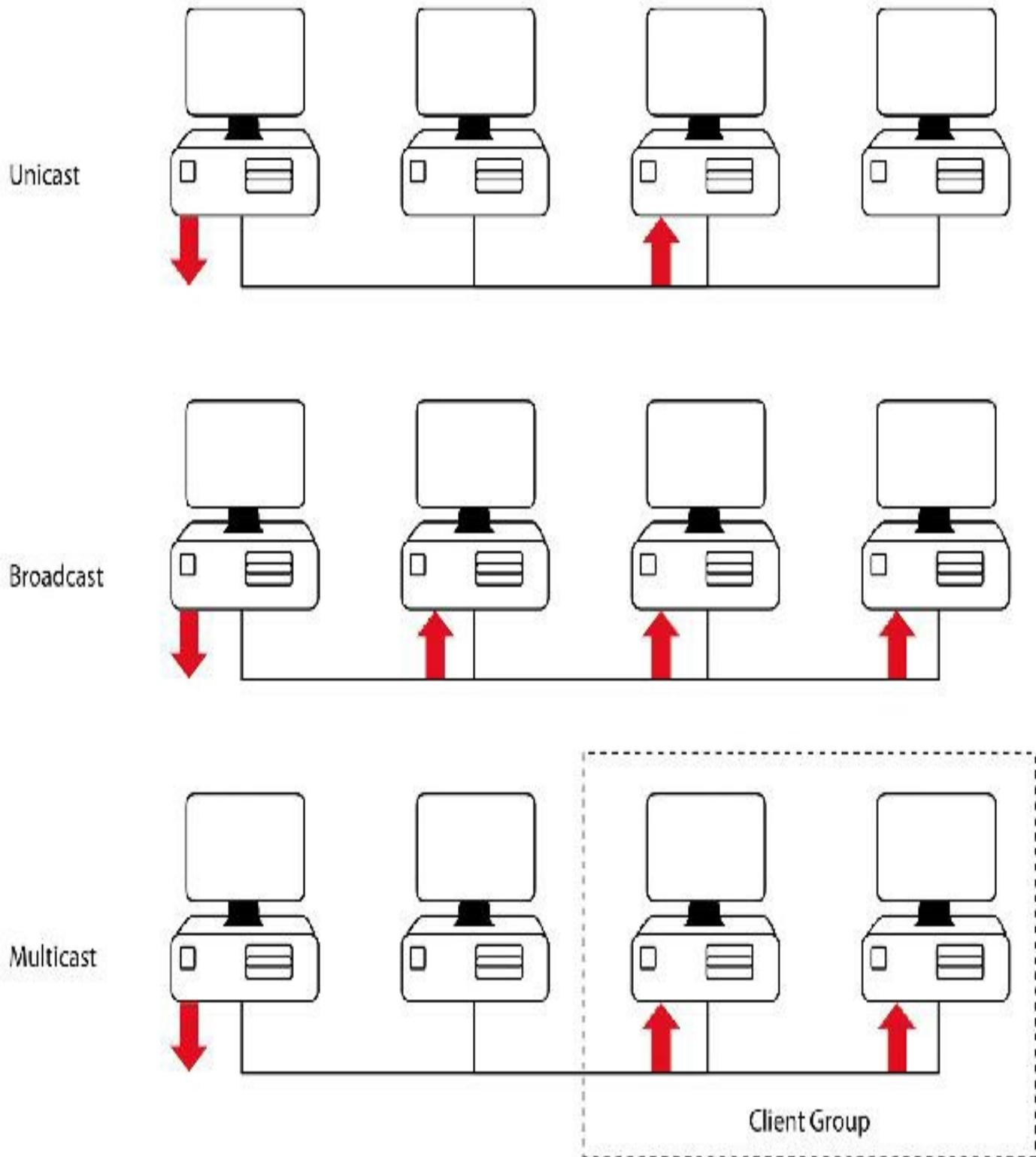


FIG 1.50 – IPv4 Ethernet traffic types

Unicast

Unicast traffic is transmitted from one host on the LAN to another host on the LAN. The switch forwards a unicast frame out of the interface the destination address is associated with. If there was no interface associated with the frame, the switch will

forward the frame out of all interfaces (as a broadcast) except the interface the frame was received on.

In the output below, if the switch receives a frame addressed to host 0001.42dd.eca2 from the host connected to F0/1, it will broadcast out of all ports except F0/1 but including all the ports it currently has a MAC address recorded.

Switch#show mac address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0001.42dd.eca1	DYNAMIC	Fa0/1
1	0001.c7d2.4eb1	DYNAMIC	Fa0/2
1	00e0.f9de.e036	DYNAMIC	Fa0/3

Broadcast

Broadcast traffic is exclusive to LANs because broadcasts are not forwarded by routers. A broadcast is sent by a device on the LAN and heard by all other devices on that segment of the LAN. A LAN can be broken into segments by routers or VLANs (more on VLANs later). A broadcast frame appears as FF:FF:FF:FF:FF:FF in hexadecimal.

A switch will forward a broadcast frame out of all interfaces except the interface the frame was received on. All hosts receive the broadcast frame and they must use CPU resources to check the destination frame to see whether it is the intended recipient. If it isn't, it will discard the frame.

Multicast

Multicast traffic is transmitted from a host and listened for by a subset of hosts in the network. Although all devices hear the frame, it is ignored by all but the devices waiting to receive the multicast. An example of a multicast is a Hello packet from the OSPF routing protocol, which is forwarded on address 224.0.0.5 (more on OSPF later).

A switch will forward a multicast in the same way that it forwards a unicast frame with an unknown destination port, unless there are explicit configurations to do otherwise.

You can see a packet capture of an OSPF multicast packet in Figure 1.51 below:

11 41.633800	c2:02:07:02:00:00	c2:02:07:02:00:00	LOOP	60 Reply
12 45.6/1270	192.168.1.2	224.0.0.5	OSPF	50 Hello Packet
13 49.306100	c2:00:07:02:00:00	CDP/VTP/DTP/PAQP/LQD CDP		361 Device ID: R1.lab.local P
Frame 12: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)				
Ethernet II, Src: c2:02:07:02:00:00 (c2:02:07:02:00:00), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)				
Internet Protocol Version 4, Src: 192.168.1.2 (192.168.1.2), Dst: 224.0.0.5 (224.0.0.5)				
<pre> Version: 4 Header length: 20 bytes Differentiated Services Field: 0xc0 (DSCP 0x30; Class Selector 5; ECN: 00; Not-ECT (Not ECN-Capable)) Total Length: 76 Identification: 0x0000 (0) Flags: 0x00 Fragment offset: 0 Time to live: 1 Protocol: OSPF TGP (89) Header checksum: 0x16ca [correct] Source: 192.168.1.2 (192.168.1.2) </pre>				

FIG 1.51 – OSPF multicast packet capture

IEEE Standards

The Institute of Electrical and Electronics Engineers (IEEE) is a non-profit group of professionals and, among other things, they are the leading authority in the field of computer engineering.

The IEEE publishes standards for networking and one such standard is called the 802 project. These standards are constantly changing to keep pace with emerging technologies and developments in the industry. IEEE 802.3 is actually a working group as well as a collection of standards that defines wired Ethernetworking at layers 1 and 2. This is mostly concerned with LAN networking, with some limited WAN applications. The connections between devices under 802.3 are copper or fiber cables. Also defined under 802.3 is CSMA/CD.

802.3 standards include 802.3u Fast Ethernet (100BASE-TX, 100BASE-FX), 802.3z Gigabit Ethernet (1000BaseX) over Fiber at 1 Gbps, and 802.3ab Gigabit Ethernet (100BASE-T) over copper wiring. IEEE 802.3bs™ “Standard for Ethernet Amendment: Media Access Control Parameters, Physical Layers and Management Parameters for 400 Gb/s Operation” is currently under development by the IEEE P802.3bs 400 Gb/s Ethernet Task Force.

FOR THE EXAM: Be very familiar with the common 802 standards. You will be expected to know them.

Table 1-4: IEEE standards

Standard	Covers
802.1	Internetworking
802.2	Logical Link Control (LLC)
802.3	Ethernet/CSMA/CD
802.4	Token Bus
802.5	Token Ring
802.6	Metropolitan Area Networks (MANs)
802.7	Broadband
802.8	Fiber Optics
802.9	Integrated Voice and Video
802.10	LAN Security
802.11	Wireless Networking (WiFi)
802.12	100BaseVG-AnyLAN

The Cisco Hierarchical Networking Model

Although not specifically addressed in the syllabus, you will benefit from understanding how Cisco likes to structure network design. This is covered in great detail in the Cisco Certified Design Associate (CCDA) exam. You will often refer to the core, distribution, or access layers in terms of the models of routers and switches available or when addressing network issues.

The Cisco approach to designing and building networks is to break networking into three distinct layers: core, distribution, and access. Each layer has very specific responsibilities to perform. You will find yourself referring to these layers in your day-to-day job as a network engineer and if your network is big enough, each layer may well have its own support team.

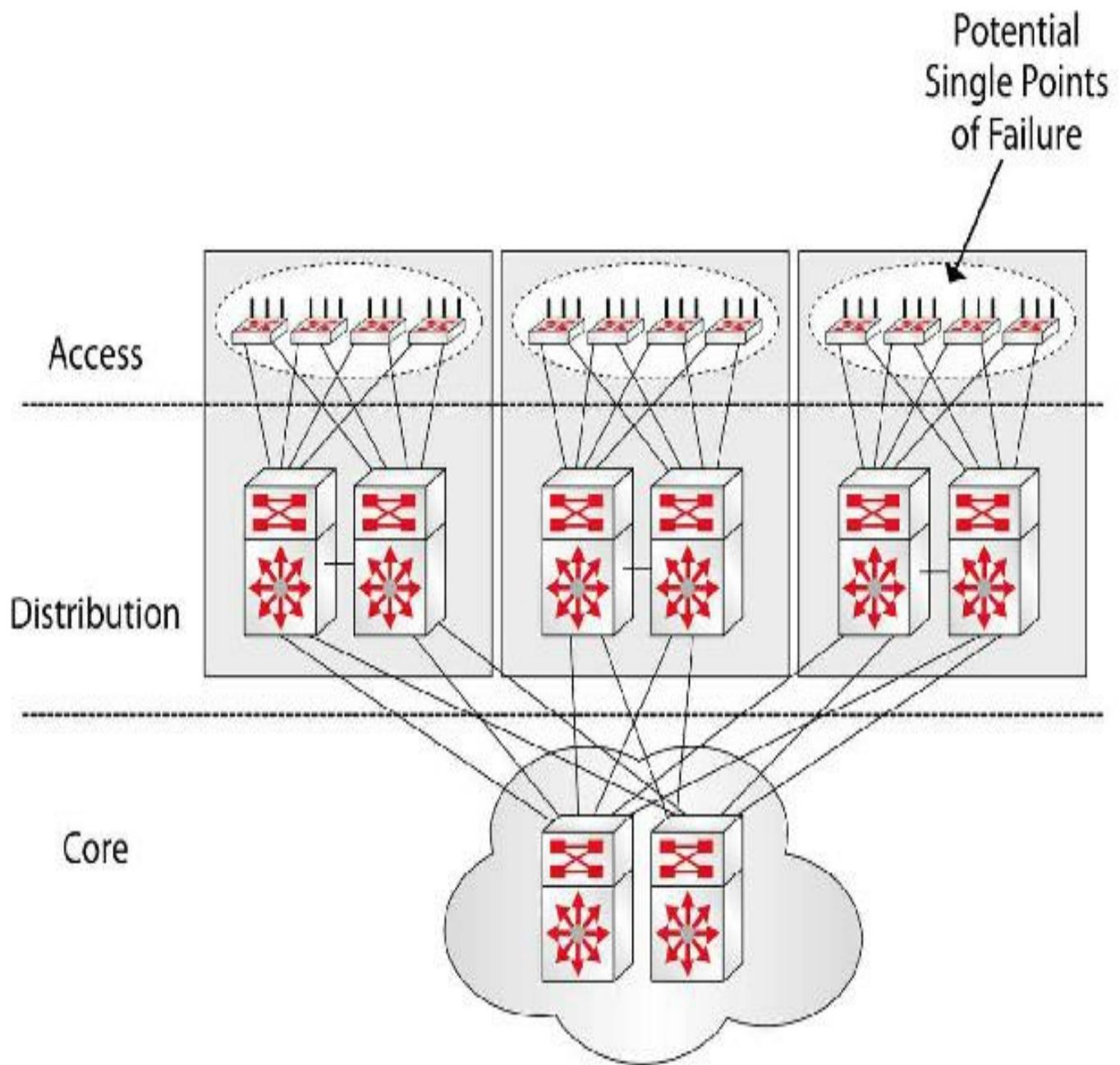


FIG 1.52 – Cisco networking model

The Core Layer

Also known as the backbone of the network, this is the heart of the network and it's responsible for switching huge amounts of traffic very reliably. In order to carry out this role, certain characteristics are associated with the core layer. It must offer high reliability, provide redundancy (i.e., if one part breaks, another can pick up where it left off), provide fault tolerance, and avoid slowing traffic down by filtering it in any way.

You will find core switches used in the largest campus LANs. Campus simply refers to LANs created to support larger buildings or a number of buildings in close proximity to one another. Core switches are very expensive and are used to connect distribution

layer switches. They are designed to forward traffic at very high speeds.

The Distribution Layer

Also known as the workgroup layer, the distribution layer allows communication between the core and access layers. Packets are manipulated at this layer. The distribution layer should perform a range of services such as security, address summarization, workgroup access routing between LANs, traffic filtering, and redistribution.

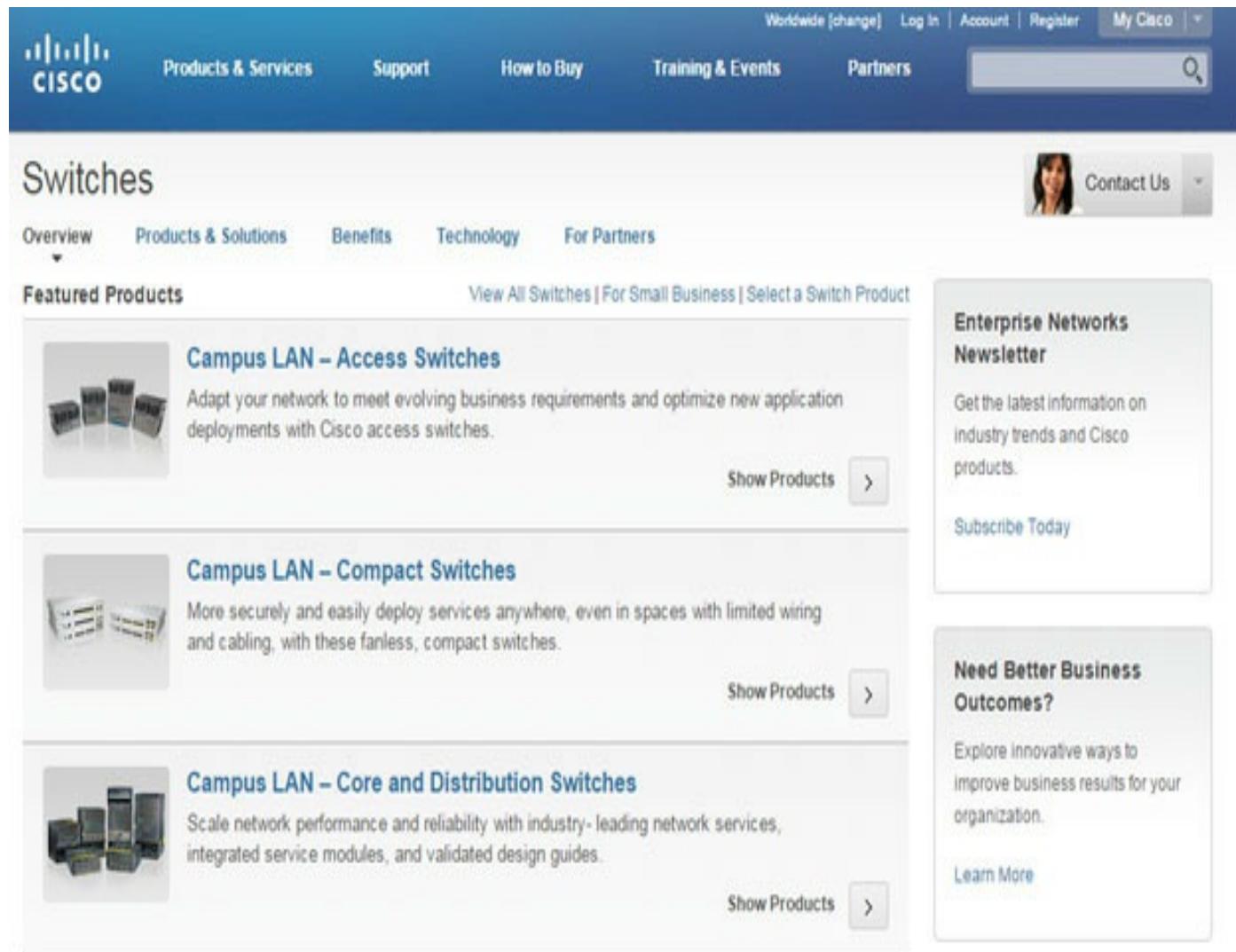
Distribution layer switches are found in larger networks and are used to forward traffic between access layer switches. Distribution layer switches usually connect to one another with two or more uplink cables, ensuring redundancy and reducing the amount of cabling on the LAN because you aren't connecting the access layer switches together. They are not used by end-user devices for network access.

The Access Layer

Also known as the desktop layer, this is the point at which end-users connect to the network. The access layer allows user access to local segments in the network. Functions performed here include the creation of separate collision domains (segmenting the network into smaller chunks), sharing bandwidth, and local policies for network traffic control.

Cisco access layer switches connect directly to end-users' PCs, network printers, and servers. The access layer switches can connect to one or more distribution layer switches. You will hear me refer to these three layers throughout this manual.

Breaking down networks into specific layers helps us to understand how they work, as well as which devices should be performing which functions and features. It's well worth visiting the Cisco.com website to review the models and features available for their core, distribution, and access layer switches.



The screenshot shows the Cisco website's 'Switches' page. At the top, there's a navigation bar with links for 'Products & Services', 'Support', 'How to Buy', 'Training & Events', 'Partners', and user account options ('Worldwide [change]', 'Log In', 'Account', 'Register', 'My Cisco'). A search bar is also at the top right. Below the header, the word 'Switches' is prominently displayed. To the right of the title is a small profile picture and a 'Contact Us' button. Under the main title, there are several navigation links: 'Overview', 'Products & Solutions', 'Benefits', 'Technology', and 'For Partners'. A 'Featured Products' section follows, featuring three categories: 'Campus LAN – Access Switches', 'Campus LAN – Compact Switches', and 'Campus LAN – Core and Distribution Switches'. Each category includes a small image of the respective switch model, a brief description, and a 'Show Products' button with a right-pointing arrow. To the right of the products is a sidebar with two sections: 'Enterprise Networks Newsletter' (describing the latest information on industry trends and Cisco products) and 'Need Better Business Outcomes?' (explaining innovative ways to improve business results). Each sidebar section has a 'Subscribe Today' or 'Learn More' button.

FIG 1.53 – Cisco device model options

Cabling the Network

PCs, hubs, switches, routers, and all other types of networking equipment use different connector types. There are many reasons for this but the main two are: (1) different networks can send data in different formats, requiring the use of different cables; and (2) as speeds have improved, certain cable types and network interfaces have become obsolete and have been replaced by newer, more efficiently designed components.

LAN Cabling

LAN cabling can use several types of cables, including coaxial and fiber optic; however, the industry standard for LAN is referred to as unshielded twisted pair (UTP) cable. UTP cable is broken down into different categories (CAT) based on what speed the cable is capable of passing data.

A UTP Ethernet connection consists of the UTP cable, an attachment for the end of the cable known as an RJ-45 male connector, and an RJ-45 female connector on the device

the cable connects to.

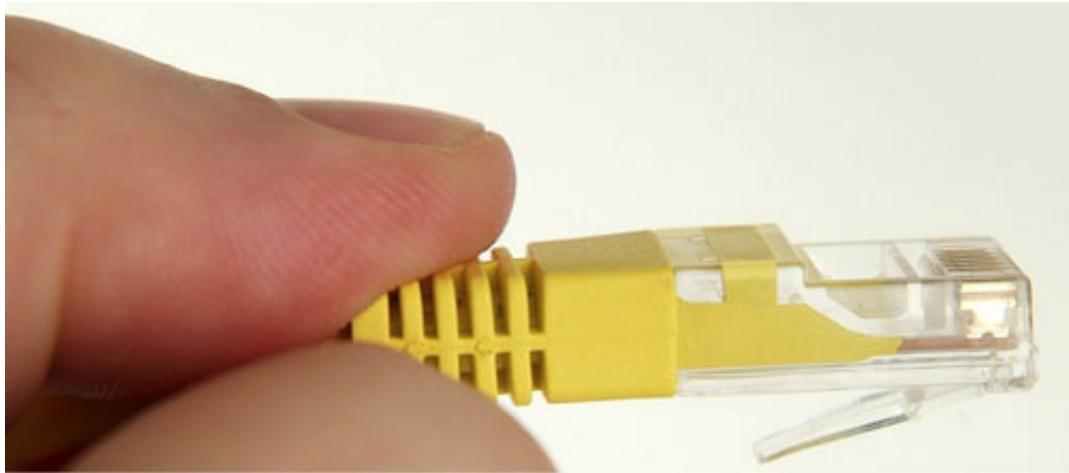


FIG 1.54 – RJ-45 connector

Inside the cable are eight smaller wires that are color-coded so they can be identified. You should not be expected to remember which color is associated with which pin, but you will be expected to know what you can do with these wires to make different types of UTP cables.

Each standard or category is rated for a different purpose. Table 1-5 below shows the IEEE speeds for the various cable specifications.

Table 1-5: Cat 1 to 7 characteristics

Category of Cable	Transmission Speed	Standards
CAT 1	N/A	Voice only (telephone cable)
CAT 2	4 Mbps	Token Ring
CAT 3	10 Mbps	Token Ring/10BASE-T/100BASE-T4
CAT 4	16 Mbps	Token Ring
CAT 5	100 Mbps	Token Ring/10BASE-T/100BASE-TX
CAT 5e	1 Gbps	10BASE-T/100BASE-TX/1000BASE-T
CAT 6	10 Gbps	1000BASE-T/1000BASE-TX/10GBASE-T
CAT 6a	10 Gbps	1000BASE-T/1000BASE-TX/10GBASE-T
CAT 7	10 Gbps	1000BASE-T/1000BASE-TX/10GBASE-T

IEEE and Cabling Standards

LAN cabling is a fairly detailed subject and for historical reasons there are many cable types. Each cable type has certain limitations and can be used only on certain

topologies.

802.3 Ethernet

Ethernet works at 10 Mbps, as referred to by the IEEE 802.3 standard. Ethernet can be either a star or a bus topology. The cabling used can be coaxial, CAT 3 or higher, using twisted pair. Twisted pair cable is referred to as XBase-T, where X refers to the speed at which the media operates, Base denotes that baseband transmission is used, and T refers to the fact that the wires are twisted into pairs to reduce crosstalk between the wires and radio frequency interference (please read any Network+ guide for more information on this).

You should already be familiar with topology types for networks, but Figure 1.55 below will serve as a memory jogger:

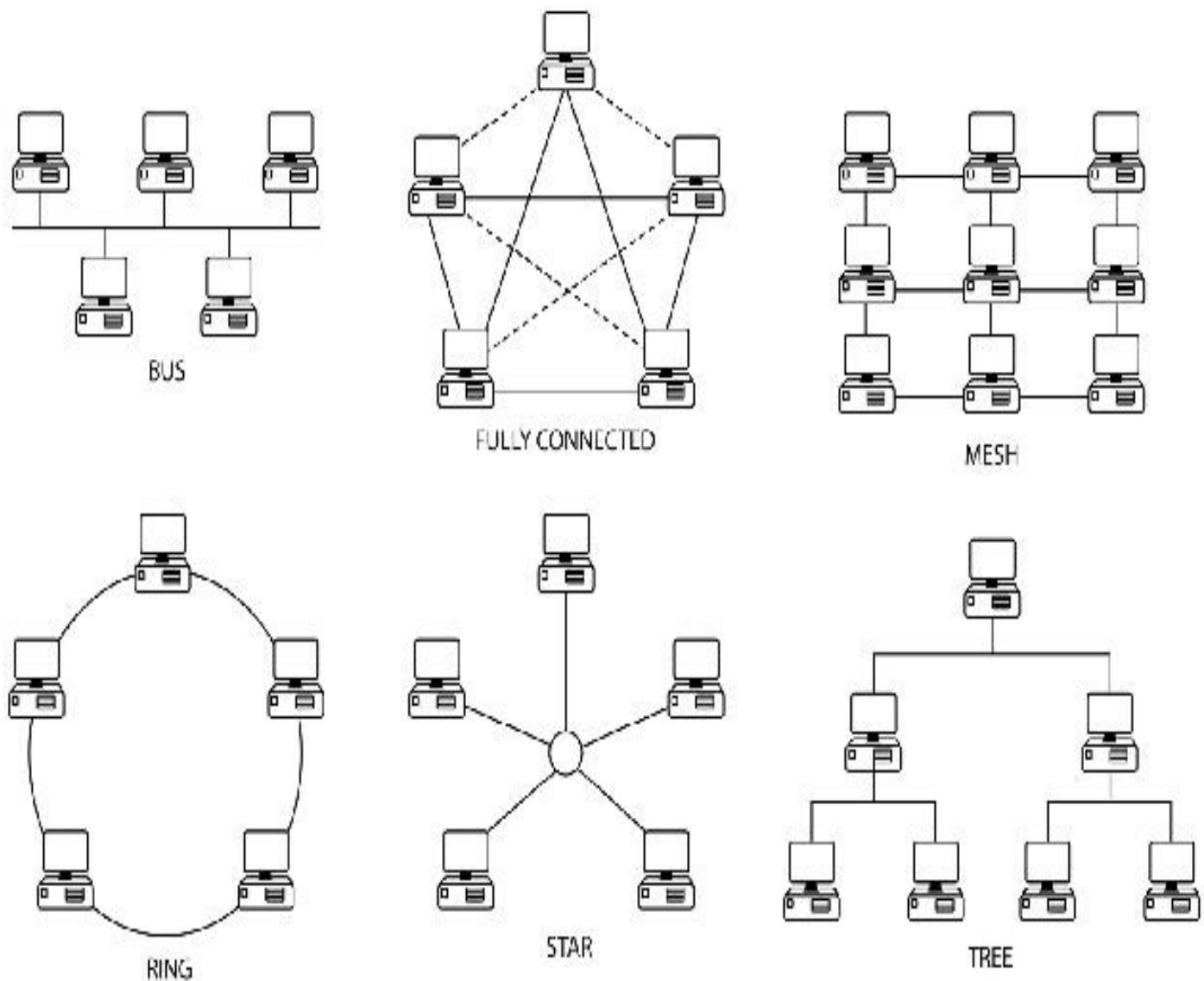


FIG 1.55 – Network topology types

The topology you use will depend on the cabling in use, the protocols, your budget, and

available equipment. As you can see, some topologies such as ring will bring the entire network down if a device or cable fails. The bus topology was the only choice in early Ethernet networks because the cable used was coaxial. With the move to network switches and UTP cables, the common topology is now star.

802.3u Fast Ethernet

The 802.3u standard is used by most modern networks. Fast Ethernet is a vast improvement on standard Ethernet. It uses a star topology and either CAT 5 twisted pair or fiber optic cable. It overcame technologies such as FDDI, 100VG AnyLAN, and ATM to become the leading solution.

Fast Ethernet provides an effective and inexpensive upgrade path from 10 Mbps Ethernet, as well as being backward compatible. It is able to operate over CAT 3, CAT 5, fiber optic, and even USB cables.

802.3z (or 802.3ab) Gigabit Ethernet

Gigabit Ethernet is an improvement on Fast Ethernet. It uses the same 802.3 frame format as Fast Ethernet in a star topology with twisted pair (802.3ab/1000Base-T) or fiber optic/shielded copper (802.3z/1000Base-X) cable.

Table 1-6: Ethernet cable specifications

Cable	Bandwidth (Mbps)	Max Cable Distance (meters)	Cable Construction
10BASE2	10	185	Thinnet coaxial
10BASE5	10	500	Thicknet coaxial
10BASE-T	10	100	UTP
100BASE-T	100	100	UTP/fiber
1000BASE-T	1000	100	UTP
1000BASE-X	1000	Varies	Fiber/shielded copper
1000BASE-CS	1000	25	STP
1000BASE-SX	1000	220 or 550	Short-wavelength laser/ MM fiber
1000BASE-LX	1000	550 (MM), 5 Km (SM), or 10 Km (SM)	Long-wavelength laser/ MM fiber/SM fiber
1000BASE-ZX	1000	70 Km	Extended-wavelength laser/SM fiber

NOTE: MM is multi-mode fiber and SM is single-mode fiber

Although the CCNA isn't a general networking exam, you may see questions about cabling specifications such as which ones use fiber.

Straight-through Cable

When the color of each wire matches on both sides of the cable, then it is known as a straight-through cable. In modern networks, hosts and devices can autosense the types of cables connected and communicate over the cable, whether it's straight-through or crossover (see next section). Historically, straight-through cables were used to connect dissimilar devices, so if you were connecting a PC to a hub/switch, you would have needed a straight-through cable.

Note the cable specifications of 568A and 568B in Figure 1.56 below. Specific colors are assigned to each of the eight internal wires, but these won't show in the printed book so do check them on Google. You won't be expected to name the colors in the CCNA exam.

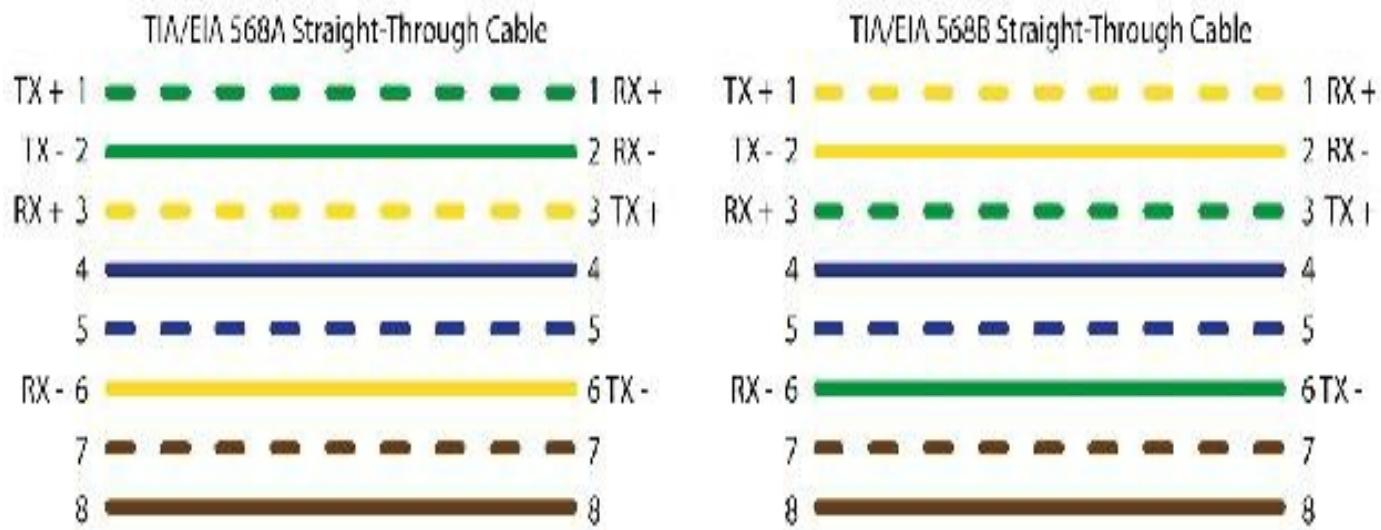


FIG 1.56 – Straight-through cable

568A Color Codes		568B Color Codes	
Pin #	Wire Color	Pin #	Wire Color
1	White/Green	1	White/Orange
2	Green	2	Orange
3	White/Orange	3	White/Green
4	Blue	4	Blue
5	White/Blue	5	White/Blue
6	Orange	6	Green
7	White/Brown	7	White/Brown
8	Brown	8	Brown

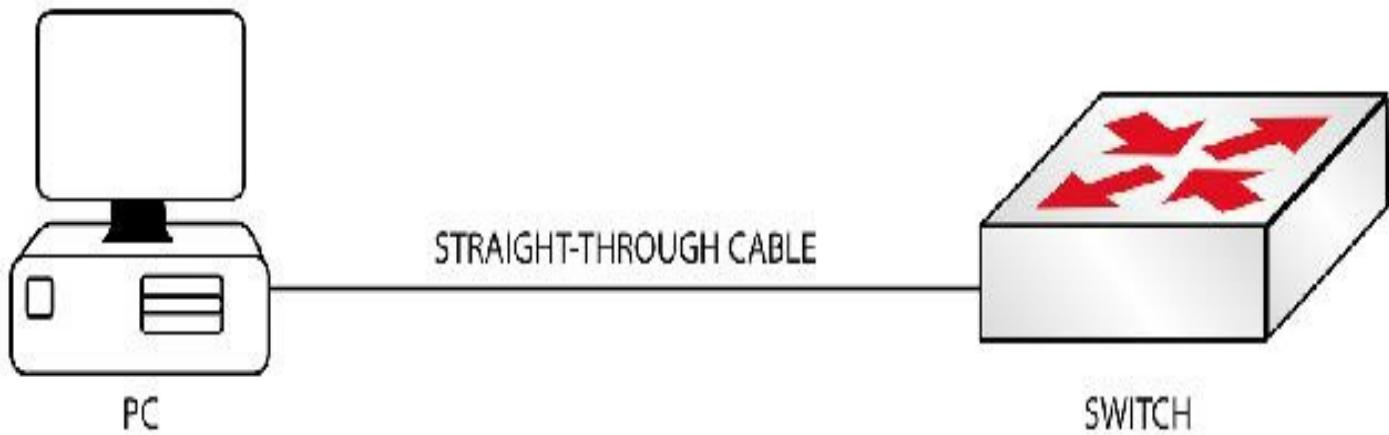


FIG 1.57 – Straight-through cable from a PC to a switch

You can compare cable ends by holding them together with the retaining clip facing down.

Crossover Cable

As mentioned earlier, the main use of a crossover cable is to connect two similar devices together, for example, connecting PCs together without having to buy a hub or switch. You will need to remember these for the CCNA exam. An easy way to remember this is that like-to-like devices use a crossover cable.

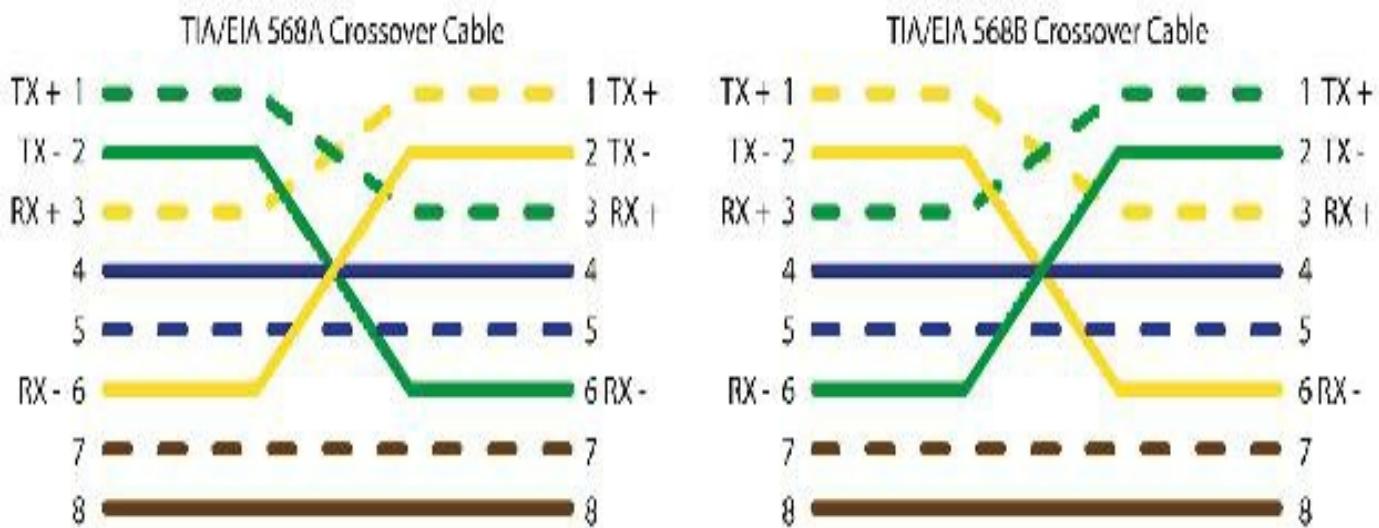


FIG 1.58 – Crossover cable

If you take the cable from pin 1 and put it into pin 3 at the other end, and then pin 2 into pin 6, you will have made a crossover cable connection.

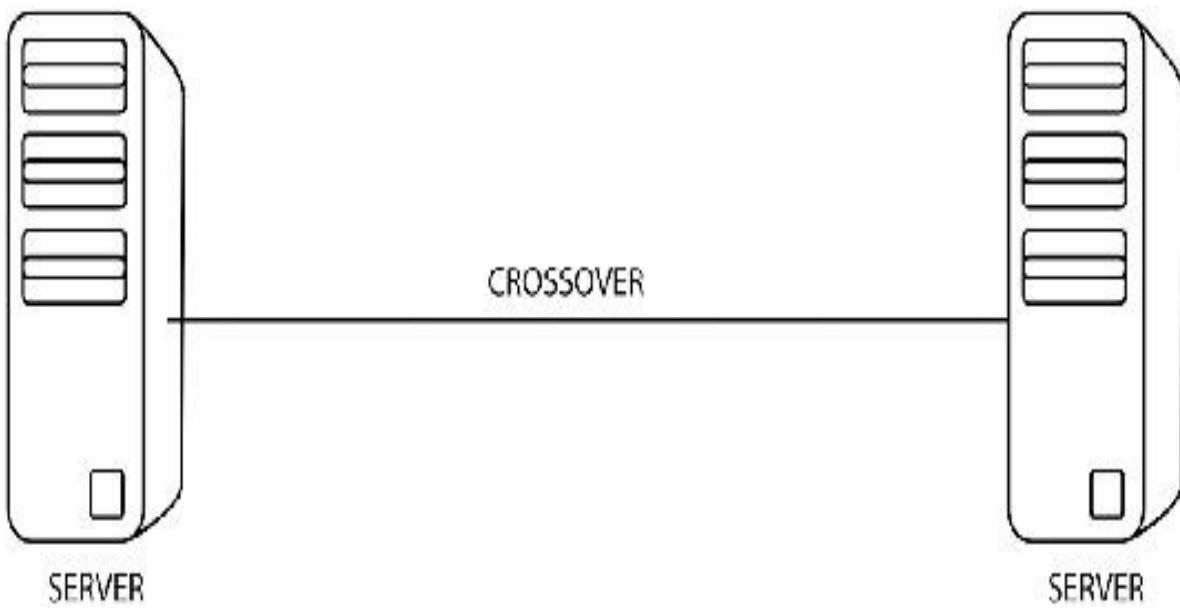
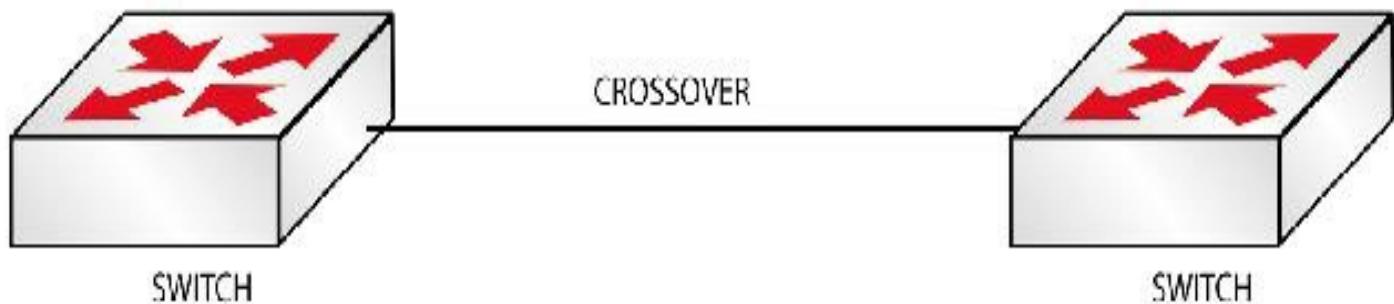
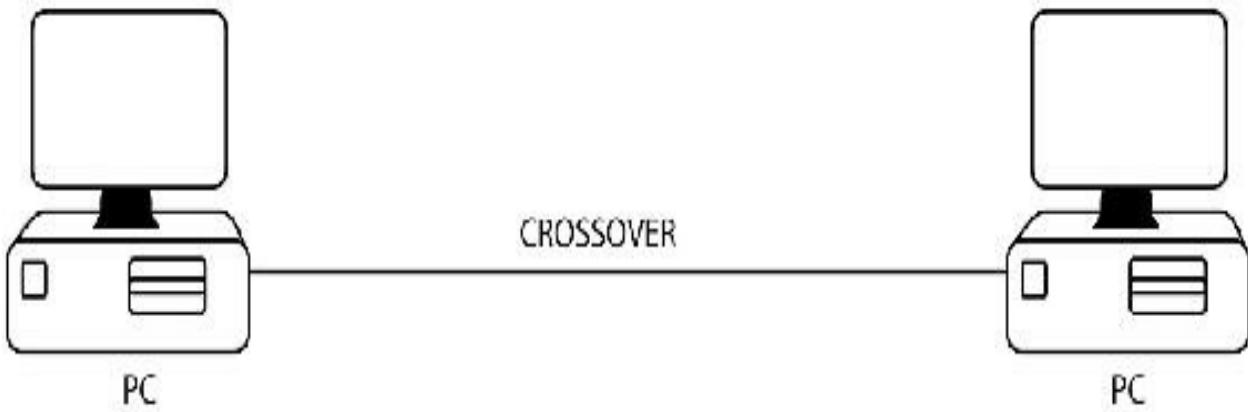


FIG 1.59 – Crossover cables in use

You should be familiar with the details of crossover and straight-through cables for the exam. The specifications are defined by the EIA/TIA. A straight-through cable has matching wires on both ends, whereas a crossover cable has T568A on one end and

T568B on the other. Typically, a network card in a PC transmits on pairs 1 and 2 and receives on pairs 3 and 6. This is why you would need a crossover cable if you were connecting two PCs together without using a switch.

Many Cisco switches support a feature known as Auto-MDIX, which stands for automatic medium-dependent interface crossover. This feature can detect an incorrectly attached cable and swap the pairs of wires on the cable used to transmit and receive. The Cisco 2960 Switch supports Auto-MDIX but it must have its speed and duplex settings configured for autonegotiation in order for the feature to work.

Rollover Cable

The last type of cable using UTP is referred to as a rollover (or sometimes called a flat or console) cable. This cable is only used to connect to a special port on routers and switches known as the console port. Connecting to the console port allows you to configure the networking equipment when you first install it or if you cannot reach it over the network.

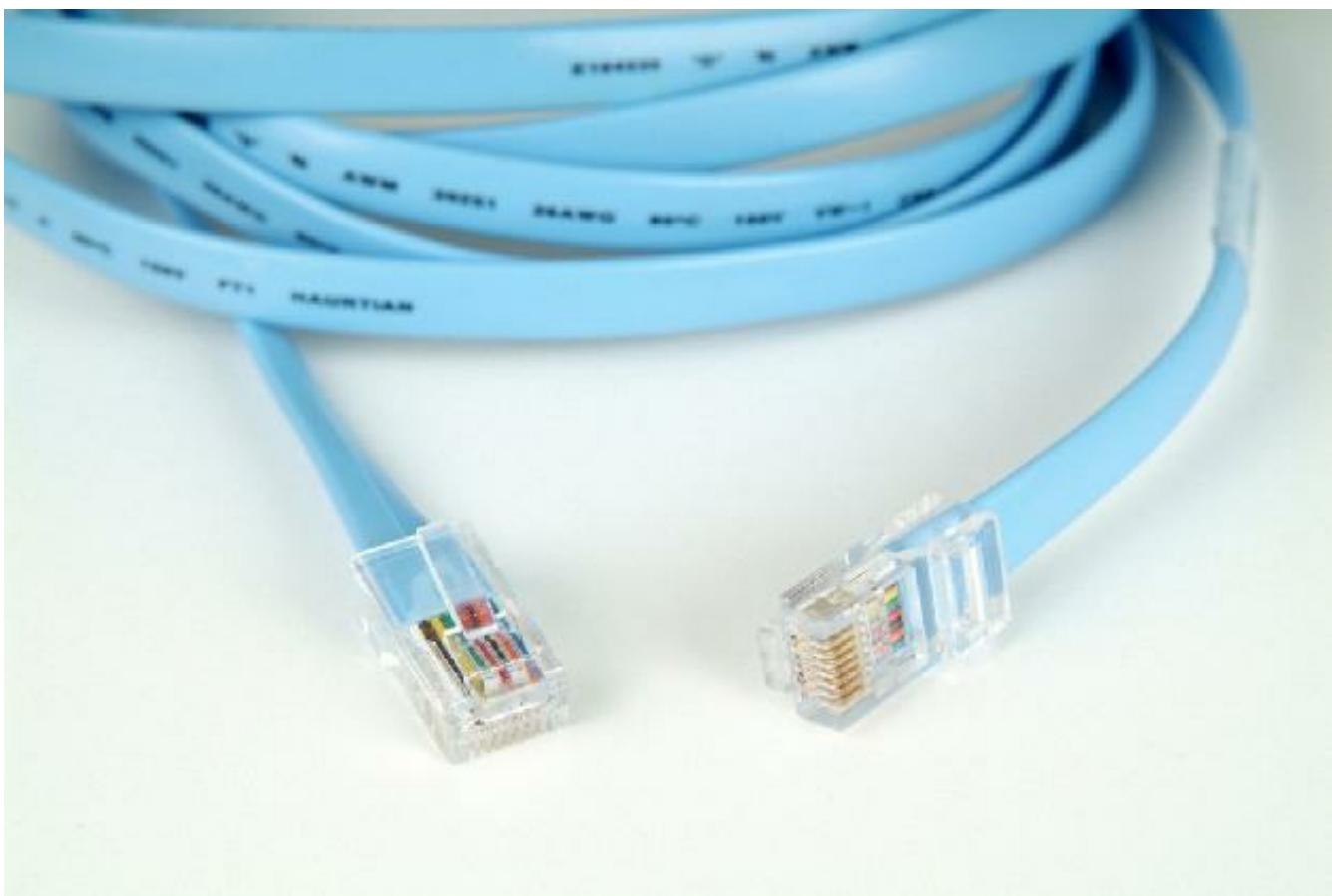


FIG 1.60 – Rollover cable

On a rollover cable, pin 1 on one side goes into pin 8 on the other, pin 2 goes into pin 7, and so on. As you hold the ends of the cables together with the retaining clips facing down, you can see that each wire is reversed.

You will be expected to know which type of cable to use between which devices in the exam (and in the real world). Table 1-7 below summarizes what you have learned so far:

Table 1-7: Which cable to use?

	PC/Server	Hub	Switch	Router
PC/Server	Crossover	Straight-through	Straight-through	Crossover
Hub	Straight-through	Crossover	Crossover	Straight-through
Switch	Straight-through	Crossover	Crossover	Straight-through
Router	Crossover	Straight-through	Straight-through	Crossover

WAN Cabling

There are many types of WAN services available, including ISDN, Frame Relay, ATM, ADSL, and others. Each type of service can have different connector types associated with it.

Most Cisco routers use a 60-pin D-shaped connector for the Serial interfaces. This usually connects to a slot containing a WAN interface card (WIC). The connector for the other side of the cable can vary, depending on who is providing you with the WAN connection and what type of service you have asked for.

It's important to discuss with your Cisco reseller what your requirements are and what type of presentation you have from your service provider to ensure that you are buying the correct modules and cables for your WAN connection.

You can see in Figure 1.61 below that the cable end has “DTE” printed on it. This indicates the Data Terminal Equipment end that connects to the Serial interface on your router. If the end had “DCE” printed on it, this would indicate that it was the Data Communication Equipment end and you would need to add a clock rate to this interface in order for it to become operational.

You can buy cheaper imported cables (and Serial cards) from China but you will find that non-Cisco products tend to have a high failure rate. I found this out at my cost when I started running my Cisco classroom training courses.



FIG 1.61 – DTE Serial cable end

The DTE cable inserts into the WIC-1T card (see [Figure 1.62](#) below), which will allow you to configure Serial connections for your router, including PPP and Frame Relay (we will cover this later in this guide).



FIG 1.62 – WIC-1T

Router Interfaces and Connectors

There are many varieties of connectors that can be used with computers and networks. Cisco devices use different types of connectors depending on which interface is used and which device is connected to the router. Next, we will look at the most commonly used connector and interface types.

RJ-45

Also known as a registered jack, RJ-45 connectors look similar to the type of cable associated with telephone connectors; however, there is a small plastic tab on the bottom to prevent it from being pulled out of the interface. There is also a connector used in some countries that can be plugged into a phone jack; this is known as an RJ-11 connector. The RJ-45 connector has eight pins, while the RJ-11 has only four.

RJ-45 connectors are used for LAN connectivity, console connections, and AUX connections. Modern CAT 5e cable is used to allow speeds of around 1 Gbps to be achieved over LANs.

Aux Connectors

Short for auxiliary, the AUX port on a router is used to dial in from a remote location. Rather than drive to the router's location, a network administrator can connect remotely using a modem and then configure the router. AUX ports use an RJ-45 connector; they can also be used for dial backup to the router via a modem.

Console Connectors

When a router is first used, the only way to configure it is to connect to the console port using a rollover cable. The console port will also be used if there is ever a problem with the router and it cannot be reached over the network. The only way to perform any sort of disaster recovery is to connect to the router via the console port.

The rollover cable connects directly into the console port; the other end of the cable connects to your COM port on your PC or laptop. If your cable end is RJ-45, you will need an adapter to allow it to interface with your nine-pin COM port (we cover this in detail later).

WAN Connectors

It would be well worth you taking the time to visit Cisco.com to review some of the network products available, in particular, the interfaces and modules.



Data Center Switches



Unified Communications



Infrastructure Software
(Cisco ONE Software)



Video



Interfaces and Modules



Wireless

FIG 1.63 – Visit Cisco.com for product research

Part of your research will be to find out which IOS versions support which model of router and switch and which cards fit in which device.

Cisco offers a wide range of router and switch models to suit home workers, small offices, and enterprise networks and beyond. Most models come in several variations to suit your budget and requirements, much the same way as cars do. The 2900 series router, for example, is available in the following models: 2901, 2911, 2921, and 2951.

The front of the router houses the power socket, the on/off switch, and various status lights, depending on the model.



FIG 1.64 – Cisco 2900 Series router

The 2911 model (shown above) features four Enhanced High-Speed WAN interface cards (EHWICs) slots. The far right slot is numbered 0, followed by 1, 2, and 3 from right to left. You need to understand this because when you insert cards into these slots, you need to ensure that you are configuring the correct interface. You can Google “EHWIC” to see the specifications and IOS requirements; however, Cisco has stated that they support up to 800 Mbps bidirectionally.

The bottom left side of the router features a large blank slot called a service module slot. Service module slots are used for router modules that run specific services, such as voice, security, Power over Ethernet, and many others.

In Figure 1.65 below, on the right rear of the router you can see three Gigabit Ethernet ports, two USB interfaces, a console and an AUX port, and a USB Serial console port. These ports give you the option of connecting the router to the USB interface on your

laptop using an RJ-45 connector or a USB 5-pin mini Type-B connector. This particular router has a voice card inserted into slot 0.

Each of the ports has one or more LED ports to indicate an active interface. Check your documentation to establish the meaning of the LEDs and colors they might display.



FIG 1.65 – Different port options

Router Interfaces and Slots

Quite possibly one of the most frustrating concepts for new Cisco engineers is understanding the nomenclature for interfaces. Why on one device do you configure Serial 0/1 but on another Serial 0/0/0? And how do you know what is attached to your device? The format has actually changed over the years, possibly because none of us, including Cisco, could see where networking would be 10 years or more into the future when they first devised the nomenclature.

Earlier models of the Cisco router featured interfaces burned onto the motherboard. They were fixed and so the numbering was easy: Serial 0, Serial 1, and so on. Cisco then updated its models to include blank slots in which you could insert modules of your choice. For example, if you had slot 0 and put two Serial interfaces into this, your interface names would be Serial 0/0 and Serial 0/1, with the first number indicating the slot and the second number the Serial interface numbering.

Next came the numbering system of slot/subslot/port. This was adopted for all devices so that the interface numbering system was universal, even if your device didn't offer slots in this configuration.

Figure 1.66 below shows a Cisco 1841 router. It is end-of-life but it would make an

ideal budget router for a CCNA home lab. I removed the card from slot 0 and issued a show ip interface brief command, or sh ip int brief for short.



FIG 1.66 – Cisco 1841 router

Router#sh ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	unassigned	YES	unset	administratively down	down
Fa0/1	unassigned	YES	unset	administratively down	down
Vlan1	unassigned	YES	unset	administratively down	down

When I insert a two-port WAN card (WIC-2T) into slot 0 and a one-port WAN card (WIC-1T) into slot 1, I can see which numbers have been allocated. All the slots are set to 0 but the subslots are numbered 0 and 1, and in subslot 0 the Serial interfaces are 0 and 1. Some cards for routers and switches are hot-swappable, which means you can insert them into a live device, but unless you have confirmed such, presume that you must power down the device before inserting or removing any cards.

Router#sh ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	unassigned	YES	unset	administratively down	down
Fa0/1	unassigned	YES	unset	administratively down	down
Se0/0/0	unassigned	YES	unset	administratively down	down
Se0/0/1	unassigned	YES	unset	administratively down	down
Se0/1/0	unassigned	YES	unset	administratively down	down
Vlan1	unassigned	YES	unset	administratively down	down

Router#

To see what hardware you have installed on your router, you can issue the show diag command. Bear in mind that you must have the correct IOS to support the cards installed in order for them to show and be configurable. I've truncated the output below to save

space. Your output may differ slightly from mine due to the IOS and platform differences, but the important thing is to correlate your show ip interface brief command with your show diag output.

Router#show diag

WIC Slot 0:

Serial 2T (12in1)

Hardware revision 18.1 Board revision A0
Serial number 16777216 Part number 00-0000-00
Version Identifier FRU Part Number
Test history 0x0 RMA number 00-00-80
Connector type PCI

WIC Slot 1:

Serial 1T WAN daughter card

Hardware revision 2.1 Board revision A0
Serial number 16777216 Part number 00-0000-00
Version Identifier FRU Part Number
Test history 0x0 RMA number 00-00-80
Connector type PCI

show inventory is also a very useful command to verify the hardware installed.



Connecting to a Router

When you buy a router it normally comes with no configuration, so the network administrator must configure it from scratch according to the particular requirements of his or her network. Because there is no IP address configured on it, you won't be able to telnet to the router, so the only option left is to use the console port. You would also use

the console port if there was a fault on the router and you couldn't reach it via Telnet or if you needed to perform a password recovery.

The traditional way of connecting to a router or switch console port involved using a rollover cable that had an RJ-45 connector for the console port and a DB9 connector for the PC or laptop. DB9 interfaces were eventually phased out and replaced with a DB9-to-USB cable to connect the rollover cable to the PC. We will cover these connections below, as well as the modern USB console ports available on routers.

A terminal emulation program allows you to configure the command line interface. This was traditionally HyperTerminal and came bundled with Microsoft Windows until Windows Vista was released. HyperTerminal has now been discontinued, and beginning with Internet Explorer version 6.0, Telnet is disabled by default. You can still find it and enable it in Windows but far superior programs are available now, such as PuTTY.

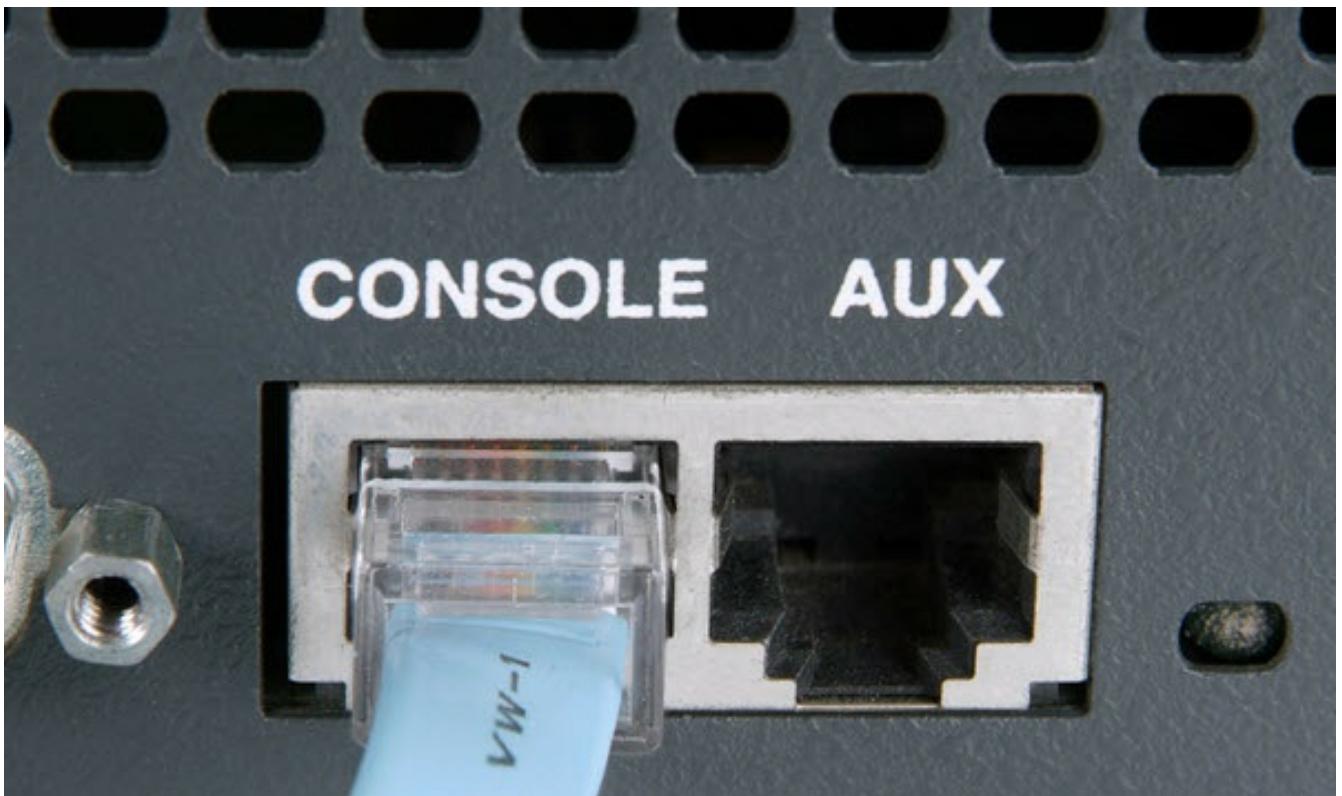


FIG 1.67 – Connecting one end of the rollover cable to the router's console port

Once you are connected to the console port on your router you can start the terminal emulator. These programs allow you to connect to network devices using Telnet or SSH, or via the console port. I'll demonstrate this action using PuTTY, which is free.

PuTTY actually defaults to the correct settings but just so you are aware, to connect to Cisco devices you need to have the terminal session settings below. You may well be asked about these in the exam so it's worth making a note of them:

- Bits per second – 9600

- Data bits – 8 is the default
- Parity – None is the default
- Stop bits – 1 is the default
- Flow control – must be set to None

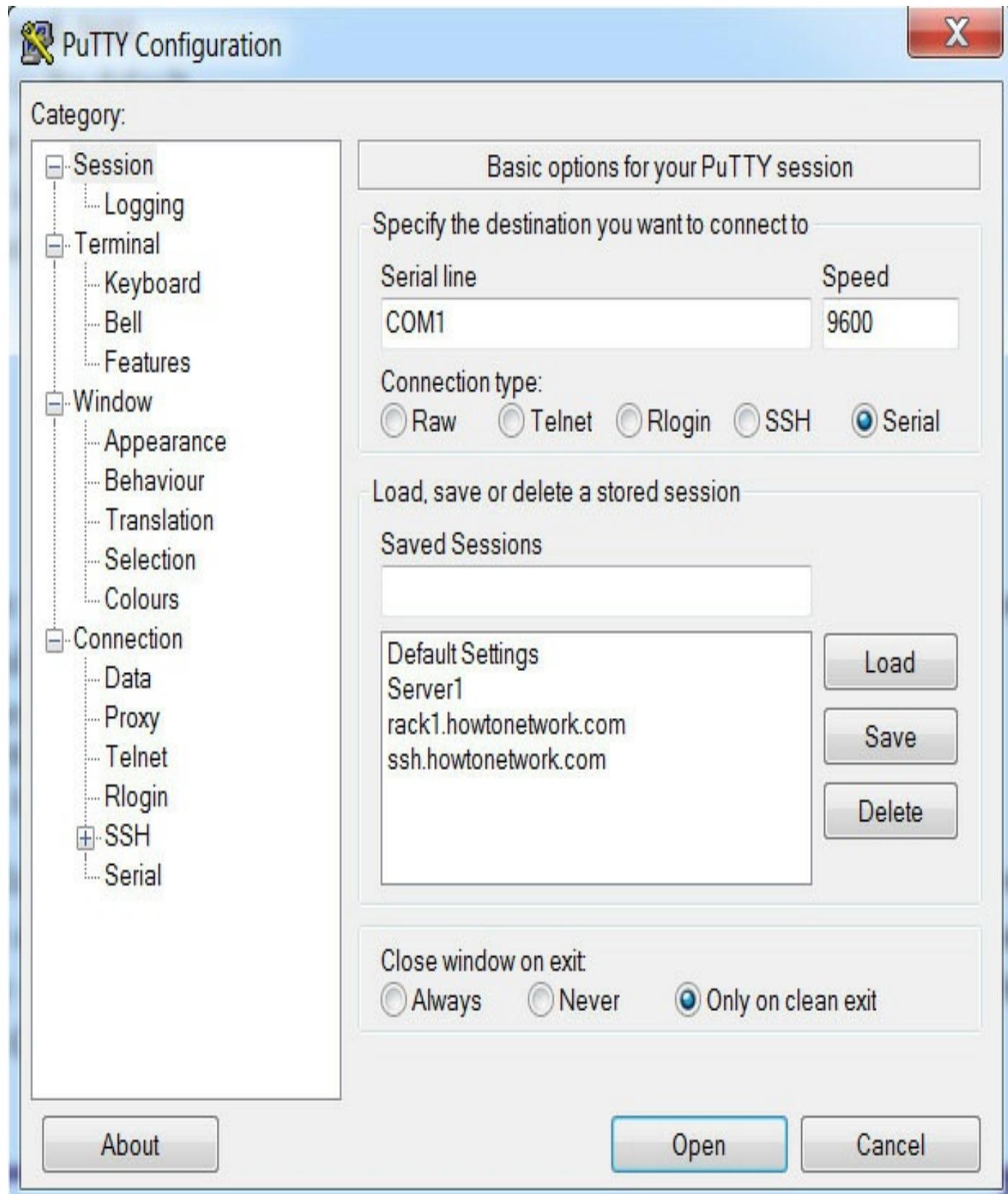


FIG 1.68 – Putty settings

Power on the router. It should then boot and you should see (after a few seconds) the boot-up text appear on the screen. If nothing appears then press the Enter key a few times; double-check that you have the correct COM port and that the cable is securely

attached at both ends.

System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)

Technical Support: <http://www.cisco.com/techsupport>

Copyright (c) 2010 by Cisco Systems, Inc.

When the router first begins to boot, it runs a diagnostic test known as power-on self-test (POST). If no problems are found, the router will then look for its operating system, which is stored in flash memory, also known as electrically erasable programmable read-only memory (EEPROM). We will cover router architecture in the ICND2 section of the study guide. The router also checks an internal setting called the configuration register, which tells the router or switch how to boot (this will also be covered later).

You should eventually see the Router> prompt. If the configuration register is set to 0x2142 or there is no startup configuration present, then you will be asked if you want to enter setup mode with the configuration dialogue. You should always type no because setup mode displays a series of questions issued by the router, after which it attempts to self-configure and this seldom gives you the desired results.

Cisco CISCO2911/K9 (rev 1.0) with 491520K/32768K bytes of memory.

Processor board ID FTX152400KS

3 Gigabit Ethernet interfaces

DRAM configuration is 64 bits wide with parity disabled.

255K bytes of non-volatile configuration memory.

249856K bytes of ATA System CompactFlash 0 (Read/Write)

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: no

You can download PuTTY for free at <http://www.putty.org>.

USB Console Connection

Most personal computers and laptops are built without a DB9 port. When this port is used it is usually allocated COM Port 1 (see CompTIA A+ for more information if you need it). It is still possible to make a console connection to the router using a special type of USB cable that connects to a console cable. The USB cable ends in a 9-pin connector and should come with driver software to allow it to be accessed via the Device Manager.



FIG 1.69 – USB to DB9 cable



FIG 1.70 – USB cable connects to the rollover cable

After the software is installed, you should go to Device Manager on your Windows PC and check which COM port the USB cable has been allocated. You can then go into PuTTY and choose that COM port.

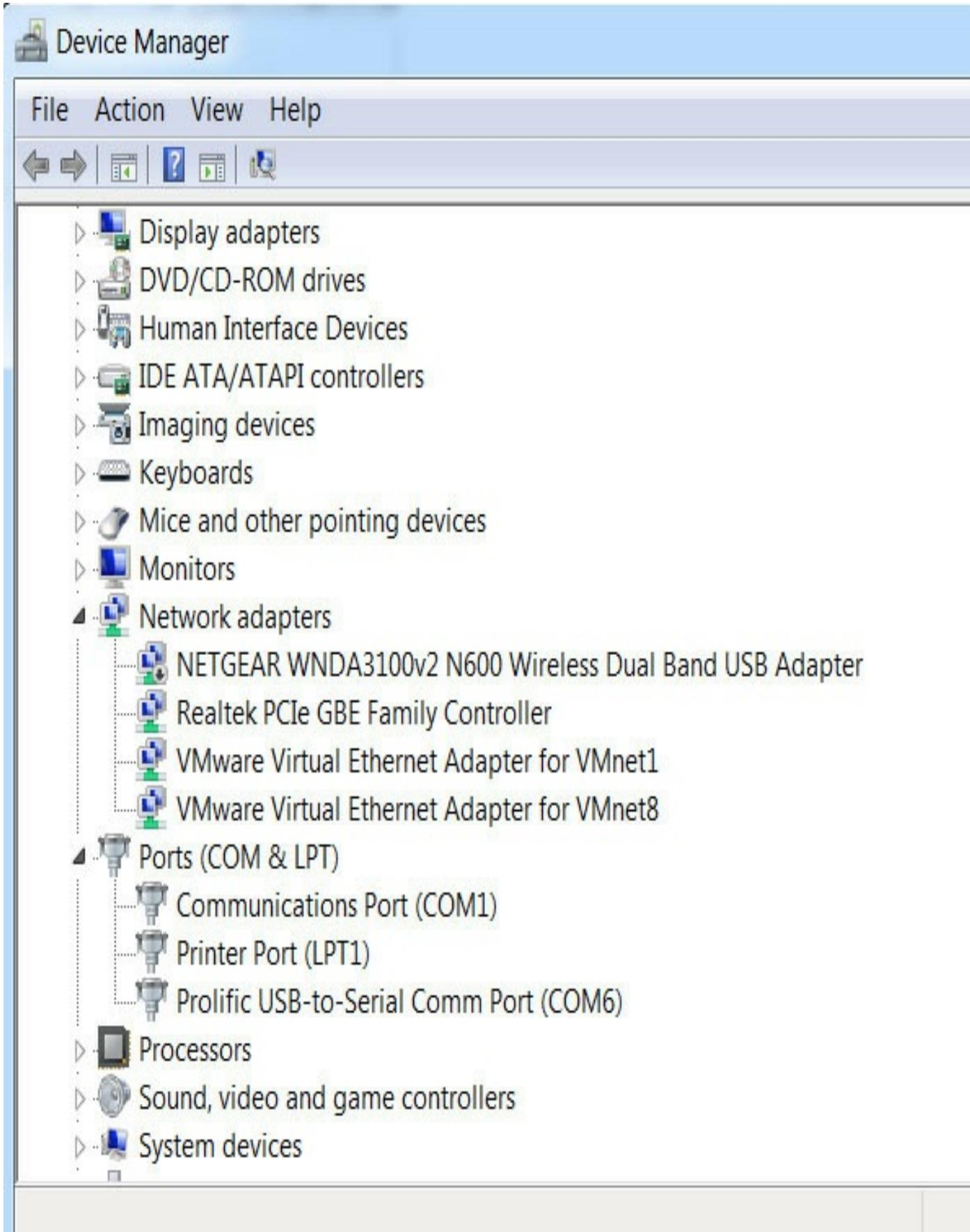


FIG 1.71 – The USB cable has been allocated to COM Port 6

On modern routers you can also connect to a mini-USB console port on your router.

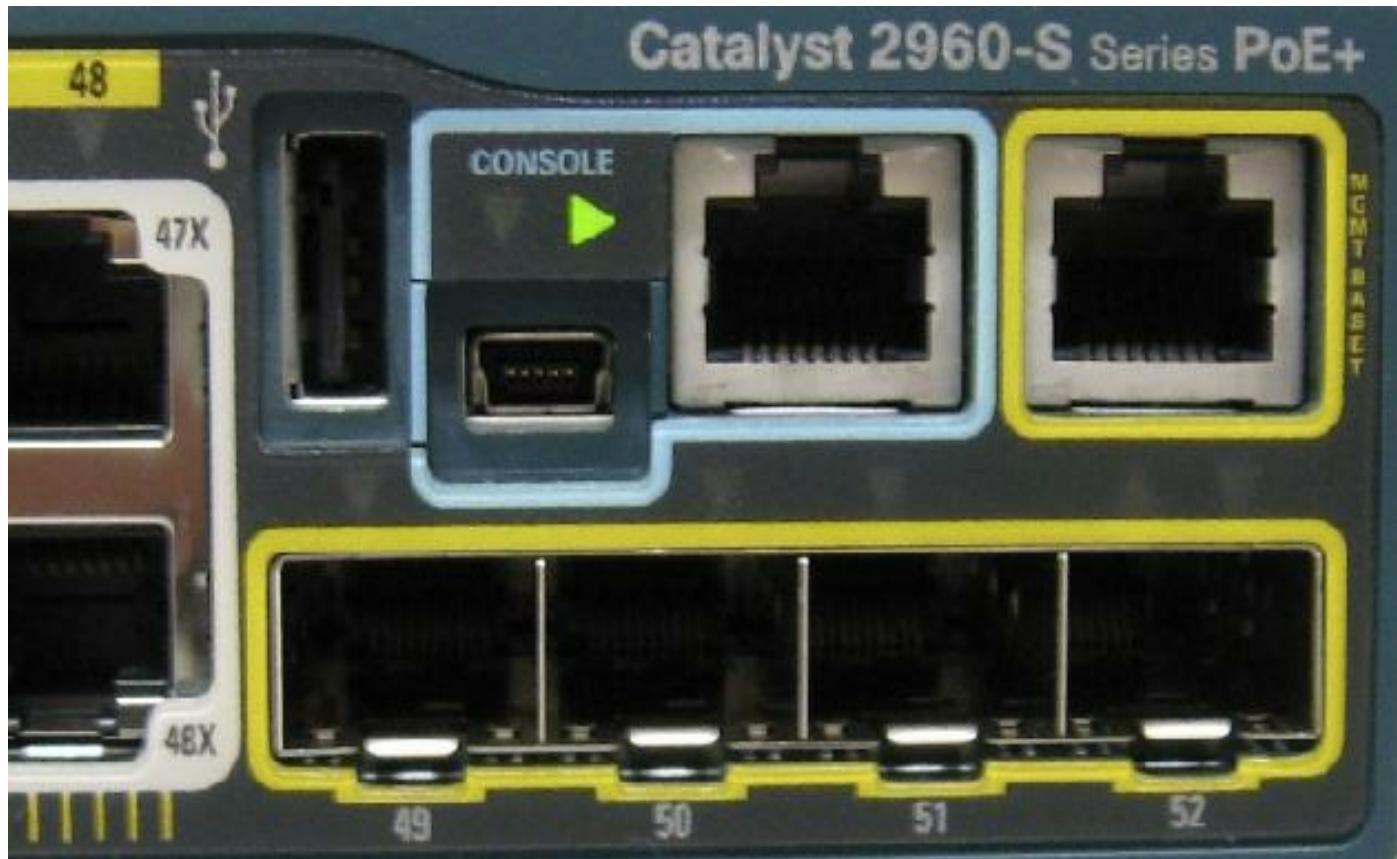


FIG 1.72 – Mini-USB console port

For this type of connection you will need the USB 5-pin mini Type-B to USB Type-A console cable.



FIG 1.73 – USB mini-cable

You will also need to download the correct USB driver from Cisco.com to match your operating system, for example, cisco_usbconsole.zip.

The screenshot shows a Cisco software download page. At the top, there's a navigation bar with the Cisco logo, a search bar, and links for 'My Profile', 'Logout', and 'Logout'. Below the navigation is a header for 'Download Software' with a sub-header for 'Catalyst 3560X-24P-S Switch'. The main content area displays a table for 'Release 3.1'. The table includes columns for 'File Information', 'Release Date', and 'Size'. Two files are listed: 'Cisco_usbconsole_driver_3_1.zip' and 'Cisco_usbconsole_driver_3_1.ipk'. To the right of the table are buttons for 'Download' and 'Add to cart'. On the left side, there's a sidebar with a search bar, a 'Search' button, and sections for 'Expand All' and 'Collapse All'. Under 'Latest', it shows '3.1' and 'All Releases'. Under '3.1', there's a link to '3.1'.

FIG 1.74 – Cisco console driver download page

For some reason, Cisco Systems seems to bury its driver software deep in its website, so you can either Google “Cisco USB console cable drivers 1911” if that is your router model, for example, or go to Cisco’s download page and drill down to all the software available for your model. Here is where I obtained it from Cisco’s download page (you will need to create a free Cisco.com account): Downloads > Home > Products > Routers > Branch Routers > 1900 Series Integrated Services > Routers 1921 Integrated Services Router > Software on Chassis > USB Console Software-3.1.

There is a timeout setting by default on the console connections. The command below may not be available for you depending on your model and IOS:

```
Router#configure terminal  
Router#(config)#line console 0  
Router#(config-line)#usb-inactivity-timeout 30
```

Or turn the timeout off:

```
Switch#(config-line)#no usb-inactivity-timeout
```

The range for timeout is 1 to 240 minutes. It may be worth checking your network security policies before configuring this value. If the timeout value is reached you will need to reseat (unplug and plug in) the USB cable.

Just as with the DB9-to-USB cable, the driver will create a COM port for you to reference in your terminal session.

Router Modes

In order to pass the CCNA exam, you will need to understand which prompt you should be at to configure various router parameters. You need to be in the correct mode to perform a specific router function. A common mistake novice network engineers make is trying to type the correct command but from the wrong mode.

For all intents and purposes, Cisco routers won’t prompt you to tell you that you are in the wrong mode. It just won’t accept your command until you are in the correct mode. This is part of what makes it hard to configure the equipment. If you find yourself stuck, you can always type ? and hit the Enter key to see the options available. The ? does work in the CCNA exam simulator, but if you have completed all the labs in this guide several times you won’t need to rely on it.

User Mode

When you log in to a router for the first time, the mode you are presented with is known as the user mode or user exec mode. There is a limited set of commands that can be run from this mode, which can be useful for looking at basic router elements. The default

name of the router is Router but this can be changed as you will see later.

Router>

Privileged Mode

Typing enable at the user prompt (or en for short) takes you into the next mode known as privileged or privileged exec mode. To get back to user mode you simply type disable, or to quit the session altogether type logout or exit (or ex for short).

Router>enable

Router#disable

Router>

The privileged mode gives you access to view the entire router configuration, inventory, and operation state of the router. In fact, it is a good idea to set a privileged mode password (also known as an enable password) to ensure that only authorized users get access to this mode.

Global Configuration Mode

To make configuration changes on the router, you have to be in global configuration mode. To get to global configuration mode, you simply type configure terminal (or conf t for short) at the privileged exec prompt. Alternatively, you can just type configure and the router will ask you which configuration mode you would like to enter. The default is terminal (the default option is shown inside brackets []). If you press Enter, the command inside the brackets will be accepted.

Router#config

Configuring from terminal, memory, or network [terminal]? i **Press Enter**

Router(config)#

Global configuration commands affect the entire router as opposed to more specific modes such as interface or routing modes, which only affect specific aspects of the router (as you will see).

Interface Configuration Mode

When you need to configure attributes specific to a particular part of the router, such as an interface, the router takes you to the specific configuration mode for that aspect. The interface configuration mode allows you to enter commands to modify attributes for individual router interfaces, such as Gigabit Ethernet, Serial, etc. On a new router, all of the interfaces will be shut down by default and no configuration will be present. You

will find out how to determine which interfaces you have available shortly.

```
Router>enable  
Router#config t  
Router(config)#interface Serial0/0/0  
Router(config-if)#
```

Line Configuration Mode

Line configuration mode is used to make changes to terminal and console lines on a router. These changes affect how a terminal of the router can be accessed to make configuration changes. You can make changes to the console, VTY (for Telnet and SSH), or AUX ports. You can also control who has access to the router via these ports, as well as set passwords or a security feature called access lists on the router, which will be covered in detail later.

```
Router#config t  
Router(config)#line console 0  
Router(config-line)#
```

You can also configure baud rates, exec levels, and a lot more in line configuration mode.

Router Configuration Mode

Just like for interface and line configuration modes, the router configuration mode allows you to configure parameters that are specific to routing protocols.

```
Router#config t  
Router(config)#router eigrp 10  
Router(config-router)#
```

Reloading the Router

You can consider reloading the router to be the same thing as a reboot. You will not have to do this due to any configuration changes because these take effect as soon as they are applied. You might need to reload the router to recover it in the event that you need to perform password recovery or if you have replaced the IOS and it needs the router to be loaded.

When the router reloads it will automatically load configurations from NVRAM, so any changes you have made to the running configuration (DRAM) won't take effect unless you copy the running configuration to NVRAM with the copy running-configuration

startup-configuration command (or copy run start for short). If you forget to do this, the router will prompt you to save any changes to the configuration.

R1#reload

System configuration has been modified. Save? [yes/no]:

You can also set a timer so that the router reloads in the specified number of minutes or hours.

R1#reload in ?

Delay before reload (mm or hh:mm)

If you want to prevent this from happening you can type reload cancel at the privileged prompt (not in configuration mode). You will usually want to reload the router and NOT save any configuration commands when using this guide because you want to become adept at configuring the router from scratch.

Abbreviating the Commands

The router will permit you to type just the first few letters of the command if you would rather abbreviate it. The proviso is that it must be the only command available that starts with those letters in that particular mode.

If you are at the Router# command, you can type the command conf t instead of configure terminal because there is no other command that starts with the letters conf. You should get used to abbreviating commands to save time, but bear in mind that some abbreviations may not work in the exam since you are configuring an emulator, not an actual router or switch.

Configuring a Router

Router configuration is accomplished via the command line interface (CLI).

Remembering all the commands might seem a bit tedious but there is some help in the form of a question mark. If you type a ? at the router prompt, you will be presented with a list of all the commands available:

Router#?

Exec commands:

access-enable Create a temporary Access-List entry

access-profile Apply user-profile to interface

access-template Create a temporary Access-List entry

alps ALPS exec commands

archive manage archive files

bfe	For manual emergency modes setting
cd	Change current directory
clear	Reset functions
clock	Manage the system clock
cns	CNS subsystem
configure	Enter configuration mode
connect	Open a terminal connection
copy	Copy from one file to another
debug	Debugging functions (see also “undebbug”)
delete	Delete a file
dir	List files on a
disable	Turn off privileged commands
disconnect	Disconnect an existing network connection
enable	Turn on privileged commands
erase	Erase a
exit	Exit from the EXEC
help	Description of the interactive help system
-- More --	

If there is too much information to display on the screen, you will see the -- More -- tag. If you want to display the next line, press Enter, and if you want to display the next page, press the space bar. If you want to exit the list of commands, hold down the CTRL+Z keys (the Control (Ctrl) and the Z key together) or press any other letter to get back to the router prompt.

There are many thousands of IOS commands available, but in your day-to-day role as a network engineer you will use only a small percentage of these. Also, if you need help completing a command, you can use the question mark to display a list of the options available.

```
Router#cl?  
clear clock
```

If you type out enough characters of a command, such that there is only one possible completion of that syntax, you can automatically complete the command using the Tab key (just like in UNIX).

```
Router#copy ru i Press the Tab key here  
Router#copy running-config
```

Loopback Interfaces

Loopback interfaces are not normally covered in the CCNA syllabus (apart from the context of OSPF Router ID, which will be discussed later) but they are very useful, both in the real world and for practice labs. A Loopback interface is a virtual or logical interface that exists only in software (it can be configured but it does not physically exist). You can assign an IP address and even test reachability to a Loopback address using pings. This is very useful when simulating networks in a lab.

An advantage of using Loopback interfaces is that they always remain UP/UP (physically up and logically up), unless an administrator shuts them down. They are not affected by cabling or clocking issues like Ethernet and Serial interfaces.

```
Router#config t
Router(config)#interface Loopback0
Router(config-if)#ip address 192.168.20.1 255.255.255.0
Router(config-if)#^z i Press Ctrl+Z
Router#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Loopback0	192.168.20.1	YES	manual	up	up

Your output for this command will show all of the interfaces available on your router. If you need to, you can shut down a Loopback interface with the shutdown command in interface configuration mode. Loopback interfaces have to be given a valid IP address. You can then use them for routing protocols or testing your router to ensure that it is permitting certain traffic. We will be using Loopback interfaces a lot throughout this manual.

Editing Commands

It is possible to navigate your way around a line you have typed rather than deleting the whole line. Certain keystrokes will move the cursor to various places around the line. These keystrokes are quite similar to the UNIX keystrokes. Please try out these shortcuts when you are configuring labs.

Table 1-8: Keyboard shortcuts

Keystroke	Meaning
Ctrl+A	Moves to the beginning of the command line
Ctrl+E	Moves to the end of the line
Ctrl+B	Moves back one character
Ctrl+F	Moves forward one character

Esc+F	Moves forward one word
Esc+B	Moves back one word
Ctrl+P or up arrow	Recalls the previous command
Ctrl+N or down arrow	Recalls the next command
Ctrl+U	Deletes a line
Ctrl+W	Deletes a word
Tab	Finishes typing a command for you
Show history	Shows the last 10 commands entered by default
Backspace	Deletes a single character

Mini-lab – Putting an IP Address on an Interface

Router interfaces must be assigned an IP address for them to be able to communicate with other devices that are connected to them. To assign an IP address to an interface, first, you need to go into the interface configuration mode. You have already seen how to discover which interfaces you have available with the show ip interface brief command. Mine says Serial 0/0/0 so that's the one I'll be configuring; yours may be different.

Router>enable **i Takes you from user to privileged mode**

Router#config t **i From privileged to config mode**

Router(config)#interface Serial0/0/0 **i And then into interface config mode**

Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shutdown **i Open the interface for traffic**

Router(config-if)#exit **i You could also hold down Ctrl and Z keys to exit**

Router(config)#exit

Router#

A description can also be added to the interface:

Router(config)#interface Serial0/0/0

Router(config-if)#description To_Headquarters

Router(config-if)#^Z **i Press Ctrl+Z to exit**

Router#show interface Serial 0/0/0

Serial0/0/0 is up, line protocol is up

Hardware is HD64570

Description: To_Headquarters

Internet address is 192.168.1.1/24

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

```
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, Loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:03, output hang never
[output truncated]
```

[END OF MINI-LAB]

Show Commands

You can verify the settings on the router by simply using the show X command in privileged mode, with X being the next command:

```
Router#show ?
access-expression List access expression
access-lists    List access lists
accounting     Accounting data for active sessions
adjacency      Adjacent nodes
aliases        Display alias commands
alps           Alps information
apollo         Apollo network information
appletalk       AppleTalk information
--More--
```

Some of the more common show commands and their meanings are listed below in Table 1-9. Please do try them out and note what information they provide.

Table 1-9: Common show commands

Show Command	Result
show running-configuration	Shows configuration in DRAM
show startup-configuration	Shows configuration in NVRAM
show flash:	Shows which IOS is in flash
show ip interface brief	Shows brief summary of all interfaces
show interface Serial0	Shows Serial interface statistics
show history	Shows last 10 commands entered

Try some out for yourself and type a ? at the end to see if there are more options available.

Examples

Router#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	administratively down	down
FastEthernet0/1	unassigned	YES	unset	up	up
Loopback0	172.16.1.1	YES	manual	up	up
Serial0/0	192.168.1.1	YES	unset	administratively down	down
Serial0/2	unassigned	YES	unset	administratively down	down

The method column indicates how the address was assigned. It can be unset, manual, NVRAM, IPCP, or via DHCP.

Routers can recall commands that were previously entered at the router prompt—the default number of commands is 10. The commands can be recalled by using the up arrow. Using this feature can save a lot of time and effort in reentering a long line. The show history command shows the buffer of the last 10 commands issued on the router:

```
Router#show history
show ip interface brief
show history
show version
show flash:
conf t
show access-lists
show process cpu
show buffers
show logging
show memory
```

You can increase the history buffer with the terminal history size command:

```
Router#terminal history ?
size Set history buffer size
[cr]
Router#terminal history size ?
[0-256] Size of history buffer
Router#terminal history size 20
```

The show commands are very powerful and are an essential part of your troubleshooting

tool bag. Ninety percent of all troubleshooting can be done without ever looking at the running configuration of the router. Learn the show commands well.

Debug Commands

A large part of the new-style CCNA exam covers troubleshooting skills. This can be thought of as your theoretical knowledge as well as your hands-on experience with the Cisco IOS. Part of your troubleshooting will be knowing which show commands to use in which circumstances. A telltale sign of a novice Cisco engineer is resorting to the show run command, which is rarely used by an experienced network engineer.

Some issues cannot be investigated using the show commands because you need to see a real-time exchange of information between network devices. Learning how to use a network sniffer is a hugely important part of your role as a network engineer, and although it is outside the CCNA syllabus, we do include some sniffer outputs to explain learning points.

But before that, you will need to understand the relevant debug commands to issue to either confirm that the network is working as expected or to troubleshoot an issue. We will cover these as we progress through the book; however, it's important to understand a few points:

- Debug output will not show on your screen if you have telnetted to a router or switch. You need to add the terminal monitor command to see it.
- Some types of debugs will generate a huge amount of output, which will quickly overload your router CPU and cause it to hang or crash. I've known of network engineers being sacked after doing this. The debug ip packet command is a major cause of router crashes.
- Issue debug commands with extreme caution on a live network and always check with an experienced engineer first. There are ways to restrict the output somewhat.
- You can turn off the debug command by typing it out again but with the word no in front, or you can turn off all debugs with the undbg all command (or un all for short).
- Debug commands will probably not work in the CCNA exam because you are working on an emulator, which doesn't actually produce live traffic.

Here is a debug command that you might use during an OSPF lab. Note that the router tells you that the debug is active and then begins printing the output on the screen. The output will differ from debug to debug; however, it should include a timestamp and date.

```
R1#debug ip ospf hello
```

OSPF Hello events debugging is on

*Mar 1 02:09:12.719: OSPF: Send Hello to 224.0.0.5 area 0 on FastEthernet0/0 from 192.168.1.1

You can disable a debug by typing it out again but with the word no in front:

R1#no debug ip ospf hello

OSPF Hello events debugging is off

If you have multiple debugs or just want to use a shorter command, you can use undebug all (or un all for short).

R1#un all

All possible debugging has been turned off

A major bugbear for Cisco engineers is router output appearing as you type commands. Although you can continue typing, seeing the output appear on the screen where you are typing can cause you to lose track of where you were in the configuration.

The logging synchronous command is very useful if you want to prevent logging information from appearing while you are entering commands on the router from a console connection. If this command is not on and a console message appears half-way through typing a command, you can simply hold down the Ctrl+L or Ctrl+R keys or hit the up arrow to redisplay the line you were typing.

RouterA(config)#line console 0

RouterA(config-line)#logging synchronous

You may find that this command is on by default depending on your IOS release. Type show run to find out which commands are turned on by default.

You will use more debug commands in the lab sections of this book. The commands make far more sense when you can see them working on a real network.

Pipes

As you've seen above, the command line interface for configuring Cisco devices using Cisco IOS has some features used by the UNIX command line. A sure sign of a confident and experienced Cisco engineer is familiarity with keyboard and command shortcuts. Part of getting quick answers to router show commands is the use of pipes, which help you cut out all extraneous output.

A pipe (shown as | on the keyboard) can be used on the Cisco command line to give you some granularity when searching for certain commands or entries in the router's configuration. Using pipes can save you time and effort and possessing this knowledge

shows professionalism in the field.

You can use several commands with pipes, such as:

```
show [command] [begin | include | exclude] [regular expression]
```

Example

```
Router#show run ?
```

```
  interface Show interface configuration  
  |      Output modifiers  
[cr]
```

```
Router#show run | ?
```

```
  begin   Begin with the line that matches  
  exclude Exclude lines that match  
  include  Include lines that match
```

```
Router#show run | include ?
```

```
  LINE Regular Expression
```

```
Router#show run | include login
```

```
  aaa authentication login default group tacacs+ line
```

```
  timeout login response 120
```

```
  timeout login response 120
```

```
Router#show run | begin interface
```

```
  interface FastEthernet0/0
```

```
  no ip address
```

```
  shutdown
```

```
!
```

```
  interface Serial0/0
```

```
  no ip address
```

```
  shutdown
```

```
!
```

```
  interface Serial1/0
```

```
  no ip address
```

```
  shutdown
```

```
!
```

```
  ip classless
```

```
  no ip http server
```

```
!
```

```
  line con 0
```

```
line aux 0
line vty 0 4
!
end
```

Don't expect any of the commands above to work on Packet Tracer.

The Configuration Register

How does the router know where to find the configuration file when it boots up? The router checks the config-register field upon booting to determine which booting option to use. You can see which option has been set when you issue the show version command. The output below is truncated to save space:

```
Router#sh ver
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version
15.1(4)M4, RELEASE SOFTWARE (fc2)
ROM: System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
cisco2911 uptime is 1 minutes, 35 seconds
System returned to ROM by power-on
System image file is "flash0:c2900-universalk9-mz.SPA.151-1.M4.bin"
```

Configuration register is **0x2102**

When using routers to practice labs, you do not want the configuration changes you make to stay there every time you boot up the routers. In order to be able to pass the practical tests in the lab you must practice over and over again. If you change the configuration register setting you can prevent the router from looking at the startup configuration file when it boots. This will boot the router with no configuration file, giving you a blank configuration to begin working on.

Having the router configuration register set to 0x2102 tells it to look at the startup configuration file when it boots, which pulls it into NVRAM. Changing it to 0x2142 tells the router to ignore it so that it will boot with a blank configuration. This setting has to be entered if you ever forget the router password because the password sits in the startup configuration. The router will boot without any of the configurations, including any passwords.

The configuration register setting isn't part of the startup configuration or running configuration so there is no need to save the change. However, you will be prompted to save it because you went to configuration mode and out again.

Changing the configuration register actually belongs in the ICND2 Router Architecture section; however, you need to change the default router settings for all the labs (if you are using live equipment), which is why we are covering it now.

Mini-lab – Changing the Configuration Register

This mini-lab was performed on a Cisco 2911 model. If you are using a different model, you may need to follow slightly different steps. Search Google for “Cisco changing configuration register” to check for your specific model.

In user exec mode go into privileged exec mode:

```
Router>enable  
Router#
```

Check the configuration register settings (your output will be slightly different from the one below if you are using a different model of router and IOS). You will need to press the space bar once to get to the bottom of the show version output:

```
Router#show version
```

Cisco CISCO2911/K9 (rev 1.0) with 491520K/32768K bytes of memory.

Processor board ID FTX152400KS

3 Gigabit Ethernet interfaces

DRAM configuration is 64 bits wide with parity disabled.

255K bytes of non-volatile configuration memory.

249856K bytes of ATA System CompactFlash 0 (Read/Write)

Configuration register is 0x2102

Enter configuration mode:

```
Router#configure terminal  
Router(config)#
```

Change the configuration register setting:

```
Router(config)#config-register 0x2142  
Router(config)#exit
```

Check the configuration register setting:

```
Router#show version  
Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.  
Processor board ID FTX152400KS
```

3 Gigabit Ethernet interfaces

DRAM configuration is 64 bits wide with parity disabled.

255K bytes of non-volatile configuration memory.

249856K bytes of ATA System CompactFlash 0 (Read/Write)

Configuration register is 0x2102 (will be 0x2142 at next reload)

Reload the router:

Router#reload

System configuration has been modified.

Save? [yes/no]: n **i Enter no or n here**

Proceed with reload? [confirm] **i Press Enter here**

00:14:47: %SYS-5-RELOAD: Reload requested

System Bootstrap, Version 11.0(10c)XB2, PLATFORM SPECIFIC RELEASE SOFTWARE (fc1)

The reload may take a few minutes so be patient. Press Enter every few seconds to see if you have a prompt.

You will be asked if you want to enter configuration dialog. Enter n for no.

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: n

The router has no configuration; hence, the Router> prompt. Check the configuration register setting now (you will need to use the space bar again):

Router>enable

Router#show version

Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.

Processor board ID FTX152400KS

3 Gigabit Ethernet interfaces

DRAM configuration is 64 bits wide with parity disabled.

255K bytes of non-volatile configuration memory.

249856K bytes of ATA System CompactFlash 0 (Read/Write)

Configuration register is 0x2142

If you reload the router now it will continue to skip the startup configuration file until you reset the configuration register back to 0x2102.

[END OF MINI-LAB]

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 1 exam.

Chapter 1 Labs

Lab 1: Basic Lab – Router Modes and Commands

There is no physical topology for this lab. Just use any Cisco router.

Purpose

Any person new to configuring Cisco routers needs to feel comfortable navigating around the various router features and modes. This lab will be a great icebreaker for a budding CCNA. We covered how to use a console cable with a router earlier in this chapter, so follow those steps before you start.

Your output and interfaces may differ from mine if you are using a different model and IOS release.

Lab Objectives

1. Connect to the console port.
2. Enter privileged mode (enable mode).
3. Enter global configuration mode (config mode).
4. Enter the interface configuration mode.
5. Enter the routing configuration mode (router mode).
6. Exit to privileged mode.
7. Execute some useful commands.
8. Exit to user exec mode.
9. Examine interface statistics.
10. Change router hostname.

Lab Walk-through

1. When connecting to the console of the router, you will typically see the following message. Always type no if asked if you want to enter System Configuration Dialog:

--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Router con0 is now available

- As instructed, you simply need to press the Return (Enter) key and enter the first mode of the router, user exec mode:

Router>

- Now you are in user exec mode. Next, enter privileged mode, or enable mode as it is more commonly known. To do this type:

```
Router>enable
```

You will now be presented with a new prompt that has a hash/pound (#) instead of the greater than (>) sign:

```
Router>enable
```

```
Router#
```

Enable mode is used to perform all the show and debug commands, which will be explained later in the lab.

- The next mode to enter is global configuration mode, or config mode as it is also known. To enter config mode type:

```
Router#config terminal
```

As you will soon learn, all the commands in the Cisco IOS (operating system) can be abbreviated; for example, you could have entered:

```
Router#conf t
```

If you just type config and press Enter you will receive the following output:

```
Router#config
```

```
Configuring from terminal, memory, or network [terminal]?
```

As you will see, terminal is the default (indicated by the square brackets []), so you can simply press Enter to go into config or privileged mode.

- Once in config mode you will be prompted with the following message:

```
Router#config terminal
```

Enter configuration commands, one per line. End with Ctrl+Z.

```
Router(config)#
```

This is telling you that when you have finished in config mode, type Ctrl+Z to exit (while holding the Ctrl key down, press the Z key).

Once in config mode, you will notice that the prompt has changed again, this time from Router# to Router(config)#, indicating that you are in config mode. There are sublayers to config mode, but we are only interested in two of them, the first being interface configuration mode. First, you need to know which interfaces you have

available:

```
Router#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	unassigned	YES	unset	administratively down	down
Fa0/1	unassigned	YES	unset	administratively down	down

I have F0/0 and F0/1 available on my router. Your options may differ.

Router(config)#interface FastEthernet0/0 i **Or use Loopback 0 if your router does not have an Ethernet interface**

```
Router(config-if)#
```

If you are not sure which interfaces you have on your router, enter the show ip interface brief command at the Router# prompt. If you do not have an Ethernet interface, replace the command above with interface Loopback 0.

You will see that the prompt has changed again: the (config-if) tells you that you are now in interface configuration mode. If you aren't sure what to type then enter a ? at the end of what you are typing.

```
Router(config)#interface ?
Dot11Radio      Dot11 interface
Ethernet        IEEE 802.3
FastEthernet    FastEthernet IEEE 802.3
GigabitEthernet GigabitEthernet IEEE 802.3z
Loopback        Loopback interface
Serial          Serial
Tunnel          Tunnel interface
Virtual-Template Virtual Template interface
Vlan            Catalyst Vlans
range           interface range command
Router(config)#interface FastEthernet0/0
Router(config-if)#
```

5. Another sublayer of config mode is the router configuration mode:

```
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#

```

When you exit from interface configuration mode and type router rip, you enter

router configuration mode. You can see that the prompt has changed again to reflect this.

6. To exit config mode and go back to privileged (enable) mode, you simply need to type:

```
Router(config-router)#^Z i Hold down the Ctrl and Z keys (together)
Router#
```

When you do this, you will get the following message displayed after a few seconds:

```
%SYS-5-CONFIG_I: Configured from console by console
Router#
```

7. Now that you are back in enable mode, you can use some useful show commands. The common ones to use are shown below:

```
Router#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	unassigned	YES	unset	administratively	down down
Fa0/1	unassigned	YES	unset	administratively	down down

```
Router#
```

The benefit of this command is that it shows the status and IP addresses of all interfaces in a table. Do not worry if your output is different from the one above.

The next command that is useful is show running-configuration, which will display the current configuration (yours may look different from the one below). The output will be cut short so that you can see it all on your monitor. You can press the Enter key to go through it line by line or press the space bar to scroll up a page at a time:

```
Router#show running-config
```

Or:

```
Router#show run i Abbreviated command
Building configuration...
Current configuration : 489 bytes
!
version 15.1
```

```
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
spanning-tree mode pvst
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
shutdown
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Vlan1
no ip address
shutdown
!
ip classless
!
line con 0
!
line aux 0
!
line vty 0 4
login
!
End

Type show version (or show ver for short):
Router#show ver
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version
15.1(4)M4, RELEASE SOFTWARE (fc2)
```

Technical Support: <http://www.cisco.com/techsupport>
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team
ROM: System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
cisco2911 uptime is 1 minutes, 35 seconds
System returned to ROM by power-on
System image file is "flash0:c2900-universalk9-mz.SPA.151-1.M4.bin"
Last reload type: Normal Reload
If you require further assistance please contact us by sending email to
export@cisco.com.
Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
3 GigabitEthernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)
Configuration register is 0x2102
To exit to the user exec mode, you simply need to type disable (opposite of
enable) or exit:
Router#disable
Router>enable
Router#

8. You can also examine the interface statistics with the show interface x command:

Router#show interface f0/0
FastEthernet0/0 is administratively down, line protocol is down (disabled)
Hardware is Lance, address is 0060.5cd9.8001 (bia 0060.5cd9.8001)
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, Loopback not set
ARP type: ARPA, ARP Timeout 04:00:00,
Last input 00:00:08, output 00:00:05, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0

Queueing strategy: fifo

Output queue :0/40 (size/max)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

0 packets input, 0 bytes, 0 no buffer

Received 0 broadcasts, 0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

0 input packets with dribble condition detected

0 packets output, 0 bytes, 0 underruns

0 output errors, 0 collisions, 1 interface resets

0 babbles, 0 late collision, 0 deferred

0 lost carrier, 0 no carrier

0 output buffer failures, 0 output buffers swapped out

Router#

9. You can change the hostname of the router by doing the following:

Router#config

Configuring from terminal, memory, or network [terminal]? i Press Enter

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#hostname RouterA

RouterA(config)#

10. Now reload the router: do not save any changes.

Router#reload

Lab 2: ARP, CDP, Ping, and Telnet Lab

The physical topology is shown in Figure 1.75 below. Connect two routers with a crossover cable or with a switch.

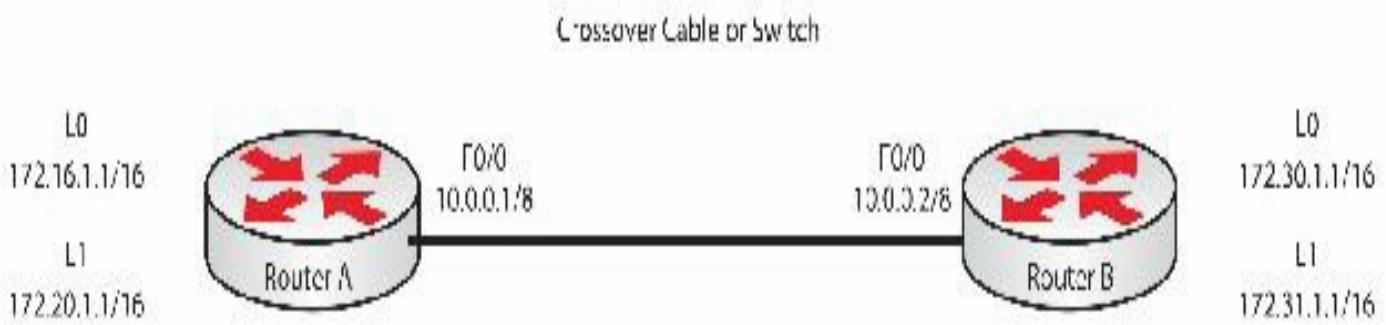


FIG 1.75 – ARP, CDP, Ping, and Telnet Lab

Lab Exercise

Your task is to configure the network referring to Figure 1.75 above to check for an ARP entry and a CDP neighbor, and to test the ping command and the telnet command. We are using Loopback interfaces here, which only exist logically. They are a perfect way to test routing and access lists without having to plug in extra hosts and cables.

Please note that if your interfaces aren't numbered F0/0, you will need to swap the interface ID for what you do have. Issue a show ip interface brief command to see what you have available. We will cover some of the commands here, which will be explained in later sections, such as static routing. Just copy them for now.

Purpose

This lab explores some TCP and CDP fundamentals. ARP issues are very common and the capacity to check ARP entries will be very useful to you in your career as a Cisco engineer.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 1.75 above. We are using Ethernet interfaces connected by a crossover cable or a switch for this lab.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Check the ARP entry on Router A. Ping Router B and check the ARP entry again.
5. Check CDP neighbor details.
6. Telnet from Router A to Router B.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```
Router#config
Router(config)#hostname RouterA
```

```
RouterA(config)#interface FastEthernet0/0
RouterA(config-if)#ip address 10.0.0.1 255.0.0.0
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback 1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#
RouterB(config)#interface FastEthernet0/0
RouterB(config-if)#ip address 10.0.0.2 255.0.0.0
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#
```

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access. You first need to check how many Telnet/VTY lines you have as each model differs, as does GNS3. To do this, type (in configuration mode):

```
RouterA(config)#line vty 0 ?
<1-903> Last Line number
<cr>
```

RouterA(config)#line vty 0 903 **Enters the VTY line configuration**
RouterA(config-line)#login local **This will use local usernames and passwords for Telnet access**
RouterA(config-line)#exit **Exit the VTY config mode**
RouterA(config)#username banbury password ccna **Creates username and password for Telnet access (login local)**

Router B:

```
RouterB(config)#line vty 0 903
RouterB(config-line)#login local
RouterB(config-line)#exit
RouterB(config)#username banbury password ccna
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco i Sets the enable password (encrypted)
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. To configure a default route, there is one simple step (in configuration mode):

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet0/0 i For all unknown addresses send the packet out of F0/0
```

Router B:

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
```

5. To test the connection, first, you will need to check whether the link is up. To do this, use the show interface command (see below):

Make sure that Fast Ethernet 0/0 is up and line protocol is up.

```
RouterA#show interface FastEthernet0/0
FastEthernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c3d.d469 (bia 0000.0c3d.d469)
  Internet address is 10.0.0.1/8
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, Loopback not set
```

Next, ping your neighbor's Ethernet interface; this will test whether the link is OK:

```
RouterA#ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!! i The first ping failed while the ARP reply came back from router A
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms.
```

Next, check your router ARP cache:

```
RouterA#show arp
Protocol Address Age (min) Hardware Addr Type Interface
```

```
Internet 10.0.0.2 0      0050.5460.f1f8 ARPA F0/0  
Internet 10.0.0.1 -      0010.7b80.63a3 ARPA F0/0
```

Your hardware address will obviously be different from the one on my routers!

6. To test CDP, you simply need to enter the show cdp neighbor command. Bear in mind that the spelling is U.S. English and that you will have a different output, depending on what device you are connected to. We will cover CDP in more detail later in this guide.

```
RouterA#show cdp neighbor
```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge S - Switch, H - Host, I - IGMP, r - Repeater

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
-----------	---------------	---------	------------	----------	---------

RouterB	F0/0 0	172	R	2900	F0/0
---------	--------	-----	---	------	------

7. Finally, Telnet from Router A to Router B. To quit a Telnet session hold down the Control (Ctrl) key, the Shift key, and the 6 key (Ctrl+Shift+6) at the same time. Then release and press the X key. Or just type exit a few times.

```
RouterA#telnet 10.0.0.2
```

Trying 10.0.0.2 ... Open

User Access Verification

Username: banbury

Password: **i Won't show as you type it**

```
RouterB>enable
```

Password:

```
RouterB# i You are now in privileged mode on Router B
```

Now issue a show run command on both routers and look at the output.

Lab 3: Traceroute from Router A to Router B

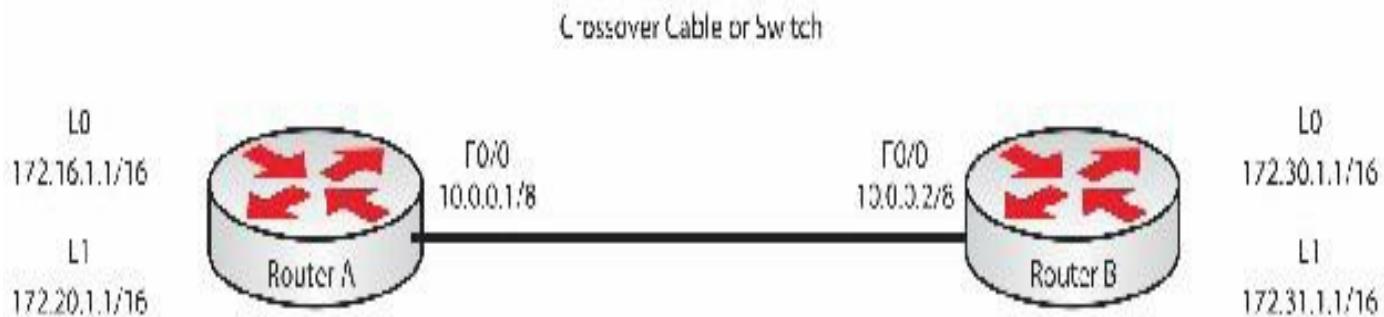


FIG 1.76 – Performing a traceroute

Lab Exercise

In this lab you will perform a traceroute From Router A to Router B using Figure 1.76 above as a reference. You wouldn't usually use the traceroute command over two routers but it's an easy way to try out some commands. You configured the network above in the previous lab so please copy all those commands.

Purpose

The traceroute command is a very valuable part of your troubleshooting toolkit. Don't mistake this for the Windows tracert command, which won't work on Cisco routers.

Lab Walk-through

In privileged mode, type in the Loopback address of Router B:

```
RouterA#traceroute 172.30.1.1
```

Type escape sequence to abort.

Tracing the route to 172.30.1.1

```
1 10.0.0.2 24 msec * 32 msec
```

Further Reading

Read Ethernet: The Definitive Guide by Charles E. Spurgeon if you want to dig into the origins of Ethernet in more detail.

Chapter 2 — LAN Switching Technologies

What You Will Learn in This Chapter

Layer 2 Switching Functions

Switching Methods

VLANs

Trunk Links and DTP

InterVLAN Routing

VLAN Trunking Protocol

Configuring a Switch

Syllabus Topics Covered

2.0 LAN Switching Technologies

2.2 Identify basic switching concepts and the operation of Cisco switches

 2.2.a Collision domains

 2.2.b Broadcast domains

 2.2.c Ways to switch

 2.2.c (i) Store

 2.2.c (ii) Forward

 2.2.c (iii) Cut-through

 2.2.c (iv) CAM table

2.3 Configure and verify initial switch configuration, including remote access management

 2.3.a Hostname

 2.3.b Management IP address

 2.3.c IP default gateway

 2.3.d Local user and password

 2.3.e Enable secret password

 2.3.f Console and VTY logins

 2.3.g Exec-timeout

 2.3.h Service password encryption

 2.3.i Copy run start

2.4 Verify network status and switch operation using basic utilities, such as

- 2.4.a Ping
- 2.4.b Telnet
- 2.5 Describe how VLANs create logically separate networks and the need for routing between them
 - 2.5.a Explain network segmentation and basic traffic management concepts
- 2.6 Configure and verify VLANs
- 2.7 Configure and verify trunking on Cisco switches
 - 2.7.a DTP (topic)
 - 2.7.b Autonegotiation
- 4.8 Configure and verify interVLAN routing (“router on a stick”)
 - 4.8.a Subinterfaces
 - 4.8.b Upstream routing
 - 4.8.c Encapsulation
- 4.9 Configure SVI interfaces

Previous versions of the CCNA exam only briefly touched on VLANs. Based on feedback from Cisco customers, the exam now heavily tests you on switch operations, configuration, security, and troubleshooting. Cisco switches use IOS but offer several unique commands since they perform a different function than routers.

This chapter will cover what you can consider to be the bread-and-butter work of any Cisco engineer.

Layer 2 Switching Functions

As you learned earlier (for the purposes of the CCNA exam), switches operate at layer 2 of the OSI model. Switches can only look at the MAC address of traffic and forward or block it based on that address. Historically, because switches didn't have to waste time examining layer 3 (IP) addresses, they were considerably faster than routers. Now, due to advances in switching and routing technology, forwarding planes are of comparable speed.

The forwarding plane is also referred to as the data plane, and on routers and switches this is where packets or frames are switched. Forwarding planes control traffic going through the device. Forwarding planes will be covered briefly later in this guide.

LAN switching is usually referred to as layer 2 switching because more advanced switching methods have now been invented that can actually operate at layer 3 and above. These types of switches are not included in the CCNA exam yet, but do check the latest syllabus before you book the exam.



FIG 2.1 – Cisco Catalyst 2960 Switch Range (Image ©Cisco Systems)

Cisco offers a huge range of switches, so please take the time to familiarize yourself with what's available on Cisco.com.

Switches are most commonly used to separate the LAN into smaller segments (microsegmentation). In practice, they can often provide far more features such as the VLAN Trunking Protocol (VTP), Virtual LANs (VLANs), and Quality of Service (QoS), which allows various types of network traffic to be prioritized over others such as video conferencing over e-mail.

Switches perform three main functions:

1. Learning MAC addresses
2. Filtering and forwarding frames
3. Preventing loops in the network

Learning MAC Addresses

When a switch is first powered up, it is not aware of the location of any of the hosts in the network. In a very short time, as hosts transmit data to other hosts, it learns and stores the source MAC address for each connected device. If an address is not currently in the switch's database it will send a broadcast message out of each port except the port that the request was received on, then when the switch receives a reply it will add the address and source port to its database. It can take only a matter of minutes to build

this database. Cisco refers to this as the CAM (content-addressable memory) table.

The switch will store a table of MAC addresses for a limited amount of time. If no traffic is heard from that port for a predefined period of time, then the entry is purged from memory. This frees up memory space on the switch and prevents entries from becoming out of date and inaccurate. This sequence of actions is known as the MAC address aging time. On the Cisco 2960 model, this time period is 300 seconds by default, but it can be configured to be between 10 and 1,000,000 seconds.

```
Switch#show mac address-table aging-time
```

```
Global Aging Time: 300
```

```
Vlan Aging Time
```

```
-----
```

```
Switch#conf t
```

```
Switch(config)#mac-address-table aging-time 600
```

The switch can also be configured to never purge the addresses.

The command you would use to see the CAM table of a switch is show mac-address-table (note the dashes in the command). Here is an example of the CAM table of a switch. You can add the command dynamic to the end if you only want to see the addresses learned dynamically by the switch. Infuriatingly, as you may discover for yourself, Cisco has changed the command slightly. One version has one dash and one has two, so find out which one you will need to use on your own switch.

```
Switch#show mac-address-table
```

```
Switch#show mac address-table
```

Mini-lab – Checking the MAC Address Table

In Figure 2.2 below, two PCs are connected to the switch. You need to add the IP address to each PC and then ping across the link. If you don't have spare PCs, then use Fast Ethernet interface routers instead and connect them with a crossover cable or a switch.

You can see in the output below that the switch has already added the MAC addresses to the table. This will only happen when traffic passes through the switch from one interface to another. Note that the switch will not store the IP address of the attached devices because it isn't concerned with IP addresses.

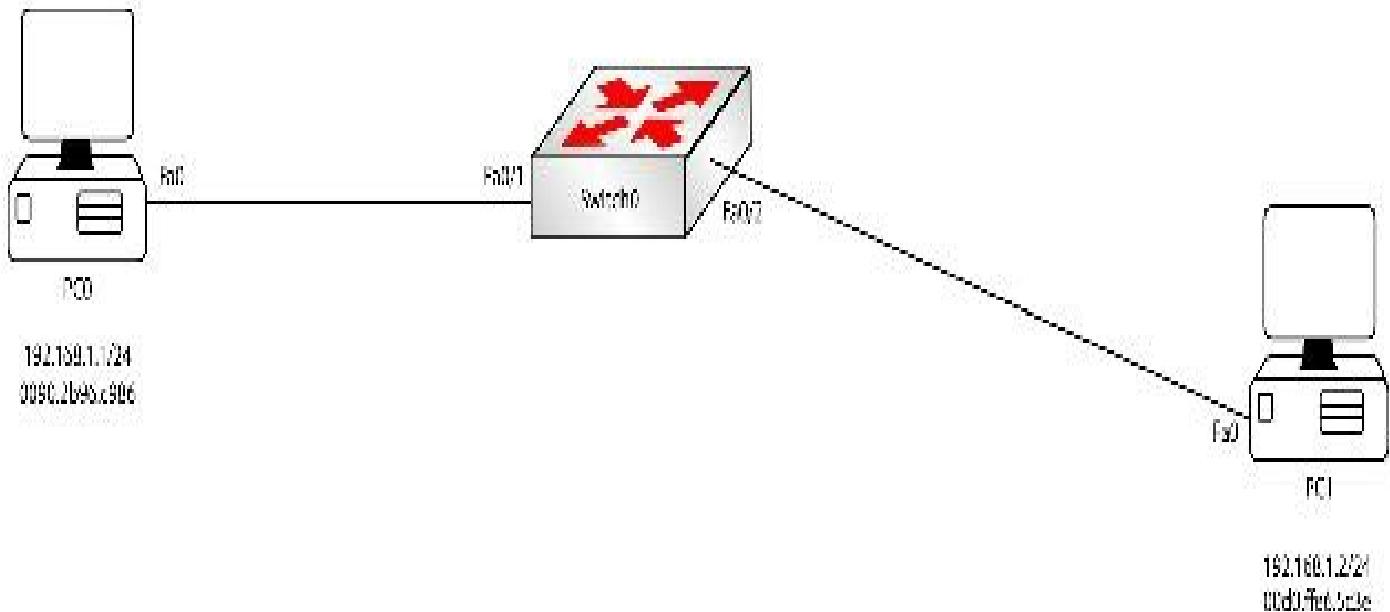


FIG 2.2 – Mini-lab: Checking the MAC Address Table

Switch#show mac-address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0090.2b96.c986	DYNAMIC	Fa0/1
1	00d0.ffe6.5c3e	DYNAMIC	Fa0/2

Vlan	Mac Address	Type	Ports
1	0090.2b96.c986	DYNAMIC	Fa0/1
1	00d0.ffe6.5c3e	DYNAMIC	Fa0/2

If you wanted to flush the CAM table on the switch you could issue the clear mac-address-table command. The switch would then repopulate the table as traffic came into the relevant interfaces (i.e., ping across the link to repopulate the MAC address table).

Switch#clear mac-address-table

Switch#show mac-address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0090.2b96.c986	DYNAMIC	Fa0/1
1	00d0.ffe6.5c3e	DYNAMIC	Fa0/2

[END OF MINI-LAB]

Note that the switch stores the MAC address and the port to which the host is connected.

Do not worry about the VLAN column at the moment, as we will cover this later in the chapter. If you want to view only the dynamic MAC addresses that the switch has discovered, use the show mac-address-table dynamic command. There are actually several options you may want to try:

Switch#show mac address-table ?

address Address to look up in the table
aging-time MAC address table aging parameters
count Number of MAC addresses in the table
dynamic List dynamic MAC addresses
interface List MAC addresses on a specific interface
move MAC Move information
multicast List multicast MAC addresses
notification MAC notification parameters and history table
secure List secure MAC addresses
static List static MAC addresses
vlan List MAC addresses on a specific vlan

It's important to note that you can see more than one MAC address allocated to a particular interface. You would also see this if the interface was a trunk (connected to another switch) or had a phone or hub attached. This is a common type of exam question.

Switch#show mac-address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
------	-------------	------	-------

1	0001.42dd.eca1	DYNAMIC	Fa0/1
1	0050.0fde.8ca1	DYNAMIC	Fa0/3
1	0090.0c63.9e31	DYNAMIC	Fa0/2
1	00e0.a30e.1c04	DYNAMIC	Fa0/1
1	00e0.f9de.e036	DYNAMIC	Fa0/1

You can see the topology for the network above in Figure 2.3 below, but my experience is that most networks have either no or out-of-date diagrams so you can't always rely on these. The output above is for the bottom switch. Fa0/1 connects via a trunk link to the top switch, which has another three hosts connected to it.

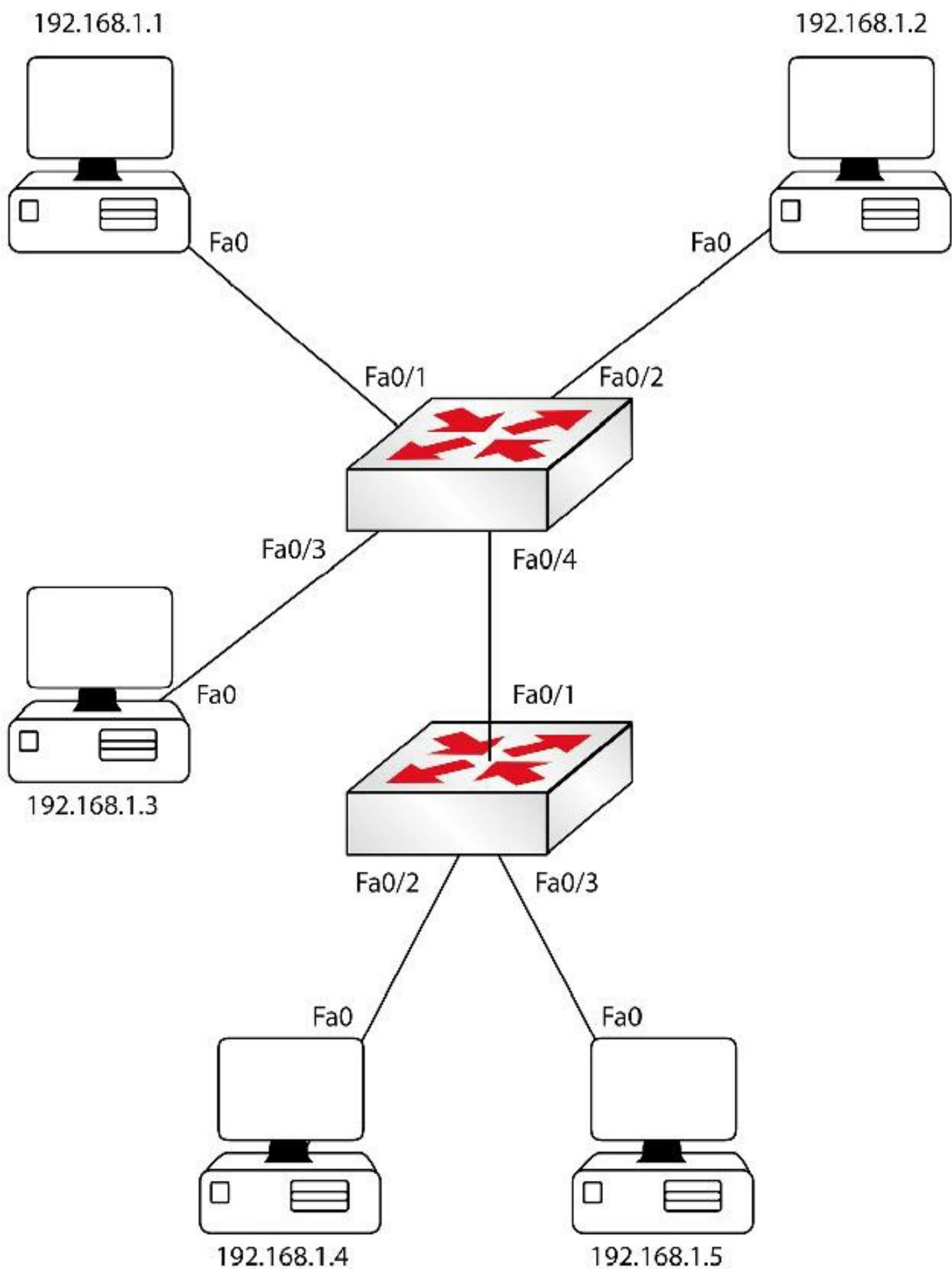


FIG 2.3 – Checking MAC address tables

If you are troubleshooting on an unfamiliar network, you could use this method to track down which switch a particular host is attached to. You would eventually find the switch it's directly connected to.

Please create your own network or use Packet Tracer, and then ping some devices to populate the CAM table. Next, issue the show mac-address-table dynamic command. It's worth noting that the table is comprised of source MAC addresses but not destination addresses. This is why you will never see a broadcast MAC address (all Fs) in the table.

Filtering and Forwarding Frames

Whenever a frame arrives at a switch port, the switch examines the destination address of the frame and then its database of MAC addresses. If the destination address is in the database, the frame will only be sent out of the interface the destination host is attached to. This process is known as frame filtering. If the address is not known, then the switch has no option but to flood the frame out of all ports other than the one on which it arrived.

Figure 2.4 below shows a capture of a frame. The Destination field contains the address c201:120c:0000 and this is what the switch will look for in its MAC address table. If this address isn't present it will broadcast for it.

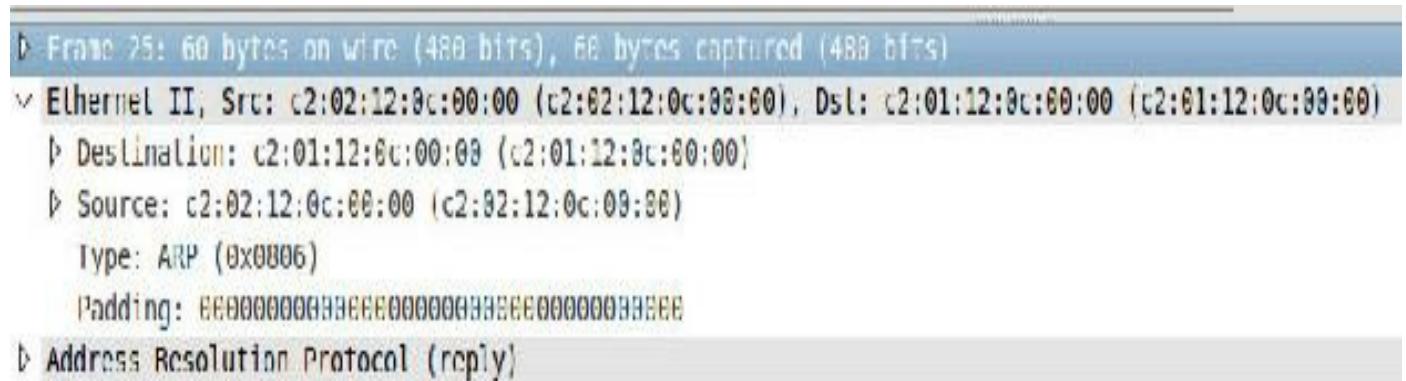


FIG 2.4 – Frame capture

Preventing Loops in the Network

Having multiple paths to destinations is very desirable in a network because if one path is no longer available, the traffic can take an alternative route. However, for switches this feature can often cause problems in the network. If a broadcast is sent out of one link, it will be flooded out of all links and could bring the network to a grinding halt due

to congestion. This situation is known as a broadcast storm.

In Figure 2.5 below, Host A sends a broadcast that is forwarded out of all ports on all switches, and each switch receiving the broadcast forwards it, quickly causing a loop.

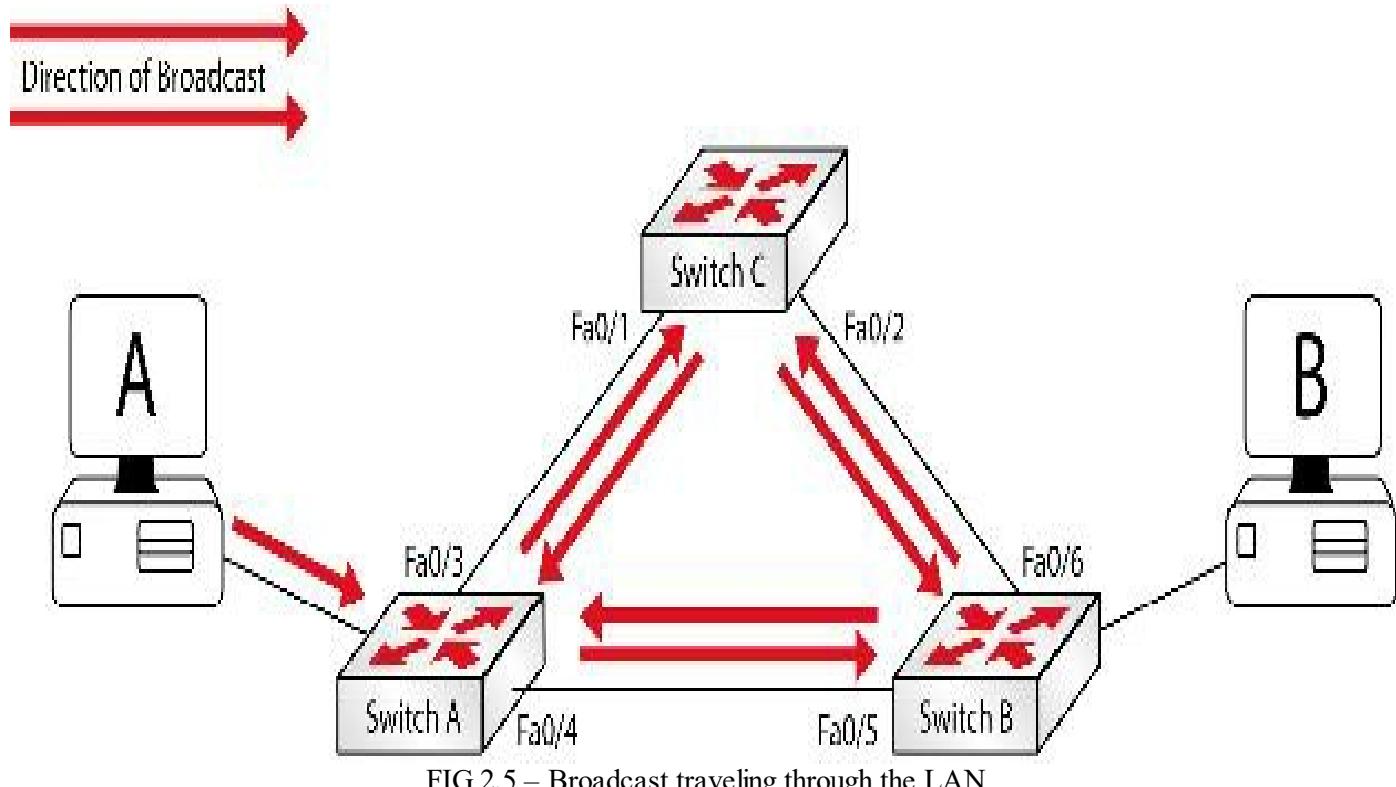


FIG 2.5 – Broadcast traveling through the LAN

The switch prevents loops using the Spanning Tree Protocol. We will look at this in depth in the ICND2 section.

Switching Methods

In theory, a frame can be forwarded by a switch as soon as the destination MAC address is read. The more of the frame that is read, the more delay (latency) is introduced. This delay can be caused by processing time on a device or congestion in the network. Of course, if the rest of the frame isn't read and checked for errors, you could be forwarding damaged frames across the network.

Cisco switches have three different switching modes, depending on how much processing is carried out on the frame. More processing usually equals more latency. The switching methods are:

- Cut-through
- Store-and-forward
- Fragment-free

Cut-through

This switching method is the fastest because the frame is only examined to determine its destination MAC address. As soon as the switch reads the destination MAC address, it looks up the MAC address table and forwards the frame to the right destination. The switch does not perform any error checks to ensure that the frame is not malformed in some way. However, if a frame is less than 64 bits, called a runt frame, cut-through switching eliminates it. This is also known as runt-free switching. The cut-through method is fast but less reliable.

Store-and-Forward

This method reads the entire frame, copies it into a buffer, performs a cyclic redundancy check (CRC), then forwards the frames only if the check passes. The CRC verifies that there is no error on the frame. The switch also checks to ensure that the frame size is between 64 and 1518 bytes. Anything outside this range is dropped. Frames with errors are also discarded. Store-and-forward switching has the highest latency of all three methods.

Store-and-forward is the default setting for the Cisco 2960 model. Modern switches offer hardware-based store-and-forward operating at wire speeds with a minimum of latency. Store-and-forward is reliable and nearly as fast as the cut-through method.

Fragment-free (Modified Cut-through/Runt-free)

Since the cut-through method cannot ensure that frames are error-free and store-and-forward increases latency, we need a method that strikes a balance between being both quick and reliable. The fragment-free method examines frames for errors but does not store all their information, resulting in reliability and speed.

As a modified variety of cut-through switching, fragment-free examines the first 64 bytes of a frame (which are the most error prone) for any errors, and if none are detected the frame is passed. Moreover, frames with less than 64 bits are dropped since they are not supported. This was the default configuration on lower-level Cisco switches.

The 3 Switching Methods

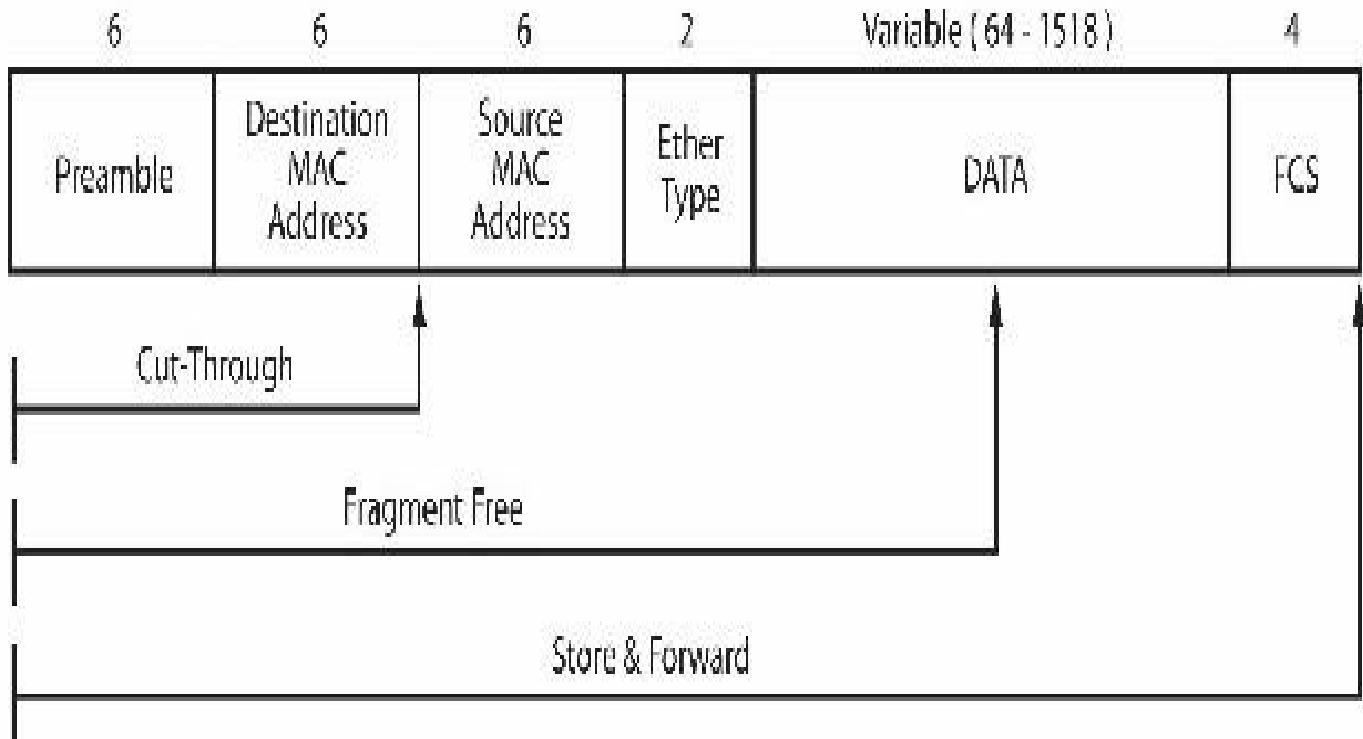


FIG 2.6 – Three switching methods

Virtual Local Area Networks (VLANs)

You learned earlier that a switch segments a collision domain and a router segments a broadcast domain. You also learned that every port (interface) on a switch will be in the same broadcast domain by default. The diagram below indicates the collision and broadcast domains in a network.

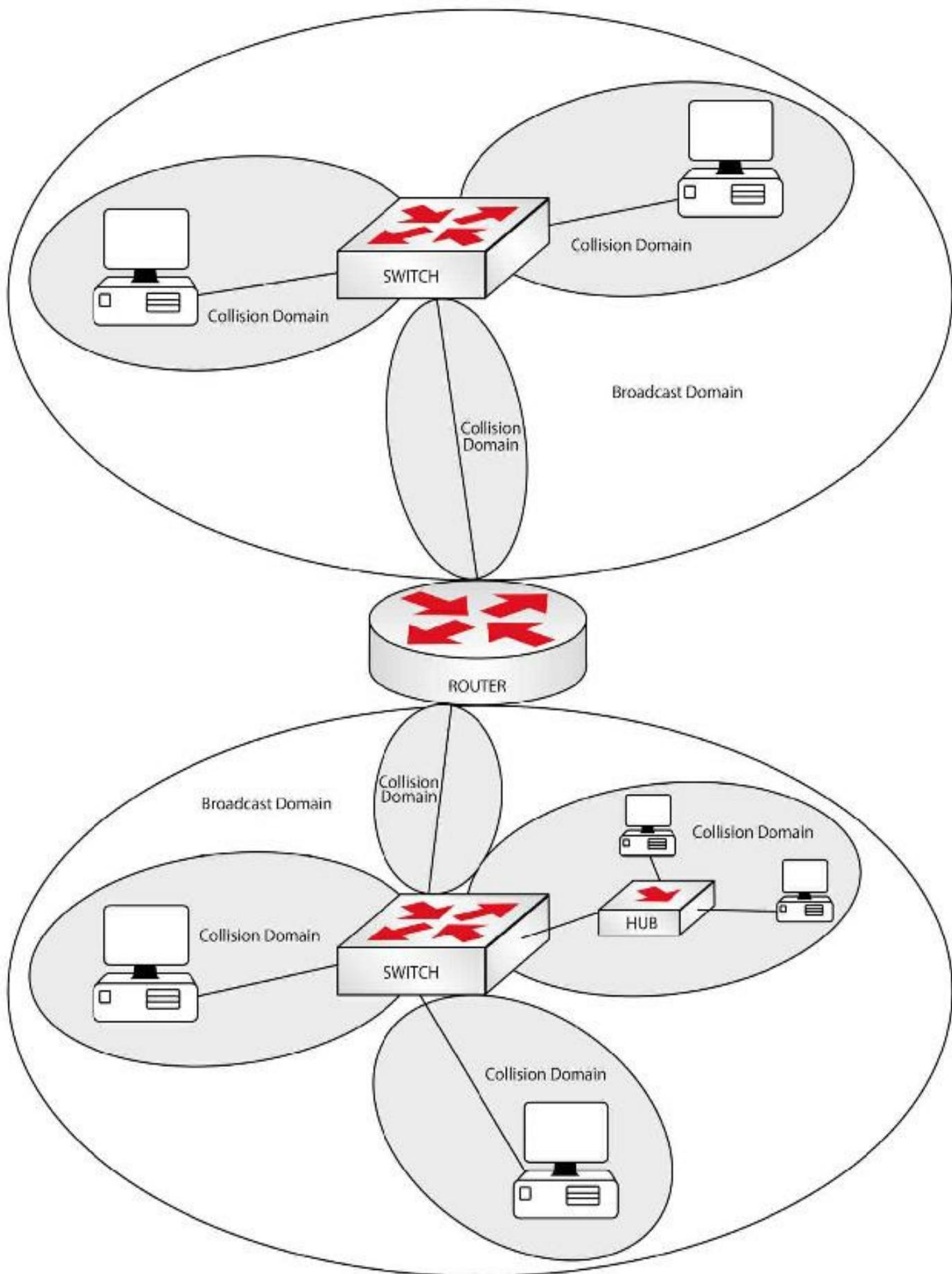


FIG 2.7 – Collision and broadcast domains

It's very important to understand what is happening in Figure 2.7 above. Note that routers break up broadcast domains, each port on a switch (and router) is a collision domain, and a hub is one big collision domain. Collision domains were discussed earlier in this guide. Please draw out your own LAN diagrams and count the number of collision and broadcast domains you see.

A LAN (or VLAN) is essentially a broadcast domain that exists on one or multiple switches and is configured by the network administrator. A VLAN is also created by the network administrator, who adds a configuration to one or more switch ports, putting them into different broadcast domains.

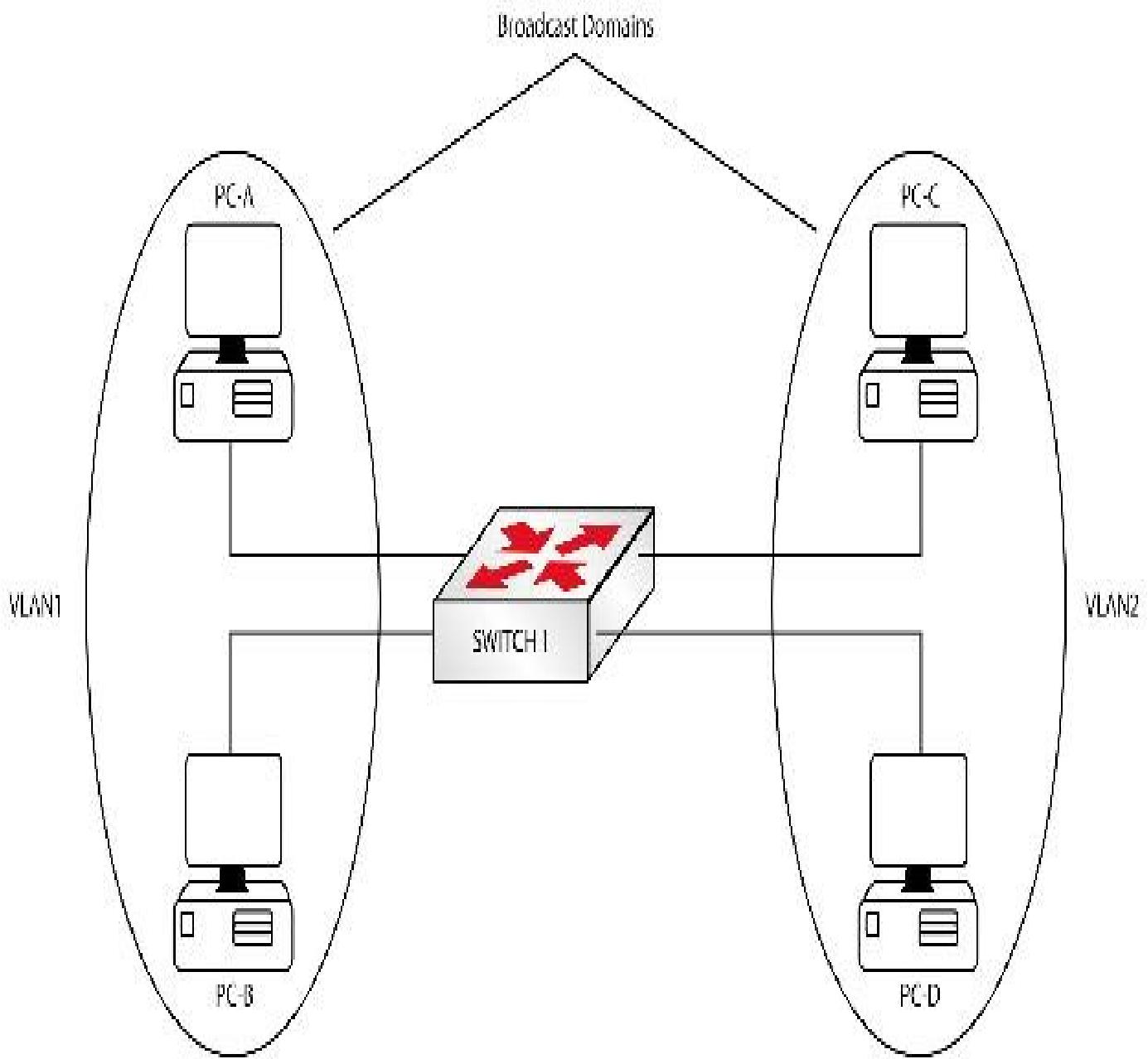


FIG 2.8 – Broadcast domains with VLANs

Figure 2.8 above shows a different network with four PCs all connected to the same switch. Without VLANs, all the ports are in the same broadcast domain and as such, a broadcast sent by PC-A would be received by PC-C and PC-D. With VLANs, a broadcast sent is forwarded only to the members of that VLAN, so a broadcast from PC-A would only reach PC-B and a broadcast from PC-C would only be received by PC-D.

By default, all ports on a switch are in VLAN 1 in the same broadcast domain, so it's up to you as the network administrator to configure more VLANs and add switch ports to them.

There are many advantages to using VLANs:

- Speed – reducing the broadcast domain makes the network faster since broadcast storms are reduced
- Resource conservation – devices no longer have to process broadcasts that are not intended for them, which saves CPU and memory resources
- Bandwidth conservation – logical networks operate on the same switch(es)
- Security – segmentation between logical domains, like departments and roles, so if a segment of the network is attacked, the attack can be contained to that logical segment
- Flexibility – since VLANs are a logical construct, they can be extended beyond a physical switch (more details on this later in this chapter), so you can have members of the same VLAN on multiple switches (which means they can be on different floors or different buildings!) as illustrated in Figure 2.9 below

VLANs also make moves, additions, and changes simpler. For example, if you need to move someone on VLAN 2 from Floor 1 to Floor 2, all you need to do is plug it into another switch and add a line or two of configuration at the most. Figure 2.9 below illustrates this concept:

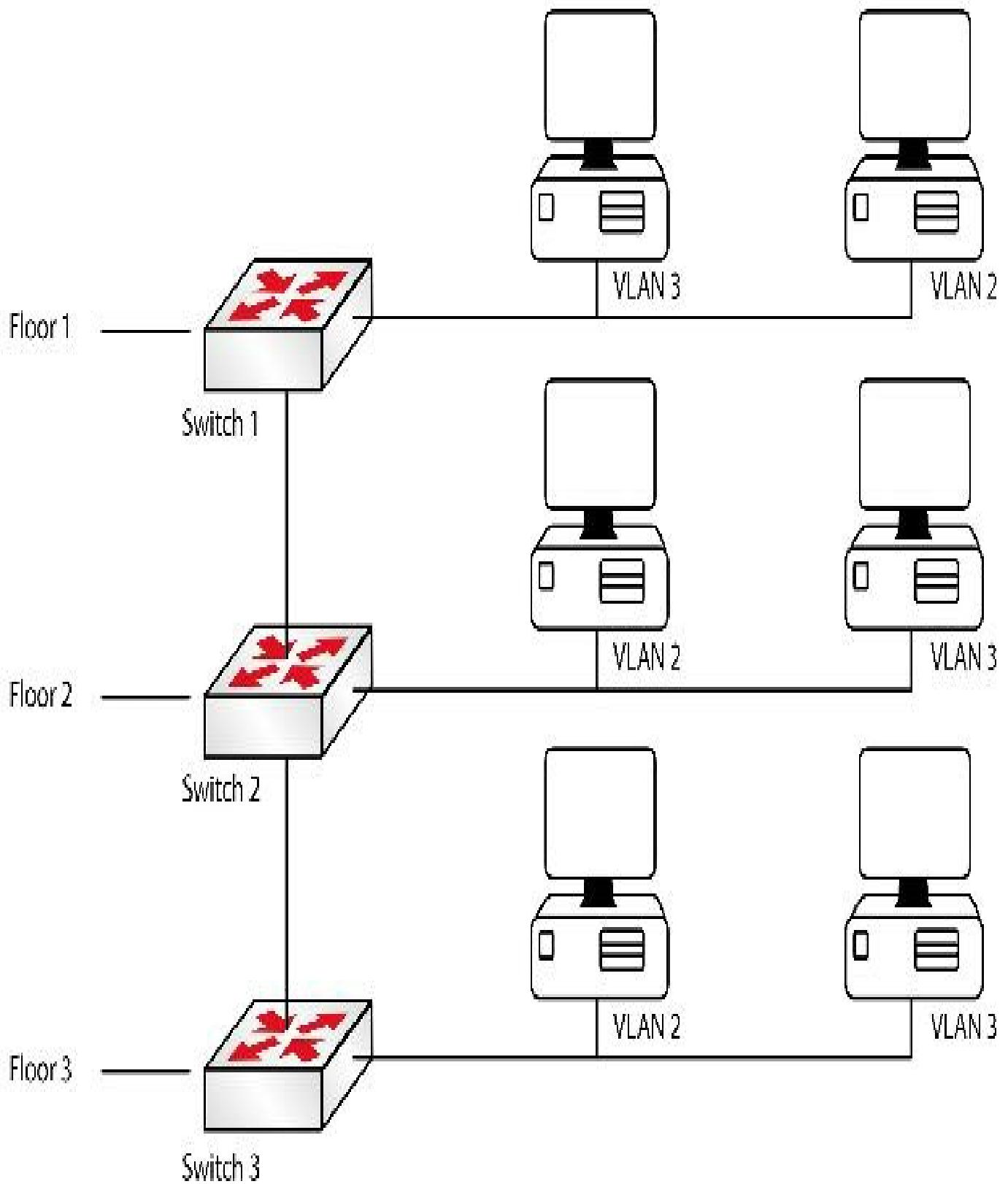


FIG 2.9 – VLANs remove the physical boundaries from a LAN

VLAN Membership

You can associate devices with VLANs either statically or dynamically. With static

assignments, the administrator assigns ports to VLANs by configuring the switch. If the user needs to move to another part of the building, the administrator has to assign the new port to the correct VLAN. We will cover this and other VLAN commands shortly. **Remember that all switch ports belong to VLAN 1 by default.** This is a security risk that you will learn how to resolve later on in the switch security chapter.

Dynamic VLAN assignment allows the automatic assignment of ports to VLANs based on the MAC addresses of the devices connected. This allows users to connect to any port without making any configuration changes. Automatic assignments are achieved using a VLAN Management Policy Server (VMPS), which is both rarely used and not in the CCNA syllabus.

For the purposes of the CCNA exam and best practices, each VLAN should have its own network or subnet address. This means:

- Communication is segmented between hosts in one VLAN and another VLAN
- Communication between two VLANs requires a layer 3 (routing) device because each VLAN has its own IP subnet (you will find more details on IP addressing in Chapter 3)

Finally, adding VLANs increases the number of broadcast domains but decreases the size of the domains (which is a good thing).

VLAN Numbers

Cisco switches have a range of numbers available for use as VLANs. By default, VLAN 1 is already in existence and all ports are set to use it. VLAN 0 isn't usable and VLANs 1002 to 1005 are reserved, meaning that you have VLANs 2 to 1001 available for use.

Switch1#show vlan brief

VLAN Name	Status	Ports

1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5 Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14, Fa0/15, Fa0/16, Fa0/17 Fa0/18, Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Fa0/24, Gig1/1 Gig1/2
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	

```
1005 trnet-default      active
```

There is an extended range of VLAN numbers available from 1006 to 4094; however, these VLANs can't be stored in switch memory and in order to use them, you must configure the switch to be in VTP transparent mode (more on this later).

VLAN Links

You have seen that a switch can have multiple VLANs and the same VLAN can span across multiple switches. In fact, this is one of the main benefits of using switches. So how does this affect traffic from one switch to another? Looking at Figure 2.9 above, If the host in VLAN 3 on Floor 1 needs to communicate with another host on a different VLAN on Switch 2 (on Floor 2), how does Switch 1 send the message to Switch 2 without losing the VLAN information? This is accomplished using a mechanism called VLAN tagging.

The switch tags the frame with a header that contains the VLAN ID. Referring to Figure 2.9, Switch 1 tags frames from VLAN 3 with VLAN ID3 before sending them on to Switch 2. Once Switch 2 sees the tags, it knows that the frames should be kept within that VLAN.

There are two kinds of layer 2 links on a switch:

- Access links
- Trunk links

Access Links

A switch port that is defined as a member of one VLAN is referred to as an access link. When a frame is received on an access link, it is tagged with its VLAN ID. The switch strips the tag of this frame at the destination before sending it on to the recipient host, so the process is transparent to the end device.

Access links are used to connect to hosts. Most ports on a switch are already set as access links, but you can hard set a port to be an access or trunk link and this is a common policy in commercial networks. I have shortened some of the commands below, which is common for network engineers to do:

```
Switch#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch(config)#int f0/1
```

```
Switch(config-if)#switchport mode ?
```

```
access Set trunking mode to ACCESS unconditionally
```

dynamic Set trunking mode to dynamically negotiate access/trunk mode

trunk Set trunking mode to TRUNK unconditionally

The dynamic setting allows the port to determine whether it should become an access or trunk link depending on the device it's connected to. You won't be able to leave the port in dynamic mode if you want to add port security, which we will look at later.

Trunk Links

A trunk link is used to carry traffic from multiple VLANs at the same time. Frames sent across a trunk link are tagged so they can be identified at the remote end. A trunk link can also be used to carry traffic between:

- Two switches
- A switch and a router
- A switch and a server

Trunk links are needed to forward traffic from multiple VLANs. Although you can use access links to connect two switches together, all the traffic from that access link would be treated as being in a single VLAN, which is configured on the port.

When multiple switches are connected using trunk links and they share information from the same VLANs, they are collectively called a Switch Fabric.

The protocol that is used for tagging VLANs on frames in a trunk link on Cisco switches is called the 802.1Q protocol.

802.1Q

802.1Q was created by the IEEE as a standard for tagging frames. 802.1Q works by inserting a 4-byte VLAN header into the original header of the Ethernet frame between the source MAC address and the Type/Length field. This 4-byte tag includes some information (including the VLAN ID of the frame).

Since 802.1Q is an IEEE standard, you can use it if you were connecting a trunk link between a Cisco switch and a non-Cisco switch. You can see 802.1Q frame tagging in action in the Wireshark capture below. Can you see the VLAN ID?

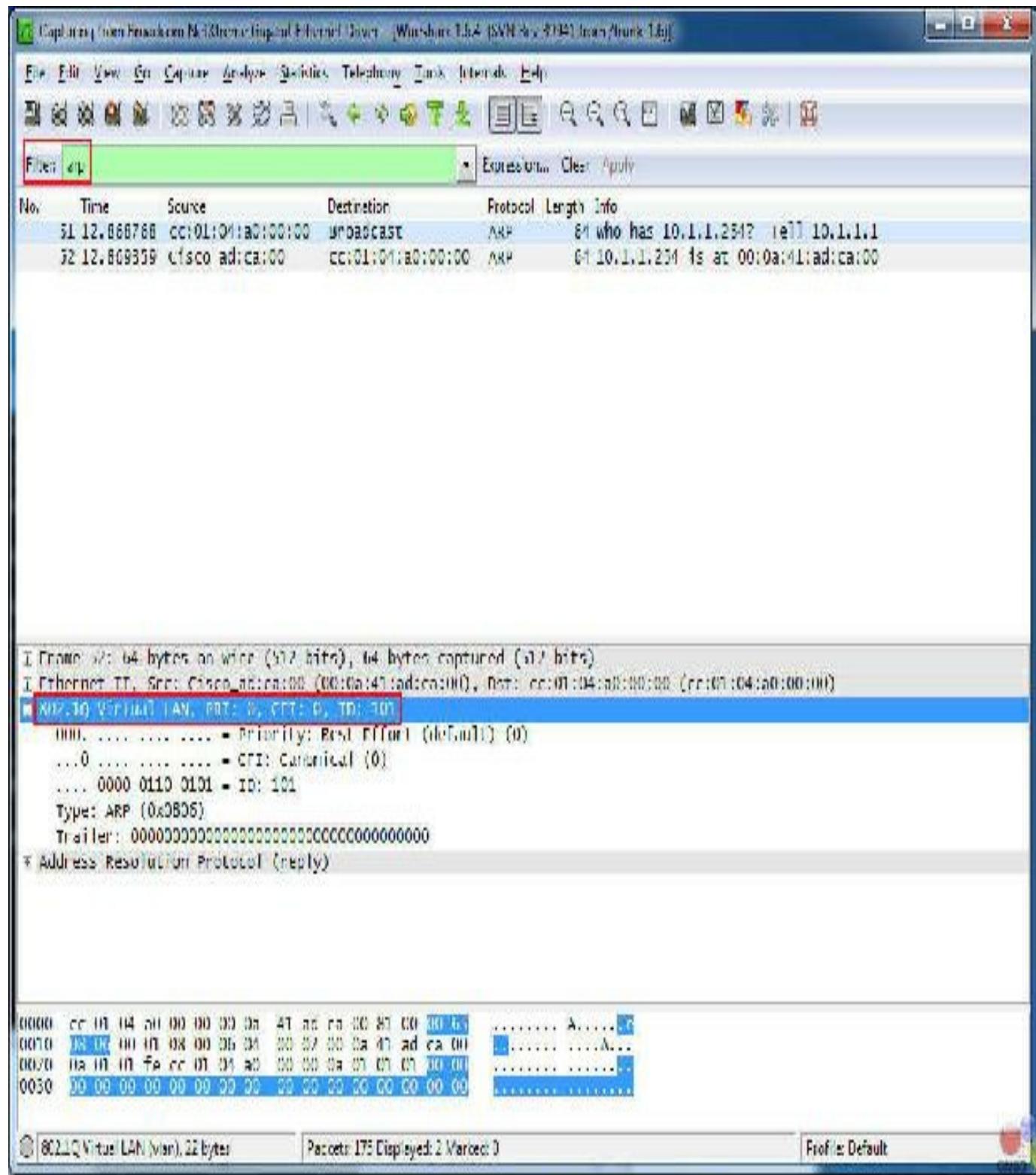


FIG 2.10 – Frame tagging

On a Cisco 3650 Multilayer Switch, the encapsulation command is still available. Previously, you could choose either 802.1Q or ISL encapsulation. On modern Cisco switches you will still see the encapsulation option but only 802.1Q is available:

```
3650Switch(config-if)#switchport trunk encapsulation ?
```

dot1q Interface uses only 802.1q trunking encapsulation when trunking
If you had a 3550 model, you would also see ISL and negotiate as options.

3550Switch(config-if)#switchport trunk encapsulation ?

dot1q Interface uses only 802.1q trunking encapsulation when trunking

isl Interface uses only ISL trunking encapsulation when trunking

negotiate Device will negotiate trunking encapsulation with peer on

On the 2960 Switch, the encapsulation command is not available, only 802.1Q is available (you will be tested on this in the CCNA exam) :

Switch(config-if)#switchport trunk ?

allowed Set allowed VLAN characteristics when interface in trunking mode

native Set trunking native characteristics when interface in trunking mode

Figure 2.11 illustrates the tag being inserted into the frame. When this is done the FCS (frame check sequence) must also be recalculated.

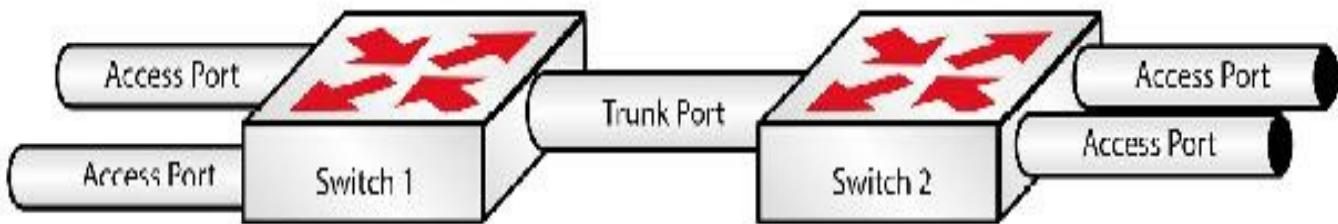
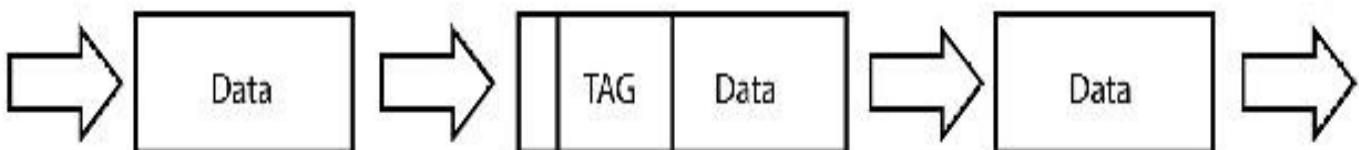
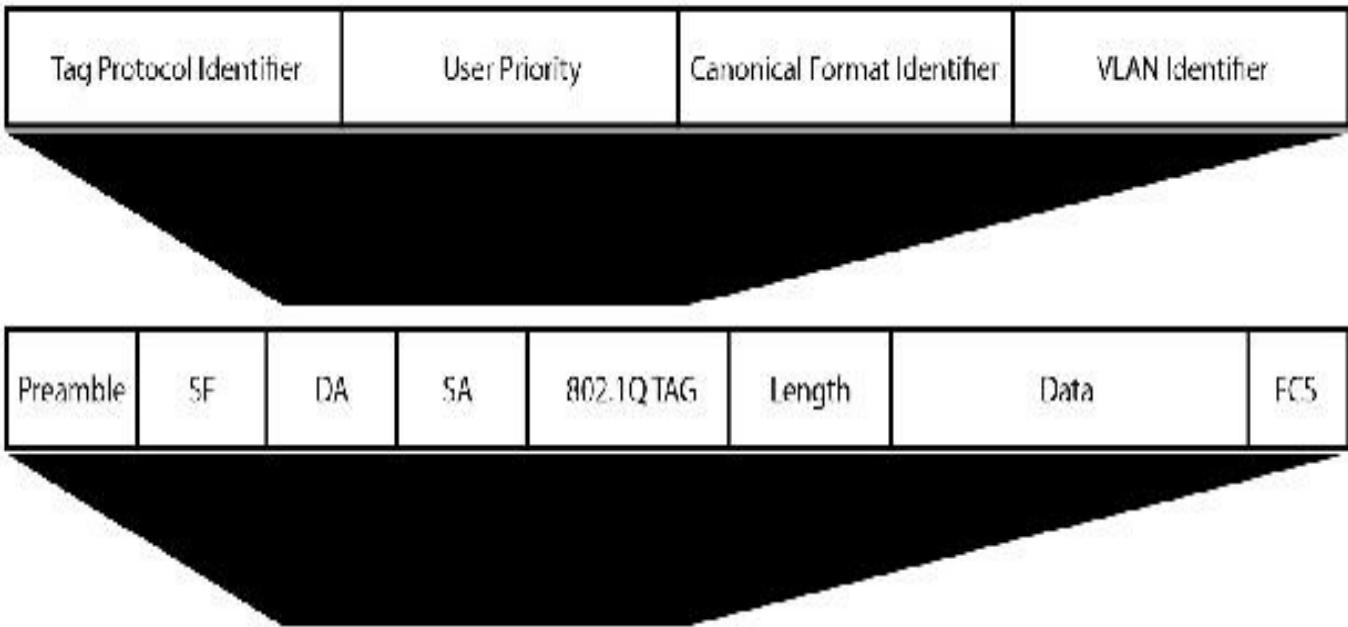


FIG 2.11 – Frames tagged and tags removed

All frames using 802.1Q are tagged with VLAN information. The exception to this is the native VLAN, which by default is VLAN 1. All frames inside the native VLAN remain untagged.

The native VLAN is nothing more than a default VLAN, given that any port in a (Cisco) switch has to be assigned to one VLAN. By default all ports belong to VLAN 1, or the native VLAN. You will learn how to change the native VLAN later.

In order for a trunk link to form, the native VLAN must match on both sides of the link. Although many study guides specify that the link must be at least 100 Mbps, you can actually create an 802.1Q trunk link over a 10 Mbps connection.

Trunk Links Continued

As a network engineer, you will want to know the settings of a switch port and how it

will form a trunk link with its neighbor because you may want it to become a certain port type rather than let the switch decide. This actually depends on the port mode. The possible modes are listed below:

- **On** – The port is configured as a trunk with the switchport mode trunk command. The connected device has to agree to also be a trunk; otherwise, the link will not work properly.
- **Off** – The port will not function as a trunk, regardless of what is configured at the remote end.
- **Auto** – The port is willing to be a trunk but will not initiate the negotiation, as the remote side has to initiate the negotiation. If both sides are set to auto, a trunk will not be formed.
- **Desirable** – The port is willing to become a trunk and will initiate the negotiation. If the remote side is desirable or auto, a trunk is formed.
- **No-negotiate** – Negotiation is disabled with the switchport nonegotiate command. The port has to be configured as a trunk or access link.

You must be familiar with all of the outcomes in Table 2-1 below for the exam:

Table 2-1: Trunk links

Switch 1	Switch 2	Is a Trunk Formed?
On	On	Yes
On	Auto	Yes (only one side passive)
Auto	Auto	No (both sides are passive)
Desirable	Desirable	Yes
Desirable	Auto	Yes
Desirable	On	Yes

We will cover how to configure the trunk links above shortly, but for now you can use the command below to see your default settings because these differ from model to model:

Switch#show interface f0/1 switchport

Name: Fa0/1

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: down

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

In order to negotiate trunking, switches use Dynamic Trunking Protocol (DTP), which used to be in the CCNP SWITCH syllabus but is also in the CCNA syllabus now.

Dynamic Trunking Protocol

DTP is used to dynamically negotiate a trunk link between two switches (if the other switch wants to perform trunking). This is a Cisco proprietary point-to-point protocol. DTP is turned on by default on a switch and a port can be in one of two DTP modes (see the previous IOS command output):

- Dynamic desirable
- Dynamic auto

Dynamic desirable puts the port in a state of active negotiation. If the remote end is configured for DTP (whether auto or desirable), a trunk will be formed. Dynamic auto is a passive negotiation state. A trunk will not be formed if both ends are configured to operate in dynamic auto mode.

Switch(config-if)#switchport mode dynamic ?

auto Set trunking mode dynamic negotiation parameter to AUTO

desirable Set trunking mode dynamic negotiation parameter to DESIRABLE

However, if one end of the link has been configured as a manual (static) trunk, then the other end will form a trunk as long as it receives DTP messages (i.e., the switchport nonegotiate command is not issued).

Table 2-2: DTP mode combinations

Switch 1	Switch 2	Result
Desirable	Desirable	Trunk Forms
Desirable	On	Trunk Forms
Desirable	Auto	Trunk Forms
Auto	Desirable	Trunk Forms
Auto	Auto	No Trunk
No-negotiate	Trunk	No Trunk
Static Access	Trunk	No Trunk



If you have the switchport nonegotiate command added to one or both ends, they will both have to be manually set to trunk to form a trunk link. If you wanted to turn off DTP for security reasons, for example, you would use the switchport nonegotiate command, which would allow you to manually configure the port as a trunk if you wanted it to become such.

```
Switch(config-if)#switchport nonegotiate
```

A link will only become a trunk through DTP or set manually with the switchport mode trunk command.

```
Switch(config-if)#switchport mode ?
```

access Set trunking mode to ACCESS unconditionally

dynamic Set trunk mode to dynamically negotiate access or trunk mode trunk Set trunking mode to TRUNK unconditionally

The show.dtp [interface [name]] command can be used to display DTP information globally for the switch or for the specified interface. The following output shows the information printed by the show.dtp command:

```
Switch#show dtp
```

Global DTP information

Sending DTP Hello packets every 30 seconds

Dynamic Trunk timeout is 300 seconds

4 interfaces using DTP

Based on the output above, the switch is sending DTP packets every 30 seconds and the DTP timeout is 300 seconds (five minutes), and four interfaces are currently using DTP. The show.dtp [interface [name]] command gives DTP information about a particular interface, which includes the type of interface (trunk or access), the current port DTP configuration, the trunk encapsulation, and DTP packet statistics, as shown in the output below. This is outside of what you need to know for the CCNA exam but it's handy to know.

```
Switch#show dtp interface FastEthernet0/1
```

DTP information for FastEthernet0/1:

TOS/TAS/TNS:	TRUNK/ON/TRUNK
--------------	----------------

TOT/TAT/TNT: 802.1Q/802.1Q/802.1Q

Neighbor address 1: 000000000000

[output truncated]

Mini-lab – Configuring VLANs and Trunk Links

It's time to see how VLANs are configured on Cisco switches. But first, it's important to note that up until IOS 12.2SX you would have used the command `vlan` database to create a VLAN. This command has now been retired, but if you have an old IOS version or an older switch model, then you will need to use this command.

In order to prepare for your CCNA exam, I recommend that you use a modern switch. You will be tested on the 2960 model in the exam (emulator); however, the 2950 model will do fine and will set you back around \$25 (\$50 for the 2960 model). If you plan to progress to the CCNP exam, then consider more powerful models that support advanced features, such as the 3560, which supports private VLANs. Cisco has full details on these features on its website.

We will configure the switches as shown in Figure 2.12 below. First, assign F0/1 on each switch to VLAN 5 and Fa0/15 will be configured as a trunk port. You already know how to change the default device name from Switch.

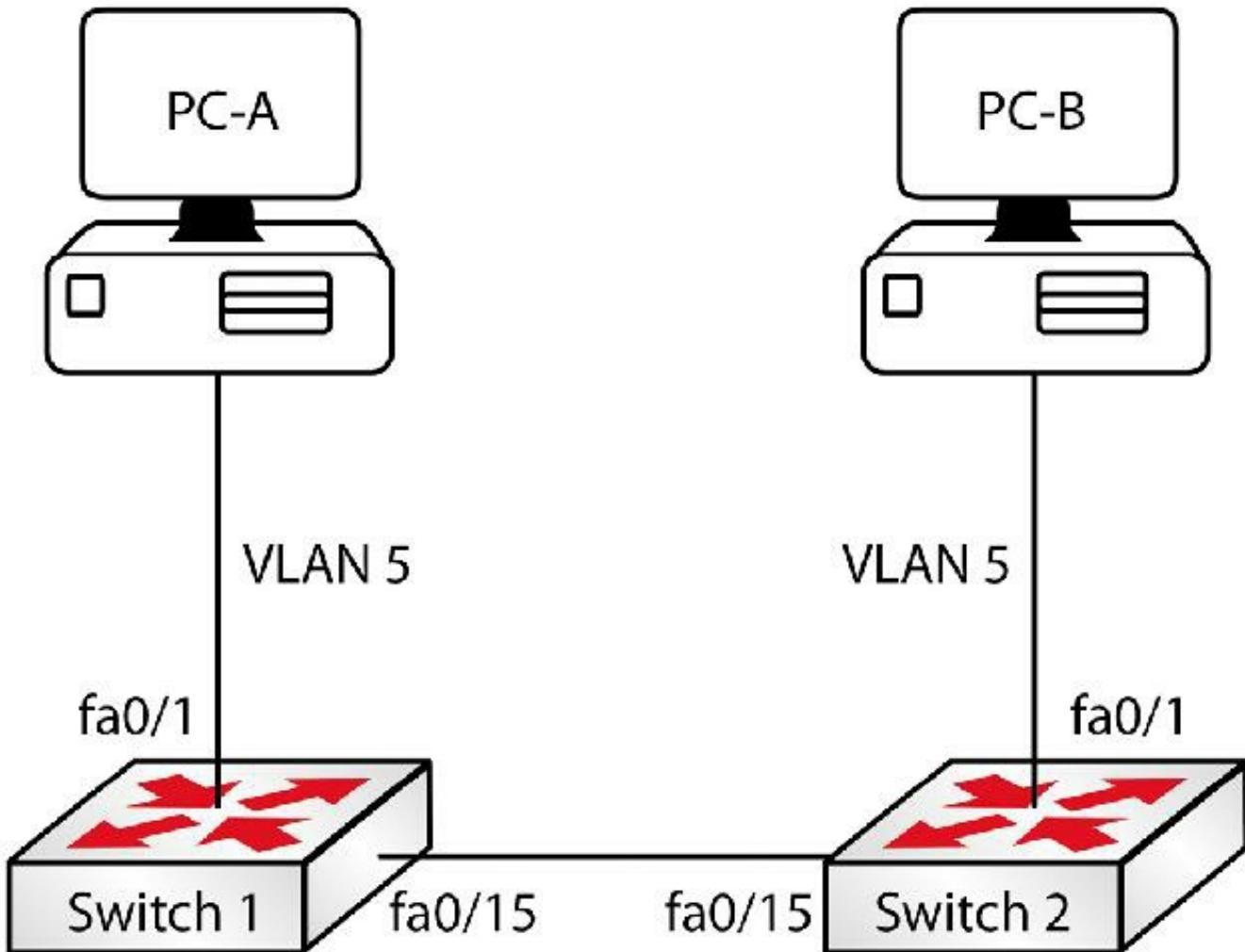


FIG 2.12 – Mini-lab: Configuring VLANs and Trunk Links

Next, you need to create the VLANs (by default, only VLAN 1 exists on a switch). VLANs are created using the `vlan [vlan#]` global configuration command. This takes you into VLAN configuration mode where you can give the VLAN a name.

On modern Cisco switch IOS versions, if a port is assigned to a VLAN that is not created, the switch will create the VLAN as you can see in the output below. If you put an interface into a VLAN that doesn't exist, the command is accepted and the VLAN is created. It is also added to the `vlan.dat` file (it won't show in the running configuration and VLAN data will remain present even after a switch reload). However, you should make it a habit to create your VLANs before assigning ports to them. See the output below, which demonstrates this point. I've truncated it to save space.

`Switch1#show vlan`

VLAN Name	Status	Ports
-----------	--------	-------

```
1 default      active  Fa0/1, Fa0/2, Fa0/3, Fa0/4  
                  Fa0/5, Fa0/6, Fa0/7, Fa0/8  
                  Fa0/9, Fa0/10, Fa0/11, Fa0/12  
                  Fa0/13, Fa0/14, Fa0/15, Fa0/16  
                  Fa0/17, Fa0/18, Fa0/19, Fa0/20  
                  Fa0/21, Fa0/22, Fa0/23, Fa0/24  
                  Gig1/1, Gig1/2
```

```
1002 fddi-default act/unsup
```

```
Switch1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch1(config)#int f0/1
```

```
Switch1(config-if)#switchport mode access
```

```
Switch1(config-if)#switchport access vlan 5
```

% Access VLAN does not exist. Creating vlan 5

```
Switch1(config-if)#end
```

```
Switch1#
```

%SYS-5-CONFIG_I: Configured from console by console

```
Switch1#show vlan
```

VLAN Name	Status	Ports
-----------	--------	-------

1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5
-----------	--------	----------------------------

[output truncated]

5 VLAN0005 **active Fa0/1**

```
1002 fddi-default act/unsup
```

You can also name your VLANs to easily identify them:

```
Switch1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch1(config)#vlan 5
```

```
Switch1(config-vlan)#name ADMIN
```

```
Switch1(config-vlan)#end
```

```
Switch#show vlan
```

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4,
[output truncated]		
5 ADMIN	active	Fa0/1

Next, assign port Fa0/1 to VLAN 5 using the switchport access vlan [vlan#] interface command. You will also create VLAN 5 on Switch 2.

```
Switch2#config t
Switch2(config)#vlan 5
Switch2(config-vlan)#name ADMIN
Switch2(config-vlan)#int fa0/1
Switch2(config-if)#switchport access vlan 5
```

Now, let's take a look at the show vlan brief output:

```
Switch1#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5 Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12, Fa0/13,Fa0/14, Fa0/16, Fa0/17,Fa0/19
5 ADMIN	active	Fa0/1
[output truncated]		

You can see that fa0/1 is now assigned to VLAN 5. The next step is to configure interface fa0/15 on both switches as trunk links. The default port mode on the switches that we are working on is dynamic auto mode, so no automatic trunk will form. This can be verified using the show interface trunk command:

```
Switch1#show interface trunk
```

```
Switch1#
```

There appears to be no trunk interfaces on either end, so you need to set the interface as a trunk. The procedure for configuring a trunk link on higher-end switches such as the 3570 is slightly different, but we will stick with the 2960 because this is the model tested in the CCNA exam. Ensure that the port is set to trunk; by default, it will be set to be accessed for use by a network host.

The configuration can be verified using the show interfaces [name] switchport command as illustrated in the following output:

```
Switch1#show interface fast0/15 switchport
```

Name: Fa0/15

Switchport: Enabled

Administrative Mode: **dynamic auto**

Operational Mode: **static access**

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Voice VLAN: none

[output truncated]

At the moment it is set to static access, but it needs to be a trunk link. The dynamic auto setting will allow the interface to become a trunk if the other end is set as a trunk or as dynamic desirable, so all you need to do is set either end of the connection between the switches as a trunk link.

```
Switch1(config)#int f0/15
```

```
Switch1(config-if)#switchport mode trunk
```

```
Switch1(config-if)#end
```

```
Switch1#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
------	------	---------------	--------	-------------

Fa0/15	on	802.1q	trunking	1
--------	----	--------	----------	---

[output truncated]

You can now repeat the earlier command to check the settings:

Switch1#show interface fast0/15 switchport

Name: Fa0/15

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Note that the settings on the other switch have changed automatically:

Switch2#show int f0/15 switchport

Name: Fa0/15

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Voice VLAN: none

[END OF MINI-LAB]

You will see different behavior on higher-model switches. Note the dynamic desirable setting on my Cisco 3550 switch below. Knowing seemingly small details like this will make you a very good Cisco engineer and save you a lot of trouble when installing and troubleshooting live networks.

Switch#show int f0/15 switchport

Name: Fa0/15

Switchport: Enabled

Administrative Mode: dynamic desirable

IEEE 802.1Q Native VLAN

Earlier in this chapter, you learned that 802.1Q tags frames with the VLAN ID, except for frames in the native VLAN. The native VLAN is created for backward compatibility with ports or equipment that do not understand VLAN tags.

VLAN 1 is the default native VLAN and this can be seen in the output of the show interfaces [name] switchport command. This is a very handy command to use on your switch because it reveals a lot of useful information.

```
Switch#show interfaces FastEthernet0/1 switchport
```

Name: Fa0/1

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Voice VLAN: none

[output truncated]

Management protocol traffic such as Cisco Discovery Protocol (CDP), Port Aggregation Protocol (PAgP), Dynamic Trunking Protocol (DTP), and VLAN Trunking Protocol (VTP) are transported in VLAN 1. We will cover these protocols in the relevant parts of this guide.

The native VLAN for a trunk interface can be set to any valid VLAN number. The caveat here is that this number must match on both sides; otherwise, the switch will issue a Spanning Tree Port VLAN ID (PVID) inconsistent state error. You will also see messages on the switch console indicating a VLAN mismatch.

Here is the error message on my 2960 Switch, which has been left at the default VLAN 1 for its Fast Ethernet 0/1 interface while the neighbor switch is configured to use this port for VLAN 10 as the native VLAN:

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/1 (1), with Switch FastEthernet0/1 (10).

Cisco error messages almost always tell you exactly what the nature of the problem is. So long as the error remains, you will see the message appear every few seconds.

The native VLAN can be changed using the switchport trunk native vlan [number] interface configuration command, as shown below. If I changed the switch interfaces above to be in the same native VLAN, the error message would cease.

```
Switch(config)#interface FastEthernet0/1
```

```
Switch(config-if)#switchport trunk native vlan ?
```

```
[1-4094] VLAN ID of the native VLAN when this port is in trunking mode
```

We will discuss this command again in the ICND2 section of the guide when we cover network security issues.

InterVLAN Routing

As mentioned, hosts in one VLAN cannot communicate with hosts in another VLAN without a layer 3 device. This is because the VLANs are, in effect, separate networks and traffic needs to be routed from one network to the other. Being able to route information between VLANs is referred to as interVLAN routing.

For the CCNA exam, you not only need to know about the available methods but also how to configure them. Because one of the methods involves configuring a layer 3 switch, I doubt that you will have a hands-on lab; however, Cisco may present you with sample configurations and ask you to choose the correct one or to spot a configuration error when shown the configurations on various switches.

There are three main methods to create interVLAN routing:

- Using physical router interfaces
- Using subinterfaces
- Using switched virtual interfaces

It's much easier to understand this process through practice labs.

Mini-lab – InterVLAN Routing Using Physical Router Interfaces

The first method to routing between VLANs is to connect a router interface to each VLAN. The router interface will serve as the default gateway for each VLAN. The router will then route between the networks since it is directly connected to both networks. This method is shown in the diagram below. You can swap the PCs for router Fast Ethernet interfaces and just use one per VLAN instead of two if you want.

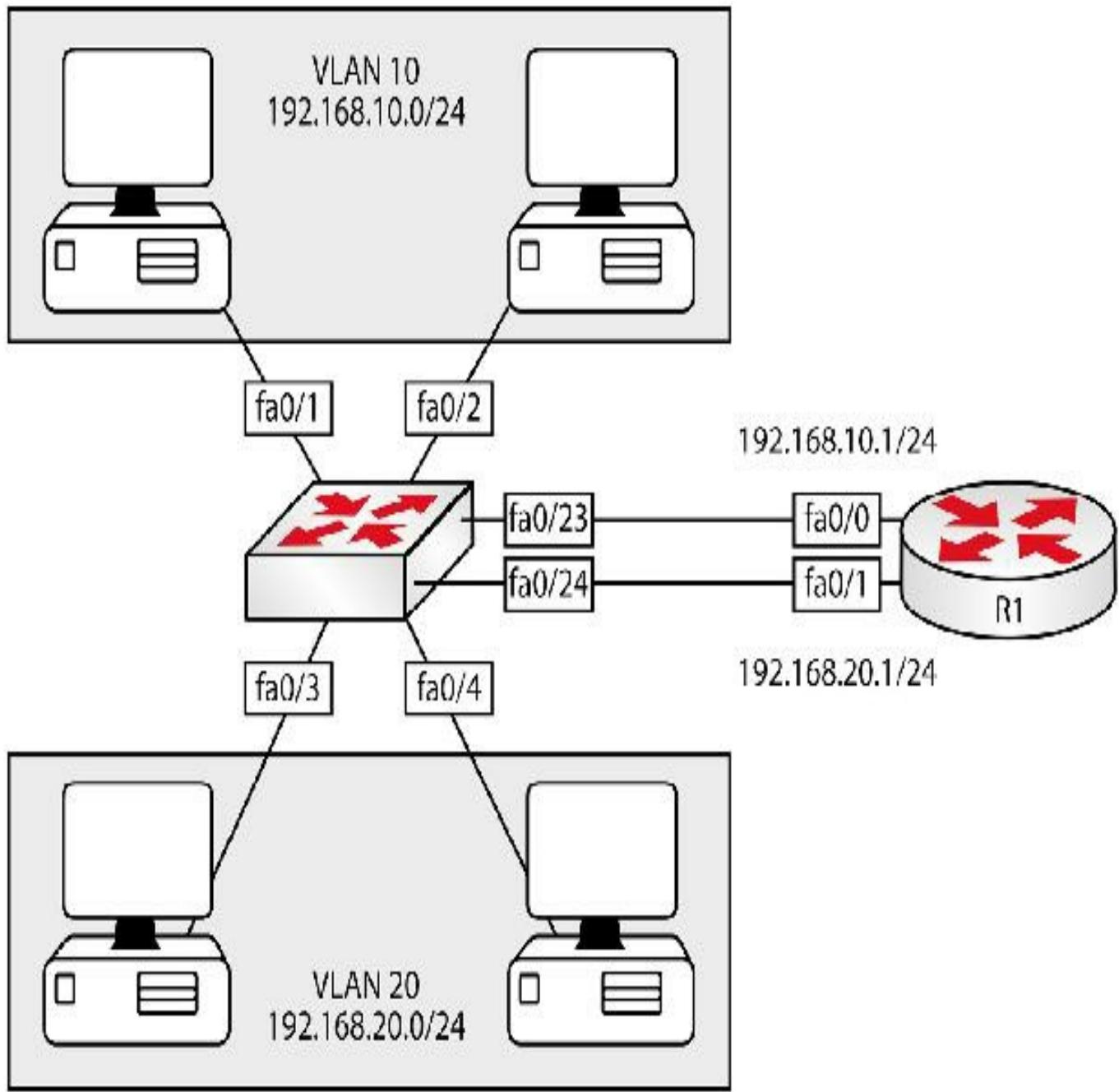


FIG 2.13 – Mini-lab: InterVLAN Routing Using Physical Router Interfaces

Figure 2.13 shows a single switch with two VLANs. VLAN 10 has subnet 192.168.10.0/24 and VLAN 20 has subnet 192.168.20.0/24. Packets going from one subnet to the other will use their default gateway. The default gateway of VLAN 10 is R1's Fa0/0 physical interface. Similarly, the default gateway of VLAN 20 is R1's Fa0/1 physical interface.

If the host needs to reach a device in its VLAN (same subnet), the host just ARPs for its MAC address and sends the traffic to that MAC address, letting the switch handle the rest. For hosts outside its subnet, they send traffic to the default gateway, which routes

the packets to the appropriate destination.

The default gateway is the address all traffic with an unresolved destination is sent (otherwise it would be dropped). IP addressing, subnetting, and routing will be covered in detail in subsequent chapters of this book.

From the switch's perspective, the router is just another host, so the port where the router is connected is assigned to the VLAN for which it serves as the default gateway. The switch configuration for this method is shown below:

```
Switch(config)#vlan 10
Switch(config-vlan)#name Example-VLAN-10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Example-VLAN-20
Switch(config-vlan)#exit
Switch(config)#interface range FastEthernet0/1 – 2, 23
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#exit
Switch(config)#interface range FastEthernet0/3 – 4, 24
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#exit
```

Note that I have used the interface range command, which is a time-saving method available on late-model switches. You have already seen that the VLAN will be created if you attempt to put ports into a non-existent VLAN. If that doesn't work for you then configure each interface individually.

```
Switch(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
```

The router's configuration is shown below:

```
R1(config)#interface fast0/0
R1(config-if)#ip add 192.168.10.1 255.255.255.0
R1(config-if)#no shut
```

```
R1(config-if)#exit  
R1(config)#interface fast0/1  
R1(config-if)#ip add 192.168.20.1 255.255.255.0  
R1(config-if)#no shut  
R1(config-if)#exit
```

You can add IP addresses to the hosts yourself. If you don't have hosts handy then use router Ethernet interfaces, or use just one router for each VLAN and add IP addresses from the correct subnet, such as:

- 192.168.10.2 (for VLAN 10)
- 192.168.10.3 (for VLAN 10)
- 192.168.20.2 (for VLAN 20)
- 192.168.20.3 (for VLAN 20)

You will need to add a default gateway for your hosts, which will be the router interface for the correct VLAN. If you find this lab to be a bit hard, come back to it later when you have more hands-on experience with the other labs.

In the output below, a PC in VLAN 10 with the IP address 192.168.10.2 is pinging a host in VLAN 20 (you can see that the first ping timed out due to an ARP lookup):

```
PC>ping 192.168.20.2
```

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out. **i Timeout due to ARP lookup**

```
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
```

```
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
```

```
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
```

Ping statistics for 192.168.20.2:

 Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),

 Approximate round-trip times in milli-seconds:

 Minimum = 0ms, Maximum = 0ms, Average = 0ms

[END OF MINI-LAB]

The advantage of this method is its ease. All you need to do is assign an IP address to the interface of a router and make that IP address the gateway of the VLAN and you are

done! The disadvantage is that it is not scalable. So what happens when you have five or 20 or more VLANs? Buying routers with multiple interfaces or multiple routers is not a viable option, both technically and economically.

This brings us to the next method of InterVLAN routing.

Mini-lab – InterVLAN Routing Using Router Subinterfaces

This method is often referred to as “router on a stick.” Instead of using a physical interface for each VLAN, you can use a single physical interface for all the VLANs on the switch. The single physical interface is logically divided into subinterfaces and a subinterface is assigned to each VLAN. This helps to address the scalability concern of the previous method. InterVLAN routing using router subinterfaces is illustrated below in Figure 2.14:

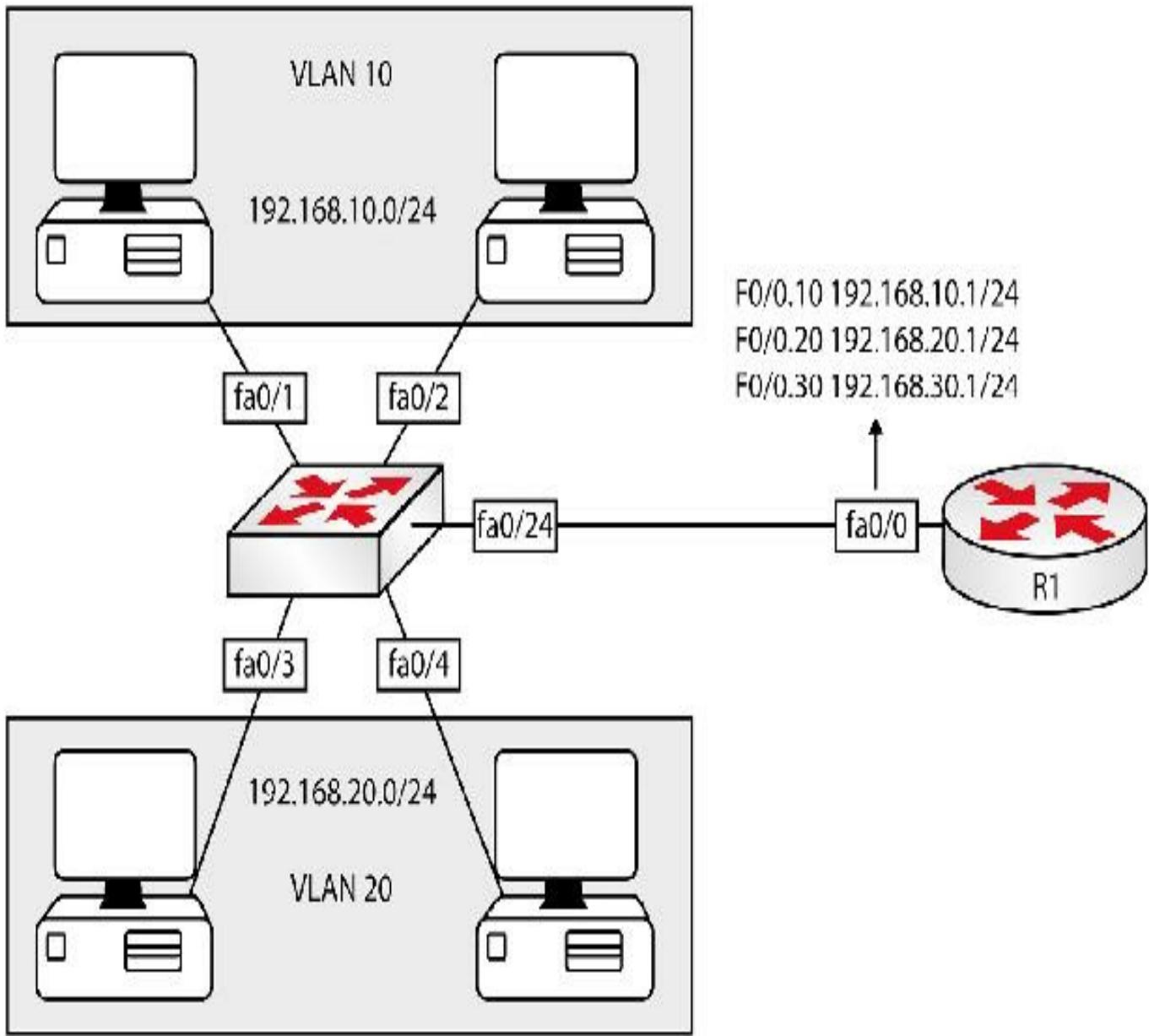


FIG 2.14 – Mini-lab: InterVLAN Routing Using Router Subinterfaces

The diagram above shows the same network as the previous one, with the difference being that rather than using a physical interface for each VLAN, now one physical interface is used with subinterfaces on each VLAN. There are three steps to configuring the subinterface:

1. Create the subinterface using the interface [name].[subinterface number] command.
2. Specify the VLAN and the encapsulation protocol using the encapsulation [isl|dot1Q] [vlan] subinterface configuration command. You will only see ISL as an option on older switch models.
3. Assign an IP address to the subinterface.

On the switch end, you now need to send traffic from multiple VLANs to the router, so you need a trunk link. You need to specify the encapsulation protocol if you are using a higher model switch and native VLAN (if you do not want to use the default). If a native VLAN other than VLAN 1 is used, then it needs its own subinterface on the router too. The configuration for the switch and router are shown below:

```
Switch(config)#vlan 10
Switch(config-vlan)#name Example-VLAN-10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Example-VLAN-20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#name Management-VLAN
Switch(config-vlan)#exit
Switch(config)#interface range FastEthernet0/1 – 2
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#exit
Switch(config)#interface range FastEthernet0/3 – 4
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#exit
Switch(config)#interface FastEthernet0/24
Switch(config-if)#switchport trunk encapsulation dot1q i Won't work on 2960
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk native vlan 30
Switch(config-if)#exit
Switch(config)#interface vlan 30
Switch(config-if)#description "This is the Management Subnet"
Switch(config-if)#ip address 192.168.30.2 255.255.255.0
```

```
Switch(config-if)#no shutdown  
Switch(config-if)#exit  
Switch(config)#ip default-gateway 192.168.30.1
```

None of this will work without the default gateway (i.e., where to send traffic for which no specific route exists) for IP traffic being configured on the switch because it's unable to route (we will cover this in detail later). Since VLAN 30 was created as the management VLAN and used as the native VLAN, you need a subinterface for VLAN 30 too. The router configuration is shown below:

```
R1(config)#interface FastEthernet0/0  
R1(config-if)#no ip address  
R1(config-if)#exit  
R1(config)#interface fast0/0.10  
R1(config-subif)#description "Subinterface For VLAN 10"  
R1(config-subif)#encapsulation dot1Q 10  
R1(config-subif)#ip add 192.168.10.1 255.255.255.0  
R1(config-subif)#exit  
R1(config)#interface fast0/0.20  
R1(config-subif)#description "Subinterface For VLAN 20"  
R1(config-subif)#encapsulation dot1Q 20  
R1(config-subif)#ip add 192.168.20.1 255.255.255.0  
R1(config-subif)#exit  
R1(config)#interface fast0/0.30  
R1(config-subif)#description "Subinterface For Management"  
R1(config-subif)#encapsulation dot1Q 30 native  
R1(config-subif)#ip add 192.168.30.1 255.255.255.0  
R1(config-subif)#exit
```

Note that each router subinterface is in a different subnet and it matches the VLAN subnet. You must also configure the correct encapsulation type (dot1Q) and the tag number for the VLAN, such as encapsulation dot1Q 10.

It is not mandatory for the subinterface number to match the tag number but it is considered good practice.



Once again, set IP addresses on the hosts in the correct VLANs and set the default gateway to the router subinterface in the respective VLAN. Ping across the VLANs now.
[END OF MINI-LAB]

Clearly, this option is more scalable than the previous one. However, this solution causes all interVLAN traffic to flow through the same interface. This can quickly lead to bandwidth constraints and create a bottleneck in the network. You would not need a routing protocol on the router because all of the networks are directly connected. Issue a show ip route to verify this for yourself.

Note that routers are backward compatible with older switches so they offer both ISL and 802.1Q encapsulation.

```
R1(config-subif)#encapsulation ?  
dot1Q IEEE 802.1Q Virtual LAN  
isl  Inter Switch Link - Virtual LAN encapsulation
```

Mini-lab – InterVLAN Routing Using Switched Virtual Interfaces

You saw earlier that modern switches can now operate at layer 3 (and above in fact). This means switches can now route. And if switches can route, you no longer need a router just to route between two VLANs. This feature is not available on the 2960 range of switches. If you want to configure the commands below you will need to find a 3550 or later model, or use a layer 3 switch in Packet Tracer.

Multilayer switches can support switch ports and routed ports. A routed port acts in the same manner as an Ethernet port on your router, allowing you to add an IP address, for example. You can make a port operate at layer 3 by issuing the no switchport command (not available on the 2960 model). This disables the port at layer 2 (and enables it at layer 3). After this, you can configure layer 3 features such as IP addressing on the port/interface.

Switches also support logical interfaces known as switched virtual interfaces (SVIs).

These interfaces are logical layer 3 interfaces (they can have an IP address). You might have noticed that I configured interface `vlan 30` in the previous mini-lab. This is an example of an SVI. You can configure an SVI for each VLAN on a multilayer switch. Since the switch can route traffic from one VLAN to the other, you only need to set the SVI's IP address as the default gateway for the VLANs.

One final step to ensuring that traffic is routed from one VLAN to another is to configure IP routing on the switch. This turns on the routing features of a multilayer switch. It's worth noting that not all Cisco switches support interVLAN routing using SVIs. It is both model and IOS (release # and feature set) dependent. Check your documentation first.

“Catalyst switch models 3560, 3750, Catalyst 4500/4000 Series with Sup II+ or later, or Catalyst 6500/6000 Series that run Cisco IOS system software support basic InterVLAN routing features in all their supported software versions. Before you attempt this configuration on a 3550 series switch, ensure that you meet these prerequisites.” (See Table 2-3 below.)

Table 2-3: ©Cisco Systems

Image Type and Version	InterVLAN Routing Capability
Enhanced Multilayer Image (EMI) – All Versions	Yes
Standard Multilayer Image (SMI) – Prior to Cisco IOS Software Release 12.1(11)EA1	No
Standard Multilayer Image (SMI) – Cisco IOS Software Release 12.1(11)EA1 and Later	Yes

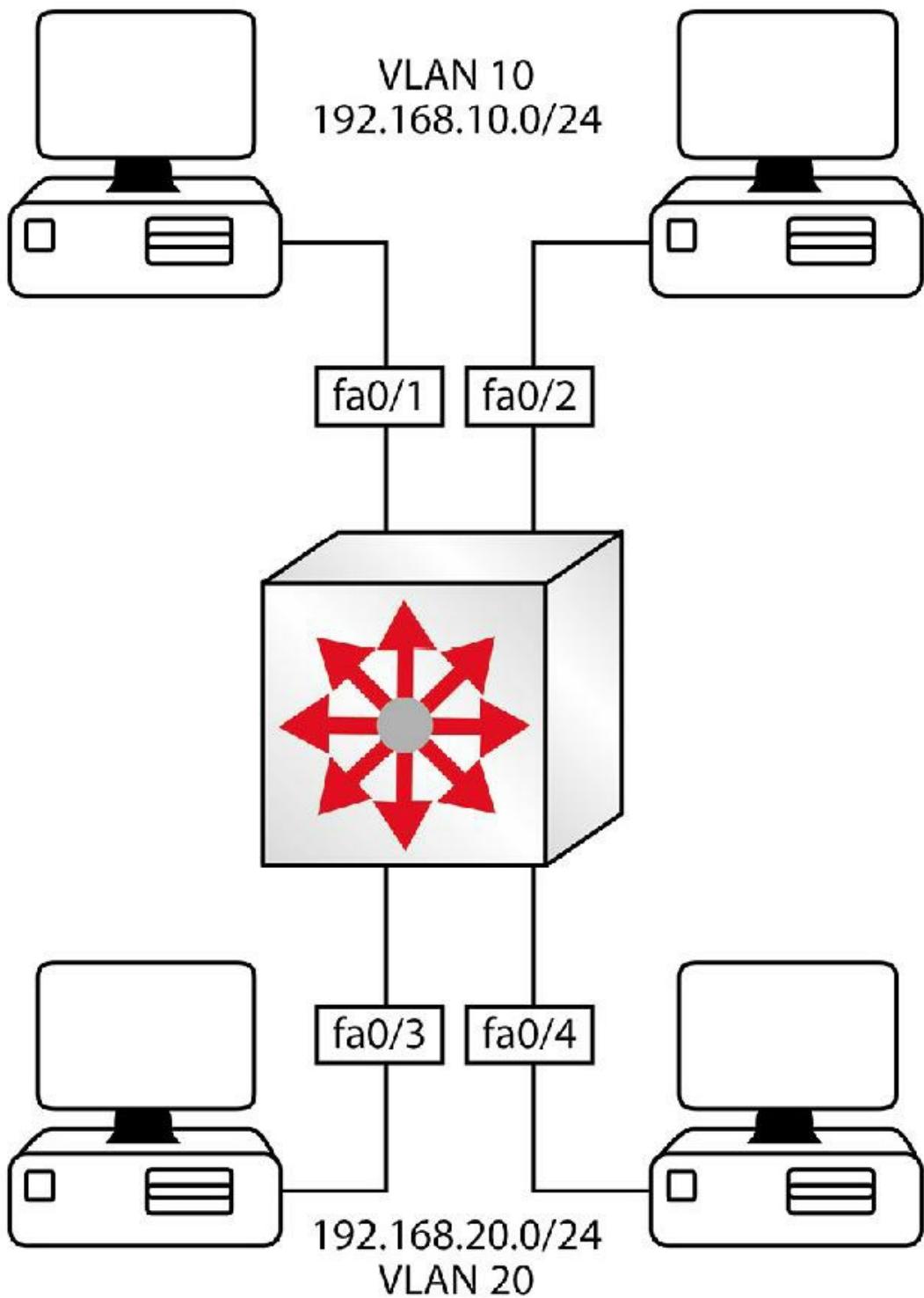


FIG 2.15 – Mini-lab: InterVLAN Routing Using SVIs

The output below shows the configuration required to support interVLAN routing on a single switch. The **ip routing** command is of particular importance to ensure that the layer 3 switch can route the traffic.

```
Switch(config)#ip routing
```

```
Switch(config)#vlan 10
```

```
Switch(config-vlan)#name Example-VLAN-10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Example-VLAN-20
Switch(config-vlan)#exit
Switch(config)#interface range FastEthernet0/1 – 2
Switch(config-if-range)#switchport
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#exit
Switch(config)#interface range FastEthernet0/3 – 4
Switch(config-if-range)#switchport
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#exit
Switch(config)#interface vlan 10
Switch(config-if)#description “SVI for VLAN 10”
Switch(config-if)#ip address 192.168.10.1 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#interface vlan 20
Switch(config-if)#description “SVI for VLAN 20”
Switch(config-if)#ip address 192.168.20.1 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
```

It's very easy to miss but ensure that you add the ip routing command. I've made this mistake many times personally and have also supported top-level Cisco experts who did the same. You should now be able to add IP addresses to the hosts in each VLAN and ping across VLANs 10 and 20.

SVIs are the preferred method for implementing interVLAN routing on a switch. As its

name suggests, the switch virtual interface is virtual, meaning this interface doesn't physically exist; instead, it's logically defined on the switch's routing logic (in its Routing Engine).

Like any other interface, you can verify the interface state using the show interface command. A sample output is shown below:

```
Switch#show interface vlan 10
```

Vlan10 is up, line protocol is down

Hardware is EtherSVI, address is c200.06c8.0000 (bia c200.06c8.0000)

Internet address is 192.168.10.1/24

MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

ARP type: ARPA, ARP Timeout 04:00:00

[END OF MINI-LAB]

SVI versus Routed Port

I don't expect this to come up in the CCNA exam but because I'm guessing you want to be a good Cisco engineer, it's worth considering the difference between a physical routed port and an SVI.

Table 2-4: SVI versus routed port

Characteristics	SVI	Routed Port
Logic	Virtual layer 3 interface working behind layer 2 interfaces. Advanced router on a stick.	True layer 3 interface
Capable of Routing	Yes	Yes
Physically Exists	No	Yes
Sample Configuration	Switch(config)#vlan 10 Switch(config)#int vlan 10 Switch(config-if)#ip address 10.0.0.10 255.255.255.0	Switch(config)#interface f0/3 Switch(config-if)#no switchport Switch(config-if)#ip address 10.0.0.10 255.255.255.0

By default, a 2960 Switch cannot route IP packets. This model is at the low end of Cisco's product range. It can't actually route packets because the Switch Database

Management (SDM) template does not support routing by default. You need to change the SDM template before routing can be supported. The output below shows your options but these will differ based on the model and IOS installed:

```
Switch(config)#sdm prefer ?  
access          Access bias  
default         Default bias  
dual-ipv4-and-ipv6 Support both IPv4 and IPv6  
ipe             IPe bias  
lanbase-routing Unicast bias  
vlan            VLAN bias
```

You need the lanbase-routing SDM template to support IP routing on the 2960 Switch. Also note that a change in the SDM template requires a reload to take effect. You can check the SDM template by using the show sdm prefer command. A sample output from a switch is shown below:

```
Switch#show sdm prefer  
The current template is desktop default template.  
The selected template optimizes the resources in  
the switch to support this level of features for  
8 routed interfaces and 1024 VLANs.
```

number of unicast mac addresses:	6K
number of IPv4 IGMP groups + multicast routes:	1K
number of IPv4 unicast routes:	8K
number of directly-connected IPv4 hosts:	6K
number of indirect IPv4 routes:	2K
number of IPv4 policy based routing aces:	0
number of IPv4/MAC qos aces:	0.5K
number of IPv4/MAC security aces:	1K

A very important point to note is that the 2960 Switch supports only static and default routing (more on this later). It is NOT a multilayer switch because it doesn't support the ip routing command. You cannot configure routing protocols on the 2960. Please Google "Configuring static IP unicast routing Cisco 2960" for more information, but feel free to

ignore that search if you want to focus only on the CCNA exam topics.

VLAN Trunking Protocol

Imagine having 20, 50, 100, or more switches on your network. If you made a VLAN change on one of the switches, you would have to configure the others manually to have the same information. This process would take many hours, and of course there would be scope for configuration errors.

As you may have already guessed, there is a way to make a change on one switch and have this change automatically propagate to the other switches. This method maintains consistency throughout the network and is known as VLAN Trunking Protocol (VTP).

VTP is a Cisco proprietary layer 2 protocol that advertises VLAN configuration information throughout the switch infrastructure (so long as certain parameters are met). VTP advertises the VLAN name, the VLAN ID, and the type of VLAN for every VLAN. One thing VTP can't do is decide which ports on each switch should be in which VLAN. VTP ensures that up-to-date and consistent VLAN information exists throughout the switching domain.

Version 2 of VTP introduced support for token ring networks, which are now obsolete so you are fine with using version 1. VTP version 3 (on the 2960 Series Switch, this is supported only when the switch is running the LAN base image) supports the following features that are not supported in version 1 or version 2:

- Enhanced authentication
- The extended range of VLAN (VLANs 1006 to 4094) database propagation (VTP versions 1 and 2 propagate only VLANs 1 to 1005; if extended VLANs are configured, you cannot convert from VTP version 3 to version 1 or 2)
- Private VLAN support
- Any database in a domain
- VTP primary server and VTP secondary servers (a VTP primary server updates the database information and sends updates that are honored by all devices in the system, while a VTP secondary server can only back up the updated VTP configurations received from the primary server to its NVRAM)
- The option to turn VTP on or off on a per-trunk (per-port) basis

Each switch has to be configured to join the same VTP domain in order to exchange VTP information. To join the domain, you simply give each switch the same VTP domain name, like "Cisco," for example (see the configuration below). Only switches in the same VTP management domain will share information. VTP information can be protected with a password. Moreover, all of the switches in the VTP domain need to have the same password in order to decrypt the VTP packets.

The benefits of using VTP include:

- Accurate monitoring and reporting of VLANs
- VLAN consistency across the network
- Ease of adding and removing VLANs

Each switch using VTP advertises the management domain, the revision number of the configuration, and known VLANs (and their parameters) out of their trunk ports. Figure 2.16 below shows a capture of a VTP frame:

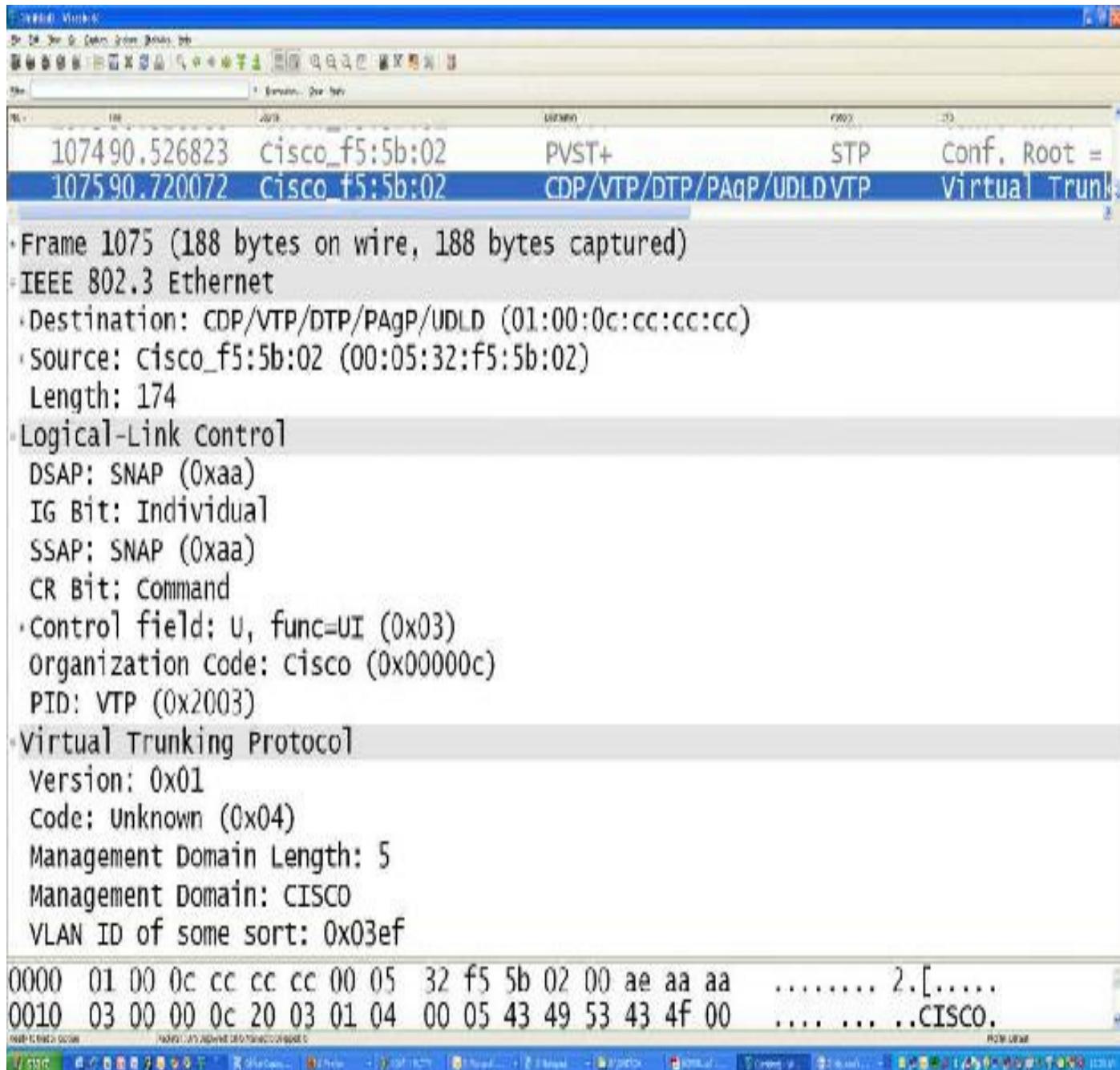


FIG 2.16 – VTP frame capture

In Figure 2.17 below you can see Switch1 sending a VTP update out of all trunk ports to

switches in the same domain. It is advertising the creation of VLAN 100 and this change is propagated throughout the switch domain.

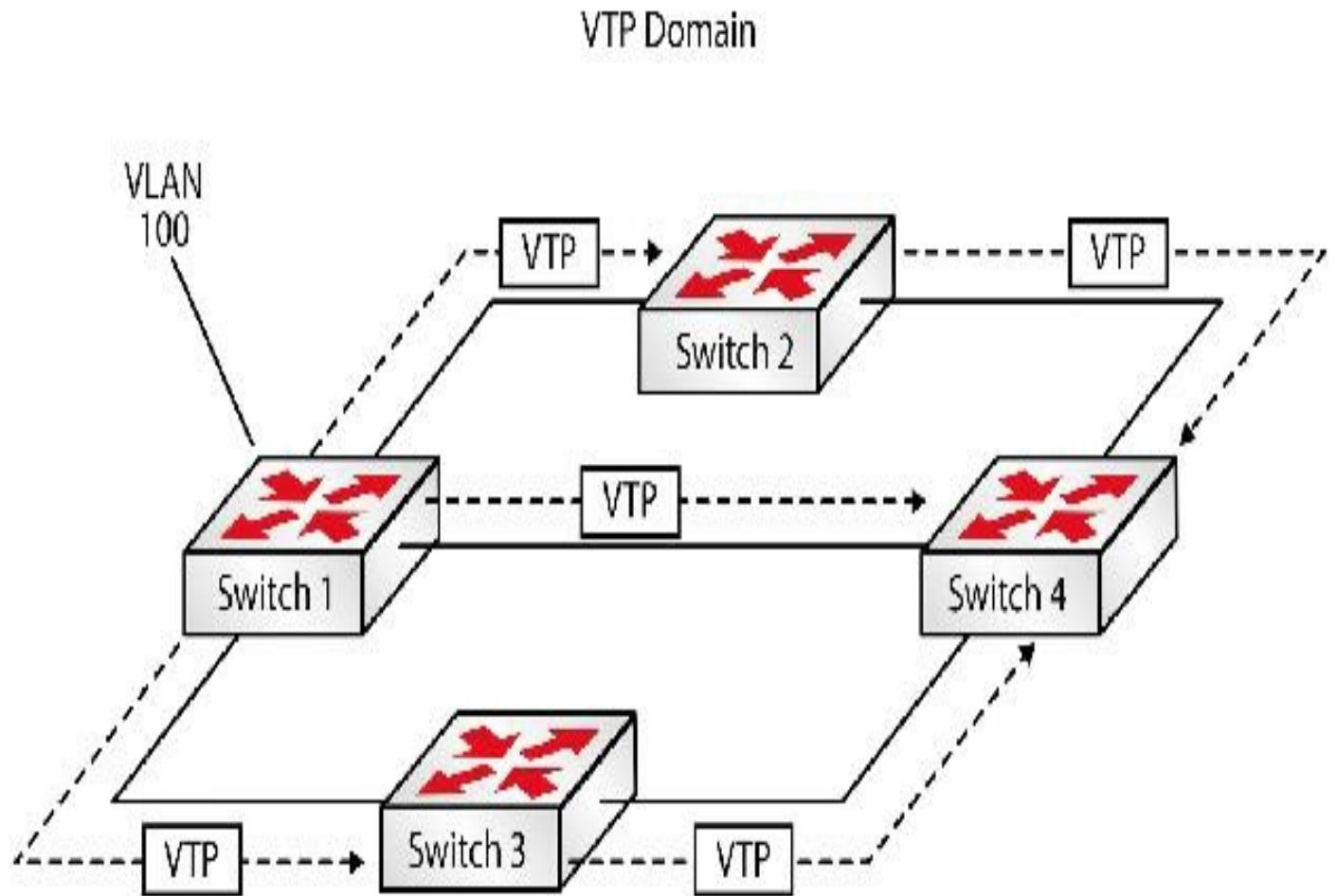


FIG 2.17 – VTP updates

In the output below I have Switch 1 and Switch 2 connected via a trunk interface (manually set to trunk on Switch 2 with the switchport mode trunk IOS command and set to auto on Switch 1). You can see that Switch 2 has the default VLANs only (which we covered earlier). The show vlan brief command will show the VLANs and interfaces in the relevant VLANs. It will not, however, show the trunk interfaces.

Switch2#show vlan brief

VLAN Name	Status	Ports
<hr/>		
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4
		Fa0/5, Fa0/6, Fa0/7, Fa0/8
		Fa0/9, Fa0/10, Fa0/11, Fa0/12
		Fa0/13, Fa0/14, Fa0/15, Fa0/16

```
Fa0/17, Fa0/18, Fa0/19, Fa0/20  
Fa0/21, Fa0/22, Fa0/23, Fa0/24  
Gig1/1, Gig1/2
```

```
1002 fddi-default active
```

```
1003 token-ring-default active
```

```
1004 fddinet-default active
```

```
1005 trnet-default active
```

You can check that the interface is actually set to trunk with the command below:

```
Switch1#show int trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/1	auto	n-802.1q	trunking	1

You can see that the encapsulation is 802.1Q and the n means that it was negotiated (this output is from the Cisco 3350 Switch, which also runs ISL, so we know that Switch 2 was set to 802.1Q and manually to trunk). Now issue the show vtp status command to give you a snapshot of the current VTP settings:

```
Switch1#show vtp status
```

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 255

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name :

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x7D 0x5A 0xA6 0x0E 0x9A 0x72 0xA0 0x3A

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 0.0.0.0 (no valid interface found)

There is no name configured yet but the default mode is server (although it will be transparent on some platforms). You can also see that no configuration modifications

have occurred yet because the configuration revision is set to zero. The only VLANs present are the default ones, which total five. I'll set a VTP domain name on Switch 2. Adding a VTP domain name triggers VTP updates to be sent from the switch. If a neighbor switch has no VTP domain name, it will accept the incoming name and add it to its own configuration (see below):

```
Switch2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch2(config)#vtp domain howtonetwork.com
```

Changing VTP domain name from NULL to howtonetwork.com

You can see that this change has been propagated to Switch 1 already:

```
Switch1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch1(config)#vtp domain howtonetwork.com
```

Domain name already set to howtonetwork.com.

I will now add VLAN 100 to Switch 1 and name it:

```
Switch1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch1(config)#vlan 100
```

```
Switch1(config-vlan)#name ADMIN
```

```
Switch1(config-vlan)#end
```

```
Switch1#
```

This change is propagated to Switch 2:

```
Switch2#show vlan brief
```

VLAN Name	Status	Ports
<hr/>		
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5
		Fa0/6, Fa0/7, Fa0/8, Fa0/9
		Fa0/10, Fa0/11, Fa0/12, Fa0/13
		Fa0/14, Fa0/15, Fa0/16, Fa0/17
		Fa0/18, Fa0/19, Fa0/20, Fa0/21

Fa0/22, Fa0/23, Fa0/24, Gig1/1
Gig1/2

100 ADMIN active
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active

You may also have noticed that the configuration revision number increased each time you made a change on a neighbor switch and the number of VLANs increased each time a new VLAN was created. (It currently shows as 2 because I may have done some testing during the commands above but not shown them.) The VTP revision number tells the switch whether an incoming update should be accepted. If the number is higher than the current revision number it will be accepted and applied.

Switch2#sh vtp status

VTP Version: 2

Configuration Revision: 2

Maximum VLANs supported locally: 255

Number of existing VLANs: 6

VTP Operating Mode: Server

VTP Domain Name: howtonetwork.com

VTP Pruning Mode: Disabled

VTP V2 Mode: Disabled

VTP Traps Generation: Disabled

MD5 digest: 0x2C 0x18 0xE9 0x5E 0x50 0x62 0xE2 0x92

Configuration last modified by 10.1.1.1 at 3-1-93 03:08:39

Local updater ID is 0.0.0.0 (no valid interface found)

If you add a switch to your network and it has a higher configuration revision number than your current one, you will run the risk of all your network switches accepting the new VTP information by wiping all your current VLANs. This can also happen even if the switch is set to client. Please Google “Adding a VTP client switch to a VTP domain” for more information from Cisco on this (this is probably not a CCNA exam

topic, so feel free to revisit this after the exam).

It is also worth noting that if your switches have been allocated a management IP address, this will show as the originator of the VTP update as you can see in the 10.1.1.1 address in the output above. If there are faulty updates being sent, you can quickly identify the source this way.

There are other outputs on the show command above; however, they are outside the scope of the CCNA exam. I recommend that you try these commands on your home CCNA rack or a remote rack if you have access.

VTP Modes

There are three possible modes a switch in a VTP domain can be in:

- Client
- Server (default mode but this is platform-dependent)
- Transparent

Client Mode

In client mode, the switch will receive VTP information and apply any changes, but it does not allow adding, removing, or changing VLAN information on the switch.

Switch#conf t

Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#vtp mode client

Setting device to VTP CLIENT mode.

Switch(config)#vlan 10

%VTP VLAN configuration not allowed when device is in CLIENT mode.

The client will also send out the VTP packet received out of its trunk ports. It cannot store any VLAN information in NVRAM so a reload will wipe any VLAN information.

Be warned that if you took the VTP client switch off the network and made it a server, added some configurations, set it back to client, and added it to the network, it would cause the changes to propagate throughout the entire VTP domain. The reason is that the revision number is higher AND the VTP server is also a client in that if it receives a configuration with a higher revision number, it will accept that information as correct.

Server Mode

In server mode, the switch is authorized to create, modify, and delete VLAN information for the entire VTP domain. Any changes that you make to a server are propagated

throughout the whole domain. Switches are in server mode by default (platform-dependent). All VLAN information is stored in NVRAM on the VTP server in a file called `vlan.dat` and remains present after a reload (so don't try to wipe the configuration files via reload).

You can have more than one server in a VTP domain; however, whichever one you add the configuration to will update the revision number and send all changes throughout the domain to other servers, clients, and transparent switches.

Transparent Mode

In transparent mode, the switch will forward the VTP information received out of its trunk ports but will not apply the changes. A VTP transparent mode switch can create, modify, and delete VLANs, but the changes are not propagated to other switches. All information is locally significant and not forwarded. VTP transparent mode also requires configuration of domain information. A VTP transparent switch is needed when a switch separating a VTP server and client needs to have a different VLAN database.

You would need to put a switch into transparent mode if you wanted to add VLAN numbers above 1024 or use private VLANs (a CCNP topic). The extended range of usable VLANs available is 1006 to 4094, inclusive. If you want to do some further reading or need to configure extended VLANs for work, please Google "Cisco extended VLAN ID."

VTP version 3 also allows you to add VLAN numbers over 1024.



```
Switch(config)#vlan 2000
```

```
Switch(config-vlan)#end
```

```
% Failed to create VLANs 2000
```

```
Extended VLAN(s) not allowed in current VTP mode.
```

```
%Failed to commit extended VLAN(s) changes.
```

```
Switch(config)#vtp mode transparent
```

Setting device to VTP TRANSPARENT mode.

```
Switch(config)#vlan 2000
```

```
Switch(config-vlan)#end
```

Table 2-5: VTP characteristics

Capability	VTP Server	VTP Client	VTP Transparent
Sending VTP Messages	Yes	Yes	No
Listening to VTP Messages	Yes	Yes	No
Creating VLANs	Yes	No	Yes (locally)
Deleting VLANs	Yes	No	Yes (locally)
Modifying VLANs	Yes	No	Yes (locally)
Storing VLANs	Yes	No	Yes (locally)

Figure 2.18 below shows VTP in action:

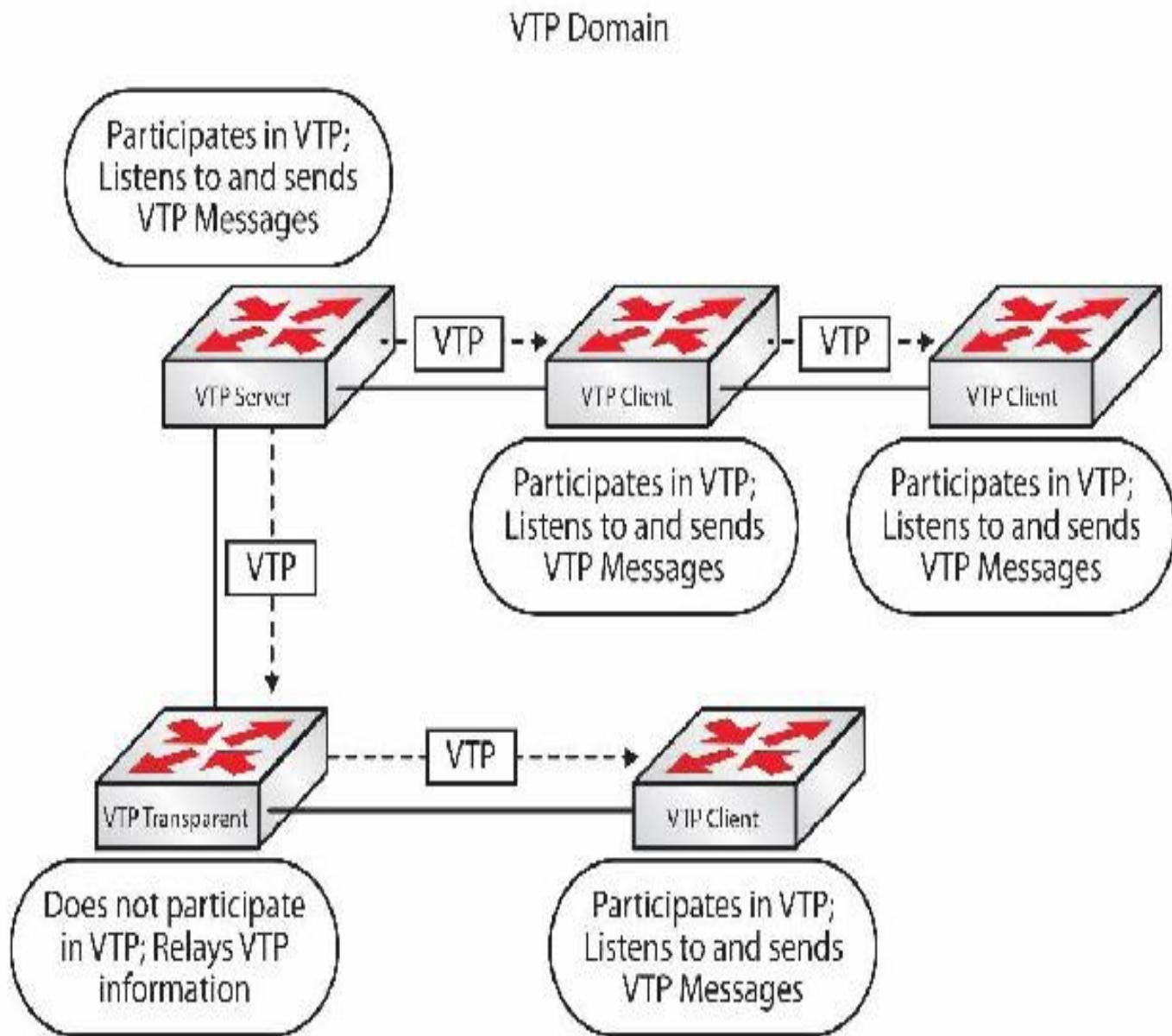


FIG 2.18 – VTP domain

Mini-lab – Configuring VTP

First, connect two switches together with a crossover cable. To enable VTP, you need to configure the VTP domain and, optionally, the VTP mode and VTP password. Note that on some platforms, I've noticed that no changes will propagate unless there is a password (this is the benefit of using live equipment—you get to discover certain glitches).

Configure a trunk link between the two switches with the switchport mode trunk command. Add VLAN 5 to Switch A and name it VTPTest.

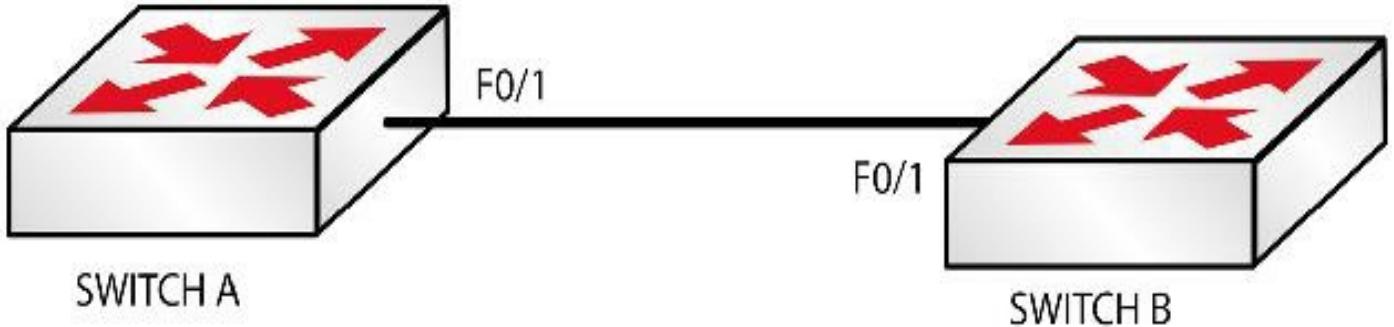


FIG 2.19 – Mini-lab: Configuring VTP

The following commands accomplish this (switches will be in server mode by default):

```

SwitchA(config)#vtp mode ?
client    Set the device to client mode.
server    Set the device to server mode.
transparent Set the device to transparent mode.
SwitchA(config)#vtp mode server
SwitchA(config)#vtp domain Cisco
SwitchA(config)#vtp password ccna
SwitchA(config)#vlan 5
SwitchA(config-vlan)#name VTPTest

```

The vtp mode [server|client|transparent] command can be used to set the desired mode. VTP configuration can be verified using the show vtp status command:

```

SwitchA#show vtp status
VTP Version: 2
Configuration Revision: 1
Maximum VLANs supported locally: 255
Number of existing VLANs: 9
VTP Operating Mode: Server
VTP Domain Name: Cisco
VTP Pruning Mode: Disabled
VTP V2 Mode: Disabled
VTP Traps Generation: Disabled
MD5 digest: 0x5D 0x16 0x1A 0x34 0x2C 0xAE 0xA5 0xB4
Configuration last modified by 0.0.0.0 at 3-1-02 00:37:56
Local updater ID is 0.0.0.0 (no valid interface found)

```

Now that the VTP server is configured on Switch A, let's take a look at both Switch A and Switch B, which is connected to Switch A using an 802.1Q trunk on port fa1/1. The show vlan outputs for both switches are shown below:

SwitchA#show vlan

VLAN Name	Status	Ports
1 default	active	Fa1/0, Fa1/2, Fa1/3, Fa1/4 Fa1/5, Fa1/6, Fa1/7, Fa1/8 Fa1/9, Fa1/10, Fa1/11, Fa1/12 Fa1/13, Fa1/14, Fa1/15
2 VLAN0002	active	
3 VLAN0003	active	
4 VLAN0004	active	
5 VTPTest	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SwitchB#show vlan

VLAN Name	Status	Ports
1 default	active	Fa1/0, Fa1/2, Fa1/3, Fa1/4 Fa1/5, Fa1/6, Fa1/7, Fa1/8 Fa1/9, Fa1/10, Fa1/11, Fa1/12 Fa1/13, Fa1/14, Fa1/15
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

Let's configure Switch B as a VTP client to Switch A and get the VTP configuration to match Switch A:

```
SwitchB(config)#vtp mode client
SwitchB(config)#vtp domain Cisco
SwitchB(config)#vtp password ccna
```

The output of the show vlan and show vtp status commands on Switch B now looks like

this:

SwitchB#show vlan

VLAN Name	Status	Ports
1 default	active	Fa1/0, Fa1/2, Fa1/3, Fa1/4 Fa1/5, Fa1/6, Fa1/7, Fa1/8 Fa1/9, Fa1/10, Fa1/11, Fa1/12 Fa1/13, Fa1/14, Fa1/15
2 VLAN0002	active	
3 VLAN0003	active	
4 VLAN0004	active	
5 VTPTTest	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

SwitchB#show vtp status

VTP Version: 1

Configuration Revision: 2

Maximum VLANs supported locally: 256

Number of existing VLANs: 9

VTP Operating Mode: Client

VTP Domain Name: Cisco

VTP Pruning Mode: Disabled

VTP V2 Mode: Disabled

VTP Traps Generation: Disabled

MD5 digest: 0x94 0xDE 0x28 0x9D 0x2D 0x95 0x96

Configuration last modified by 0.0.0.0 at 3-1-02 00:18:45

[END OF MINI-LAB]

In Figure 2.20 below, I've connected two switches together and created a trunk link between them. Both have matching VTP domain names and passwords. Switch A has management IP address 192.168.1.1 (so we can connect to it across the network in order to manage it) and Switch B has 192.168.1.2. I added a VLAN to Switch B and the update was sent to Switch A. You can see the IP address of Switch A at the bottom of

the output.

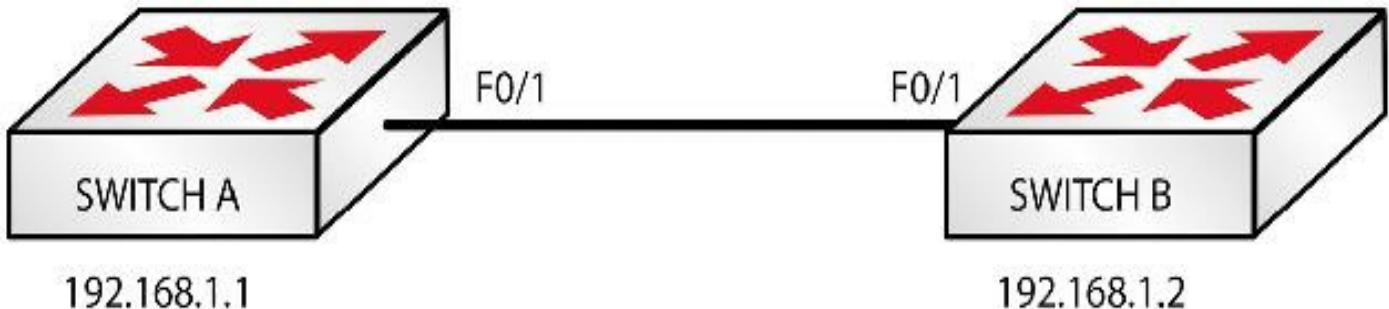


FIG 2.20 – VTP across two switches

```
SwitchA#show vtp status
```

VTP Version: 2

Configuration Revision: 1

Maximum VLANs supported locally: 255

Number of existing VLANs: 6

VTP Operating Mode: Server

VTP Domain Name: Cisco

VTP Pruning Mode: Disabled

VTP V2 Mode: Disabled

VTP Traps Generation: Disabled

MD5 digest: 0x18 0xD3 0x67 0x97 0x7E 0xD3 0xAA

Configuration last modified by 192.168.1.2 at 3-1-93 00:03:11

Local updater ID is **192.168.1.1** on interface Vl1 (lowest numbered VLAN interface found)

```
SwitchA#
```

The vlan.dat files were referred to earlier. In the output below you can see it in flash memory (NVRAM):

```
Cat2960-Switch#show flash
```

Directory of flash:/

```
2 -rwx 2888547 Mar 01 1993 00:03:33 c2960-i6q412-mz.151-13.EA1.bin
```

```
4 -rwx 924 Mar 01 1993 02:28:44 vlan.dat
```

7741440 bytes total (2867200 bytes free)

If you want to remove the configuration, you can issue the delete flash:vlan.dat command and then reload the switch. If you want to remove the current configuration on a switch without rebooting, you can set it to transparent mode and then back to server mode with the vtp mode transparent command, or by changing the VTP domain name to

something else and then back again with the vtp domain [name] command.

A gotcha that you may have already spotted is that if you erase vlan.dat, the switch can still add the old configuration if it receives it via VTP from a connected switch.

```
Switch1(config)#vtp mode transparent
```

Setting device to VTP TRANSPARENT mode.

```
Switch1(config)#vtp domain cisco
```

Changing VTP domain name from howtonetwork.com to cisco

VTP Pruning

VTP information can be passed around the domain and reach the same switch from several paths. To add to this, every switch that has a trunk link forwards broadcasts to every other switch, even if it has no ports in that VLAN. The end result is broadcasts to switches that will eventually discard the traffic.

VTP pruning prevents VLAN information, broadcasts, multicasts, and unicasts from flooding other trunk ports when there is no need. It also reduces unnecessary broadcast, multicast, and flooded unicast traffic, which in turn increases available bandwidth. To understand when you would use this, let's look at Figure 2.21. Since Switch B has ports only in VLAN 5, it would be wise to ensure that traffic not belonging to this VLAN doesn't traverse the trunk link on Switch A.

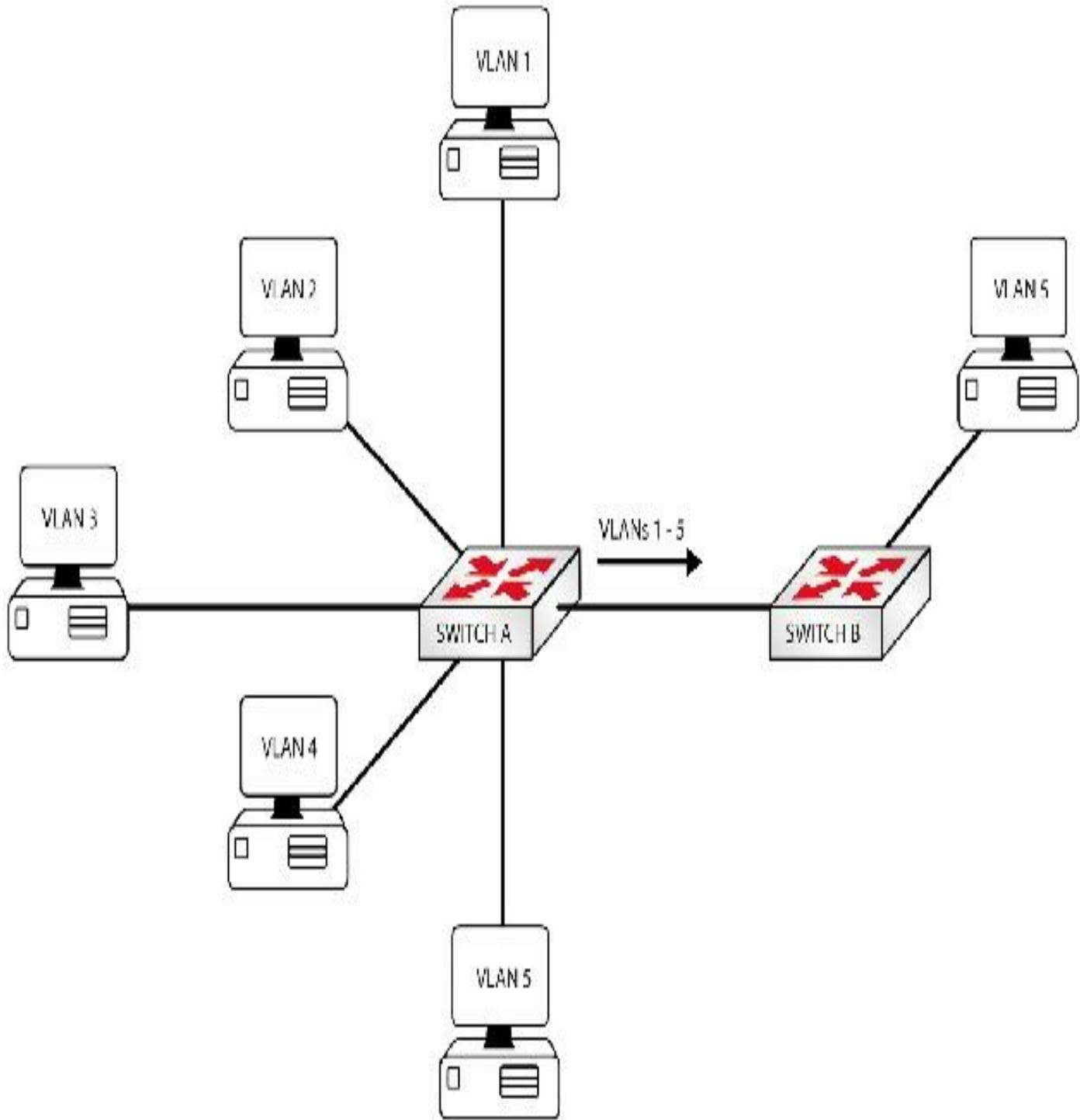


FIG 2.21 – All VLANs advertised

VTP pruning can only be enabled from the VTP server. Once enabled there, it will automatically be enabled on all the VTP clients. VTP pruning can be enabled using the `vtp pruning` command on the VTP server. This in turn will enable the pruning capability in other switches in the VTP domain.

`SwitchA#show vtp status`

VTP Version: running VTP1 (VTP2 capable)

Configuration Revision: 0

Maximum VLANs supported locally: 1005

Number of existing VLANs: 5

VTP Operating Mode: Server

VTP Domain Name:

VTP Pruning Mode: **Disabled**

SwitchA(config)#vtp pruning

Pruning switched on

SwitchA(config)#end

SwitchA#show vtp status

VTP Version: running VTP1 (VTP2 capable)

Configuration Revision: 1

Maximum VLANs supported locally: 1005

Number of existing VLANs: 5

VTP Operating Mode: Server

VTP Domain Name:

VTP Pruning Mode: **Enabled**

Let's see the output of show interface trunk on Switch A after VTP pruning is enabled:

SwitchA#show interface trunk

Port Mode Encapsulation Status Native vlan

Fa1/1 on 802.1q trunking 1

Port Vlans allowed on trunk

Fa1/1 1,5

Port Vlans allowed and active in management domain

Fa1/1 1,5

Port Vlans in spanning tree forwarding state and not pruned

Fa1/1 1,5

The output of the show interface trunk command shows the interfaces that are operating as trunks, as well as the VLANs that are allowed on the trunk and the VLANs that are in

forwarding state and are not pruned. You can't prune VLAN 1.

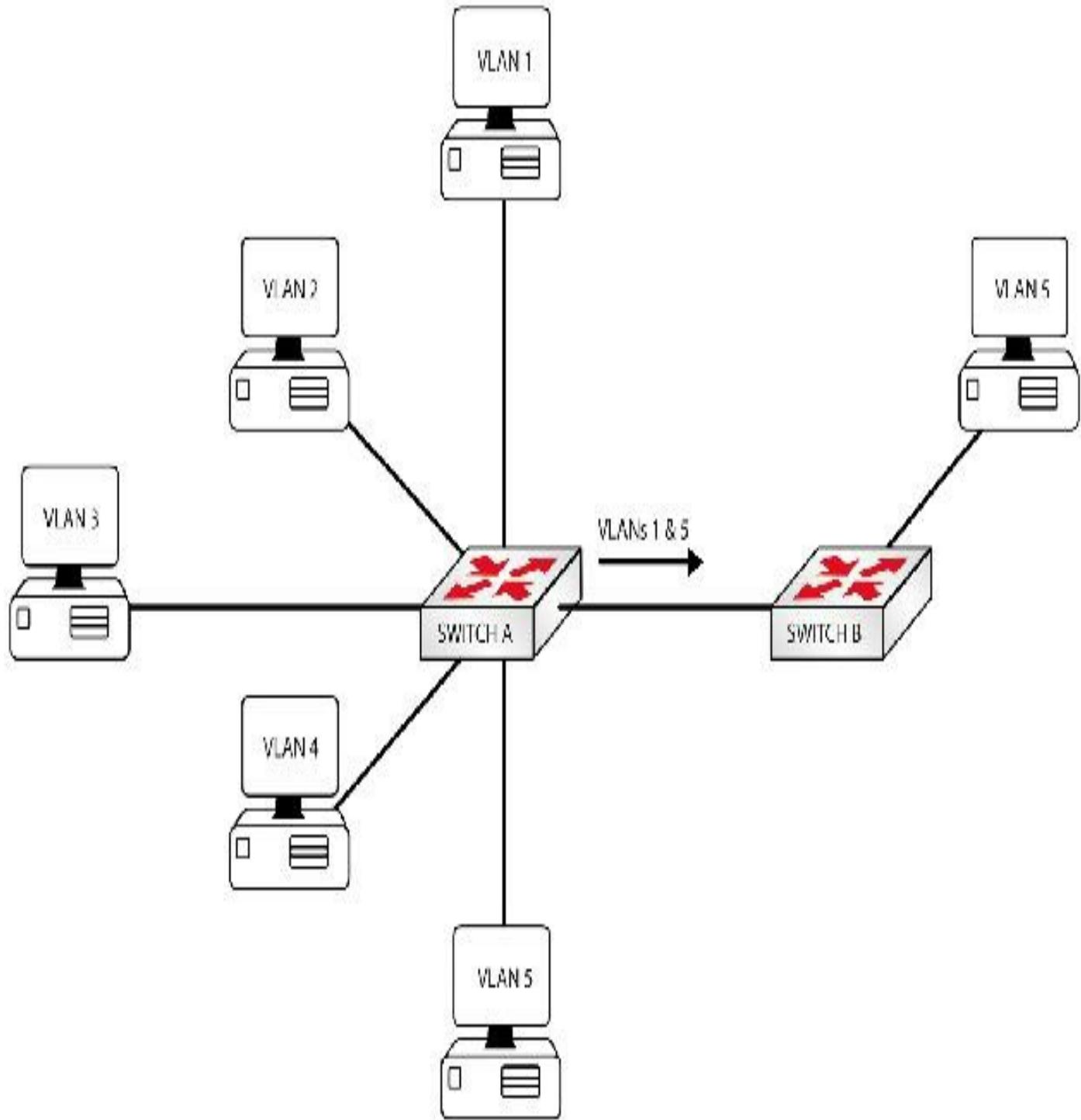


FIG 2.22 – VLANs pruned

Note that VLANs 2,3, and 4 are pruned on interface Fa0/1 since Switch B does not have any hosts in those VLANs. There is a manual method to prevent VLANs from crossing a trunk link, which will be covered in the switch security section in Chapter 8.

If there is a VTP transparent switch in between the VTP server and client, then pruning will not work. You will cover this in more detail if you progress to the CCNP RS exam.

Configuring a Cisco Switch

Let's cover a few basic configuration commands for your switch. Some of these commands have already been mentioned and others will be revisited as we progress through this guide. Just type them out for now so you get some hands-on experience using them.

You can set the hostname of the switch:

```
Switch>enable
```

```
Switch#config t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch(config)#hostname 2960
```

```
2960(config)#exit
```

And look at the version of IOS running on it:

```
2960#show version
```

Cisco Internetwork Operating System Software

IOS (tm) C2960 Software (**C2960-I6Q4L2-M**), Version **15.1(20)EA1a**, RELEASE SOFTWARE (fc1)

Copyright (c) 1986-2004 by Cisco Systems, Inc.

Compiled Mon 19-Apr-04 20:58 by yenanh

Image text-base: 0x80010000, data-base: 0x805A8000

ROM: Bootstrap program is C2960 boot loader

Switch uptime is 11 minutes

System returned to ROM by power-on

System image file is flash:/**c2960-i6q4l2-mz.151-20.EA1a.bin**

cisco WS-C2960G-12-EI (RC32300) processor (revision E0) with 20713K bytes of memory.

Processor board ID FHK0652X0PY

Last reset from system-reset

Running Enhanced Image

12 FastEthernet/IEEE 802.3 interface(s)

2 GigabitEthernet/IEEE 802.3 interface(s)

32K bytes of flash-simulated non-volatile configuration memory.

Base Ethernet MAC Address: 00:0C:BE:D4:3C:40

Motherboard assembly number: 34-7410-05

Power supply part number: 34-0475-02

Motherboard serial number: FUY00LWXZ
Power supply serial number: PHI0648897W
Model revision number: E0
Motherboard revision number: A0
Model number: WS-C2960G-12-EI
System serial number: FHK0652X0PY
Configuration register is 0xF

Switches can be managed remotely in the same way a router can (using the management IP address). A default gateway can also be configured on the switch. Since the switch ports are layer 2 ports, they cannot be assigned an IP address. For this reason, an SVI is created on the switch. The SVI is a layer 3 interface and every VLAN can have one (the number depends on your model and IOS). SVIs are named after the VLAN ID, such as “Interface VLAN 1.” SVIs can be assigned an IP address.

The default management VLAN is VLAN 1. You can, however, change this and we’ll do this later on.

For a management interface to become active you must create the VLAN, add an interface to the VLAN, or configure a trunk link containing this particular VLAN, and then configure an IP address on the relevant SVI and a default gateway for IP traffic. You saw this earlier in the SVI configuration commands, but we’ll cover it again.

```
2960#conf t
Enter configuration commands, one per line. End with CNTL/Z.
2960(config)#interface vlan 2
2960(config-if)#ip address 172.16.100.1 255.255.0.0
2960(config-if)#no shut
2960(config-if)#exit
2960(config)#ip default-gateway 172.16.1.1
2960(config)#^Z
2960#
00:22:51: %SYS-5-CONFIG_I: Configured from console by console
2960#conf t
Enter configuration commands, one per line. End with CNTL/Z.
2960(config)#interface FastEthernet0/1
2960(config-if)#switchport mode access
2960(config-if)#switchport access vlan 2
2960(config-if)#exit
```

With the configuration above you can ping the switch and the switch will know to send

all IP traffic to 172.16.1.1 via the ip default-gateway 172.16.1.1 command. Note that, because the switch cannot route as such, the default gateway should be in the same subnet as the SVI. You must also apply the no shut command to the SVI to make it operational. This command will also enable you to telnet to the switch in order to manage it over the network.

You can set a local username and password so that any users telnetting into the switch will need to authenticate themselves. You can also apply encryption to the password so that it won't display if a user issues the show run command:

```
Switch(config)#username paul password cisco
```

```
Switch(config)#service password-encryption
```

If you want the console line to terminate any idle sessions you can apply a timeout value to it:

```
Switch(config)#line console 0
```

```
Switch(config-line)#exec-timeout ?
```

```
<0-35791> Timeout in minutes
```

```
Switch(config-line)#exec-timeout 5
```

```
Switch(config-line)#
```

You can configure the port to be either an access port or a trunk port. We will take a look at the access port first.

```
Switch(config)#int f0/1
```

```
Switch(config-if)#switchport mode access
```

```
Switch(config-if)#switchport access vlan 5
```

The first command tells the port to be an access port. This port will not try to establish a connection as a trunk. The second switchport command tells the port which VLAN it belongs to, in this case VLAN 5.

Now, let's configure a port for trunking. You will need to be connected to another switch at this point.

```
Switch(config)#int fast0/1
```

```
Switch(config-if)#switchport mode trunk
```

It's worth noting that if the other side of the trunk link has been left on auto for the encapsulation type, you will see the letter n in front of the encapsulation type. Note that

different switch models will have different default settings.

```
Switch#sh int trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/11	desirable	n-802.1q	trunking	1

If you wanted to telnet to the switch to configure it remotely, you would need to add line VTY configuration and enable secret/password configuration similar to a router, as well as a management IP address and default gateway (where to send all the IP traffic to). We will cover all of this as we progress through this guide, but in the meantime, please just follow along with the commands.

Mini-lab – IP Default Gateway on a Switch

Figure 2.23 below demonstrates a simple router-to-switch connection using the default VLAN 1. I added an IP address to VLAN 1 on the switch, creating an SVI and pointing all traffic to the router. I also added some basic configuration commands, which will be addressed as we progress through the guide.

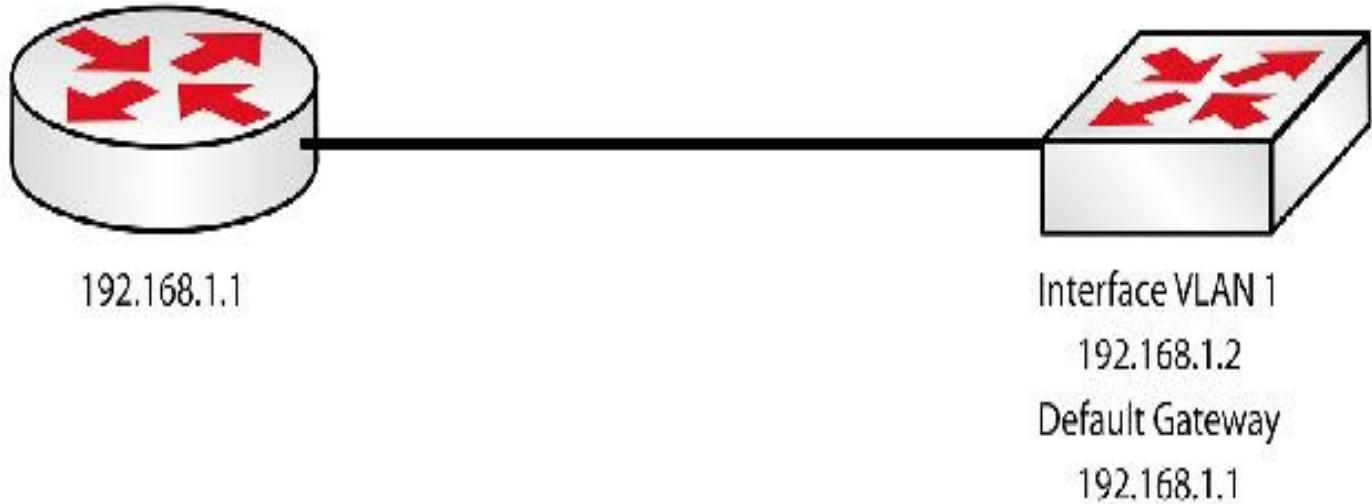


FIG 2.23 – Mini-lab: IP Default Gateway on a Switch

```
Switch#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Switch(config)#interface vlan 1
```

```
Switch(config-if)#ip add 192.168.1.2 255.255.255.0
```

```
Switch(config-if)#no shut
```

```
Switch(config-if)#exit
```

```
Switch(config)#ip default-gateway 192.168.1.1 i The default gateway
```

```
Switch(config)#enable secret cisco123 i Protects enable mode
Switch(config)#line vty 0 ?
<1-15> Last Line number
<cr>
Switch(config)#line vty 0 15 i Permits Telnet
Switch(config-line)#password cisco i Protects incoming Telnet sessions
Switch(config-line)#login
Switch(config-line)#end
Switch#copy run start i Saves the configuration
Destination filename [startup-config]?
Building configuration...
[OK]
Switch#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int f0/0
Router(config-if)#ip add 192.168.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#end
Router#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
...!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 0/0/0 ms
Router#telnet 192.168.1.2
Trying 192.168.1.2 ...Open
User Access Verification
Password: i Won't show when typed
Switch>en
```

Password:

Switch#

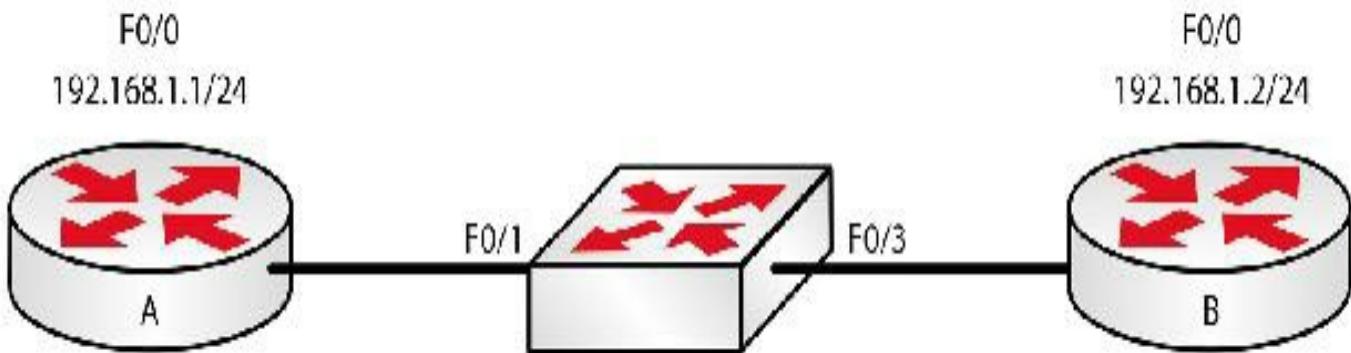
End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 2 exam.

Chapter 2 Labs

Lab 1: VLANs on an IOS Switch

The physical topology is shown in Figure 2.24 below:



Lab Exercise

Your task is to configure the network in Figure 2.24 above. All the router and switch interfaces are in VLAN 2. If you do not want to use routers, you can use PCs and configure the IP address on the Ethernet card instead. Any switch running IOS will do for this lab.

Purpose

Creating VLANs on an IOS switch is one of the core competencies of a Cisco engineer. Make sure you are very familiar with doing this.

Lab Objectives

1. Configure the switch to use VLAN 2, name “Cisco.”
2. Place two interfaces on the switch in VLAN 2.
3. Configure the routers’ interfaces.
4. Ping across the LAN on VLAN 2.

Lab Walk-through

1. To configure the IP address on the routers, do the following:

```
Router>enable
```

```
Router#
```

```
Router#configure terminal
```

```
Router(config)#hostname RouterA
```

```
RouterA(config)#interface FastEthernet0/0
```

```
RouterA(config)#ip address 192.168.1.1 255.255.255.0
```

```
RouterA(config-if)#no shut  
RouterA(config-if)^Z  
RouterA#
```

Router B:

```
Router>enable  
Router#config t  
Router#hostname RouterB  
RouterB(config)#interface FastEthernet0/0  
RouterB(config)#ip address 192.168.1.2 255.255.255.0  
RouterB(config-if)#no shut  
RouterB(config-if)^Z  
RouterB#
```

If you have plugged directly into the switch, you will be able to ping from router to router or switch to switch. This is because the switch will use VLAN 1 by default. Give the switch ports 30 seconds to come up.

```
RouterA#ping 192.168.1.2  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms  
RouterA#
```

2. To configure both routers to connect to VLAN 2 on the switch, enter the following commands:

```
Switch#config t  
Switch(config)#vlan 2 i Creates VLAN 2  
Switch(config-vlan)#name Cisco  
Switch(config-vlan)^Z  
Switch#
```

3. To configure the interfaces on the switch to use VLAN 2, use the following commands:

```
Switch#config t  
Switch(config)#interface FastEthernet0/1  
Switch(config-if)#description ToRouterA i Sets the description  
Switch(config-if)#switchport mode access  
Switch(config-if)#switchport access vlan 2 i Adds to VLAN 2
```

```
Switch(config-if)#^Z
```

4. You can try to ping from Router B to Router A now. Since you have put only one interface into VLAN 2 (Router A), the second (Router B) remains in VLAN 1 (by default) and the ping will fail:

```
RouterB#ping 192.168.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

5. You now put the Ethernet interface connecting to Router B into VLAN 2:

```
Switch#config t
```

```
Switch(config)#interface FastEthernet0/3
```

```
Switch(config-if)#description ToRouterB
```

```
Switch(config-if)#switchport mode access
```

```
Switch(config-if)#switchport access vlan 2
```

```
Switch(config-if)#^Z
```

6. You can now ping across the LAN from Router A to Router B. The first one or two pings will fail until the switch ports have started to forward traffic:

```
RouterA#ping 192.168.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

```
RouterA#ping 192.168.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 4/4/4 ms

Show Runs

```
RouterA#show run
```

Building configuration...

Current configuration : 428 bytes

!

version 15.1

!

```
hostname RouterA
!
ip subnet-zero
!
interface FastEthernet0
ip address 192.168.1.1 255.255.255.0
!
end
```

- - -

```
RouterB#show run
Building configuration...

Current configuration : 428 bytes
!
version 15.1
no service single-slot-reload-enable
service timestamps debug uptime
no service password-encryption
!
hostname RouterB
!
ip subnet-zero
!
interface FastEthernet0
ip address 192.168.1.2 255.255.255.0
!
end
RouterB#
```

- - -

```
Switch
Switch#show run
Building configuration...

Current configuration:
!
version 15.0
no service password-encryption
```

```
!
hostname Switch
!
ip subnet-zero
!
interface FastEthernet0/1
description ToRouterA
switchport access vlan 2
!
interface FastEthernet0/2
!
interface FastEthernet0/3
description ToRouterB
switchport access vlan 2
!
!
end
```

Lab 2: Trunking across IOS Switches

Lab Exercise

Your task is to configure the network below. All the router interfaces on the switch and router are in VLAN 2. You will need two switches running Cisco IOS; I have used two 2960 Switches.

If you do not want to keep swapping between switches, then follow the entire configuration for one 2960 Switch. It's been broken down into smaller chunks to make the various stages more understandable. Also, feel free to swap the routers for PCs.

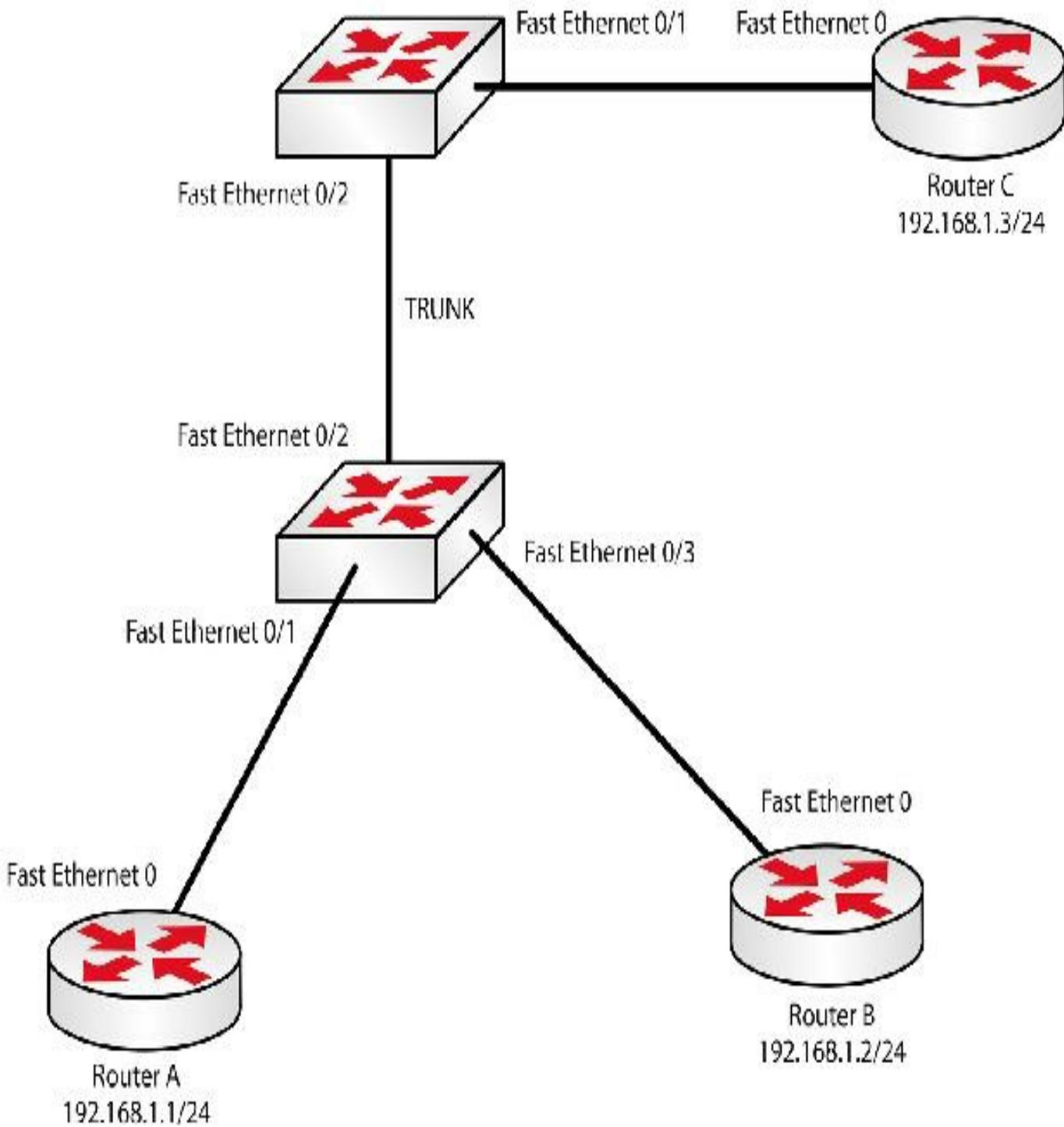


FIG 2.25 – Trunking across switches

Purpose

Being able to trunk between switches is an essential skill for a Cisco engineer.

NOTE: Issue a show run on the switches to check which interfaces they have and which one you are plugged into. You can also go through the entire switch configuration at once rather than in phases (as shown below) to save time.

Lab Objectives

1. Configure the switch to use VLAN 2, name “Cisco.”
2. Place two interfaces on the switch in VLAN 2 on one 2960 Switch and one interface in VLAN 2 on the other 2960 Switch.
3. Configure the routers’ interfaces with the IP addresses as in Figure 2.25 above.
4. Ping across the LAN on VLAN 2.

Lab Walk-through

1. To configure the IP address on the routers, do the following:

```
Router>enable
```

```
Router#
```

```
Router#configure terminal
```

```
Router(config)#hostname RouterA
```

```
RouterA(config)#interface FastEthernet0
```

```
RouterA(config-if)#ip address 192.168.1.1 255.255.255.0
```

```
RouterA(config-if)#no shut
```

```
RouterA(config-if)#^Z
```

```
RouterA#
```

Router B:

```
Router>enable
```

```
Router#config t
```

```
Router(config)#hostname RouterB
```

```
RouterB(config)#interface FastEthernet0
```

```
RouterB(config-if)#ip address 192.168.1.2 255.255.255.0
```

```
RouterB(config-if)#no shut
```

```
RouterB(config-if)#^Z
```

```
RouterB#
```

Router C:

```
Router>enable
```

```
Router#config t
```

```
Router(config)#hostname RouterC
```

```
RouterC(config)#interface FastEthernet0
```

```
RouterC(config-if)#ip address 192.168.1.3 255.255.255.0
```

```
RouterC(config-if)#no shut
```

```
RouterC(config-if)#^Z
```

```
RouterC#
```

If you have plugged directly into the switch, you will be able to ping from Router A to Router B and Router C. This is because they are all in VLAN 1, by default. If you have just booted up the switch, it may take a few moments for the database to be built.

```
RouterB#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
RouterB#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 4/4/4 ms
RouterB#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
RouterB#
```

2. Configure VLAN 2 on the IOS switches:

```
Switch>enable
Switch#config t
Switch(config)#hostname Top2960
Top2960(config)#vlan 2
Top2960(config-vlan)#name Cisco
Top2960(config-vlan)#^z
```

Now configure VLAN 2 on the bottom 2960 Switch:

```
Switch>enable
Switch#config t
Switch(config)#hostname Bottom2960
Bottom2960(config)#vlan 2
Bottom2960(config-vlan)#name Cisco
Bottom2960(config-vlan)#^z
```

3. Put the relevant ports in VLAN 2 on each switch:

```
Top2960#config t
Top2960(config)#interface fast0/1
Top2960(config-if)#switchport access vlan 2
Top2960(config-vlan)#^z
Top2960#
```

You could also add the switchport mode access command to the interface if you want to hard set it to access:

- - -

```
Bottom2960#config t
Bottom2960(config)#interface fast0/1
Bottom2960(config-if)#switchport access vlan 2
Bottom2960(config-vlan)#int fast0/3
Bottom2960(config-if)#switchport access vlan 2
Bottom2960(config-if)#^z
Bottom2960#
```

4. Turn trunking on—on the interfaces between the switches, do the following:

```
Bottom2960(config-if)#interface FastEthernet0/2
Bottom2960(config-if)#switchport mode trunk
Bottom2960(config)#exit
Bottom2960#
```

- - -

```
Top2960(config-if)#interface FastEthernet0/2
Top2960(config-if)#switchport mode trunk
Top2960(config)#exit
Top2960#
```

5. Ping from Router C to Router A:

```
RouterC#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.!!!! i One ping fails due to the ARP lookup
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/4/4 ms
RouterC#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
RouterC#
```

Show Runs

```
RouterA#show run
```

```
Building configuration...
```

```
!
version 15.1
!
hostname RouterA
!
ip subnet-zero
!
interface FastEthernet0
ip address 192.168.1.1 255.255.255.0
!
end
```

```
---
```

```
RouterB#show run
Building configuration...
```

```
!
Current configuration : 428 bytes
!
version 15.1
!
hostname RouterB
!
ip subnet-zero
!
interface FastEthernet0
ip address 192.168.1.2 255.255.255.0
!
end
```

```
---
```

```
RouterC#show run
```

Building configuration...

Current configuration:

```
!
version 15.0
!
hostname RouterC
!
interface FastEthernet0
ip address 192.168.1.3 255.255.255.0
!
end
```

Switch Show Runs (truncated)

Top 2960

Top2960#show run

Building configuration...

Current configuration:

```
!
hostname Top2960
!
ip subnet-zero
!
interface FastEthernet0/1
switchport access vlan 2
!
interface FastEthernet0/2
switchport mode trunk
!
```

- - -

Bottom2960#show run

Building configuration...

Current configuration:

```
!
hostname Bottom2960
!
```

```
ip subnet-zero
!
interface FastEthernet0/1
switchport access vlan 2
!
interface FastEthernet0/2
switchport mode trunk
!
interface FastEthernet0/3
switchport access vlan 2
```

Lab 3: InterVLAN Routing

Lab Exercise

Your task is to configure the network such that PC-A in VLAN 2 can ping PC-B in VLAN 3 across the switches. In the topology shown in Figure 2.26 below, you can always swap the PCs for routers and use the Fast Ethernet interfaces to connect to the switches.

Purpose

This topology is known as a router on a stick. When you have VLANs in your network, they must each reside in their own subnet. In order for subnets to be able to communicate, you must have a layer 3 device.

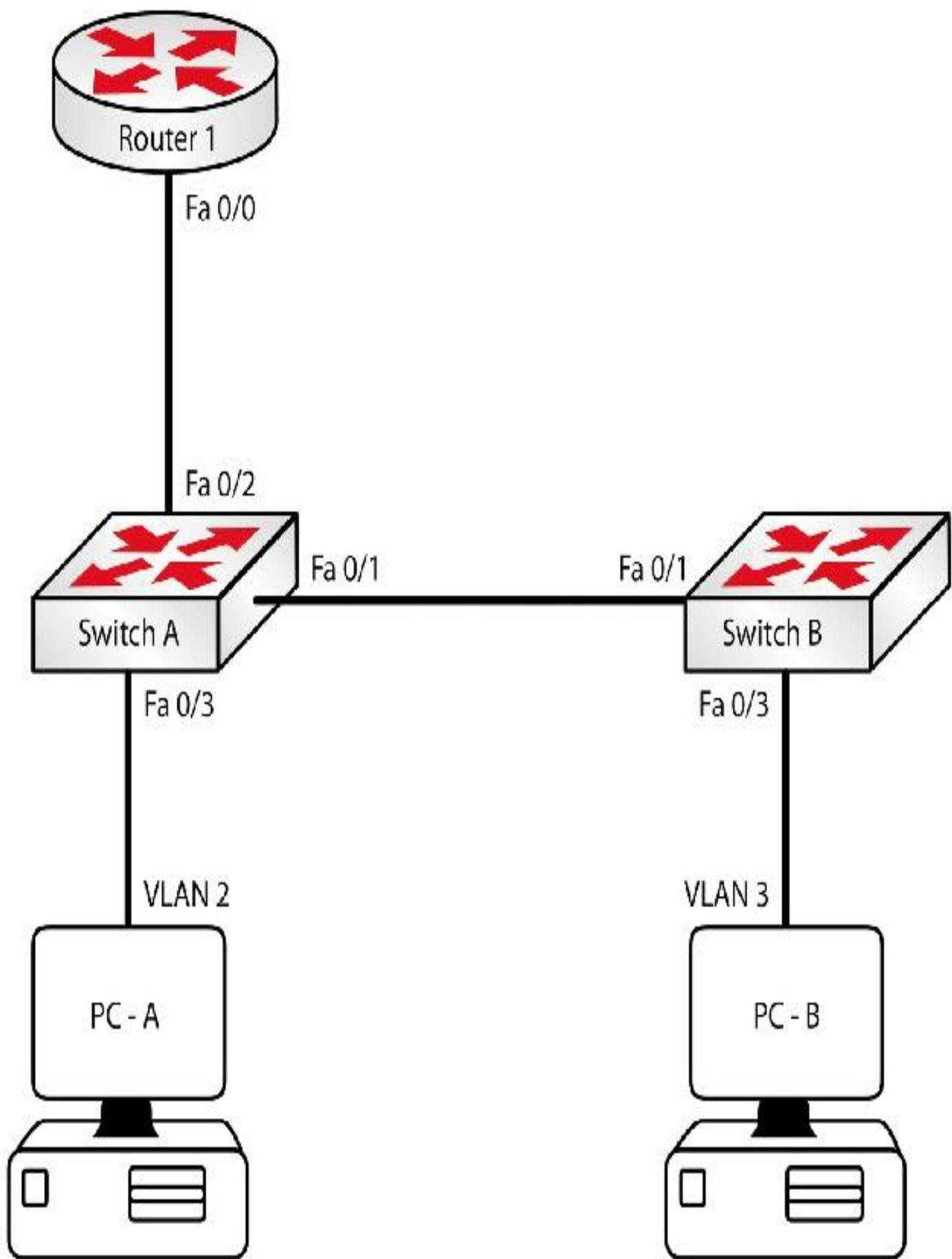


FIG 2.26 – InterVLAN routing

VLAN 2 – 192.168.2.0/24, PC-A 192.168.2.2/24

VLAN 3 – 192.168.3.0/24, PC-B 192.168.3.2/24

Because you may have only one available Fast Ethernet interface on your router, you will need to divide the physical interface into logical subinterfaces. For example, if your router interface is Fast Ethernet 0/0, then the subinterface would be 0/0.1. It is common practice to name your subinterface to match your VLANs, so you would name the subinterface for VLAN 2 Fast Ethernet 0/0.2.

In order to use the router Ethernet interface as a trunk connection to the switches, the encapsulation type must be set to dot1q. Depending on your switch model, this may be your only encapsulation choice so the encapsulation command may not work (this applies to 2960 Switches).

One last note: in this lab you may swap PCs for Fast Ethernet interfaces on routers. You will need to add static routes on the routers though (see below).

Lab Objectives

1. Configure Switch A to trunk with Switch B and Router 1 using 802.1Q.
2. Configure Switch B to trunk to Switch A using 802.1Q.
3. Configure port fa0/3 to be in the correct VLANs on both switches.
4. Configure Router 1's fa0/0 interface with two subinterfaces in the correct VLANs and with correct IP addresses (.1 in the respective subnets).
5. Configure the PCs with respective gateway addresses (address of the subinterface on Router 1).
6. Ping from PC-A to PC-B.

Lab Walk-through

1. Configure Switch A for trunking on relevant ports. You need trunking ports because they can carry multiple VLAN information. Do the following:

```
Switch#configure terminal
```

```
Switch#(config)#hostname SwitchA
```

```
SwitchA(config)#interface range fa0/1 - 2
```

```
SwitchA(config-if-range)#switchport mode trunk
```

Please note: you need a space between the interface numbers if you are using the range command, and the command may not work if your IOS version does not support it. If you are using different switch models, your interfaces may show as

1/1, etc.



This is platform/IOS version-dependent. Some platforms/versions require a space before and after the hyphen, some require it only before the hyphen, and some don't require a space at all.

You can check encapsulation type with the show interface trunk command:

```
SwitchA#show interface trunk
Port    Mode     Encapsulation  Status      Native vlan
Fa0/1   on      802.1q        trunking    1
Fa0/2   on      802.1q        trunking    1
```

2. To configure Switch B for trunking, do the following:

```
Switch#config t
Switch#(config)#hostname SwitchB
SwitchB(config)#int fa0/1
SwitchB(config-if)#switchport mode trunk
```

3. Access ports are used to connect hosts to the switch. They should actually be access ports by default, but it is useful to know the command. To configure the access ports, do the following:

```
SwitchA(config)#interface fa0/3
SwitchA(config-if)#switchport mode access
SwitchA(config-if)#switchport access vlan 2
SwitchB(config)#interface fa0/3
SwitchB(config-if)#switchport mode access
SwitchB(config-if)#switchport access vlan 3
```

Please note: you have put the interfaces into the respective VLANs 2 and 3 with the commands above. Also, if VLAN 2 or 3 is not already created on the switch, you may see:

```
SwitchB(config-if)#switchport access vlan 2
% Access VLAN does not exist. Creating vlan 2
```

You can see from earlier labs that VLANs are created with the Switch(config)#vlan 2 command.

4. To configure the router port, you need to add subinterfaces and configure dot1q encapsulation so the interface can trunk the VLANs. Do the following:

```
Router#config t
Router(config)#hostname Router1
Router1(config)#interface fa0/0
Router1(config-if)#no shut
Router1(config-if)#interface fa0/0.2
Router1(config-subif)#encapsulation dot1q 2
Router1(config-subif)#ip address 192.168.2.1 255.255.255.0
Router1(config-subif)#interface fa0/0.3
Router1(config-subif)#encapsulation dot1q 3
Router1(config-subif)#ip address 192.168.3.1 255.255.255.0
```

(Please note: interface fa0/0.2 is used for VLAN 2. The correct VLAN number after the dot1q command was also added. You can see the output the router expects below.)

```
Router1(config-subif)#encap dot1q ?
[1-4095] IEEE 802.1Q VLAN ID required, range 1 - 0xFFFF.
```

5. (Optional) Depending on the operating system of the PCs, configure their gateway to be 192.168.2.1 and 192.168.3.1, respectively.

In this example, I used the switching rack at <https://www.howtonetwork.com> and used two routers (connected to Switches A and B) with Fast Ethernet interfaces instead of PCs. They are only required to prove that the VLANs can communicate. Fast Ethernet 0/0 on Router A is given an IP address in VLAN 2 and Router B in VLAN 3. If you do use routers, please remember to add a static route (e.g., on the router with the IP address 192.168.3.2, add static route 0.0.0.0 0.0.0.0 192.168.3.1.).

```
Router(config)#hostname RouterA
RouterA(config)#int fast0/0
RouterA(config-if)#ip add 192.168.2.2 255.255.255.0
RouterA(config-if)#no shut
RouterA(config-if)#^Z
RouterA#
Router(config)#hostname RouterB
```

```
RouterB(config)#int fast0/0
RouterB(config-if)#ip add 192.168.3.2 255.255.255.0
RouterB(config-if)#no shut
RouterB(config-if)#^Z
RouterB#
```

6. Now ping 192.168.3.2 from PC-A (or Router A if you are using routers):

```
RouterA#ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
!!!!
```

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Show Runs

```
Running Configuration
Router1#sh run
Building configuration...
[output truncated]
hostname Router1
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
!
interface FastEthernet0/0.2
encapsulation dot1Q 2
ip address 192.168.2.1 255.255.255.0
!
interface FastEthernet0/0.3
encapsulation dot1Q 3
ip address 192.168.3.1 255.255.255.0
!
SwitchA#sh run
Building configuration...
[output truncated]
hostname SwitchA
!
```

```
interface FastEthernet1/0
!
interface FastEthernet1/1
switchport mode trunk
!
interface FastEthernet1/2
switchport mode trunk
!
interface FastEthernet1/3
switchport mode access
switchport access vlan 2
!
SwitchB#sh run
Building configuration...
[output truncated]
hostname SwitchB
!
interface FastEthernet1/0
!
interface FastEthernet1/1
switchport mode trunk
!
interface FastEthernet1/2
!
interface FastEthernet1/3
switchport mode access
switchport access vlan 3
!
```

Chapter 3 — IP Addressing

What You Will Learn in This Chapter

- How Binary Works
 - How Hexadecimal Works
 - IPv4 Addressing
 - Subnetting
 - VLSM and Route Summarization
-

Syllabus Topics Covered

3.0 IP Addressing (IPv4/IPv6)

- 3.1 Describe the operation and necessity of using private and public IP addresses for IPv4 addressing
- 3.3 Identify the appropriate IPv4 addressing scheme using VLSM and summarization to satisfy addressing requirements in a LAN/WAN environment

7.0 Troubleshooting

- 7.1 Troubleshoot and correct common problems associated with IP addressing and host configurations

We will now look at the core subject of CCNA certification and, in fact, pretty much any networking certification—IP addressing and learning to subnet. You will need to have this down cold no matter your chosen destination, including security engineer, network designer, server engineer, network architect, cloud engineer, or voice/video networking.

Every device in a network must have a unique IP address, but many of the addresses can't be used because they are in an excluded range, already belong to a network, or are a broadcast in a network. I can't tell you how many times I've had to clean up problems caused by other IT engineers who didn't understand IP addressing or subnetting and allocated the wrong IP address to the wrong subnet, causing untold issues to network equipment and servers.

What you will learn in this chapter will serve you well for years to come and will feature heavily in the exam, so take your time and drill it over and over until it becomes second nature.

How Binary Works

In order to understand how IP addressing works, you need to understand binary mathematics. Electronic equipment does not communicate using decimal values.

Humans use decimal because it's a numbering system that uses ten digits and we have ten fingers on our hands.

Digital electronic equipment works by interpreting the presence or absence of an electrical (or optical) signal. As such they can only be in two states—On or Off—which are represented by the numbers 1 and 0, respectively. Computers work using a numbering system with two digits, the binary system. A 0 indicates the absence of a signal while a 1 indicates its presence.

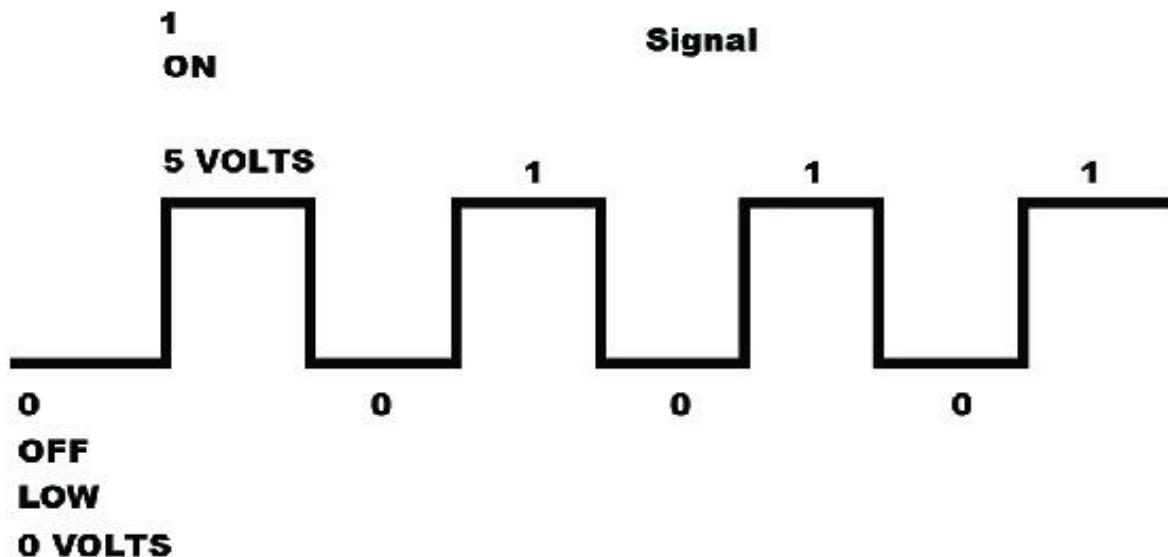


FIG 3.1 – On/Off signal status

Any decimal number can be converted from binary values. The larger the number, the more binary bits required to express it. A bit is short for BInary digiT. For every binary value added, it doubles the next number (i.e., 1 to 2 to 4 to 8 to 16 and so on into infinity). With two binary digits you can count up to 3; just place a 0 or a 1 in the column to decide whether you want to use that value or not.

We will start with only two binary values, in columns 2 and 1:

2	1
0	0

$$0 + 0 = 0$$

2	1
0	1

$$0 + 1 = 1$$

2	1
1	0

$$2 + 0 = 2$$

2	1
1	1

$$2 + 1 = 3$$

With eight bit places (an octet), any number from 0 up to 255 can be expressed.

128	64	32	16	8	4	2	1	

Adding a 0 to each of these columns will give you a value of 0 in decimal. If you add a 1 in any of the respective columns, you will add that value to the final total, while a 1 in each column will add up to 255 as shown below:

128	64	32	16	8	4	2	1	Total
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	255
0	0	1	0	1	1	0	0	44

Pay special attention to the binary table below because the values can be used for any subnet mask (more on that later). If you start adding 1 to each column from left to right, you will get the following values:

Table 3-1: Binary values

Binary	Decimal
10000000	128
11000000	192
11100000	224

11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

Make up some of your own binary numbers to ensure that you understand this concept fully.

How Hexadecimal Works

In previous versions of the CCNA exam, having a cursory understanding of hexadecimal, or hex, was a must. Because IPv6 is now a major part of the syllabus, you need to have an even deeper understanding of this numbering system. We will cover IPv6 later in Chapter 4.

In 1859, Johan Vilhelm Nyström, a Swedish-American civil engineer, inventor, and author, proposed a hexadecimal (base 16) system of notation, arithmetic, and metrology called the Tonal System. Rather than counting in twos or tens, 16 characters are used. Hex numbering starts at 0 and goes up to F:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Each hexadecimal digit actually represents four binary digits, as shown below:

Table 3-2: Decimal to hex to binary conversion

Decimal	0	1	2	3	4	5	6	7
Hex	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Decimal	8	9	10	11	12	13	14	15
Hex	8	9	A	B	C	D	E	F
Binary	1000	1001	1010	1011	1100	1101	1110	1111

Converting from binary to hex is fairly simple, as illustrated in the example below:

Decimal	13	6	2	12
Hex	D	6	2	C
Binary	1101	0110	0010	1100

Hex is a more manageable counting system for humans compared with binary but is close enough to binary to be used by computers and networking equipment. As shown below, one hex character is equal to four binary digits. Any number can be expressed using hex, as it can use binary or decimal; it's just counting in multiples of 16, as in 1 then 16 then 16 multiplied by 16 (256), then 256 multiplied by 16 (4096) and so on.

Hex	4096	256	16	1
			1	A

Hexadecimal numbering therefore is represented as 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 and so on. 1A (above), for example, is a 1 in the 16 column and an A in the 1 column: $1A = 10 + 16 = 26$.

When converting binary to hex, you can make the task easier by breaking down the number into groups of four bits, starting from the right. So, 11110011 becomes 1111 0011. 1111 is $8 + 4 + 2 + 1 = 15$ and 0011 is $2 + 1 = 3$. 15 is F in hex, and 3 is 3, giving us the answer F3.

Hex to binary is the same process. Each hex digit represents four bits. 7C can be split into 7, which is 0111 in binary, and C, which is 1100 in binary (or 12 in decimal). The answer is 01111100.

Have a Try

Here are some examples for you to try. A very useful thing to do would be to write out the charts for working out hex and binary (i.e., for hex a 1 column, then a 16 column, then a 256 column, and so on). The answers are at the end of this chapter (no peeking).

- Convert 1001 (binary) to hex and decimal
- Convert 11011 (binary) to hex and decimal
- Convert 10001 (binary) to hex and decimal
- Convert 29 (decimal) to binary and hex
- Convert 33 (decimal) to binary and hex
- Convert 102 (decimal) to binary and hex
- Convert C7 (hex) to binary and decimal
- Convert FE (hex) to binary and decimal
- Convert B5 (hex) to binary and decimal

In the exam you are allowed a small whiteboard and marker pen, which will help you work out any binary to hex conversions.

IP Version 4

The current version of Internet Protocol (IP) in wide deployment is version 4, and it is explained in RFC 791. IP version 6 (IPv6) is currently in use on a number of networks, either solely or alongside IPv4, and will ultimately replace IPv4 in a gradual phasing-out process. You can still expect to be tested on IPv4 and, in particular, subnetting using IPv4 in the exam so ensure that you know it well.

IPv4 uses four groups of octets to make an IP address, and each octet is made up of eight bits (also known as 1 byte). Therefore, every IP address is 32 bits ($4 \times 8 = 32$), or 4 bytes. An example of how an IPv4 address appears in binary would be:

11000011.11110000.11001011.11111100

1st octet 2nd octet 3rd octet 4th octet

It is worth noting that routers do not see an IPv4 address as four octets; they just see 32 bits. Octets just make things easier for us to see.

When IPv4 addressing was designed in 1981, the thinking was that it would provide enough IP addresses for the foreseeable future. No one predicted the huge growth in home computing (or the invention of mobile devices) that was to come, so this scheme in its initial incarnation had to be amended to cater to the demand. We will cover some of the techniques used to extend the life of IPv4 addresses, such as NAT and VLSM, later on. VLSM stands for Variable Length Subnet Masking and it allows you to change the default subnet mask value, for example, for Class A using /8 to /12. As stated, we will cover this later in the manual.

Powers of Two

In order to really understand IP addressing, you should understand the powers of two. While it may appear confusing initially, you simply start with the number 2 and keep doubling the previous number. Understanding the powers of two will really help you master subnetting and network design using VLSM, and you will surely be tested on this in the exam. We will apply the powers of two in an easy subnetting system so you can answer network addressing and VLSM questions in a matter of seconds.

The important thing is not the multiplication, it is what is happening to the answers: each time, the number doubles. If you want to work out the powers of two, you would write it like this:

$$2^1 = 2 \times 1 = 2$$

$$2^2 = 2 \times 2 = 4$$

$$2^3 = 2 \times 2 \times 2 = 8$$

$$2^4 = 2 \times 2 \times 2 \times 2 = 16$$

$$2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$$

and so on...

We will come back to the powers of two when we look at easy subnetting and designing subnets, both of which are heavily tested in the CCNA exam and beyond.

IP Addressing

As mentioned, IP addresses consist of binary numbers that are grouped into octets. IPv4 addresses are further divided into classes, from Class A to Class E. IP addresses are assigned by a group called the IANA (Internet Assigned Number Authority). You can also buy one from an ISP who has in turn bought a block from the IANA. The size of your business organization determines which class of IP address you are given.

Class A Addresses

Historically, these were given to the very largest organizations, as they needed a tremendous number of IP addresses because they owned more computers than everyone else. Class A addresses use only the first octet to identify the network number, while the remaining three octets identify the hosts in the network, as shown below:

Network.Host.Host.Host

10 .2 .5 .4

In the example above, the network is 10, and 2.5.4 is a host in that network. You would pronounce this as ten dot two dot five dot four if reading it out to another engineer.

In binary the address would be:

00001010.00000010.00000101.00000100

nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

Class A addresses are numbered from 1 to 126 in the first octet. Network equipment identifies a Class A address because the very first bit on the first octet has to be a 0. A Class A address cannot have a 1 in this bit position. So, the first network number is 1 because you can't have a network with the number 0.

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	1

The last possible network number is 127. (Check by adding all the values together.) Network number 127 cannot actually be used because the value 127.0.0.1 is reserved for troubleshooting (normally found on a logical interface called Loopback or Localhost on devices). You can ping the Loopback address to check whether TCP/IP is working on the host.

128	64	32	16	8	4	2	1
0	1	1	1	1	1	1	1

Because you are not permitted to use 0 or 127 as a network number, this leaves you with 126 available networks for Class A addresses.

A 0 in each of the host address spaces indicates the network number. You need to have at least one digit in there to designate a host address. Hosts can start at number 1 until (almost) every single possible value is used up. To illustrate this concept, we'll use a network address starting with 10:

10.0.0.1 is the first host. In binary:

00001010.00000000.00000000.00000001

10 . 0 . 0 . 1

(as a decimal)

10.0.0.2 is the second host. In binary:

00001010.00000000.00000000.00000010

10 . 0 . 0 . 2

If you were allocating the addresses to hosts in the network sequentially, you would eventually use up all the available bits in the three octets, but you can't use every single host bit (see below).

10.255.255.254 is the last host. In binary:

00001010.11111111.11111111.11111110

Now you can see why we use decimal. It would take a long time to write out addresses in binary and it would be almost impossible to remember them. You can change the router to display all addresses in binary, but there is no reason to do this.

Why can't 10.255.255.255 be a host? Because when all the binary values have a 1 turned on, on the host part of the address, this tells the network that it is a broadcast

packet. As you learned earlier, a broadcast packet is addressed to every host in the network or subnet so it can't be allocated to an individual host.

If you think further about this, you now know that for the host part of any network address, two addresses can never be used—the network address and the broadcast address. This will become important later in this chapter when you will need to work out the number of hosts per network.

Class B Addresses

Class B addresses were reserved for large organizations that needed a lot of host numbers, but not as many as the largest ones. Unfortunately, when a Class B address was assigned to an organization, it resulted in thousands of wasted host numbers.

Class B addresses must have the first two binary values on the first octet reserved with a 1 and a 0 next to them. So, the first network number is 128 and you can see from the row of zeros that all the available network bits on the first octet are turned off:

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0

The last available Class B network number is 191 (add the values together) and, as shown below, the available network bits have been turned on (on the first octet):

128	64	32	16	8	4	2	1
1	0	1	1	1	1	1	1



If this is all a bit confusing just stick with it and go through a few rereads, because it takes a while to sink in. It took six weeks for me when I started out!

For Class B addresses, the first two octets for the network address are used, so for the address 130.24.5.2. 130.24 is the network number and 5.2 is a host in that network. The rule is still the first number you see in the first octet will always be between 128 and 191, inclusive.

Using the powers of two rule, the first two octets you will see can have a possible $65,536$ ($2^{16} = 65,536$) networks. However, you are not allowed to use the first two bits of the first octet because they are reserved for showing the 10 value, remember? So this leaves you with $6 + 8$ digits: 2^{14} gives you 16,384 networks.

There are two full octets to use for hosts, so $8 + 8$ bits gives you $2^{16} = 65,536$ hosts per Class B network, but you have to take two away from this value for the broadcast and subnet (more on this later), so technically there are 65,534 host addresses.

Class C Addresses

These were originally reserved for any other organization that was not large enough to warrant having a Class A or Class B address. A Class C address has the first three bits reserved so the network device can recognize it as such. The first three bits must show as 110.

The first network number is 192 (128+64) and all the other network bits are off (0):

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

The last network number is 223 and this time all the network bits are on (on the first octet):

128	64	32	16	8	4	2	1
1	1	0	1	1	1	1	1

An example of a Class C address is 200.2.1.4, where 200.2.1 is the network address and .4 is a host in that network. While there are lots of available network numbers to assign to companies, there are a limited amount of numbers free to use for the hosts in the networks.

For networks, the first three bits (110) from the first octet are unavailable, giving you $5 + 8 + 8 = 21$ network bits: $2^{21} = 2,097,152$.

For the hosts you have 2^8 (the last full octet), giving you 256 (only 254 are usable, though, because of the two taken away for the broadcast and subnet).

Class D and Class E Addresses

Class D addresses are reserved for multicast traffic and cannot be used for hosts on

your network. Multicast traffic is sent to multiple hosts using one IP address. Some routing protocols use multicast addressing for updates as you will see later in this guide.

Class D addresses range from 224.0.0.0 to 239.255.255.255

Class E addresses are reserved for experimental research only. They are numbered from 240.0.0.0 to 255.255.255.255.

Summary

To summarize class addresses:

Class A – first bit set to 0

Address range, from 1 to 126 (127 is reserved for testing)

Network.Host.Host.Host

Class B – first bits set to 10

Address range, from 128 to 191

Network.Network.Host.Host

Class C – first bits set to 110

Address range, from 192 to 223

Network.Network.Network.Host

Class D – first bits set to 1110

Address range, from 224 to 239.

Class E – first bits set to 11110

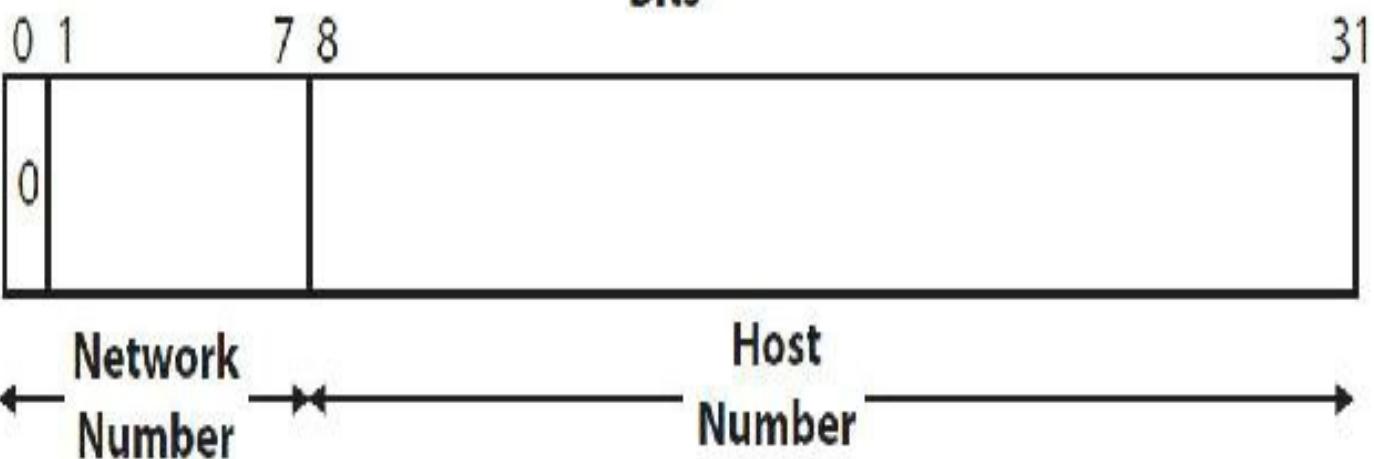
Address range, from 240 to 255.

Looking at the number in the first octet will tell you which class of address you are dealing with:

- 10.1.2.1 = Class A
- 190.2.3.4 = Class B
- 220.3.4.2 = Class C

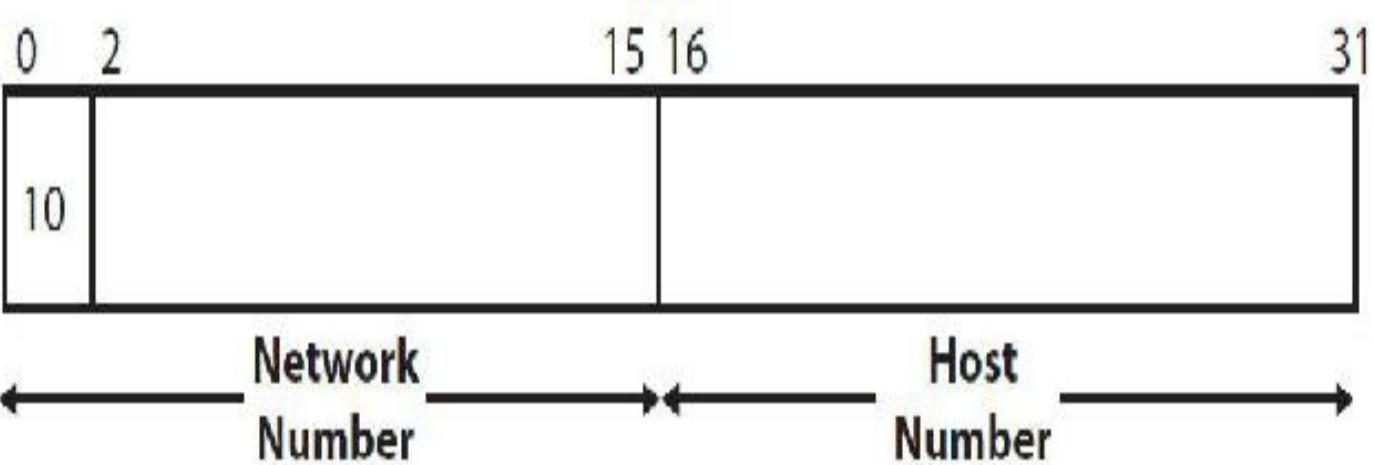
Class A

Bits



Class B

Bits



Class C

Bits

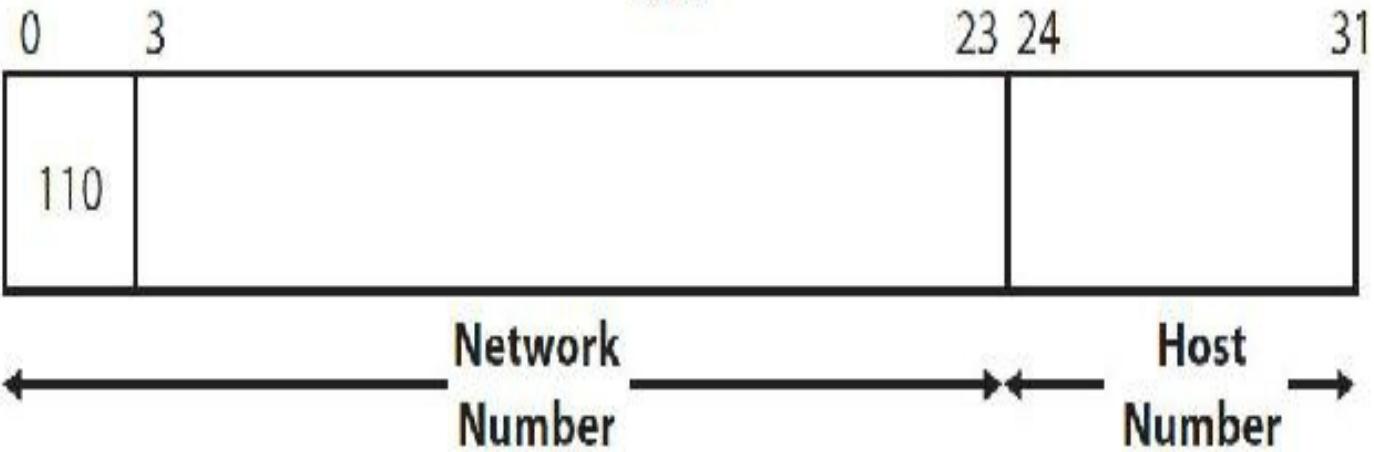


FIG 3.2 – IP addressing classes summarized

Although you need to understand address classes, it is really only for historical purposes because due to enhancements in IPv4 addressing using variable length subnet masking (VLSM), any class of address can now be allocated to any organization, large or small. We will cover VLSM shortly.

Private IPv4 Addresses

To help prevent wastage of IP addresses, certain addresses are reserved for use on private networks. Any individual or company can use these addresses in their network provided they do not try to get out to the Internet to use them. The address allocation scheme was suggested in RFC 1918: Address Allocation for Private Internets.

The reserved addresses are:

- 10.x.x.x – any IP address beginning with 10
- 172.16.x.x to 172.31.x.x – any IP address starting with 172.16 to 172.31, inclusive
- 192.168.x.x – any IP address starting with 192.168

This idea proved to be an ideal way to avoid wasting thousands of IP addresses. It's important for you to be able to recognize private addresses, not only for the exam but also in the real world, because you may find customers trying to get out to the Internet to use them and wondering why their ISP is blocking their traffic. If you want hosts using private IP addressing to go out to the Internet, then you need to use Network Address Translation (NAT) to swap it to a routable address. We will cover NAT later in this guide.

Classless Inter-Domain Routing

RFC 1517 through 1520 specifies Classless Inter-Domain Routing (CIDR) features. CIDR removes the need for classes of IP addresses (classless routing), which is yet another solution to the problem of the depletion of IP addresses. CIDR allows for route aggregation whereby a single route in a routing table can represent several network addresses, saving space and routing table size as well as introducing VLSM. All of these are covered in this guide.

Using CIDR, you no longer need to worry about the class system for addressing networks. Network administrators can now allocate address spaces on an as-needed basis, rather than having to use a Class B address and wasting thousands of spare addresses.

CIDR allows the use of the slash system to represent subnet masks (e.g., /26 instead of 255.255.255.192). Table 3-3 below shows a Class C subnetting chart with the CIDR representation:

Table 3-3: CIDR

Masked Bits	Subnet Mask	CIDR	Subnets	Hosts
0	255.255.255.0	/24	1 network	254
1	255.255.255.128	/25	2	126
2	255.255.255.192	/26	4	62
3	255.255.255.224	/27	8	30
4	255.255.255.240	/28	16	14
5	255.255.255.248	/29	32	6
6	255.255.255.252	/30	64	2

Subnetting

Subnetting can be one of the most difficult subjects to master for a CCNA candidate. There is a long way to subnet and a very short and easy way. It is vital for you to understand how the long way works first, but then in real life and to save time in the exam you will use the easy way. Please also refer to the free subnetting lessons I provide on www.howtonetwork.com/ccnasimplified, which supplement the lessons in this book.

When working on live networks over the years, I've encountered a large number of network engineers who can't solve simple subnetting issues. They often try to allocate a network address to a host or use a host number in the wrong subnet and then wonder why the host or server can't reach the default gateway. Make it a priority to master subnetting and then refresh your knowledge every few weeks to stay sharp.

Address Depletion

Shortly after the IPv4 addressing scheme was implemented, it became apparent that there were not enough addresses to meet demand. More and more organizations were using computers and networking equipment and the then-current scheme was wasting thousands of addresses.

Say, for example, that a company was given a Class A address. Remember that Class A addresses could only be allocated to 126 companies, and the first octet was used for the

network while the other three octets were free for use as hosts in the network.

If every combination of numbers on the remaining 24 digits (3×8) was used, that would equal over 16 million hosts (16,777,216, in fact) per network. You can actually work out how many networks or hosts using the powers of two. For example, three octets are free for hosts in the network, giving you 24 bits ($3 \times 8 = 24$):

2^{24} , or $2 \times 2 \times 2$.

$$2^{24} = 16,777,216$$

Historically, what happened was that a very large organization would be allocated a Class A address. They would need around 10,000 host addresses and waste the other 16 million plus. Because they owned the network number it could not be shared with other organizations, so the other addresses were wasted.

Another issue was network broadcasts. You couldn't split the network into smaller portions (at the time), so the entire network would have to be attached to one router interface and all the hosts would be in the same broadcast domain.

For a Class B address, the story was very similar:

2^{16} is $2 \times 2 = 65,536$ hosts

Class C addresses were more reasonable with 2^8 host per network address, which equals 256 hosts. I have not started taking two of the host addresses away for the broadcast and subnet yet because we will cover this shortly.

How to Subnet

As you have learned in the previous sections, the initial way of using IP addresses was very rigid: Class A addresses were fixed with 8 bits for the network and 24 for the hosts; Class B addresses were fixed with 16 bits for the network and 16 for the hosts; and Class C addresses were fixed with 24 bits for the network and 8 for the hosts. There had to be some way to avoid wasting host addresses. The answer came with the introduction of subnetting.

Subnetting involves using bits that are normally used for the host part to be used for the subnet part of the address. In order for routers or PCs to know that subnetting is being used, another number has to be applied to the IP address. This number is known as the subnet mask and it is also in binary.

Think of subnetting as taking a cake and cutting it into slices. You can cut any of those slices into smaller slices (using VLSM). Before subnetting you were not permitted to do

this. Now if you have a portion of the network that needs more host addresses, you can give it a bigger slice (subnet). Other parts with a requirement for less hosts can have a smaller slice.

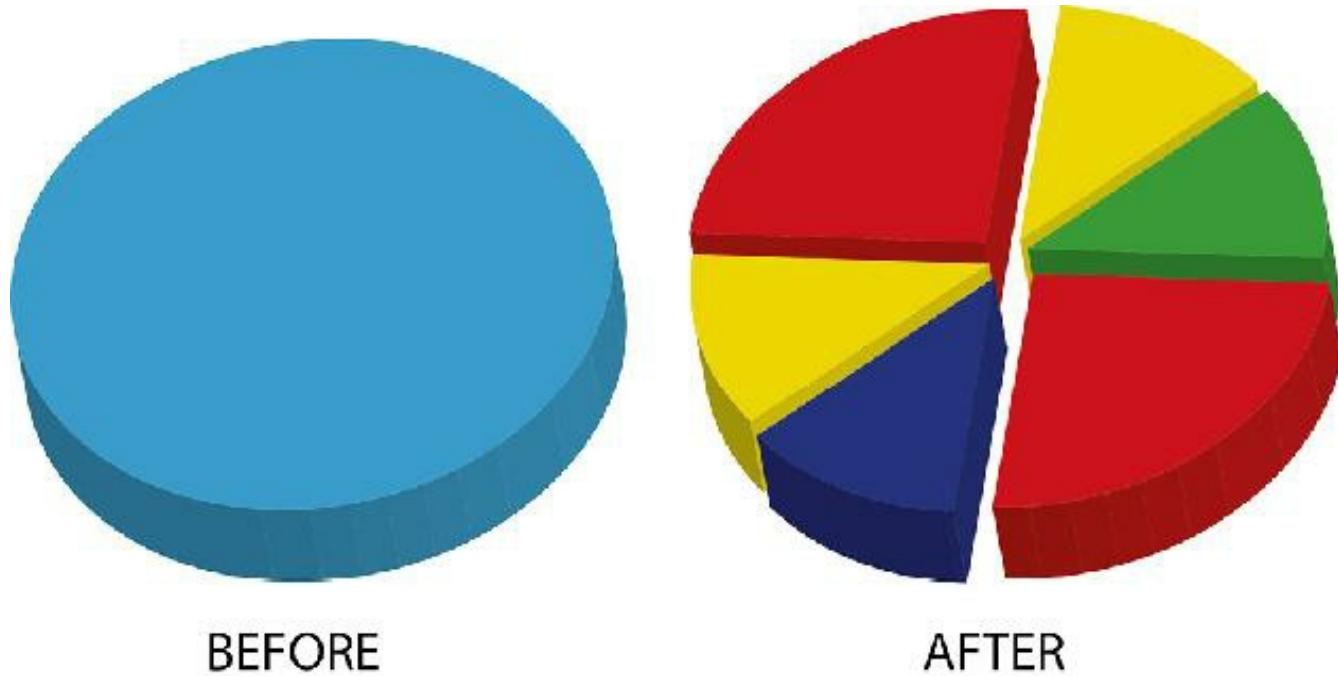


FIG 3.3 – Subnetting: Before and after

Today, network devices check each bit on the subnet mask, compared with the bits on the IP address, to determine which parts belong to the network and which belong to the host. A default subnet mask is allocated to each class of address. It is not possible to enter an IP address onto a PC or router without also entering the subnet mask.

Default subnet masks are as follows:

- Class A – 255.0.0.0, or in binary: 11111111.00000000.00000000.00000000
- Class B – 255.255.0.0, or in binary: 11111111.11111111.00000000.00000000
- Class C – 255.255.255.0, or in binary: 11111111.11111111.11111111.00000000

In the default subnet masks above, the first octet for Class A addresses is reserved for the network number, as are the first two octets for Class B and the first three for Class C.

One rule for subnet masks is that the 1 and 0 network and host bits must be contiguous (i.e., connected), without a break from left to right. So you can have

11111111.11111111.00000000.00000000

but you cannot have

1111111.0001111.0000000.0000000

When you start to add host addresses from 1,2,3, and upward, you will start from the far right:

1111111.1111111.0000000.0000001

Each part of the IP address is matched with the subnet mask to determine which bits are part of the network identification and which bits are part of the host identification.

Example

10001100.10110011.11110000.11001000 – 140.179.240.200 (Class B)

11111111.11111111.00000000.00000000 – 255.255.0.0 (subnet mask)

10001100.10110011.00000000.00000000 – 140.179.0.0 (network address)

How did we get this number? The router performs something called logical ANDing. It compares the 1s and 0s to establish which numbers belong to the network and which belong to the host, as shown below:

AND	0	1
0	0	0
1	0	1

All the values are compared and, as you can see, anything apart from a 1 and 1 equals 0. Check the example above again to make sure that you understand how it works.

Because you now know which are the network bits and which are the hosts, you can start assigning IP addresses to hosts on your network. If all the host bits are 0 then you cannot use this for a network host. The all 0s represent the subnet; you will see why and how shortly.

140.179.0.0 is your network address (in binary) and it has all the host bits turned off:

10001100.10110011.00000000.00000000

Network. Network. Host. Host

140.179.0.1 can be used for your first host:

10001100.10110011.00000000.00000001

140.179.0.2 can be used for your second host:

10001100.10110011.00000000.00000010

You can see from the bold font in the example above that you are adding host addresses from the far right.

You can keep adding hosts until both the second and third octet are (almost) full.

- 140.179.0.255 is still a valid host number
- 140.179.1.255 is still okay
- 140.179.255.254 is the last host number you can use.

Here is the last host number in binary:

10001100.10110011.**11111111.11111110** – not every host bit is turned on

Why can't you use the last bit portion above for a host? As you already know, an IP address with all 1s in the host portion is reserved to tell the network that the packet is a broadcast packet. A broadcast packet is a packet that must be examined by all hosts in the network (or, more specifically, all the hosts in this portion of the network, i.e., the subnet). The number below is a broadcast packet to every host in the 140.179 network:

10001100.10110011.**11111111.11111111** – 140.179.255.255 in binary has all of the host bits turned on.

Now you can see that you are not permitted to use all 0s for the hosts since this is the network address, and you cannot use all 1s because this is reserved for a broadcast. With this information you will be able to decide how many available hosts you have per network or subnet.

You can use the power-of-two formula to work out how many hosts you will get on the subnet. Simply multiply two to the power of how many host bits you have and take away two, one for the network/subnet of all 0s and one for the broadcast address of all 1s.

In the example 140.179.0.0 255.255.0.0, you can see that you have the last two octets free (the 0.0) to allocate to hosts in the network. That is, two octets of eight binary bits each, giving you 16 bits.

The formula for the maximum number of hosts (per subnet) is: 2^{n-2} . In the example above, $n = 16$, so $2^{16} - 2 = 65,534$.

You could also just take the number 2 and double it 16 times, so 2, 4, 8, 16, 32, and so on up to 65,536, and then subtract 2. We will actually do this to around the power of 10 when we look at the Subnetting Cheat Chart, which you will be allowed to write out on

your whiteboard in the CCNA exam. Doing this will save you a ton of time and will literally make the answer pop out before your eyes.

Do you think it would be practical to have a network with over 65,000 hosts on it? If you had a broadcast in the network, each and every single host in the network would have to stop what it was doing to listen to the broadcast packet to see whether it was the intended recipient.

Let's steal some bits from the host part of the address and make a subnetwork (or subnet) from those bits. I will write out the network address in longhand to make it easier to understand:

140.179. 00000 000.00000000
[16 bits] [5 bits] [11 bits]
[network][subnet][host bits]

I have stolen five of the host bits to make the subnet. The advantage of this is having more than one subnet to use and fewer hosts per subnet. Now you can use the powers-of-two formula to work out how many subnets you have and how many hosts per subnet. You do not have to take two away for the subnets because it is a network number (you only take two away for host numbers).

2^5 (or $2 \times 2 \times 2 \times 2 \times 2$) = 32 subnets, each with

2^{11} (or $2 \times 2 - 2$) = 2,046 hosts per subnet

Why would you want to do this? Because you will have fewer hosts using the bandwidth on the network segment. It is far easier to administer smaller subnets than one huge network. Additionally, it is desirable to limit the number of broadcasts on a given subnet because each and every host on a subnet must examine the contents of a broadcast packet, whether it is the intended recipient or not.

In an environment with an excessive number of hosts, the number of broadcasts can grow significantly, and while not immediately measurable, this broadcast traffic will lower the overall performance of all the networked systems.

The more host bits you steal, the more subnets you get, but each of those subnets is capable of supporting a lesser number of hosts because there are fewer host bits available. Deciding how many hosts you need and how many hosts per subnet is part of the network design phase. The more host bits you steal means you have more and more subnets and less host bits available; this is the tradeoff.

Table 3-4 below illustrates a Class B network. Remember that for Class B addresses, you are looking at the third and fourth octets for the bit pattern, as the first two octets are used for the network address and cannot be stolen.

Table 3-4: Subnetting in a Class B network

Bit Pattern (3rd/4th Octet)	CIDR	Masked Bits	Subnets	Hosts per Subnet (2 to the Power of n - 2)
00000000.00000000	/16 255.255.0.0	0	1 (network)	65,534
10000000.00000000	/17 255.255.128.0	1	2	32,766
11000000.00000000	/18 255.255.192.0	2	4	16,382
11100000.00000000	/19 255.255.224.0	3	8	8,190
11110000.00000000	/20 255.255.240.0	4	16	4,094
11111000.00000000	/21 255.255.248.0	5	32	2,046
11111100.00000000	/22 255.255.252.0	6	64	1,022
11111110.00000000	/23 255.255.254.0	7	128	510
11111111.00000000	/24 255.255.255.0	8	256	254
11111111.10000000	/25 255.255.255.128	9	512	126
11111111.11000000	/26 255.255.255.192	10	1,024	62
11111111.11100000	/27 255.255.255.224	11	2,048	30
11111111.11110000	/28 255.255.255.240	12	4,096	14
11111111.11111000	/29 255.255.255.248	13	8,192	6
11111111.11111100	/30 255.255.255.252	14	16,384	2
11111111.11111110	/31 255.255.255.254	15	32,768	0*

*In modern networks, a /31 subnet mask can be used on point-to-point links between two routers since a broadcast and a network address is not needed. Try it for yourself between two routers. You will require IOS 12.2(2)T or later.

R1(config)#int s0/0

R1(config-if)#ip add 192.168.1.1 255.255.255.254

R1(config-if)#end

R1# show int s0/0

Serial0/0 is up, line protocol is up

Hardware is GT96K Serial

Internet address is 192.168.1.1/31

How to Write Subnet Masks

As with IP addresses, subnet masks are expressed in dotted decimal notation (each of the octets in a subnet mask is converted to decimal and separated with a dot).

If you steal five host bits from the third octet, you have to add the binary values together:

128	64	32	16	8	4	2	1
1	1	1	1	1	0	0	0

So, you have $128 + 64 + 32 + 16 + 8 = 248$.

Remember that you are using a Class B example here and are working with the third octet, and you are not allowed to alter the first two octets because they are fixed. This results in the following:

255.255.248.0

This tells the router that you are subnetting and that you are using the first five hosts bits to carve out the subnets.

Things can get a little bit (more) complicated, and you can no longer rely on what your eyes are telling you because the router is looking at a binary value and you are looking at a decimal value. Do not worry too much though, as we will look at easy subnetting later.

In order for the router to know whether a host is on a certain subnet, it looks at the masked bits. If all of the masked bits match, then it follows that the host must be in the same subnet. If the subnet bits do not match, then the hosts are in different subnets. This all happens in a matter of milliseconds on the router.

Let's examine the following IP address: 129.10.147.0 255.255.248.0

Again, this is a Class B address and you are stealing five bits for subnetting. You know you have stolen five bits because 248 in binary is 11111000, which is five masked bits:

10000001.00001010.10010011.00010000 129.10.147.32

10000001.00001010.10010100.01010101 129.10.148.85

You can see that the subnet bits in the example above both match. This means that both

host addresses are in the same subnet. Here is another example:

```
10000001.00001010.10011010.00000010 129.10.154.2
```

This time one of the subnet bits has changed, so the router or PC can see it is a different subnet. Unfortunately, when you write it out in decimal, it is not very easy to see that this third IP address is in a different subnet.

Changing the Subnet Representation

Although subnet masks are displayed as dotted decimal, the format can be changed to bit count or hex with the ip netmask-format bit command. It is very unlikely that you would ever need or in fact want to do this, but the command is available should you need it.

```
Router#show interface Serial0/0
```

Serial0/0 is administratively down, line protocol is down

Hardware is HD64570

Internet address is 192.168.1.1 255.255.255.0

```
Router#terminal ip netmask-format ?
```

bit-count Display netmask as number of significant bits

decimal Display netmask in dotted decimal

hexadecimal Display netmask in hexadecimal

```
Router#terminal ip netmask-format bit-count
```

```
Router#show interface Serial0/0
```

Serial0/0 is administratively down, line protocol is down

Hardware is HD64570

Internet address is 192.168.1.1/24

Bit count shows the address with a slash value, such as 192.168.1.1/24. Decimal displays the subnet mask in dotted decimal, such as 255.255.255.0. Hex shows the mask in hex, such as 0XFFFFF00.

There are only certain values available to use as a subnet mask due to binary mathematics. If you calculate a subnet mask and it is some other value such as 160, it is clearly wrong!

Table 3-5: Class B subnet mask values

Binary Value	Subnet Value
00000000	0
10000000	128
11000000	192

11100000	224
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

How Many Subnets and How Many Hosts?

Understanding my easy subnetting formula is crucial to passing the CCNA exam, as well as to designing and troubleshooting subnets in real-life networking. Many networking emergencies can be avoided by planning for future requirements during the design stage.

When planning a network addressing scheme, always ask clients what their expected growth for the next few years is and account for that. Never design a network addressing scheme for what they have currently.

The typical types of exam questions will be something like you are given a network ID and subnet mask and are asked to calculate how many subnets you can form and how many hosts there are per subnet. Or you will be given an IP address and subnet mask and asked which subnet the address is in. You might also be asked to indicate the broadcast address associated with the IP address and subnet mask.

It all boils down to the powers of two:

255.255.224.0

11111111.11111111.11100000.00000000
[16 bits] [3 bits][13 bits]
[Network] [Subnet][Host]

In the Class B example above, you can see that 224 has been converted to its binary value of 11100000 and so you have three subnet bits to use. You can use every combination of three binary numbers to make up different subnets:

000, 001, 010, 011, 100, 101, 110, 111

Up until fairly recently, we would have had to disregard the network 000 because this was the subnet, and we could not use the network 111 because it was for the broadcast. This is no longer the case (as of RFC 1878). I only mention it for those readers who have been in IT for 10 or more years and learned the old method. The command that allows us to use the first and last subnet is ip subnet-zero, which is now turned on by default on Cisco routers.

Alternatively:

How many subnets?

$$2^3 = (2 \times 2 \times 2) = 8$$

How many hosts? (you have 13 bits left for the host addresses)

$$2^{13} - 2 = 8,190$$

So for this subnet mask, you can see that you have eight subnets and each subnet has 8,190 hosts available for use.

Another Example

What are the first host and broadcast address for 131.107.32.0 255.255.224.0?

This is a Class B address and you are taking three of the host bits (three binary bits is 11100000, which is $128 + 64 + 32 = 224$). The last bit is 32 and you will use this as the count for subnets (i.e., they are all increments of 32 starting at zero, for example, 0, 32, 64, 96, etc.). There is no room to fit in every subnet permutation so you will have to shorten the output to the subnet you are being asked about:

1111111.1111111.11100000.00000000 255.255.224.0 (subnet mask)

10000011.01101011.00100000.00000000 131.107.32.0* (this is the subnet)

10000011.01101011.00100000.00000001 131.107.32.1 (this is the first host)

10000011.01101011.00100000.00000010 131.107.32.2 (this is the second host)

10000011.01101011.00100000.00000011 131.107.32.3 (this is the third host)

10000011.01101011.00100001.11111111 131.107.33.255 (keep counting up)

10000011.01101011.00100011.11111111 131.107.35.255 (keep going)

10000011.01101011.00100111.11111110 131.107.39.254 (keep going)

10000011.01101011.00111111.11111110 131.107.63.254 (this is the last host)

10000011.01101011.00111111.11111111 131.107.63.255 (broadcast address)

So, the hosts are 131.107.32.1 to 131.107.63.254 (8,190 in total)

131.107.32.0 = the subnet and 131.107.63.255 = the broadcast address for this subnet

Because the IP addresses changed from 32.1 and then 33.255 all the way up to 63.254, it is easy to look at them and mistake them for different subnets. When you write out the addresses correctly you can see that all of the hosts above are on the same subnet. Also, note that the third octet always starts with 001, so you know that all the host addresses are in the same subnet.

This means that they will all have to be attached to one router interface. You cannot decide to put half of your addresses on one side of the router and half on the other. Many engineers have made this mistake and then wasted hours trying to troubleshoot the problem.

In the example above, the first subnet would be 131.107.0.0, which is the zero subnet. We will look at this shortly.

Shortcut Method

Writing out IP addresses and subnets in binary is very time-consuming. There is a quicker way to do this, which we'll cover before we look at an even easier method. Just follow five simple steps:

1. How many subnets?

2^x to the power of masked bits, or 2^x

2. How many hosts per subnet?

2^y to the power of unmasked bits - 2

3. What are the valid subnets?

256 – the right-most non-zero subnet to give you the subnet increment

4. What are the valid hosts per subnet?

5. What is the broadcast address of the subnet?

Example

Which subnet is 131.107.32.1 255.255.224.0 in?

255.255.224.0 is 11111111.11111111.11100000.00000000 in binary.

You could actually write this out with a slash mask of /19 (19 masked network bits or 1s). You can see that you are subnetting on the third octet here.

1. How many subnets?

You have stolen three bits ($128 + 64 + 32 = 224$ subnet), so:

$2^3 = 8$ subnets (or $2 \times 2 \times 2 = 8$)

2. How many hosts per subnet?

You have 13 bits left for hosts, so:

$2^{13} - 2 = 8,190$ hosts

3. What are the valid subnets?

Take the right-most non-zero subnet (224) away from 256.

$256 - 224 = 32$

This part is crucial. If you get it wrong, then all of your subnetting will be off, so double-check it. Some people make the mistake of doubling the numbers (i.e., 32, 64, 128, etc.). The Subnetting Cheat Chart will actually make this step easier.

You now know that you have eight valid subnets and each subnet will be an

increment of 32. You can start at 0 if subnet zero is permitted: 0, 32, 64, 96, 128, 160, 192, 224. (0 and 224 are valid because subnet zero is allowed.)

4. What are the valid hosts per subnet?

It is best to write out the subnet number and the next subnet number. From this you can work out the first and valid hosts.

1st subnet: 131.107.0.0 i **This is the zero subnet**

2nd subnet: 131.107.32.0 i **131.107.32.1 is in this subnet**

3rd subnet: 131.107.64.0

4th subnet: 131.107.96.0

5th subnet: 131.107.128.0

6th subnet: 131.107.160.0

7th subnet: 131.107.192.0

8th subnet: 131.107.224.0

So, now you know that you can use any number, including the 224 subnet. It is fairly clear from the list above that the host address you are looking for is in the second subnet, so you can look at the available host numbers in this subnet.

Subnet: 131.107.32.0

1st host: 131.107.32.1

2nd host: 131.107.32.2

3rd host: 131.107.32.3 ... You could keep going, but there are over 8,000 hosts.

Last host: 131.107.63.254 (take one away for the broadcast to get this value)

Broadcast: 131.107.63.255 (take one away from the next subnet [131.107.64.0] to get this value).

For the third or .64 subnet, follow the same process of writing out the subnet and broadcast address. You can work out the broadcast address by writing out the next subnet. The subnet starting 131.107.64 is the third of the eight subnets. Take one away from that to determine the broadcast address for the previous subnet.

Subnet: 131.107.64.0 (take one away to get the broadcast for the .32 subnet)

1st host: 131.107.64.1

Last host: 131.107.95.254

Broadcast: 131.107.95.255

You could go on, but can you see that you've gone past the IP address you were trying to find, which is 131.107.32.1.

5. What is the broadcast address?

The broadcast address is 131.107.63.255

It's pretty easy to work this out if you write out the subnets first.

What just happened there?

Without being able to see the whole thing in binary, it does look a little strange. You just have to have confidence that the method works and if in doubt go back to binary. You can prove that the broadcast address is such by checking to see whether it's all 1s in binary. The last host will be the last number you can use without having all 1s (255 in decimal):

10000011.01101011.001111111111110 131.107.63.254 (one host bit not on)

10000011.01101011.001111111111111 131.107.63.255 (all host bits on here)

The underlined bits are the subnet bits. The .254 has all the host bits apart from one turned on. The broadcast address has all the host bits turned on. This tells the network that it is a broadcast packet to the subnet.

If you follow the process for each subnet using a subnet number of 255.255.224.0 going up in increments of 32, you will get the following host address ranges:

Subnet 1: 131.107.0.1 to 131.107.31.254

Subnet 2: 131.107.32.1 to 131.107.63.254 **i You can see that host 32.1 is in this subnet**

Subnet 3: 131.107.64.1 to 131.107.95.254

Subnet 4: 131.107.96.1 to 131.107.127.254

Subnet 5: 131.107.128.1 to 131.107.159.254

Subnet 6: 131.107.160.1 to 131.107.191.254

Subnet 7: 131.107.192.1 to 131.107.223.254

Subnet 8: 131.107.224.1 to 131.107.255.254

Another Example

Which subnet is host 10.20.1.23 255.240.0.0 in?

Take four bits (240 in binary is 11110000 or 128 + 64 + 32 + 16) to make subnets from since Class A addresses normally have the default of eight bits:

255.240.0.0 is 11111111.11110000.00000000.00000000 in binary. Or you can shorten it to 10.20.1.23 /12.

1. How many subnets?

$$2^4 = 16$$

2. How many hosts per subnet?

You have 20 bits left for hosts, so:

$$2^{20} - 2 = 1,048,574 \text{ hosts per subnet}$$

3. What are the valid subnets?

$256 - 240 = 16$ (the increment, so the subnets go up in increments of 16)

10.0.0.0 (the zero subnet)

10.16.0.0

10.32.0.0

10.48.0.0 ...all the way up to...

10.224.0.0

10.240.0.0

4. What are the valid hosts per subnets/broadcasts?

10.0.0.0 – Hosts 10.0.0.1 to 10.15.255.254

10.16.0.0 – Hosts 10.16.0.1 to 10.31.255.254 **i 10.20.1.23 is in this subnet**

10.32.0.0 – Hosts 10.32.0.1 to 10.47.255.254

10.48.0.0 – Hosts 10.48.0.1 to 10.63.255.254

10.64.0.0 and so on...

10.224.0.0 – Hosts 10.224.0.1 to 10.239.255.254

10.240.0.0 – Hosts 10.240.0.1 to 10.255.255.254

What are the broadcast addresses?

This is the last address before each subnet

10.15.255.254 is the last host on the first subnet

10.15.255.255 is the broadcast address for the first subnet

10.16.0.0 is the next subnet

10.16.0.1 is the first host on the next subnet

10.31.255.254 is the last host

10.31.255.255 is the broadcast address

10.32.0.0 is the next subnet

10.47.255.255 is the broadcast address on the third subnet

10.48.0.0 is the next subnet, etc.

You've already gone past the point you need to answer the question. Now you will learn how to shorten these five steps.

Example

Which subnet is host 192.168.21.41/28 in?

The first and most important task is to work out how to change the /28 mask into a full subnet mask. You already know that each octet is eight binary bits and $8 + 8 + 8 = 24$ binary bits, which is 255.255.255.0. You need to add 4 to 24 to get 28, which is $128 + 64 + 32 + 16$ binary places, or 11110000, or 240. The Subnetting Cheat Chart (shown later) will make this task so easy you will wonder what all the fuss was about.

With four bits of masking, you now have $2^4 = 16$ subnets. This leaves four bits for the hosts, which is $2^4 - 2 = 14$ hosts per subnet. You weren't asked how to answer this question but the exam could easily tag this on as part of the question above.

The subnet mask is 255.255.255.240 (four masked bits on the last octet).

$256 - 240 = 16$ increments

Now count up in subnets until you find the one where host 41 is residing (subnet 3 below):

Subnet 1: 192.168.21.0 hosts 1–14 (broadcast = 15) **i This is the zero subnet**

Subnet 2: 192.168.21.16 hosts 17–30 (broadcast = 31)

Subnet 3: 192.168.21.32 hosts 33–46 (broadcast = 47) **i 41 is in this range**

Subnet 4: 192.168.21.48 hosts 49–62 (broadcast = 63)

Subnet 5: 192.168.21.64 hosts 65–78 (broadcast = 79)

and so on ... up to ...

Subnet 16: 192.168.21.240 hosts 241–254 (broadcast = 255)

In the exam I always found the correct subnet the host was in and counted one more up just in case I made a mistake. However, there are no extra points for counting up higher and time is ticking away.

Example

Which subnet is host 10.65.2.5/10 in?

Turn the /10 into a subnet mask. 255 is eight binary bits, so you need to add 2 to get to 10.

Two binary bits is $128 + 64$, which is 192, or 11000000.

Two masked subnet bits = 255.192.0.0.

$2^2 = 4$ subnets.

You have $2^{22} - 2 = 4,194,302$ hosts per subnet.

$256 - 192 = 64$ increments (for the subnets).

Subnet 1: 10.0.0.0 hosts 10.0.0.1–10.63.255.254 (broadcast = 255)

Subnet 2: 10.64.0.0 hosts 10.64.0.1–10.127.255.254 (broadcast = 255) **i Host 10.65.2.5 is in this range**

Subnet 3: 10.128.0.0 hosts 10.128.0.1–10.191.255.254 (broadcast = 255)

Subnet 4: 10.192.0.0 hosts 10.192.0.1–10.255.255.254 (broadcast = 255)

Let's use the Subnetting Cheat Chart to answer the next question. If this is the first time you've read this chapter, then do the next example the second time you read it.

Example

Which subnet is 192.168.100.203/27 in?

You know that three binary octets is 24 bits, and to get to 27 you need to add three bits. If you have already done a read-through of this chapter and have seen the Subnetting Cheat Chart (below), you will know you can easily tick down three numbers on the top subnetting part of the chart. This will give you the value 224.

So the subnet /27 is 255.255.255.224. You can then tick three across the top to get the subnet increment, which is 32 (or just take 224 away from 256, if you prefer).

You are subnetting on the fourth octet, so just add up the subnets until you get to the one with the number 203 in it. I prefer to start with a multiple of 32 to save time (160, for example), but you may prefer to start with 0, 32, 64, etc. for now.

Just remember that in the exam time is of the essence.

Subnet 1: 192.168.100.0 hosts 1–30 (broadcast = 31)

Subnet 2: 192.168.100.32 hosts 33–62 (broadcast = 63)

Subnet 3: 192.168.100.64 hosts 65–94 (broadcast = 95)

Subnet 4: 192.168.100.96 hosts 97–126 (broadcast = 127)

Subnet 5: 192.168.100.128 hosts 129–158 (broadcast = 159)

Subnet 6: 192.168.100.160 hosts 161–190 (broadcast = 191)

Subnet 7: 192.168.100.192 hosts 193–222 (broadcast = 223) **i Host 203 is in this subnet**

Subnet 8: 192.168.100.224 hosts 225–254 (broadcast = 255)

The host 192.168.100.203 is in the subnet 192.168.100.192.

In the exam you will have one of two types of subnetting questions. The first type you

have just worked through: given a certain network number and subnet mask, you need to determine which subnet the IP address is in. The second type of question will ask you to design a subnet mask to give a customer a certain number of hosts and a certain number of subnets.

Working out How Many Hosts and How Many Subnets

The special chart below will enable you to answer any question in the exam. I call it the Subnetting Cheat Chart. This chart will ensure that you can answer any subnetting question in the exam within 30 seconds after practice; just watch and see for yourself. In the exam you will be given a small whiteboard and pen that you can use to write out the chart below:

Subnetting Cheat Chart

8192								
16,384								

What the Subnetting Cheat Chart helps you to do is easily and quickly work out how many bits are being used for subnetting, which subnet the host is in, how many hosts per subnet, and how many subnets. Let's look at a design example.

How many subnets and hosts does 192.168.2.0/26 give you?

First, take an extra two bits from the normal 24-bit mask. Tick off two numbers down in the upper portion of the chart (128 and then 192), giving you the mask of 192; or to be more specific, 255.255.255.192.

You can work out that it is two bits being used if you remember that each octet count is eight. 255.0.0.0 is eight binary bits, 255.255.0.0 is 16, and 255.255.255.0 is 24. If you have a /26 mask, then you need to add two onto the 255.255.255.0 mask, which is 24 bits plus two more (or 255.255.255.192).

You've taken two bits for the subnet, so in the Subnets column on the bottom, tick down two numbers (2 and then 4). This gives you four subnets.

Now you know you have six bits left for the hosts ($8 - 2 = 6$ bits remaining), so tick off six places down in the Hosts Minus 2 column to get the number of hosts. You will always use the bottom part of the chart first when trying to work out how many subnets and hosts per subnet. If you're trying to work out which subnet a host address is in, use the top part of the chart.

Six down in the Hosts Minus 2 column gives you 64; take two away for the subnet and broadcast and that gives you 62 hosts per subnet. You can see I also ticked across two at the top bits row so you can see that our subnets go up in increments of 64.

Easy, isn't it?

Subnetting Cheat Chart

	Bits	128	64	32	16	8	4	2	1
Subnets		ü	ü						
128	ü								
192	ü								
224									
240									
248									
252									
254									

255									
Powers of Two	Subnets	Hosts Minus 2							
2	ü	ü							
4	ü	ü							
8		ü							
16		ü							
32		ü							
64		ü							
128									
256									
512									
1024									
2048									
4096									
8192									
16,384									

Another Example

You are given a network address of 192.168.5.0 255.255.255.0. You are the network administrator and you need to subnet this address to make six subnets, each with at least 15 hosts per subnet.

Go back to the chart and tick down the Subnets column on the bottom until you get to a number that gives you six subnets: 2 and 4 will not be enough, but 8 gives you six subnets with two extra (it is okay if you have to go just over). So now you need to steal three bits for subnetting. Count down three in the upper portion of the Subnetting Cheat Chart, starting at 128, then 192, and finally 224.

Stealing three bits leaves you five bits left for hosts per subnet. If you tick down five places in the Hosts Minus 2 column, you will see that this gives you 30 (32 - 2) hosts per subnet, which are more than enough for your requirements.

The answer is you need subnet 255.255.255.224 (or a /27 mask) to get your six subnets. If you tick down three in the bottom Subnets column (which is 8), you can tick down three in the upper portion of the chart to determine the correct subnet mask (which is 224). On the top bits row you can see that our subnets go up in increments of 32.

Subnetting Cheat Chart

	Bits	128	64	32	16	8	4	2	1
--	------	-----	----	----	----	---	---	---	---

Subnets		ü	ü	ü				
Powers of Two	Subnets	Hosts Minus 2						
128	ü	ü						
192	ü	ü						
224	ü	ü						
240								
248								
252								
254								
255								
2	ü	ü						
4	ü	ü						
8	ü	ü						
16		ü						
32		ü						
64								
128								
256								
512								
1024								
2048								
4096								
8192								
16,384								

Let's do one final example, this time back to which subnet a certain host is in:

Which subnet is host 172.16.100.119/29 in?

This is a Class B address but for subnetting you don't need to concern yourself with that. You just need to look at which octet the subnetting is happening in.

Because the third octet is already filled in with binary 1s, we will disregard it (to save time) and look only at the last octet.

/29 is five places into the last octet, so start by ticking five places across in the upper portion of the chart and then five down. You can now see that you have a subnet mask of 255.255.255.248 and the subnets are going up in increments of 8.

Subnetting Cheat Chart

	Bits	128	64	32	16	8	4	2	1
Subnets		ü	ü	ü	ü	ü			
128	ü								
192	ü								
224	ü								
240	ü								
248	ü								
252									
254									
255									
Powers of Two	Subnets	Hosts Minus 2							
2									
4									
8									
16									
32									
64									
128									
256									
512									

Unfortunately, you are counting up in increments of 8, meaning that if you were creating a chart at work for allocating subnets, you would start with 172.16.0.0, then 172.16.0.8, and so on (until you've worked out all 8,192 subnets), but you need to answer the exam question quickly. With this in mind, let's focus only on the last octet since this is where you'll find the host address:

172.16.100.0

172.16.100.8

This will still take too long, so jump up from values of 8 to 80.

172.16.100.80

172.16.100.88

172.16.100.96

172.16.100.104

172.16.100.112 **i Host 172.16.100.119 is in this subnet**

172.16.100.120

With practice you should be able to answer most subnetting questions in around 30 seconds if you use the chart. There are some free examples using the subnetting chart at the resources URL below, so please refer to them also:

www.howtonetwork.com/ccnasimplified

Secondary IP Address

You are not restricted to using just one IP address on an interface of a Cisco router. You can assign an unlimited number of addresses using the ip address [ip address] [subnet mask] secondary command:

```
Router#config t  
Router(config)#interface Serial0/0  
Router(config-if)#ip address 192.168.1.1 255.255.255.0  
Router(config-if)#ip address 192.168.2.1 255.255.255.0 secondary
```

You may need to do this because you need to use more than one subnet on the same physical segment of the network. An example of this would be an insufficient amount of host addresses available for the subnet you are using. In the real world, secondary IP addresses should be used only as a temporary workaround. For a permanent solution you should use subinterfaces or SVIs for each subnet.

It's also worth noting that each interface should be on a separate subnet. If you try to use two addresses from the same subnet on different interfaces, the router will throw an error message. Try it for yourself:

```
R1(config)#int f0/0  
R1(config-if)#ip add 192.168.1.1 255.255.255.0  
R1(config-if)#no shut  
R1(config-if)#int f0/1  
R1(config-if)#ip add 192.168.1.2 255.255.255.0  
R1(config-if)#no shut  
% 192.168.1.0 overlaps with FastEthernet0/1  
FastEthernet0/0: incorrect IP address assignment
```

Route Summarization

There are many millions of routes on the Internet. If these routes all had to be stored individually, the Internet would have come to a stop many years ago. Route summarization, also known as supernetting or address aggregation, was proposed in RFC 1338, which you can read online by visiting www.faqs.org/rfcs/rfc1338.html.

Route summarization allows a router to advertise a summary of a group of networks it can reach. This conserves network resources by reducing routing table sizes and bandwidth because fewer routes are advertised.

For example, if you drive along a street with 10 avenues attached, each avenue could have a sign at the end showing a summary of the houses you will find, such as 1–10 Alphabet Ave., 11–20 Alphabet Ave., etc. At the very end of the street there could be another sign showing you 1–100 Alphabet Ave., with a summary of the 10 Alphabet Avenues.

Working Out Summary Routes

The easy way to establish the summary route you can advertise is to write out the common binary bits and write out the subnet mask, along with the lowest subnet:

192.168.100.128/27

192.168.100.160/27

192.168.100.192/27

192.168.100.224/27

In the example above, all the binary bits in the first three octets will agree because the numbers are all exactly the same. The last octet does differ though:

128 = 10000000

160 = 10100000

192 = 11000000

224 = 11100000

The only bit that matches on all of the above subnets is the first one. If you add that to the first three octets, you will have 25 matching bits: 192.168.100.128/25. You would advertise 192.168.100.128/25 for these routes as a summary route.

Route Summarization Prerequisites

Many network engineers put little or no thought into their IP addressing, and when it comes to sending out a summary route they find that it isn't possible. If you chose random subnets for your network, you would not be able to summarize them because the numbers are not contiguous.

As well as having contiguous addressing of subnets, you must use a routing protocol that supports VLSM, such as RIPv2 (Routing Information Protocol version 2), EIGRP (Enhanced Interior Gateway Routing Protocol), or OSPF (Open Shortest Path First).

This is not usually a problem because classful protocols such as RIP and IGRP are rarely used.

Once you have passed your CCNA exam, I urge you to take the Cisco Certified Design Associate (CCDA) exam, which includes many of the subjects from the CCNA RS, because it will give you a very strong understanding of network design principles. Not many engineers are good at both, which means you will have a big advantage over them.

Breaking the Subnet Boundary

You know that you can't apply a Class A subnet mask to a Class C IP address such as 192.168.1.1/11 because you are trying to use network bits for host bits. But, if you are advertising a summary route, you can reduce the subnet mask so that a group of Class C networks can be advertised with a mask of /12, for example.

In the network addresses below, you only need to find the common bits in the subnet mask and apply that to the lowest available subnet.

CIDR also allows for supernetting, which enables you to advertise a summary of your network addresses, providing you have a contiguous block. For example, if you owned the networks 172.16.20.0/24 up to 172.16.23.0/24, then you could advertise the single network 172.16.20.0/22 out to the Internet. The advantage of this is saving bandwidth and greater efficiency. This is also known as route summarization.

Route summarization only works if you calculate the addresses in binary first:

11111111.11111111.11111111.00000000 = 24-bit mask

10101100.00010000.00010100.00000000 = 172.16.20.0

10101100.00010000.00010101.00000000 = 172.16.21.0

10101100.00010000.00010110.00000000 = 172.16.22.0

10101100.00010000.00010111.00000000 = 172.16.23.0

All of the underlined parts of the address are common and can be aggregated with one subnet mask to advertise them all. There are 22 common bits, so you can use the mask 255.255.252.0 or /22 to advertise the entire block of addresses. Figure 3.4 below shows the summarized network being advertised:

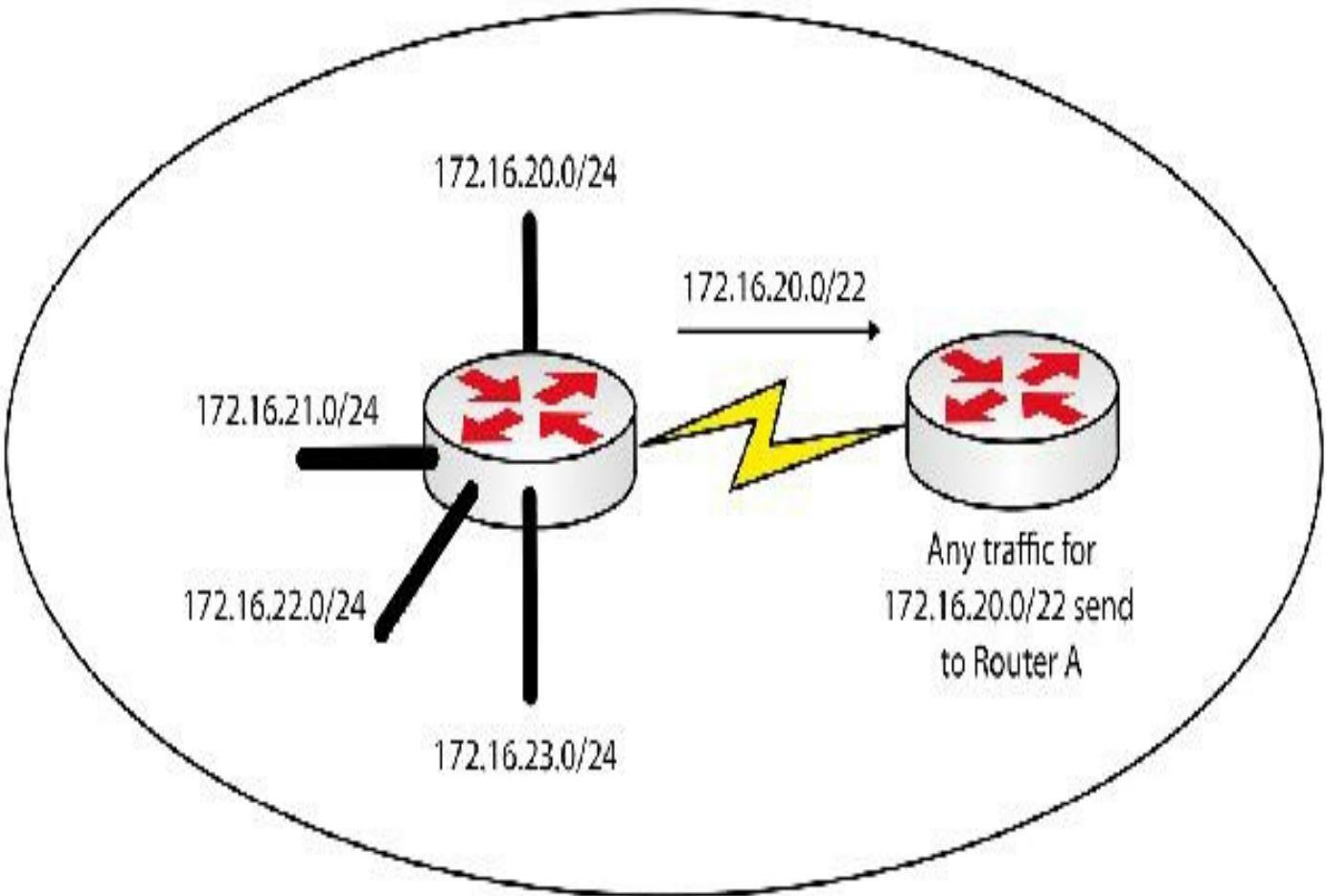


FIG 3.4 – Supernetting reduces the amount of routes advertised

A carefully planned IP addressing scheme is designed with summarization in mind. The advantages are many but primarily they are reduced routing tables (saving router CPU and bandwidth) and easier network management and troubleshooting, making your life as a network administrator far easier.

VLSM

Although subnetting provides a useful mechanism to improve the IP addressing issue, in the past network administrators were only able to use one subnet mask for an entire network. RFC 1009 addressed this issue by allowing a subnetted network to use more than one subnet mask.

Think of it as being given a large slice of cake. You can cut that slice of cake into two (or more) pieces, or you could take one of the two slices and cut that into smaller slices. It's basically subnetting a subnet.

Today a network administrator can have a Class B address with a 255.255.192.0 mask and further break down that subnet into smaller units with more masks, such as 255.255.224.0. Instead of writing out the subnets in decimal, engineers in the real world use something called a slash address, writing out how many bits are used for subnetting.

Some examples of this are shown below:

255.255.0.0 can be expressed as /16 because there are 16 binary bits masked.

11111111.11111111.00000000.00000000 = 16 on or masked bits.

255.255.192.0 can be expressed as /18 because there are 18 binary bits masked.

11111111.11111111.11000000.00000000 = 18 on or masked bits.

255.255.240.0 can be expressed as /20 because there are 20 binary bits masked.

11111111.11111111.11110000.00000000 = 20 on or masked bits.

It is best to illustrate this with an example, as in Figure 3.5 below:

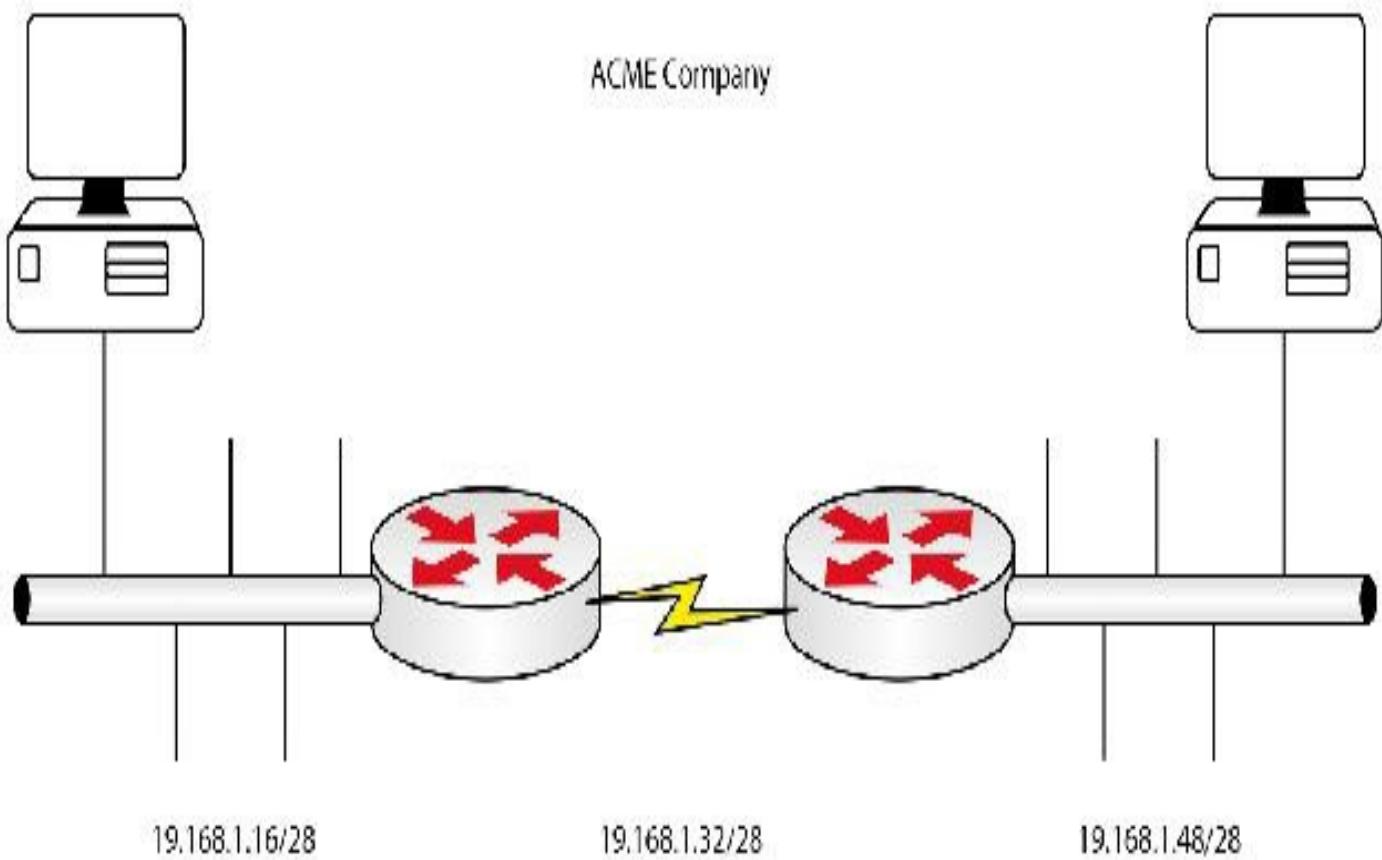


FIG 3.5 – ACME Company with no VLSM

You may have spotted a few problems with the addressing scheme above. The most important issue is the breach of the conservation of IP addresses. If you are using RFC 1918 addresses (non-routable such as 10.x.x.x), then perhaps you may not be worried about address wastage, but this is very bad practice and, for Cisco exams, you can guarantee an expectation that you will conserve IP addresses. With a /28 mask (or 255.255.255.240) you have 14 hosts per subnet. This may be fine for your LAN on either end, but for your WAN connection you only need two IP addresses, which wastes 12 addresses. You could change the masks to /30 (or 255.255.255.252), but then for

your LANs you will obviously need more than two hosts.

The first workaround is to buy a separate network address for each network (two LANs and one WAN) but this would prove expensive and unnecessary. The other alternative is to break down the subnet further using VLSM, which is actually what it was designed to do!

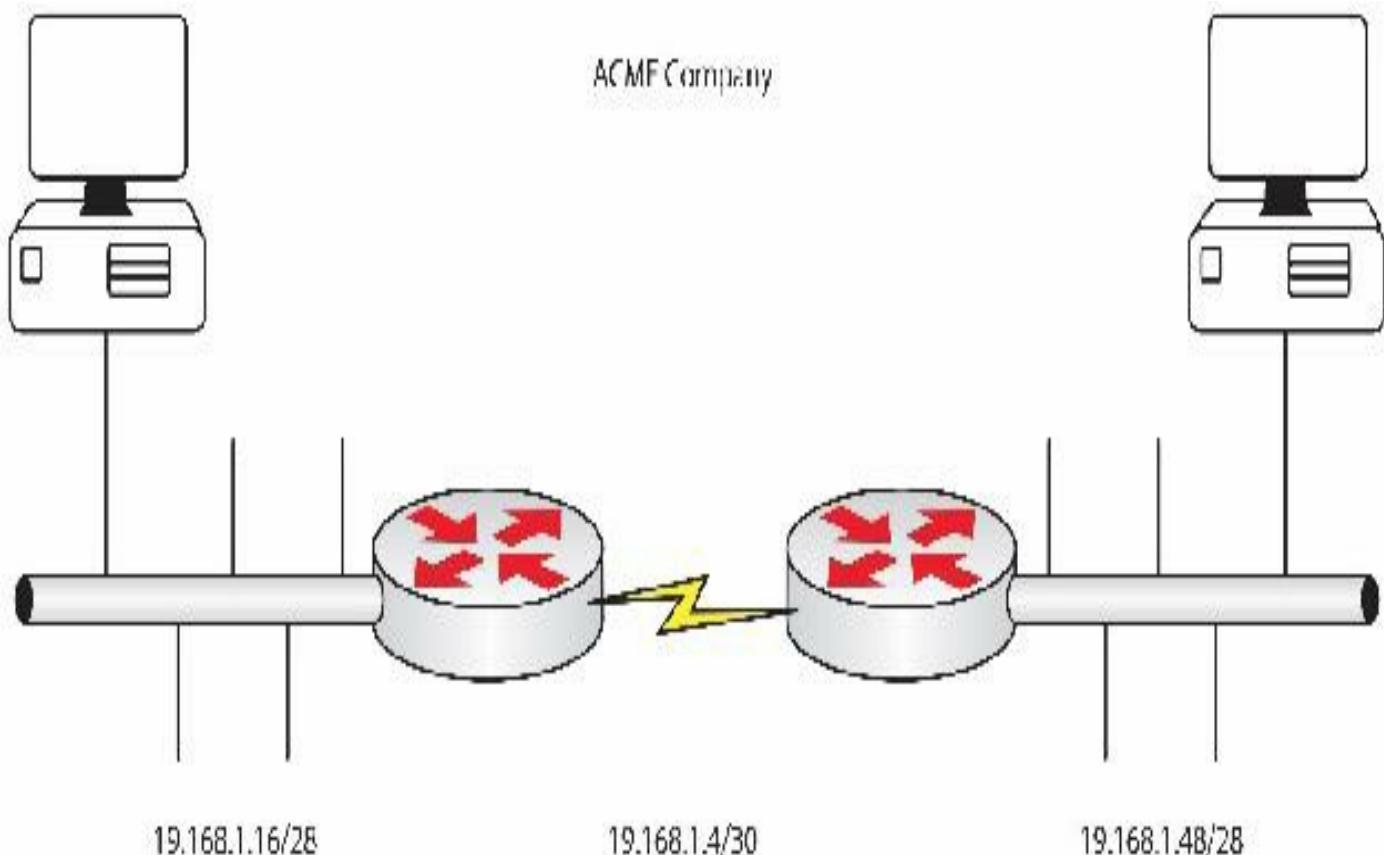


FIG 3.6 – ACME Company with VLSM

In Figure 3.6 above, you can see that the WAN link now has a /30 mask, which produces two usable hosts. You also have a tighter addressing allocation. If ACME Company expands (as companies often do) you can easily allocate further WAN links and LANs.

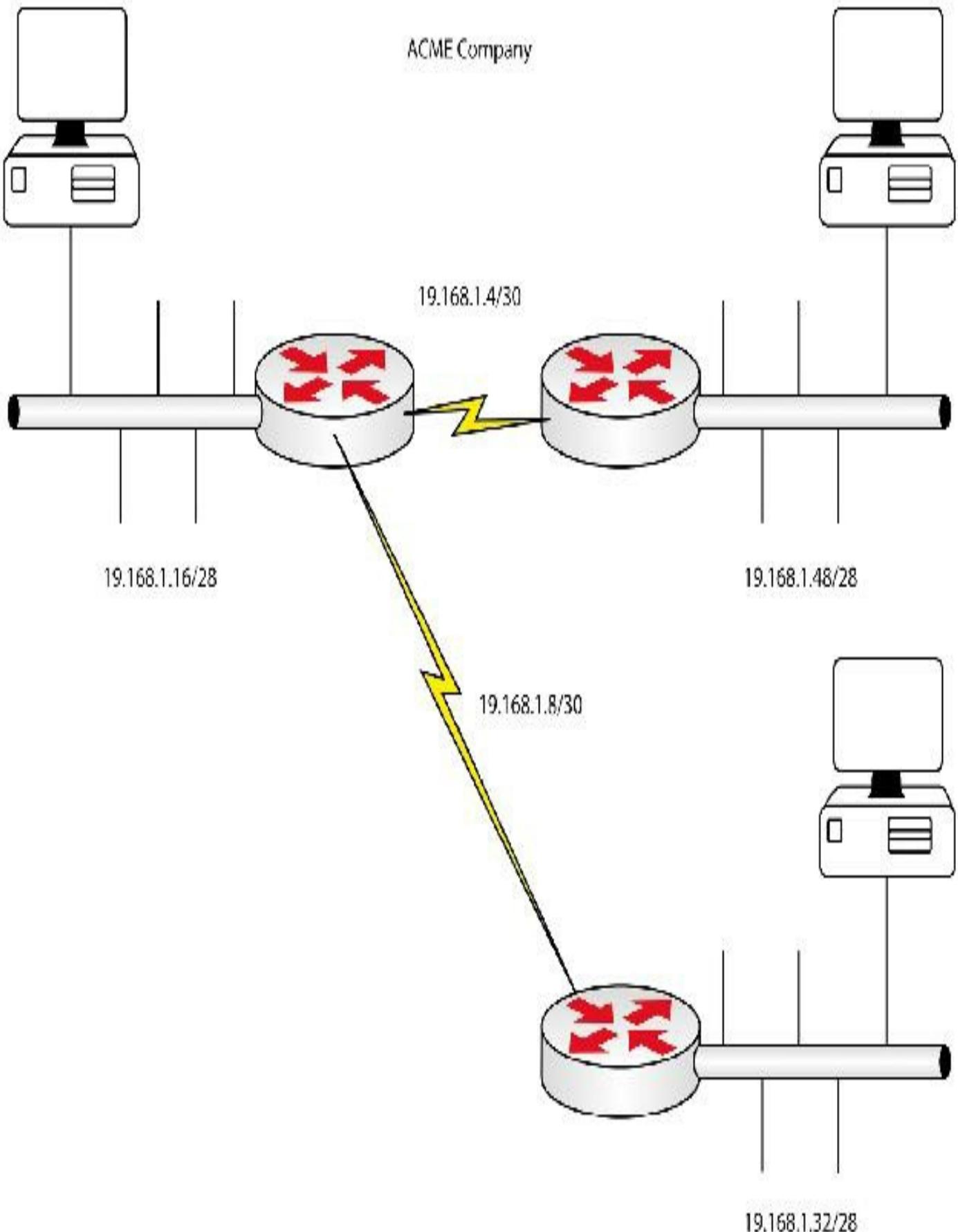


FIG 3.7 – ACME Company with a new office

In Figure 3.7 above, you can see that ACME Company has now grown and added a remote office. Because you have taken the time to plan and allocate a carefully thought out VLSM scheme, you can simply allocate the next block of IP addresses.

But will the IP addresses clash? This is a very common question and it's a valid one. Let's say you have address 19.16.1.1/28 for one of your LANs. You will not, therefore, be able to use the IP address 19.16.1.1 with any other subnet mask. The IP address can only be used once, no matter which subnet mask is attached to it.

It is a bit of a head scratcher for people who are new to networking or subnetting, but it does work. The general concept here is that VLSM does not remove the need for unique IP addresses, it just helps to efficiently manage their use.

VLSM Practice for the CCNA Exam

Here is a network you have been asked to design an addressing scheme for:

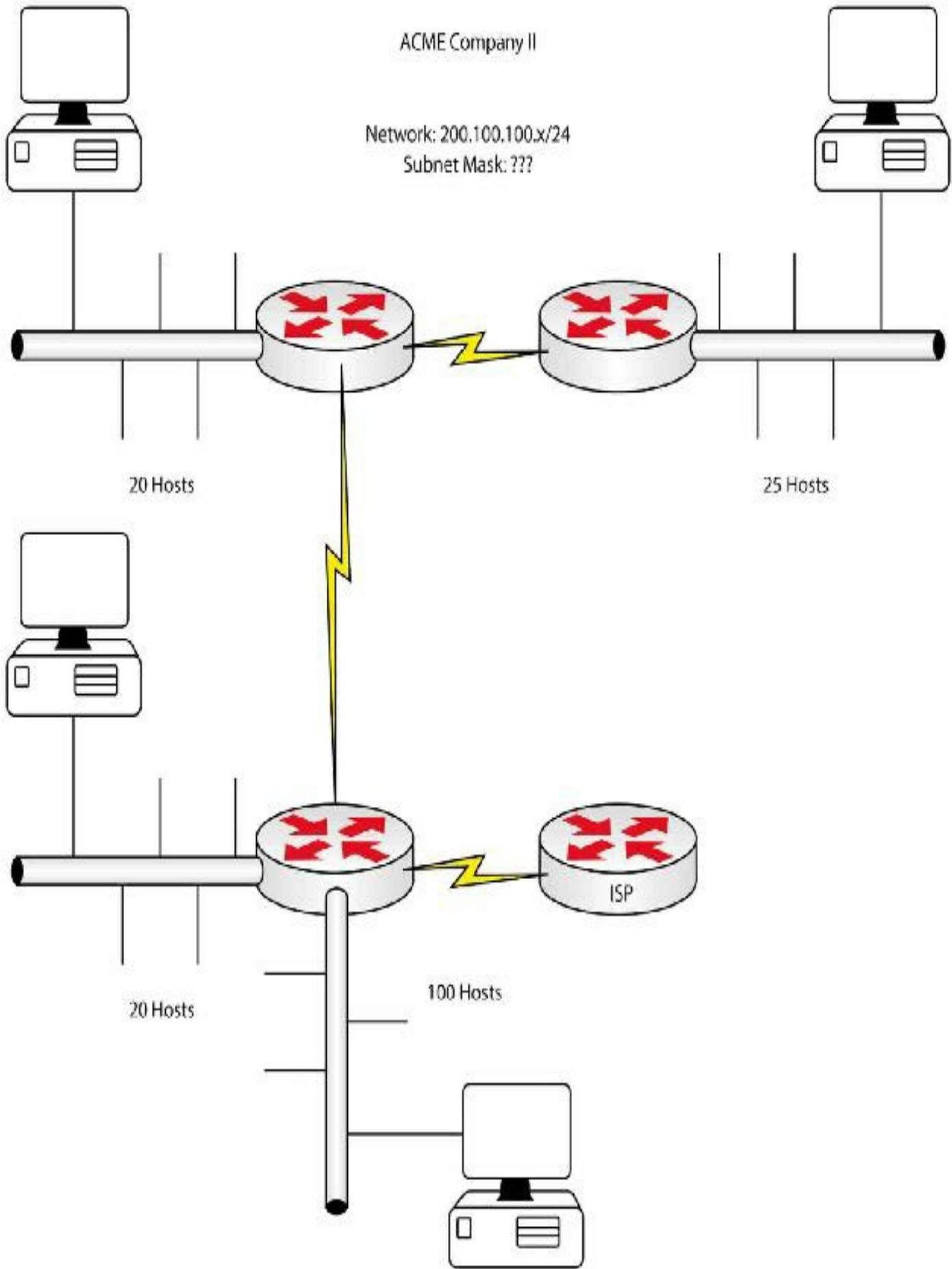


FIG 3.8 – ACME II Company

In Figure 3.8 above, ACME II Company has been allocated the 200.100.100.x network, with a default mask of 255.255.255.0. If you keep the standard mask, you will be left with one network with 254 usable hosts. Using the bottom half of the Subnetting Cheat Chart introduced earlier, tick down eight places in the Hosts Minus 2 column, which would give you one subnet with 256 - 2 hosts, or 254 hosts.

The challenge is this: You have three Serial connections and each requires only two usable host addresses. You also have four LANs that need between 20 and 100 hosts. If you design a mask to give you 20 to 100 hosts, you will be wasting a lot of addresses. To get 100 hosts, tick down seven places in the Hosts Minus 2 column, which would give you a mask of 255.255.255.128 (because you have only one bit left to tick down in the Subnets column). This gives you 126 hosts (128 - 2). You would then have two networks: one starting at 200.100.100.0 and one starting at 200.100.100.128. Not great, to be honest, because you need seven subnets (three WAN and four LAN) and some require only 20 host—so why waste 108 addresses?

Referencing the bottom half of the Subnetting Cheat Chart below, tick down in the Hosts Minus 2 column until you find a number close enough to give you 100 hosts. The only number you can use is 128, which is seven ticks down, so you are stealing seven bits from the host portion, leaving you one bit for subnetting.

Powers of Two	Subnets	Hosts Minus 2
2	ü	ü
4		ü
8		ü
16		ü
32		ü
64		ü
128		ü
256		
512		

Using the upper portion of the Subnetting Cheat Chart, tick down one place to reveal the subnet mask of 128.

Subnets	

128	ü
192	
224	
248	
252	
254	
255	

When you use the 128 subnet with ACME II Company's IP address, you get subnet 200.100.100.0 and subnet 200.100.100.128, both with a mask of /25, or 255.255.255.128. For the network needing 100 hosts, you can use the 200.100.100.128 subnet. For the first host, you will use 200.100.100.129 and so on up to 200.100.100.229. So, now you have:

200.100.100.128/25 – LAN (hosts 129–254)

200.100.100.0/25 – available for use or for VLSM

You need to allocate hosts to the three remaining LAN networks and the three WANs. The other three LANs all need between 20 and 30 hosts. If you tick down five places in the Hosts Minus 2 column, you will get $32 - 2$, or 30 hosts. If you steal five bits from the host portion, you are left with three bits for the subnet (because there are eight bits in every octet).

Powers of Two	Subnets	Hosts Minus 2
2	ü	ü
4	ü	ü
8	ü	ü
16		ü
32		ü
64		
128		
256		
512		

Tick down three places in the upper portion of the Subnetting Cheat Chart to reveal a subnet mask of 224. This mask will give you eight subnets (you only need three for the

LANs) and each subnet will have up to 30 available host addresses. Can you see how this will fit ACME II Company's requirements?

Subnets	
128	ü
192	ü
224	ü
240	
248	
252	
254	
255	

If you tick across three places in the upper portion of the Subnetting Cheat Chart you will see that the subnets go up in increments of 32, so the subnets will be 0, 32, 64, and 96; you cannot use 128 because this was used for the large LAN.

Bits	128	64	32	16	8	4	2	1
	ü	ü	ü					

So, now you have:

200.100.100.0/27 – Reserve this for the WAN links

200.100.100.32/27 – LAN 1 (hosts 33–62)

200.100.100.64/27 – LAN 2 (hosts 65–94)

200.100.100.96/27 – LAN 3 (hosts 97–126)

Next, you need IP addresses for the three WAN connections. WAN IP addressing is fairly easy because you only need two IP addresses if it is a point-to-point link. In the Hosts Minus 2 column, tick down two places to get 4 - 2, or two hosts. This leaves six bits for the subnet.

Powers of Two	Subnets	Hosts Minus 2
2	ü	ü
4	ü	ü
8	ü	

16	ü	
32	ü	
64	ü	
128		
256		
512		

Tick down six places in the upper portion of the Subnetting Cheat Chart to get 252 as the subnet mask.

Subnet	
128	ü
192	ü
224	ü
240	ü
248	ü
252	ü
254	
255	

Network Addresses

As a network administrator, you will need to keep a record of the IP addresses and subnets used. So far, you have allocated the following addresses:

WAN links

- 200.100.100.0/30 – WAN link 1 (hosts 1–2)
- 200.100.100.4/30 – WAN link 2 (hosts 5–6)
- 200.100.100.8/30 – WAN link 3 (hosts 9–10)

LAN hosts

- 200.100.100.32/27 – LAN 1 (hosts 33–62)
- 200.100.100.64/27 – LAN 2 (hosts 65–94)
- 200.100.100.96/27 – LAN 3 (hosts 97–126)

Large LAN hosts

200.100.100.128/25 – LAN (hosts 129–254)

Chopping Down

VLSM principles will let you take a network and slice it down into smaller chunks. Those chunks can then be sliced into smaller chunks and so on. You will reach the limit only when you get to the mask 255.255.255.252, or /30, because this gives you two usable hosts, which is the minimum you would need for any network.

Consider network 200.100.100.0/24. If you change the mask from /24 to /25, this is what happens:

Original mask (last octet)	00000000	1 subnet	254 hosts
New mask (Subnet 1)	00000000	200.100.100.0 – Subnet 1	126 hosts
New mask (Subnet 2)	10000000	200.100.100.128 – Subnet 2	126 hosts

Now you have two subnets. If you take the new Subnet 2 of 200.100.100.128 and break it down further by changing the mask from /25 to /26, you get this:

Original mask (last octet)	10000000	1 subnet	126 hosts
New mask (Subnet 1)	10000000	200.100.100.128 – Subnet 1	62 hosts
New mask (Subnet 2)	11000000	200.100.100.192 – Subnet 2	62 hosts

If you take the second subnet and break it down further by changing the mask from /26 to /28 (for example), you get this:

Original mask (last octet)	11000000	1 subnet	62 hosts
New mask (Subnet 1)	11000000	200.100.100.192 – Subnet 1	14 hosts
New mask (Subnet 2)	11010000	200.100.100.208 – Subnet 2	14 hosts
New mask (Subnet 3)	11000000	200.100.100.224 – Subnet 3	14 hosts
New mask (Subnet 4)	11100000	200.100.100.240 – Subnet 4	14 hosts

In Summary

Hopefully, this has helped you understand a bit more about VLSM. It's no mystery, really. Please take the time to go over the examples above again, and then have a go at the challenge below.

ACME Company II has been allocated the address 200.10.200.x/24, as shown in Figure

3.9. You are required to design an addressing system so that hosts can be given IP addresses and the WAN links can be addressed with no wastage.

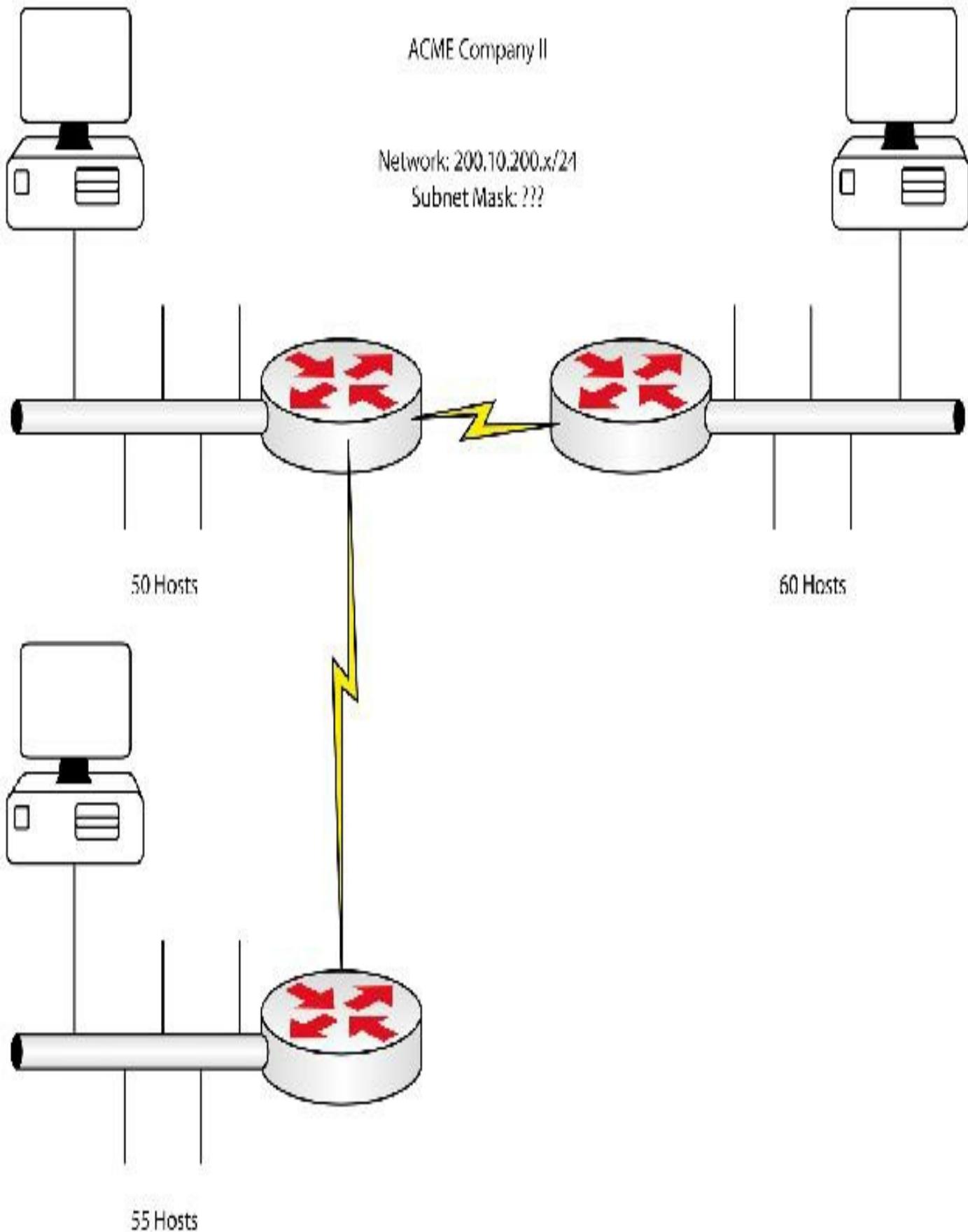


FIG 3.9 – VLSM address allocation challenge

End of Chapter Questions

Please also visit www.howtonetwork.com/ccnasimplified to take the free Chapter 3 exam.

Reality Press Limited has hired you to configure summary routes for their network. The previous network engineer has correctly allocated contiguous subnets to the network, but it's your job to aggregate these internally and then send just one summary network out to the Internet according to the summarization challenge shown in Figure 3.10 below:

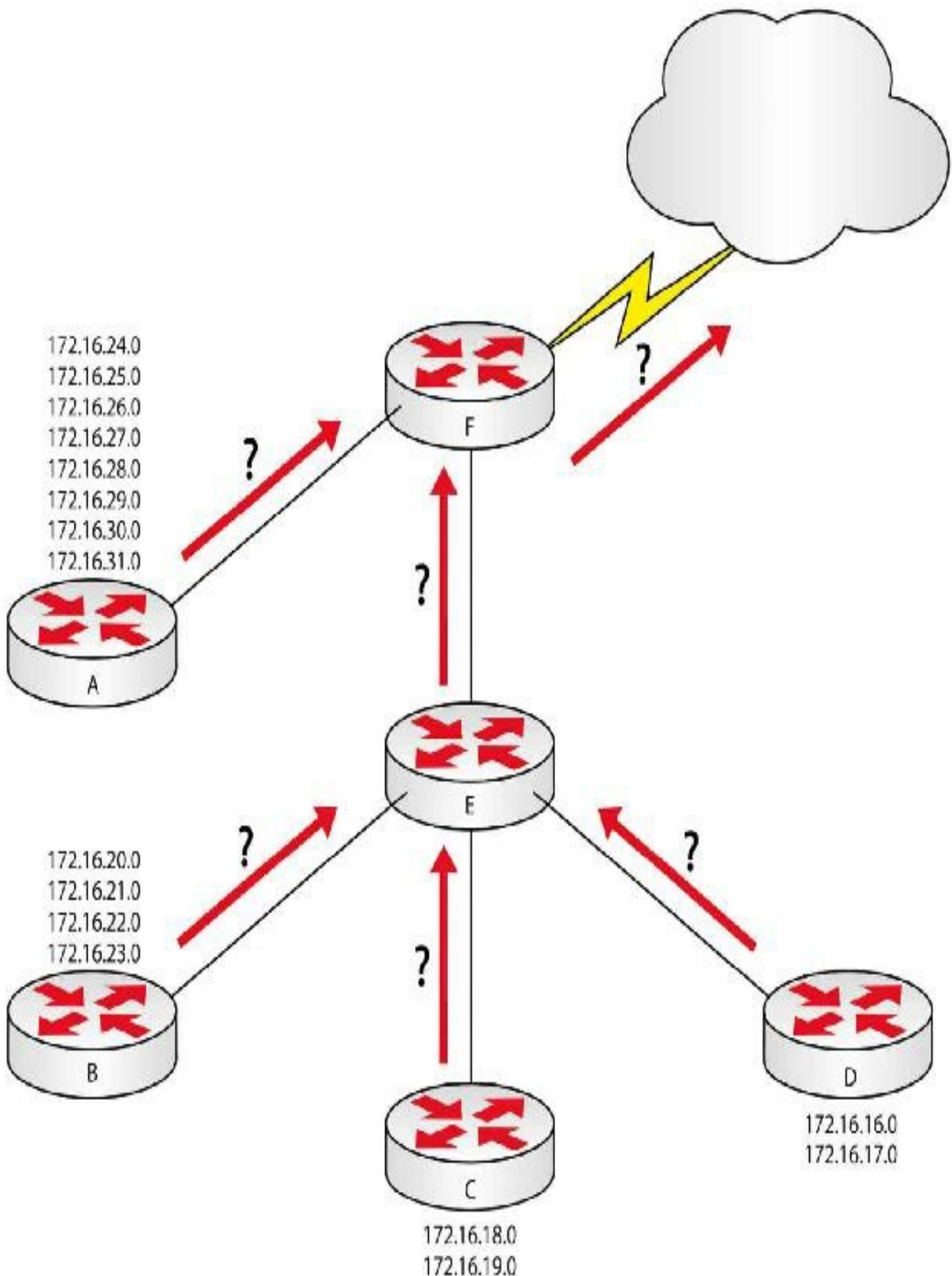


FIG 3.10 – Summarization challenge

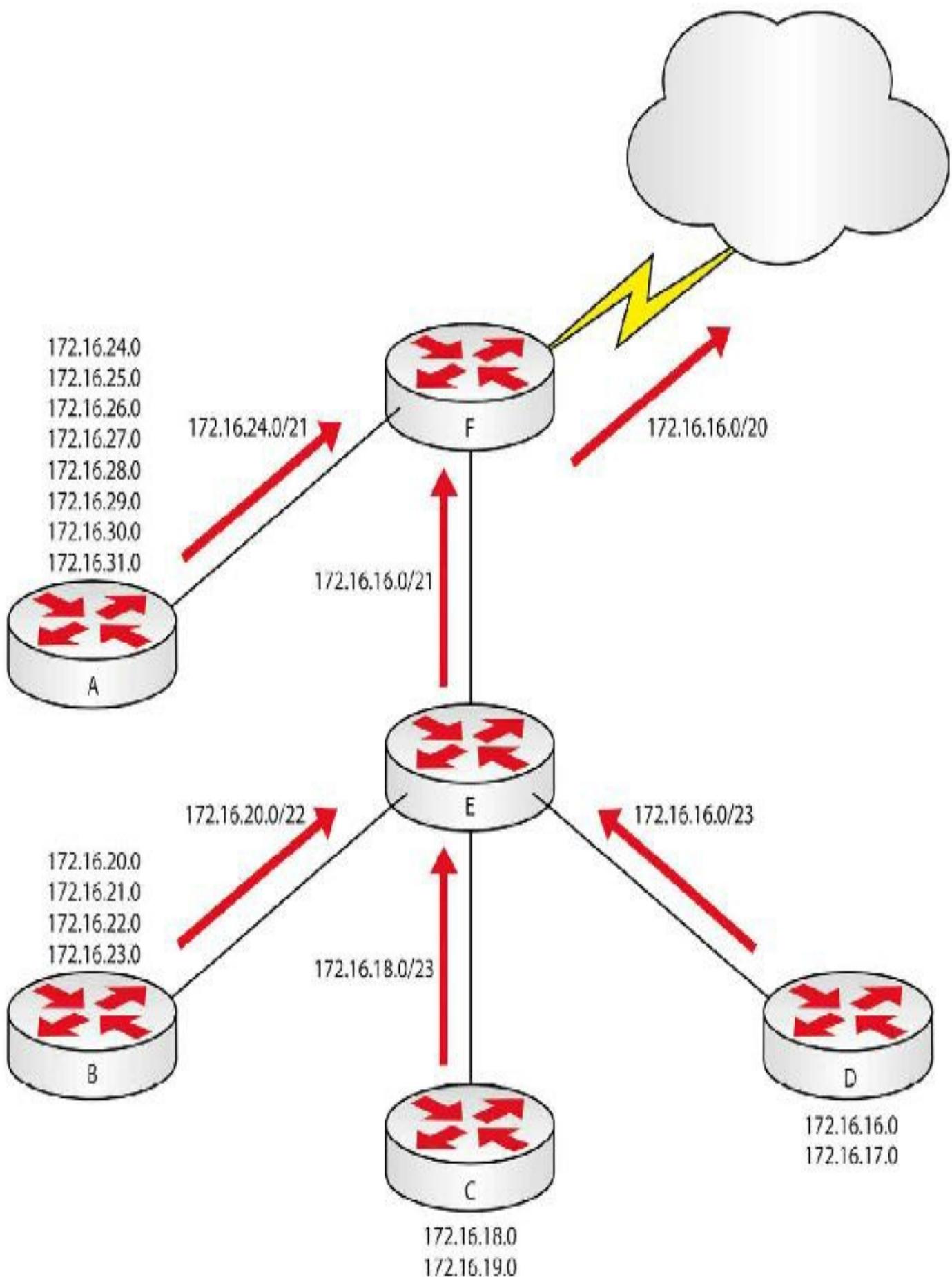


FIG 3.11 – Summarization solution

Mini-lab – Troubleshooting IP Addressing

Cisco expects you to be able to resolve IP addressing issues in the CCNA exam. This could be presented in the form of theory questions, diagrams with questions, or having to log on to a router or switch and use various show commands to diagnose and resolve the issue.

Most IP addressing issues can be resolved using the shortcut subnetting steps outlined in this chapter. This method will quickly tell you whether an IP address is in the wrong subnet, is a subnet address, or is a broadcast address for the subnet.

If you have to troubleshoot IP addressing as part of a hands-on lab, this may be presented as an isolated issue or as part of a larger troubleshooting issue, such as an issue involving routing or access lists. We will cover troubleshooting these topics in relevant chapters later on.

Check any diagrams or documentation you have access to in the exam on the screen. There will often be several windows you can open and close in addition to the question window, as well as console windows for routers and switches. It can all become overwhelming if you let it.

Check the diagram, which should show you the correct IP address and subnet mask for the interface or network. Just play along with me and configure 172.16.1.1/16 on R1 and 172.16.1.2/28 on R2, referencing Figure 3.12 below:

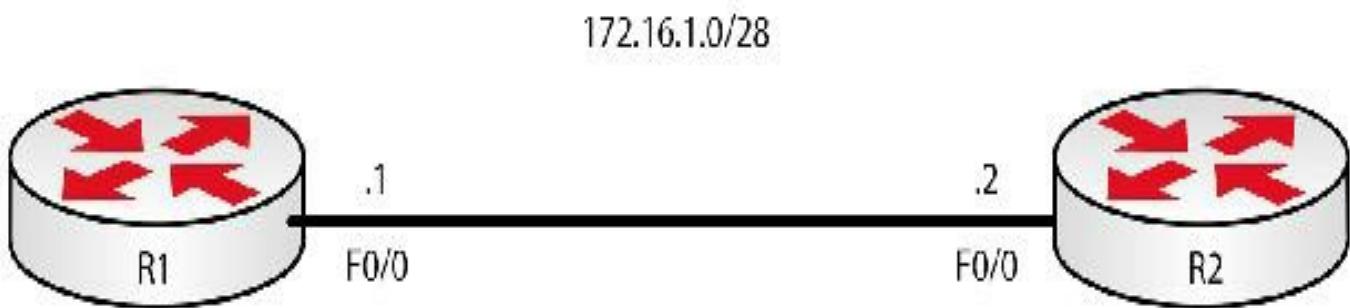


FIG 3.12 – Mini-lab: Troubleshooting IP Addressing

If you apply the usual subnetting methods to the network above, you should expect to see IP address 172.16.1.1 255.255.255.240 on R1 and 172.16.1.2 255.255.255.240 on R2.

You could be lazy and just ping from R1 and think that all is well. You already know that the first ping will fail due to the ARP process.

R1#ping 172.16.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 20/28/40 ms

Even if you issued a show ip interface brief command, all would appear normal.

R1#show ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	172.16.1.1	YES	manual	up	up
Fa0/1	unassigned	YES	unset	administratively down	down

The interface is up/up so it can't be the IP address, right? Well, you haven't seen the subnet mask yet so you need to check this before you know.

R1#show int f0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 172.16.1.1/16

Clearly this isn't correct because you know that the mask should be /28. The default subnet mask has been applied and it has been correctly applied to R2.

R2#show int f0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 172.16.1.2/28

R1#conf t

R1#int f0/0

R1(config-if)#ip address 172.16.1.1 255.255.255.240

Now check all the show commands again.

The incorrect mask won't be an issue for pinging the directly connected interface, but if you added a classless routing protocol then it could very well cause problems. You already know how to configure an IP address on an interface so that would resolve this particular issue.

The IP addressing issue may be part of a larger issue, including missing clock rates, shutdown interfaces, routing configuration, or more; however, I wanted to stick to troubleshooting IP addressing in this section. We will cover more advanced troubleshooting later on.

[END OF MINI-LAB]

Chapter 3 Labs

Lab 1: IP Addressing

The physical topology is shown in Figure 3.13 below:

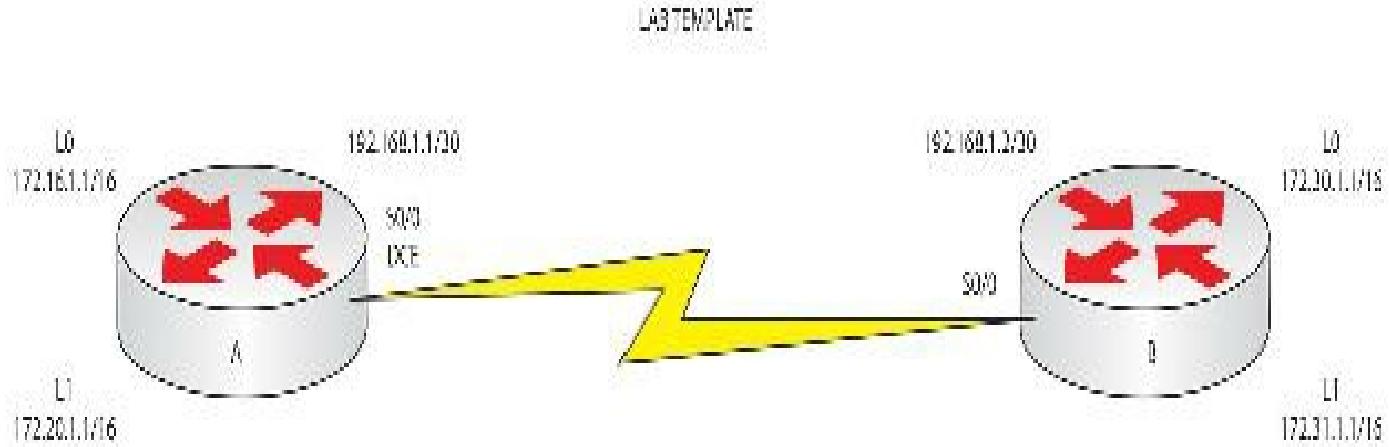


FIG 3.13 – IP Addressing Lab

Lab Exercise

Your task is to configure the IP addressing as specified in Figure 3.13. Some of the commands you will use will be discussed later, but you will enter them for practice.

Purpose

Every router will need to be configured with IP addresses and will have to connect to another router, either on the LAN or WAN. Being able to do this without looking at the walk-through is your eventual goal. This lab is the springboard for every other lab in the book.

Lab Objectives

1. Configure the enable password to be cisco.
2. Set the console password to be letmein.
3. Configure Telnet access to use the local username banbury with the password ccna.
4. Set the IP subnets for the Loopback 0 and Loopback 1 interfaces.
5. Set the IP subnet for the Serial 0 interface.

Lab Walk-through

1. To configure the enable password command on a router, you need to do the following:

Router>enable

Router#configure terminal

```
Router(config)#hostname RouterA
```

RouterA(config)#enable password cisco **i Configures the enable password or use an enable secret password (but not both)**

RouterA(config)enable secret cisco **i Configures a more secure enable password**

2. To configure the console password, you need to configure the console port. To do this, enter the following commands (in global configuration mode):

```
RouterA(config)#line console 0
```

```
RouterA(config-line)#password letmein
```

RouterA(config-line)#login **i Whoever connects to the console port will have to enter the password letmein**

```
RouterA(config-line)#exit
```

```
RouterA(config)#
```

3. To set Telnet access, you need to configure the VTY lines to allow Telnet access (make sure you check how many you have by using line vty 0 ?; to do this, type (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration
```

RouterA(config-line)#login local **i This will use local usernames and passwords for Telnet access**

```
RouterA(config-line)#exit i Exits the VTY config mode
```

RouterA(config)#username banbury password ccna **i Creates username and password for Telnet access (login local)**

4. You now need to start setting the IP addresses on the Loopback interfaces. As you will see from Figure 3.13, the subnet mask for the networks is represented by a /16 notation. This is actually a Class B mask (255.255.0.0)—the default for the address ranges you are using for the Loopback interfaces. To apply an IP address to an interface, do the following:

```
RouterA(config)#interface Loopback0
```

```
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
```

```
RouterA(config-if)#interface Loopback1
```

```
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
```

5. Next, you need to configure the IP address for the interface Serial 0 (or you may have Serial 0/0). As this is a WAN link, it is good practice to conserve as many addresses as possible. So, as you will see in the network diagram in Figure 3.13, the Serial interface is using a /30 mask; this is a 255.255.255.252 mask when written in dotted decimal format. By using this subnet mask, you reduce

the available addresses from 255 per subnet to two available host addresses per subnet.

If the router does not accept this mask you may be running an old IOS version, so you may need to enter the ip subnet-zero command to the router in configuration mode. However, as you will remember from the IP addressing/subnetting presentation, you cannot use an address that is all 1s or all 0s, so the actual usable addresses are two. This mask is perfect for a WAN link, such as the one you are using in this lab, because no addresses are being wasted:

```
RouterA(config)#interface Serial0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000 i If this has the DCE cable attached
RouterA(config-if)#no shutdown i This will initialize the interface from the default state of down. This is only necessary on physical, not logical (Loopback), interfaces.
```

The following message should appear on the console session for Router A:

```
00:15:51: %LINK-3-UPDOWN: Interface Serial0, changed state to down
```

The interface will not come up until Router B's Serial interface is up.

6. You will now follow all of the steps above for Router B.

```
Router>enable
RouterB#config t
Router(config)#hostname RouterB
RouterB(config)#enable password cisco
```

Or use an enable secret password command (but not both):

```
RouterB(config)#enable secret cisco
```

Do not use the same password for both; otherwise, you will receive a warning message. In fact, it is best practice to just use the enable secret command:

The enable secret you have chosen is the same as your enable password.
This is not recommended. Re-enter the enable secret.

Configure the console password:

```
RouterB(config)#line console 0
RouterB(config-line)#password letmein
RouterB(config-line)#login
```

```
RouterB(config-line)#exit  
RouterB(config)#
```

Configure the Telnet password:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit  
RouterB(config)#username banbury password ccna
```

Configure the Loopback addresses:

```
RouterB(config)#interface Loopback0  
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0  
RouterB(config-if)#interface Loopback1  
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
```

Configure the Serial addresses:

```
RouterB(config-if)#interface Serial0  
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252  
RouterB(config-if)#no shutdown
```

7. You should see the interface come up on the console session:

```
00:31:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,  
changed state to up
```

If Router B has the DCE interface attached, then set the clock rate on that. You can issue the show controllers serial 0 command to see whether the cable is DTE or DCE.

You should now be able to ping Router A from Router B and vice versa.

```
RouterB#ping 192.168.1.1  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms  
RouterB#
```

If pings are not working, check the Serial interfaces to make sure they are both up/up with the show ip interface brief command. You won't be able to ping the Loopback addresses on the opposite routers because you haven't configured any

routing. We will cover this later.

```
RouterA#show ip interface brief
Interface IP-Address OK? Method Status Protocol
Ethernet0 unassigned YES unset administratively down down
Loopback0 172.16.1.1 YES manual up up
Loopback1 172.20.1.1 YES manual up up
Serial0 192.168.1.1 YES manual up up
Serial1 unassigned YES unset administratively down down
RouterA#
```

Make sure that you have put a clock rate on the correct interface (i.e., the DCE side). You can check this with the show controllers serial 0 command (if you are plugged into Serial 0, that is). If you are using GNS3 you won't need to worry about clock rates.

```
RouterA#show controllers Serial0
HD unit 0, idb = 0x162890, driver structure at 0x168C18
buffer size 1524 HD unit 0, V.35 DCE cable, clockrate 64000
cpb = 0x2, eda = 0x2940, cda = 0x2800
RX ring with 16 entries at 0x4022800
00 bd_ptr=0x2800 pak=0x16A10C ds=0x4026108 status=80 pak_size=0
```

Make sure that you have put the correct IP address on both sides and that the subnet masks are the same.

If all else fails and you cannot find what is wrong, look at the show run for both routers.

You can copy and paste the configurations into your router at the Router(config)# prompt. Bear in mind that your router may have different interfaces. If, for example, your Serial cable is plugged into Serial 1/0, you will have to make the necessary changes to my configurations.

The exclamation marks are there to make the reading easier when you look at the show run.

Show Runs

```
RouterA#show run
Building configuration...
!
Current configuration : 709 bytes
```

```
version 15.1
!
hostname RouterA
!
enable secret 5 $1$rujI$BJ8GgiK8U9p5cdfXyApPr/
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Loopback1
ip address 172.20.1.1 255.255.0.0
!
interface Serial0
ip address 192.168.1.1 255.255.255.252
clockrate 64000
!
ip classless
no ip http server
!
line con 0
password letmein
login
line aux 0
line vty 0 4
login local
!
end
```

RouterA#

Router B:

```
RouterB#show run
Building configuration...
```

```
Current configuration : 697 bytes
!
version 15.1
!
hostname RouterB
!
enable secret 5 $1$ydeA$MyfRKevoCkjm7w/0ornnB1
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Serial0
ip address 192.168.1.2 255.255.255.252
!
ip classless
no ip http server
!
line con 0
password letmein
login
line aux 0
line vty 0 4
login local
!
end
```

RouterB#

Answers to binary/hex/decimal conversions:

- Convert 1001 (binary) to hex and decimal – 9 / 9
- Convert 11011 (binary) to hex and decimal – 1B / 27
- Convert 10001 (binary) to hex and decimal – 11 / 17

- Convert 29 (decimal) to binary and hex – 11101 / 1D
- Convert 33 (decimal) to binary and hex – 100001 / 21
- Convert 102 (decimal) to binary and hex – 1100110 / 66
- Convert C7 (hex) to binary and decimal – 11000111 / 199
- Convert FE (hex) to binary and decimal – 11111110 / 254
- Convert B5 (hex) to binary and decimal – 10100101 / 165

Chapter 4 — IPv6 Addressing

What You Will Learn in This Chapter

IPv6 Addressing

Migrating from IPv4 to IPv6

IPv6 Functionality Protocols

IPv6 is receiving a lot of attention at the moment in the IT community, even though development on it started back in 1991.

In my previous CCNA manuals, I said that IPv6 was coming so get prepared. Up until the latest syllabus change, you would only need to know what an IPv6 address looked like and how to legally compress an IPv6 address. Well, IPv6 is now in common usage and you need to understand a whole lot more, including having a good grasp of IPv6 address formats, how various IPv6 mechanisms work, and some of the IPv6 routing protocols.

I know many IT engineers have avoided learning IPv6 because they think that it's a complicated subject, but rest assured that once you've read about it and start some configurations, you will find that it actually makes sense and is just as understandable as IPv4, if not more so.

Syllabus Topics Covered

3.0 IP Addressing (IPv4/IPv6)

3.2 Identify the appropriate IPv6 addressing scheme to satisfy addressing requirements in a LAN/WAN environment

3.4 Describe the technological requirements for running IPv6 in conjunction with IPv4

 3.4.a Dual stack

3.5 Describe IPv6 addresses

 3.5.a Global unicast

 3.5.b Multicast

 3.5.c Link-local

 3.5.d Unique local

 3.5.e EUI 64

 3.5.f Autoconfiguration

Why Do We Need IPv6?

Nobody could have predicted the exponential explosion in the growth of the Internet when it was first created. Who could possibly have imagined even just a few years ago that nearly every household in the world would have a PC in it? Or that every person will require an IP address for their work PCs, home PCs, mobile phones, mobile IP devices, and even remote IP management of such things as home intruder alarms, ovens, garage doors, and TVs? Experts now agree that estimates of each individual requiring over 250 IP addresses is well within the bounds of possibility.

IPv4 was developed when only large companies required IP addresses. These addresses were cut into blocks from Class A to Class C, with Class D being reserved for multicasting and Class E for experimental use. The original incarnation of IPv4 created huge waste; for example, for Class A addresses, potentially thousands of addresses were wasted, and for Class C addresses, smaller companies were forced to buy several blocks of network addresses for use in their networks. Often, the addresses were non-contiguous, which added to route summarization problems.

Work on IPv6 began as soon as the scale of the IPv4 problem was fully realized (1991). At the time of writing this section of the manual, various Internet Service Providers were issuing press releases to announce that they had finally run out of IPv4 addresses. This means that there is a massive opportunity for engineers to help companies make the transition from IPv4 to IPv6.

The development of IPv6 has addressed some of the shortfalls of IPv4, which include:

- **LAN latency** – When IPv4 is used on Ethernet segments, there has to be a layer 3-to-layer 2 mapping; in addition, IPv4 uses an ARP broadcast to perform address resolution, which involves an ARP broadcast packet being sent to and received by all stations on an Ethernet segment that is processed as an interrupt on the Ethernet port.
- **Autoconfiguration** – IPv4 lacks a simple autoconfiguration addressing system.
- **Security** – IPv4 has no built-in security parameters, as this function is left to PC and router firewalls.
- **Mobility** – IPv4 has no facility to allocate IP addresses to mobile devices.
- **Routing** – IPv4 addressing can lead to huge routing tables and vast amounts of routing update packets traversing the Internet. Moreover, changes made to Domain Name System (DNS) entries can take up to 48 hours to propagate, leading to network downtime.

IPv6 has some similarities to IPv4, along with many new features, as illustrated in Figure 4.1 below:

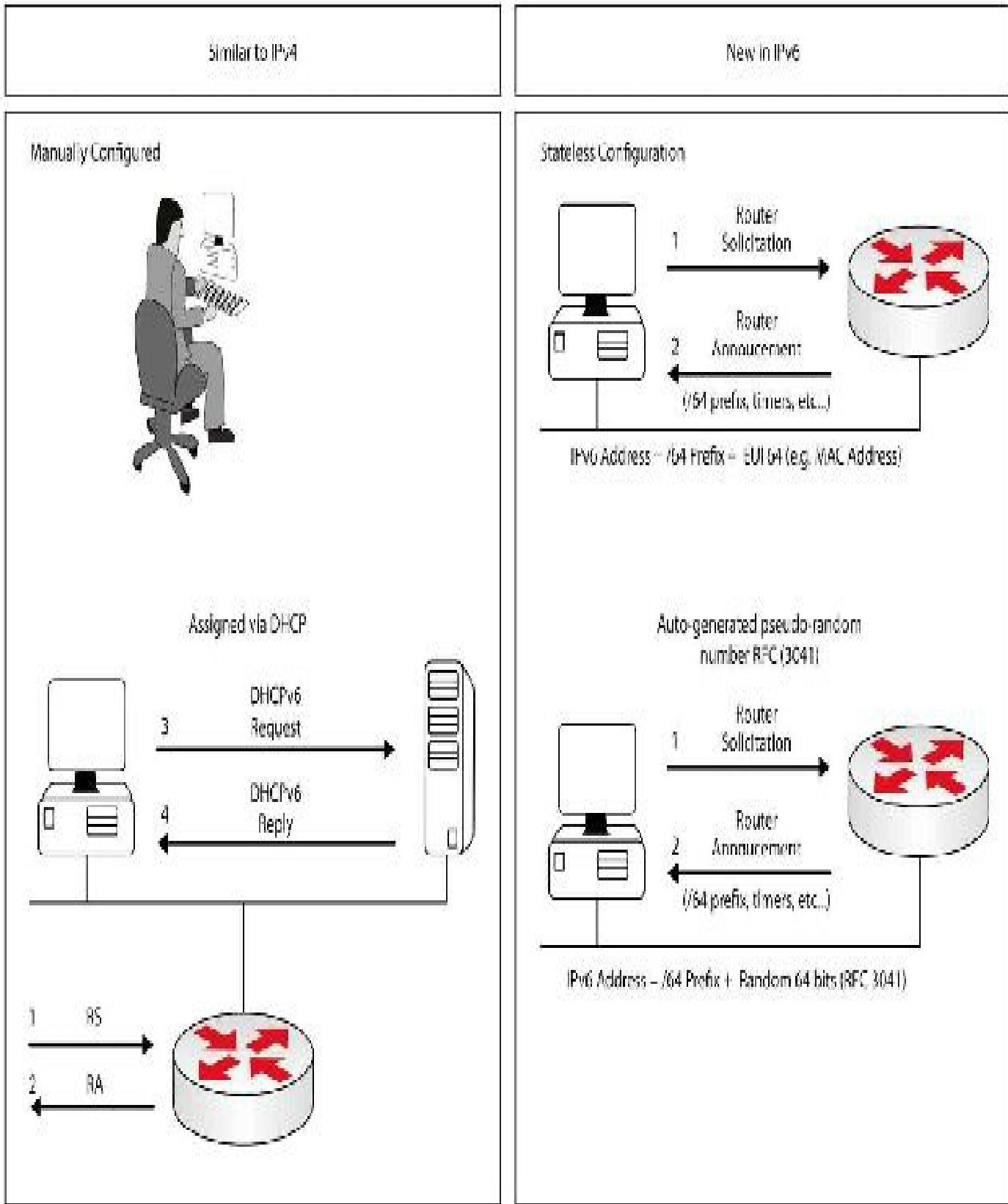


FIG 4.1 – IPv4 compared to IPv6

Anatomy of an IP Packet

The design of a new IP addressing scheme has given network architects a clean slate

and the ability to incorporate a wish list into the design of the IPv6 packet. The requirements were a pure design for the header with as few fields as possible, as opposed to the IPv4 header shown in Figure 4.2 below:

Version	IHL	Type of Service	Total Length	
Identification		Flag		Fragment Offset
Time to Live	Protocol	Header Checksum		
Source Address				
Destination Address				
Options			Padding	

FIG 4.2 – IPv4 header: 20 octets

IPv4 allows a unique network number to be allocated to every device on the Internet, but IPv6 takes this one step further, as can be seen in Figure 4.3 below:

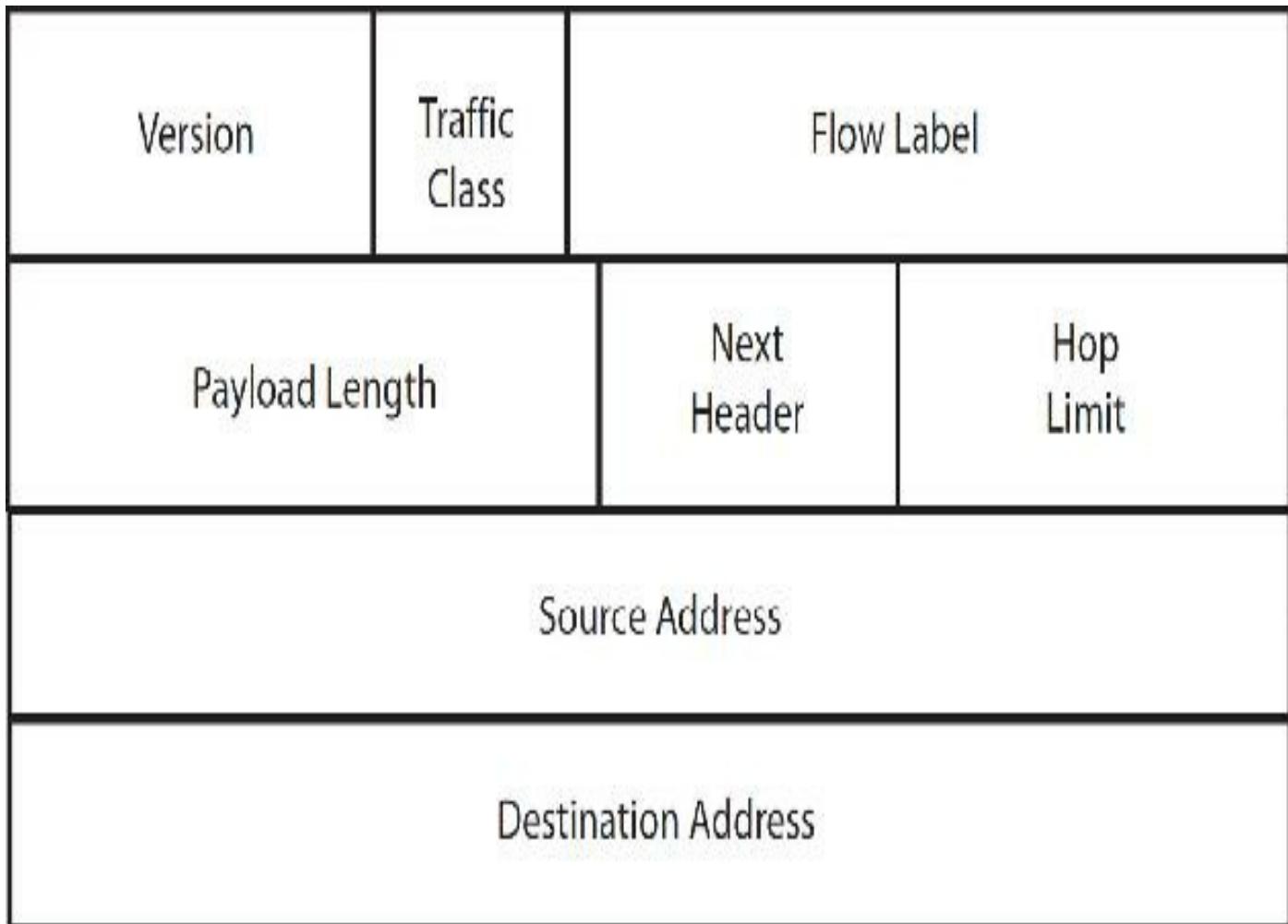


FIG 4.3 – IPv6 header: 40 octets

The Version field is set to 6, of course, or in binary 0110. While going into the field properties is beyond the CCNA syllabus, it is worth noting that the Flow Label field is unique to IPv6. This field allows routers to identify a flow by looking at just the packet header as opposed to having to dig deeper into TCP/UDP headers to find this information. A flow is a packet that matches the same source/destination address and service/port, thus speeding up the passing of the packet.

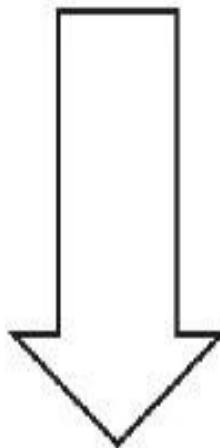
There is no Checksum field in the IPv6 packet due to the reliability of modern devices and the fact that upper-layer protocols usually carry out their own error checking.

IPv6 assigns a 128-bit numerical address to each interface in a network. This, of course, will lead to extreme difficulty for network administrators tracking which interface is using which address. For this reason, IPv6 works hand in hand with DNS. There is no requirement for a subnet mask in IPv6; instead, a prefix is used.

An IPv6 address has two parts. The first part is the data link layer address, which identifies the host destination within the subnet. This is the layer 2 address. The second address is layer 3 and it identifies the destination network the packet must reach. IPv6

uses the Neighbor Discovery Protocol (NDP), not ARP, for layer 2-to-layer 3 address mapping.

IPv4 - 32 Bits for Addressing



IPv6 - 128 Bits for Addressing

FIG 4.4 – IPv6 address size

There are 2^{128} addresses available with IPv6, which is exactly 340,282,366,920,938,463,374,607,431,768,211,456 addresses. That is over 5×10^{28} addresses for every person in the world! These addresses are available without the need for private address translation or any other techniques required for address conservation (such as NAT).

RFC 1884 recommends that IPv6 syntax for the 128 bits is represented in eight groups of hexadecimal digits (so eight groups of 16 bits). Each group is divided by a colon so the syntax is referred to as coloned hex, as shown in the example below:

EEDE:AC89:4323:5445:FE32:BB78:7856:2022

IPv6 Address Representation

IPv6 addresses can be represented in three ways:

- The preferred or complete address representation or form
- Compressed representation

- IPv6 addresses with an embedded IPv4 address

Although the complete address representation (preferred form) is the most common method for expressing 128-bit IPv6 addresses, you should be familiar with the other methods, which are described in the following sections.

The Preferred Form

The preferred form expresses the 128 bits as 32 hexadecimal digits grouped into eight 16-bit fields (each represented by four hexadecimal digits). This is expressed by separating each group of four hex digits with a colon, for example, 3FFF:1234:ABCD:5A78:020C:CDFF:FEA7:F3A0. This is the longest format for expressing an IPv6 address.

Each 16-bit field can have a value between 0x0000 and 0xFFFF. As will be described later, some bits in the first field have been reserved and not all the possible values of the first field are used. Hexadecimal characters are not case sensitive, so 2001:ABCD:0000 and 2001:abcd:0000 are the same thing. The preferred form of IPv6 address representation is illustrated in Figure 4.5 below:

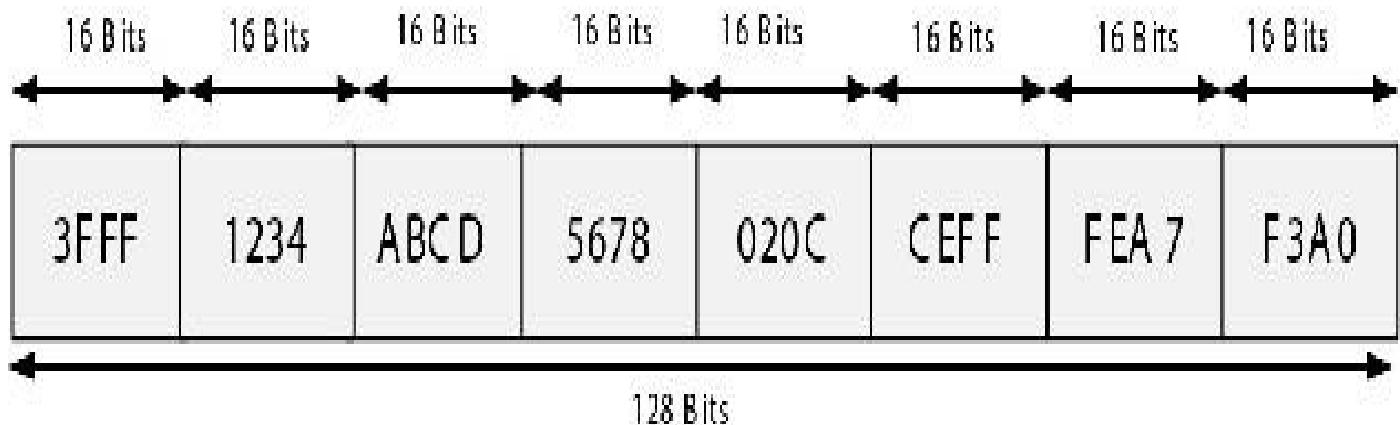


FIG 4.5 – The preferred form for IPv6 address representation

The following IPv6 addresses are examples of valid IPv6 addresses in the preferred form:

0000:0000:0000:0000:0000:0000:0003

2001:0000:0000:ABCD:0000:5678:af23:bcd5

3FFF:0000:0000:1010:12AB:9000:0B00:DE09

fec0:2004:ab10:00cd:1234: 0000:0000:6AE9

0000:0000:0000:0000:0000:0000:0000

Compressed Representation

Compressed representation allows IPv6 addresses to be shortened using two methods. The first uses a double colon (::) to represent consecutive zero values or leading zeros in an IPv6 address. The caveat here is that the double colon can be used only once. Each node or device can then expand the value by calculating the number of bits missing and replacing them with zeros. Table 4-1 below illustrates this representation:

Table 4-1: Representing complete IPv6 addresses in the preferred compressed form

Complete IPv6 Address Representation	Compressed IPv6 Address Representation
0000:0000:0000:0000:0000:0000:000D	::000D
2001:0000:0000:12A0:0000:5678:af23:bcdd	2001::12A0:0000:5678:af23:bcdd
3FFF:0000:0000:1010:1A2B:5000:0B00:DE0D	3FFF::1010:1A2B:5000:0B00:DE0D
FEC0:2004:AB10:00CD:1234:0000:0000:6789	FEC0:2004:AB10:00CD:1234::6789
0000:0000:0000:0000:0000:0000:0000	::

Note that in the example with 2001:0000:0000:12A0:0000:5678:af23:bcdd, the double colon was used only once, even though there are two sets of consecutive strings of zeros. This is because it would be impossible for a device to convert the address further without losing its unique value, so a value of 2001::12A0::5678:af23:bcdd would be wrong. However, a value of 2001:0000:0000:12A0::5678:af23:bcdd would still represent the same IPv6 address.

The second method allows the leading zeros in each IPv6 field to be omitted. The only caveat here is that when the field is all zeros, then you need to represent the field with one zero so as not to lose its value. The second method is illustrated in Table 4-2 below:

Table 4-2: Representing complete IPv6 addresses in the alternative compressed form

Complete IPv6 Address Representation	Compressed IPv6 Address Representation
0000:0123:0abc:0000:04b0:0789:f000:0001	0:123:abc:0:4b0:789:f000:1
2001:0000:0000:5678:0000:1234:af23:bcdd	2001:0:0:5678:0:1234:af23:bcdd
3FFF:0000:0000:1010:1A2B:6000:0B00:DE0D	3FFF:0:0:1010:1A2B:6000:B00:DE0D
fec0:2004:ab10:00cd:1234:0000:0000:6789	fec0:2004:ab10:cd:1234:0:0:6789
0000:0000:0000:0000:0000:0000:0000	0:0:0:0:0:0:0

The two methods of compressing IPv6 addresses can be used together if an IPv6 address has both consecutive strings of zeros and leading zeros in other fields. This is illustrated in Table 4-3 below:

Table 4-3: Representing complete IPv6 addresses using both compressed form methods

Complete IPv6 Address Representation	Compressed IPv6 Address Representation
0000:0000:0000:0000:1a2b:000d:f123:0456	::1a2b:d:f123:456
FEC0:0004:AB10:00CD:1234:0000:0000:6789	FEC0:4:AB10:CD:1234::6789
3FFF:0c00:0000:1010:1A2B:0000:0000:DE0F	3FFF:c00:0:1010:1A2B::DE0F
2001:0000:0000:1234:0000:5678:af23:00d5	2001::1234:0:5678:af23:d5

You can expect to be quizzed on IPv6 address compression in the CCNA exam.

IPv6 Addresses with an Embedded IPv4 Address

The third method of representing an IPv6 address is to embed an IPv4 address within the IPv6 address. Although this method is valid, it is important to note that it is on the path to deprecation, since it is only applicable to migrations from IPv4 to IPv6.

The Different IPv6 Address Types

IPv4 supports four different classes of addresses, which are anycast, broadcast, multicast, and unicast. An anycast address is an IP address that is assigned to multiple devices that provide the same function in such a way that the closest device (based on the routing protocol metric) responds when that IP address is queried for the same function. This is an easy way to provide load balancing and redundancy in networks. Common uses of anycast include DNS server load balancing in networks.

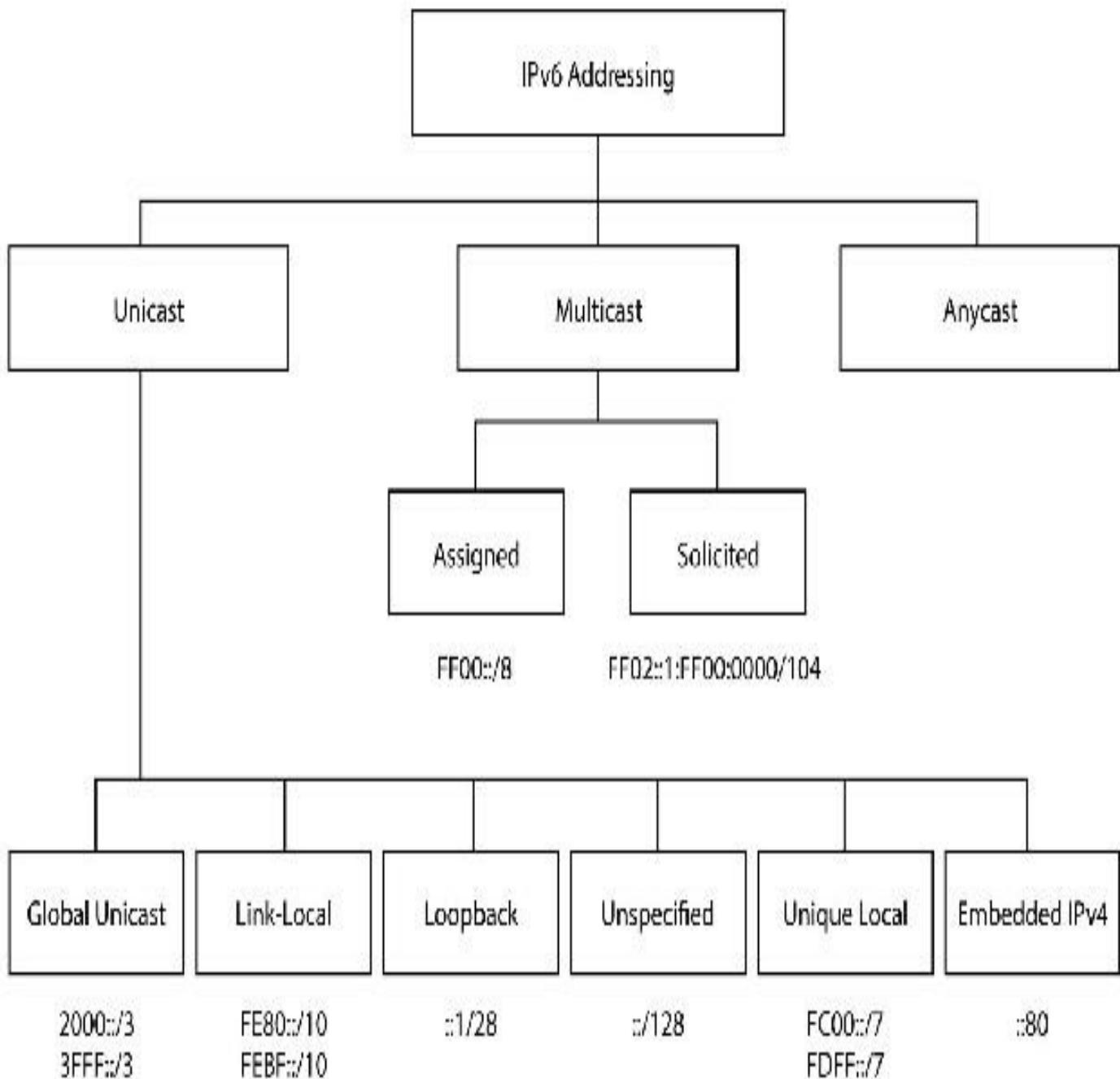
NOTE: IPv6 allows multiple addresses to be allocated per interface. There is no concept of primary and secondary IP addressing in IPv6.

At this point you should be familiar with IPv4 broadcast, multicast, and unicast addresses. In IPv6, broadcast addresses are no longer supported. The following types of addresses are supported in IPv6:

- Link-local addresses
- Site-local addresses
- Aggregate global unicast addresses

- Multicast addresses
- Anycast addresses
- Loopback addresses
- Unspecified addresses

IPv6 Address Types:



NOTE: There are no broadcast addresses in IPv6

FIG 4.6 – IPv6 address types

Link-local Addresses

IPv6 link-local addresses as defined in section 2.5.6 of RFC 4291 are valid only on the local link (the shared segment between devices). They are automatically assigned to each IPv6-enabled interface. Link-local addresses are assigned from the FE80::/10 prefix (fe80:: through to febf::). This means that the first 10 bits must be 1111 1110 10. Also, the next 54 bits must be set to 0. The remaining 64 bits are the Extended Unique Identifier 64 (EUI-64) address. The EUI-64 address will be covered in detail later in this chapter.

The format for a link-local address is illustrated in Figure 4.7 below:

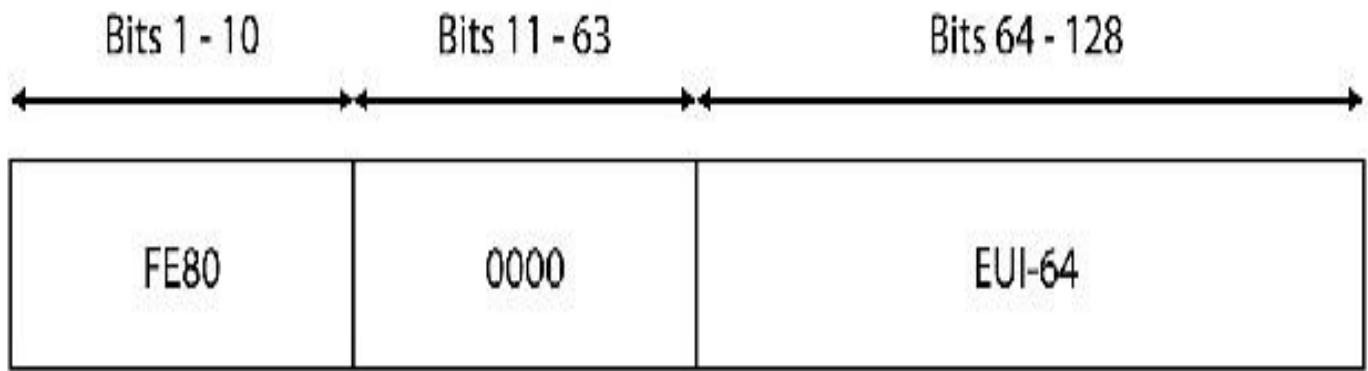


FIG 4.7 – IPv6 link-local addressing

It's important to note that every IPv6 interface, be it Ethernet, PPP, Frame Relay, or other interface, will be assigned a link-local address for use on that segment as soon as IPv6 is enabled on the interface. An example of a link-local address is FE80::211:77FF:FE80:72B7. These addresses can be automatically created using stateless address autoconfiguration (SLAAC), which will be discussed later. If you wanted to manually configure a link-local address, you would use the commands below:

```
R1#conf t  
R1(config)#ipv6 unicast-routing  
R1(config)#int f0/0  
R1(config-if)#ipv6 address fe80::211:77ff:fe80:72b7 link-local  
R1(config-if)#end
```

```
R1#show ipv6 interface f0/0  
FastEthernet0/0 is administratively down, line protocol is down  
  IPv6 is enabled, link-local address is FE80::211:77FF:FE80:72B7  
[output truncated]
```

The link-local address is created when you enable IPv6 on an interface, such as by adding a global unicast address or with the `ipv6 enable` command. IPv6 must be enabled globally on the router before this command will take effect.

```
R1(config)#ipv6 unicast-routing
R1(config)#int f0/0
R1(config-if)#ipv6 enable
R1(config-if)#end
R1#
*Mar 1 00:02:17.227: %SYS-5-CONFIG_I: Configured from console by console
R1#show ipv6 interface f0/0
FastEthernet0/0 is administratively down, line protocol is down
  IPv6 is enabled, link-local address is FE80::C000:6FF:FEFF:0 [TEN]
  No Virtual link-local address(es):
  No global unicast address is configured

[output truncated]
```

Link-local traffic is not forwarded from the local link (it's non-routable); instead, it's used for routing protocol neighbor communication and other local operations.

You should note that IPv4 also supports link-local IP addressing, although it is less commonly used than in IPv6. The link-local range for IPv4 addresses is 169.254.0.0/16. The most common use of IPv4 is when a device automatically assigns itself an IP address from this range after an unsuccessful attempt to obtain an IP address from a DHCP server.

Site-local Addresses and Unique Local Addresses

Site-local addresses are unicast IPv6 addresses that are used only within a site. This serves as the equivalent of RFC 1918 (private) IPv4 addresses, meaning they are not guaranteed to be unique globally and are therefore not routed on the IPv6 Internet.

Although it is possible to perform Network Address Translation (NAT) for IPv6, it is not recommended; hence, the reason for the much larger IPv6 addresses. Site-local addresses are assigned from the FEC0::/10 prefix (the first 10 bits are 1111 1110 11). The next 54 bits are a subnet ID, while the remaining 64 bits are an interface identifier in the EUI-64 format. The format for the Site-local address is illustrated in Figure 4.8 below:

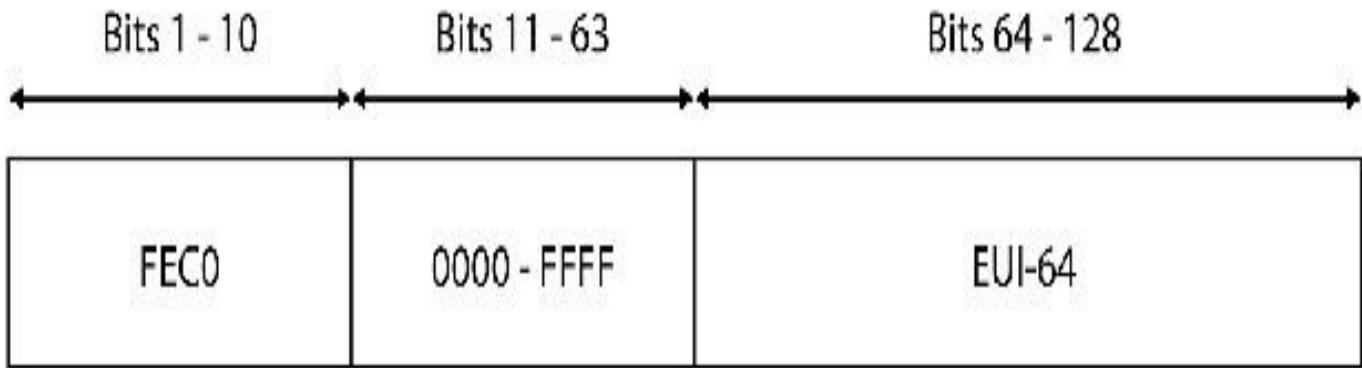


FIG 4.8 – IPv6 site-local addressing

Site-local addresses are described in this section because they are still supported in Cisco IOS Software. However, it is important to note that these addresses have since been deprecated by RFC 4193, which describes unique local addresses (ULAs); these addresses serve the same function as site-local addresses and are also not routable on the IPv6 global Internet.

Unique local addresses are assigned from the FC00::/7 prefix, which is further subdivided into two /8 address groups referred to as the assigned and random groups. These two groups are the FC00::/8 and the FD00::/8 IPv6 address blocks. The FC00::/8 block is managed by an allocation authority for /48s in use, while the FD00::/8 block is formed by appending a randomly-generated 40-bit string to derive a valid /48 block.

Aggregate Global Unicast Addresses

Aggregate global unicast addresses are the IPv6 addresses used for generic IPv6 traffic and the IPv6 Internet. These are equivalent to the public IPv4 addresses (i.e., host-to-host communication). Each IPv6 address is made up of three parts: a 48-bit prefix received from a provider, a 16-bit site prefix, and the host portion, which is 64 bits. Figure 4.9 below shows you how the address is comprised:

3 bit current global reservation for IPv6 Global Unicast addresses

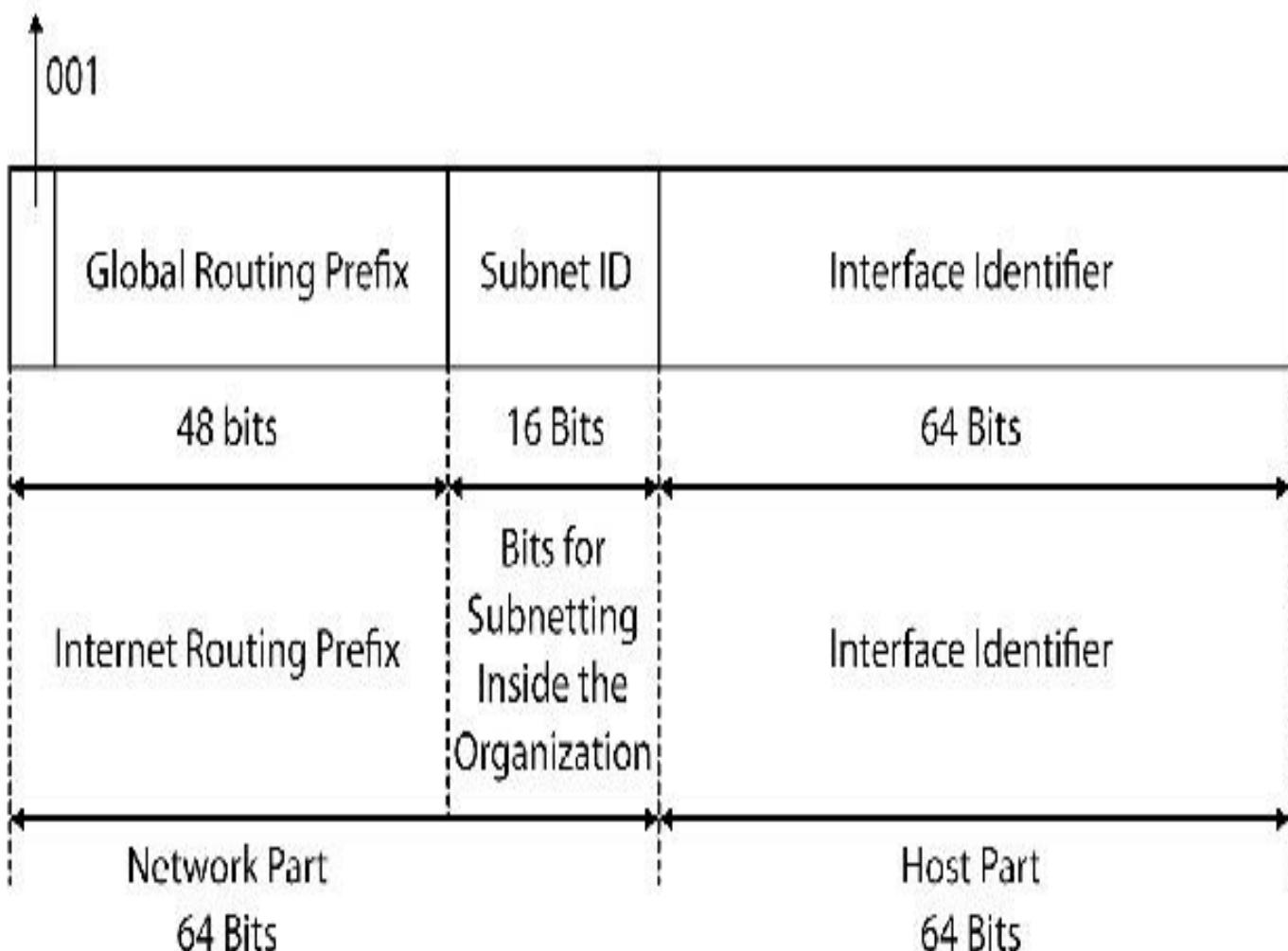


FIG 4.9 – IPv6 global unicast address

Providers have a larger /32 prefix from which they assign /48 prefixes to organizations. These /32s are unique to the provider and other providers have to assign them to their own allocated /32 prefixes.

Within an organization, the /48 prefix can be further subdivided into 64-bit site prefixes. This allows for up to 65,536 different 64-bit subnets to be used. The remaining 64 bits are used for the host portion of the network.

A /64 prefix will provide any company with 2^{16} (or 65,536) unique networks, with each network having 2^{32} (or 4,294,967,296) addresses.

Cisco routers can derive the interface ID using a variety of methods, which will be covered later.

The aggregate global unicast addresses are assigned by the Internet Assigned Numbers Authority (IANA) from the IPv6 prefix 2000::/3 (the first three bits are 001). The range

of the IPv6 aggregate global unicast addresses is shown in Table 4-4 below:

Table 4-4: IPv6 aggregate global unicast addresses

Description	Address
First Address in Range	2000:0000:0000:0000:0000:0000:0000
Last Address in Range	3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
Binary Notation	The three high-order bits are set to 001

You can find a list of the globally allocated unicast prefixes via the URL below:

<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml>

Multicast Addresses

You already know that multicast traffic is used for one-to-many or even many-to-many communications. Because IPv6 has no broadcast capability, multicast takes over this function in IPv6. Multicast will always be a destination address, never a source address.

IPv6 multicast addresses are assigned from the FF00::/8 IPv6 prefix (1111 1111). IP multicast is used extensively in IPv6 to perform multiple operations. IPv6 multicast has replaced ARP. In addition, it is used in IPv6 for prefix advertisements and renumbering, as well as for Duplicate Address Detection (DAD), which, as the name suggests, checks that the address is used only once in the network.

IPv6 does not use the TTL value to restrict multicast packets to the local network segment. Instead, the scope of an IPv6 multicast packet is defined within the multicast address itself via the use of the Scope field. This allows all nodes on an IPv6 segment to know about all other neighbors on that same segment. The format for multicast addresses used in IPv6 networks is illustrated in Figure 4.10 below:

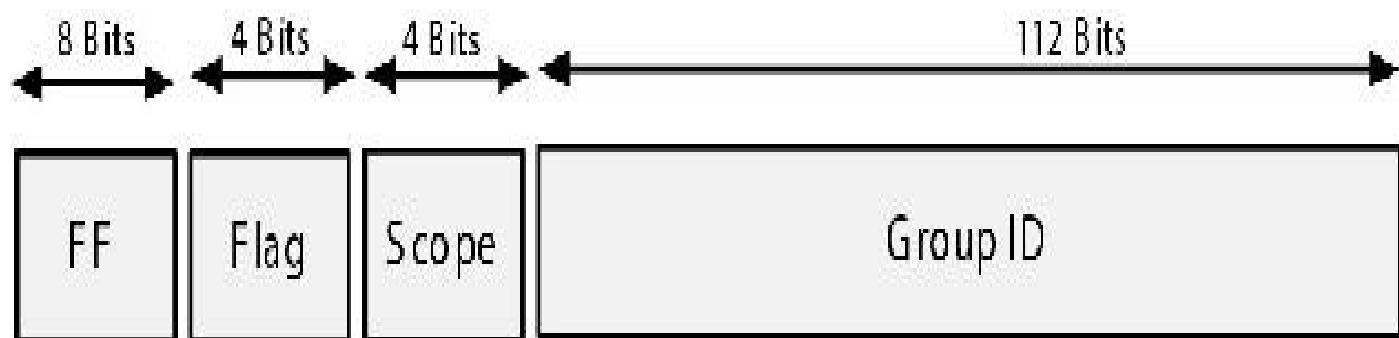


FIG 4.10 – IPv6 multicast addressing

The first eight bits of the IPv6 multicast address represent the multicast prefix FF::/8. The Flag field in the IPv6 multicast address is used to indicate the type of multicast address, either permanent or temporary.

Permanent IPv6 multicast addresses are assigned by the IANA, while temporary IPv6 multicast addresses can be used in pre-deployment multicast testing. The Flag field can contain one of the two possible values, as illustrated and described in Table 4-5 below:

Table 4-5: IPv6 permanent and temporary multicast addresses

Type of Multicast Address	Binary Representation	Hexadecimal Value
Permanent	0000	0
Temporary	0001	1

The next four bits in the multicast address represent the Scope field. In IPv6 multicasting, this field is a mandatory field that restricts multicast packets from being sent to other areas in the network. This field specifies the domain where the multicast traffic can be sent. The IPv6 multicast address scope types are listed in Table 4-6 below:

Table 4-6: IPv6 multicast address scope types

Scope Type	Binary Representation	Hexadecimal Value
Interface-local	0001	1
Link-local	0010	2
Subnet-local	0011	3
Admin-local	0100	4
Site-local	0101	5
Organization	1000	8
Global	1110	E

Within the IPv6 multicast prefix, certain addresses are reserved. These reserved addresses are well-known multicast addresses that represent specific multicast groups. They are described in Table 4-7 below and you should memorize these for the exam:

Table 4-7: IPv6 reserved multicast addresses

Address	IPv4 Equivalent	Description
FF02::1	Subnet broadcast	All hosts on the link-local scope

FF02::2	224.0.0.2	All routers on the link-local scope
FF02::5	224.0.0.5	OSPFv3 routers
FF02::6	224.0.0.6	OSPFv3 designated routers
FF02::9	224.0.0.9	All RIP routers
FF02::A	224.0.0.10	EIGRP routers

All routers must join the all-hosts multicast group of FF02::1 and the all-routers multicast group of FF02::2. I have demonstrated this by enabling IPv6 on a Fast Ethernet link between two connected routers as shown below:

```
R1(config)#ipv6 unicast-routing
```

```
R1(config)#int f0/0
```

```
R1(config-if)#ipv6 enable
```

```
R1(config-if)#no shut
```

```
R1(config-if)#end
```

```
R1#show ipv6 interface f0/0
```

FastEthernet0/0 is up, line protocol is up

IPv6 is enabled, link-local address is FE80::C000:6FF:FE95:0

No Virtual link-local address(es):

No global unicast address is configured

Joined group address(es):

FF02::1

FF02::2

Anycast Addresses

Anycast, as mentioned earlier, can be described as using the nearest address based on routing protocol metrics for one-to-nearest communication and can be considered in the same way that we think of unicast traffic. In IPv6, global unicast, site-local, or even link-local addresses can be used for anycast. However, there is also an anycast address reserved for special use. This anycast address is referred to as the subnet-router anycast address and is formed with the subnet's 64-bit unicast prefix, with the remaining 64-bits set to zero (e.g., 2001:1a2b:1111:d7e5:0000:0000:000:0000).

These addresses are commonly used by protocols such as Mobile IPv6 and they represent a service rather than a device. The same address can be found on multiple devices that are providing the same service. Any router receiving anycast advertisements from a group of servers (for example) would not be aware that they are

coming from a group of devices. Instead, the router assumes it has three routes to the same device and therefore chooses the route with the lowest cost. This concept is illustrated in Figure 4.11 below:

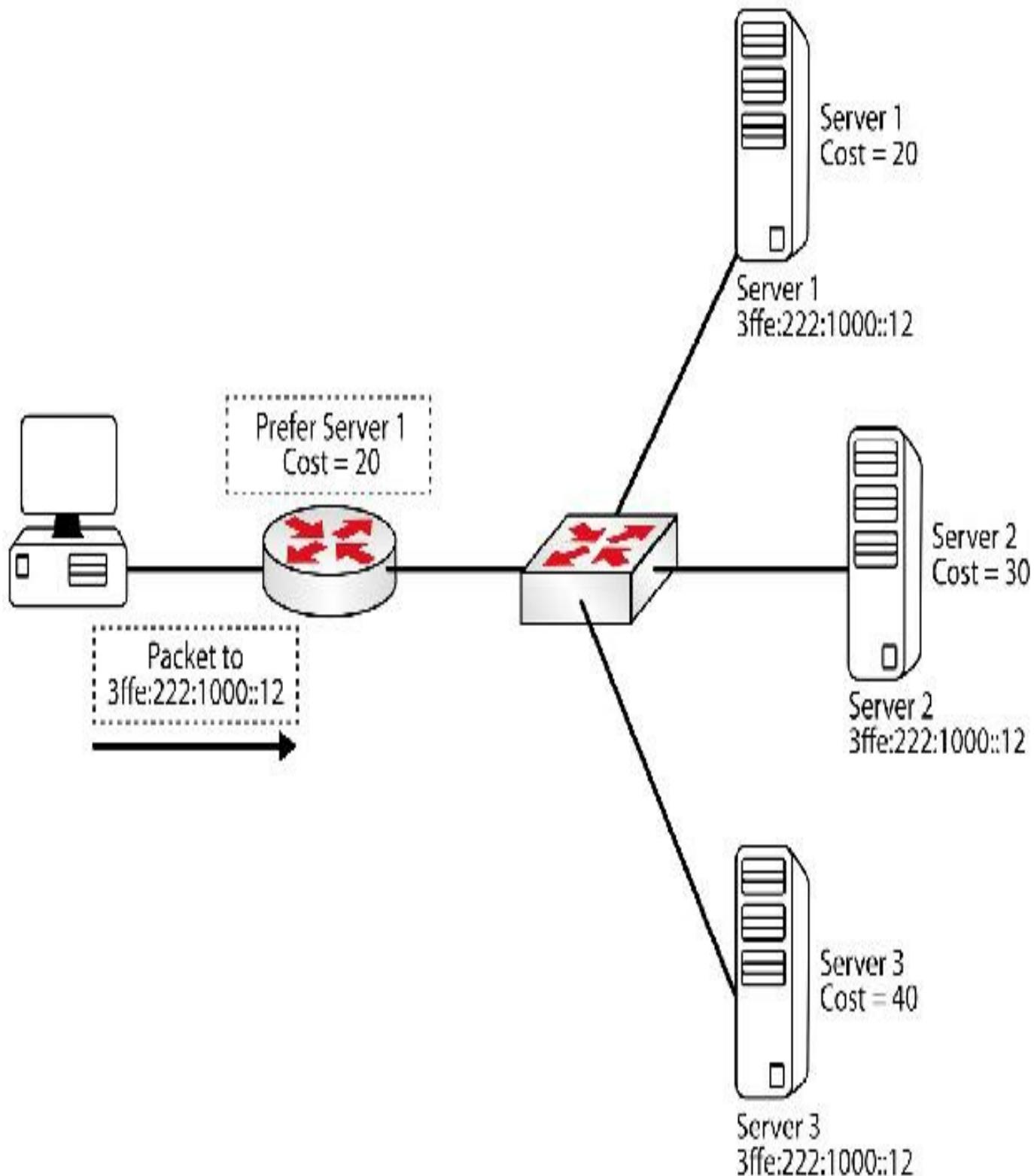


FIG 4.11 – Anycast addressing

Address Autoconfiguration

IPv6 offers an easy addressing solution for network administrators faced with the mammoth task of having to allocate addresses from an unimaginably huge range available. Automatic address configuration allows an IPv6 host to self-allocate the complete address or just the host portion. The methods available are manual, stateful, and stateless autoconfiguration.

Manual autoconfiguration is, of course, not autoconfiguration. In stateful autoconfiguration, the router uses DHCPv6 to obtain an IPv6 address, and DHCP assigns either the host portion or the entire 128-bit address. Finally, in stateless autoconfiguration (SLAAC), the interface configures its own address using Router Solicitation (RS) and Router Advertisement (RA) messages. Stateless autoconfiguration dynamically assigns the interface a 64-bit prefix, and for the host portion the EUI-64 addressing process (described below) is used. There is also stateless DHCP; however, we won't be discussing this method. Figure 4.12 below shows the stateless autoconfiguration process. We will cover RA messages shortly.

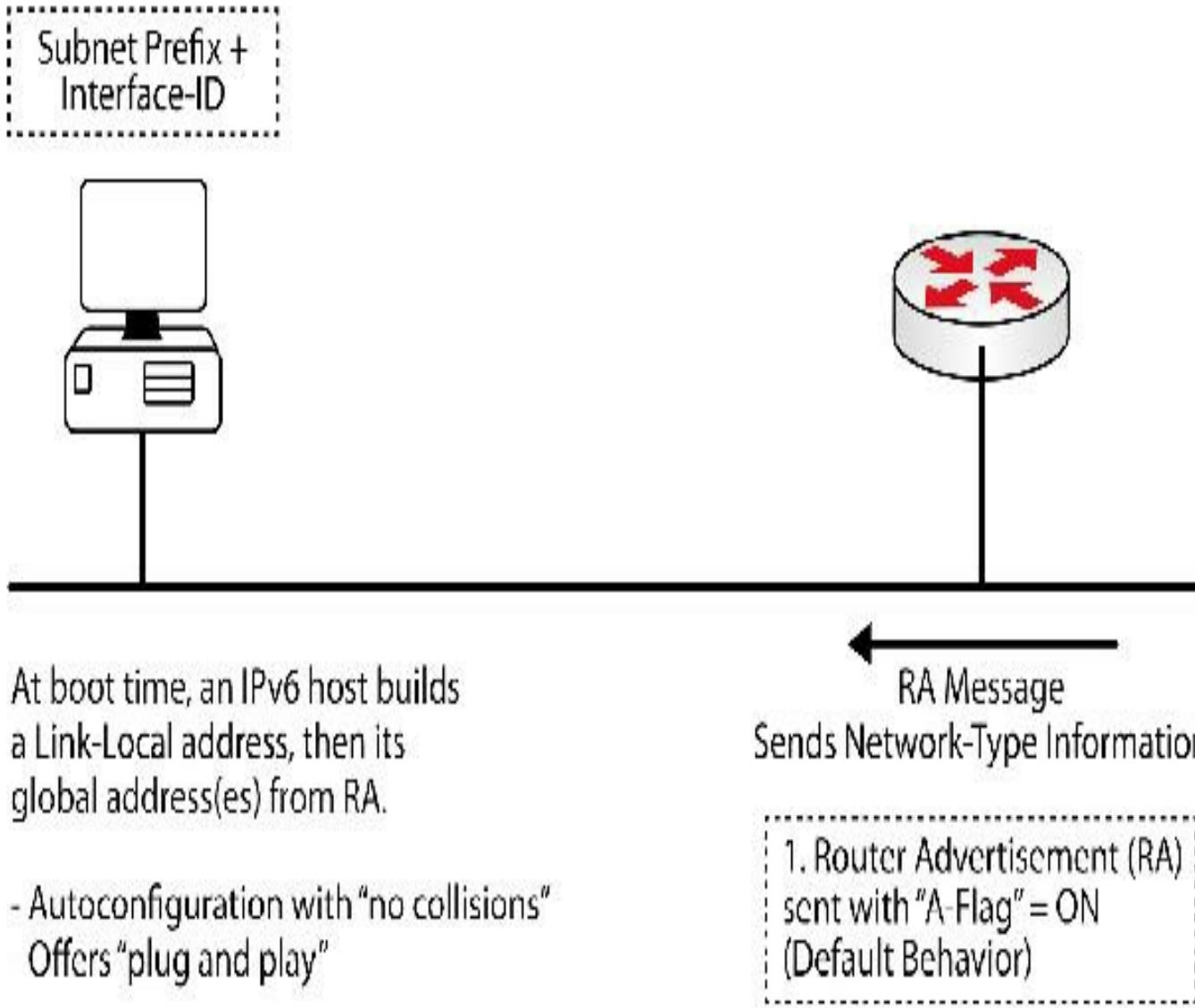


FIG 4.12 – Stateless autoconfiguration

EUI-64 Addressing

One of the coolest features of IPv6 is automatic address assignment without the need for configuration by the administrator. This immediately allows devices to communicate across an IPv6 link. A device can automatically assign itself an IPv6 address using the IEEE Extended Unique Identifier-64 (EUI-64) format. This address is generated from the unique 48-bit MAC address of the device. Because 48 bits are too short to create an IPv6 address, additional hex digits are added (see below). EUI-64 allows autoconfiguration of IPv6 addresses without depending on DHCP or manual configuration.

The host portion of an EUI-64 address is generated in two steps. First, a 16-bit field, FFFE, is inserted in the middle of the MAC address. For example, inserting FFFE in the middle of MAC address 00.11-AA.BB.CC.DD would give you this 64-bit address:

00.11.AA.FF.FE.BB.CC.DD.

The next step is to invert the U/L flag, which is the seventh bit in the 64-bit Host field. It may sound somewhat complicated but it's simply flipping one of the bits. Going back to the previous example, inverting the seventh bit in 00.11.AA.**FF.FE**.BB.CC.DD would give you **02**.11.AA.**FF.FE**.BB.CC.DD. This is because the first 8 bits (00 in hex) are 0000 0000. Inverting the seventh bit would give you 0000 0010, which is 02 in hex. This is then used as the interface ID in the EUI-64 address.

A router interface that has a MAC address of 00.11.AA.BB.CC.DD would have the following EUI-64 autogenerated IPv6 address:

Router#show interface f0/0

FastEthernet0/0 is up, line protocol is down

Hardware is Gt96k FE, address is 0011.aabb.ccdd (bia 0011.aabb.ccdd)

Router(config)# interface f0/0

Router(config-if)#ipv6 address 2001:aa::/64 eui-64

Router(config-if)#do show ipv6 interface f0/0

FastEthernet0/0 is up, line protocol is down

....

Global unicast address(es):

2001:AA::211:AAFF:FEBB:CCDD, subnet is 2001:AA::/64 [EUI]

You can expect to see a question on EUI-64 addressing in the exam because Cisco has specifically included it in the syllabus.

Loopback Addresses

Loopback addresses used in IPv6 are used in the same manner as in IPv4. Each device has an IPv6 Loopback address, which is the equivalent of the 127.0.0.1 Loopback address used in IPv4, and this address is used by the device itself. IPv6 Loopback addresses use the ::1 prefix, which is represented as 0000:0000:0000:0000:0000:0000:0001 in the preferred address format. This means that in IPv6 Loopback addresses, all bits are set to 0 except for the last bit, which is always set to 1.

These addresses are always assigned automatically when IPv6 is enabled on a device and they can never be changed. Figure 4.13 below shows me pinging the IPv6 Loopback address on my Windows 7 PC:

```
C:\Users\owner>ping ::1

Pinging ::1 with 32 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
```

```
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

FIG 4.13 – Pinging the IPv6 Loopback address on a PC

Unspecified Addresses

In IPv6 addressing, unspecified addresses are simply unicast addresses that are not assigned to any interface. These addresses indicate the absence of an IPv6 address and are used for special purposes that include IPv6, DHCP, and DAD. Unspecified addresses are always used as a source address by an interface that has yet to learn its unicast address and are represented by all 0 values in the IPv6 address, which can be written using the :: prefix. In the preferred format, these addresses are represented as 0000:0000:0000:0000:0000:0000, or compressed as 0:0:0:0:0:0:/128, or ::/128 for short. The unspecified address cannot be a destination address and cannot be assigned to an interface.

Migrating from IPv4 to IPv6

You will not find Internet users all over the world using IPv4 one day and then switching to IPv6 the next. The change will take place over a number of years in a phased approach. You will find that the address allocation is done in batches of addresses using a combination of DNS and DHCP or DHCPv6 autoconfiguration

scripts. To manually assign IPv6 addresses to nodes in a network would be an almost impossible task. DHCPv6 operation is described in RFC 3736.

There are a few methods available to phase IPv6 addressing into networks, including tunneling, dual stack, Automatic 6to4, ISATAP, and NAT-PT.

Tunneling

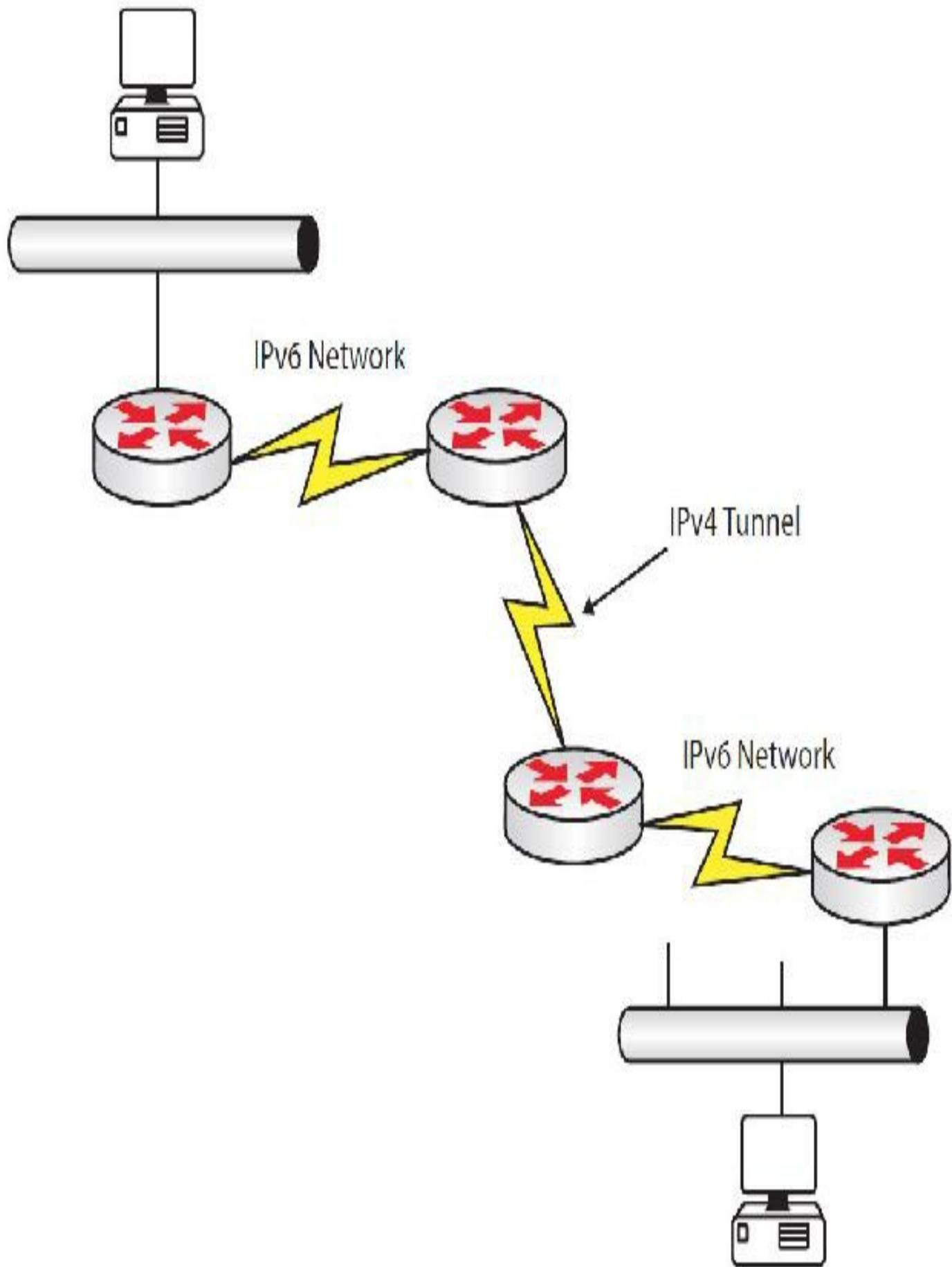


FIG 4.14 – Tunneling

Tunneling in internetworking usually refers to one type of packet being encapsulated in another type of packet. In this instance an IPv6 packet is encapsulated inside an IPv4 packet. In order for tunneling to work here, the routers must support dual stack so both IPv4 and IPv6 are running. Because almost every major network in existence is built on IPv4, tunneling already existed before IPv6 was created.

IPv6 packets up to 20 bytes can be transmitted because the IPv4 header is 20 bytes in length. The IPv4 header is appended to the packet and removed at the destination router.

IPv6 tunneling allows current IPv4 addresses to be used in conjunction with IPv6 addresses in much the same way that dual protocols can be run in a network that is transitioning from one to another.

IPv6 tunneling is defined in RFC 3056 and 2893, among others. Although Teredo tunneling is one method available, it won't be covered in this guide. Just be aware that it is available.

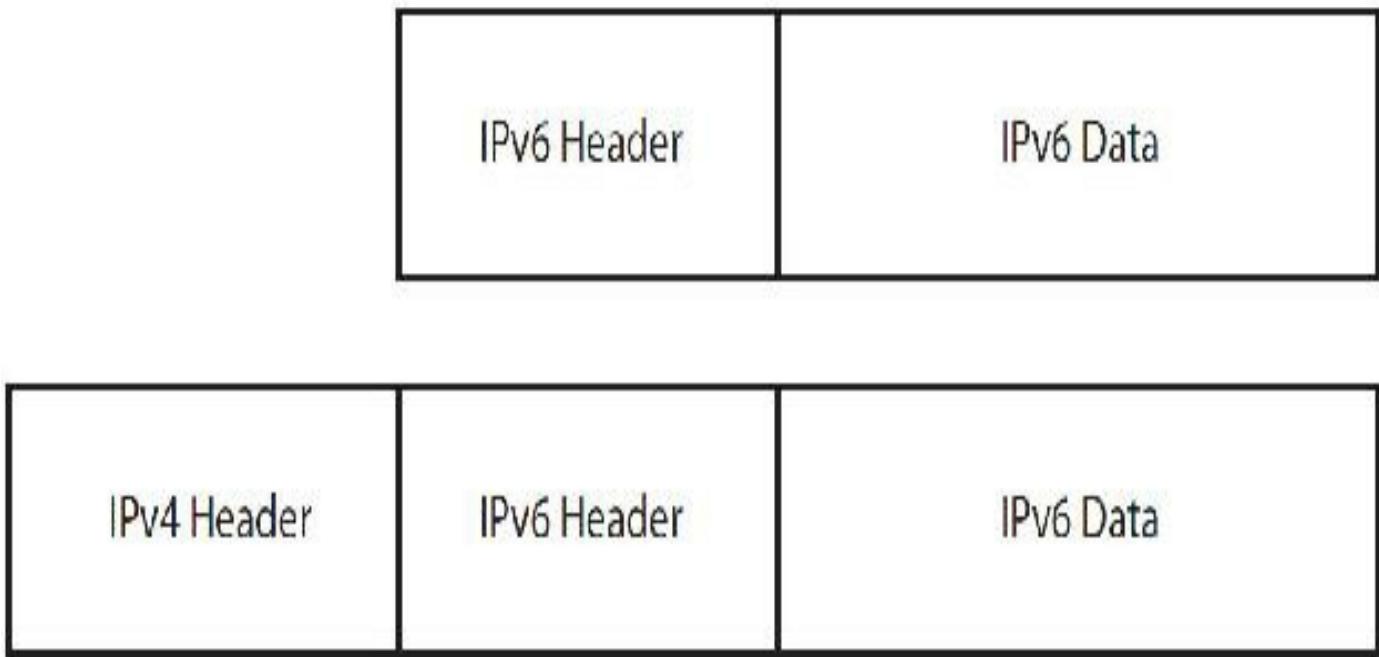


FIG 4.15 – IPv4 header on an IPv6 packet

For the purposes of the CCNA exam, we will look at:

- Dual stack
- Automatic 6to4
- ISATAP
- NAT-PT

Manually Configured Dual Stack

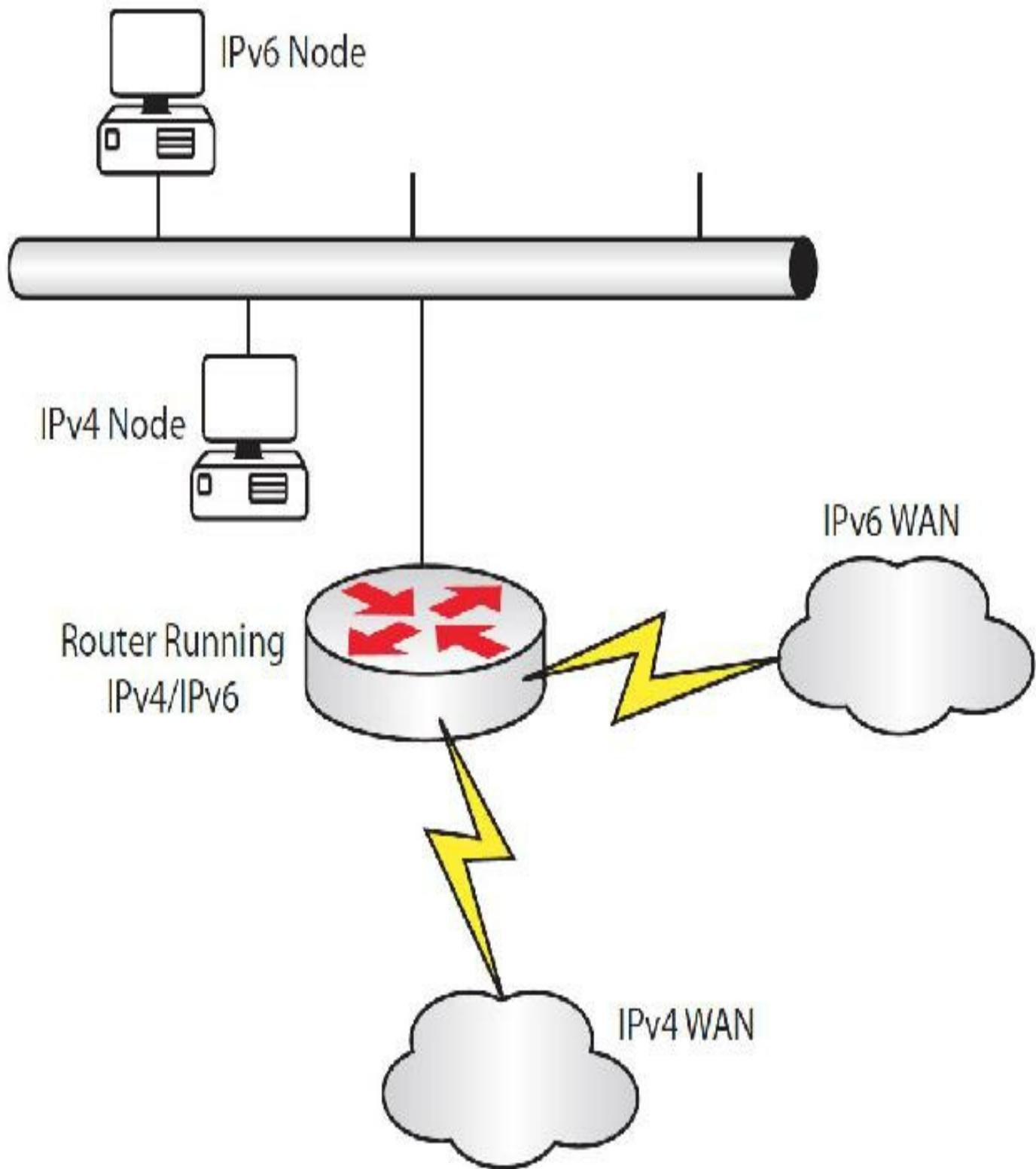


FIG 4.16 – Dual stack

Dual stack is where two IP protocol stacks run on a network device—IPv4 and IPv6. The dual stack method is the preferred migration method for networks transitioning from IPv4 to IPv6 because they can continue to run both seamlessly while the transition takes place. Dual stack can operate in the same network node interface and chooses which version of IP to use based on the destination address.

This process has been thoroughly tested by a project team referred to as 6-Bone. The only requirement for implementing IPv6 addressing in a network is connectivity to a DNS server.

Automatic 6to4

Automatic 6to4 is outlined in RFC 3056 and it enables IPv6 packets to be encapsulated within IPv4 packets. This method treats the underlying network as a Non-Broadcast Multi-Access (NBMA) network (which will be covered in the WAN section) and allows traffic to be tunneled without having to specifically configure a tunnel.

Any IPv6 address that begins with the 16-bit 2002::/16 prefix is known as a 6to4 address. The first 16 bits of the prefix are always 2002:, the next 32 bits are the IPv4 address, and the last 16 bits of the prefix are available for addressing multiple IPv6 subnets in the same 6to4 router.

ISATAP Tunnels

Intra-Site Automatic Tunnel Addressing Protocol (RFC 4214) also treats the underlying network as an NBMA cloud. ISATAP addressing requires the address 0000:5EFE to be sandwiched between the 64-bit link-local address and the IPv4 address on the ISATAP link.

NAT-PT

Network Address Translation (NAT)-Port Translation (PT) for Cisco software is based on RFC 2766 and RFC 2765. NAT-PT is a migration tool that helps customers transition their IPv4 networks to IPv6 networks. Its purpose is to facilitate bidirectional connectivity between IPv4 and IPv6 domains.

Although NAT-PT is still being used as a translation method between the two protocols, it has been deemed deprecated by IETF because of its tight coupling with DNS and general limitation in translation. NAT-PT is still supported by Cisco and other vendors and is still considered a viable option. However, Cisco recommends not using NAT-PT but instead supporting its replacement, NAT64.

IPv6 Functionality Protocols

IPv6 uses a number of underlying protocols in order to function (much in the same way IPv4 does). Some are enhancements of already familiar protocols, such as DNS, CDP, and DHCP, but a key protocol is Neighbor Discovery Protocol, which will be covered in some detail due to its importance.

DHCP for IPv6

As you already know, DNS does for IPv6 what it does for IPv4, namely, it resolves hostnames to IP addresses. DHCP is used for stateful autoconfiguration of IPv6 interfaces. You can read more about DHCPv6 in RFC 3315.

Hosts can be configured to use DHCPv6 to obtain configuration settings, or an IPv6 router can state that it wants to use DHCPv6 in an outgoing RA message. If this is the case, one of two well-known multicast addresses are used (via UDP port 547)—FF02::1:2 (all DHCP relay agents and servers) or FF05::1:3 (all DHCP servers).

DHCPv6 then replies with the relevant configuration settings using UDP port 546. In addition to the usual information you would obtain for IPv4, it can also send information for multiple subnets. Configuring a Cisco router as a DHCPv6 server is outside the scope of the CCNA exam.

ICMPv6

In much the same way that IPv4 needs a protocol to control and forward informational messages, IPv6 uses ICMP (Internet Control Message Protocol) for this service. ICMPv6 was designed exclusively for use with IPv6 and was specified in RFC 2463. Many values from ICMP match ICMPv6, such as “Destination Unreachable” and “Time Exceeded”; however, other mechanisms have been added for use by another protocol developed for IPv6 called Neighbor Discovery Protocol (NDP).

Neighbor Discovery Protocol

NDP is known as the plug-and-play aspect of IPv6. Some of these features include:

- Router discovery – a router can discover when it is at the end of another IPv6 link
- Prefix discovery – a router can discover the prefix of the other side of an IPv6 link
- Address autoconfiguration – without using DHCP, a router can self-assign an IPv6 address
- Parameter discovery – a router can discover the MTU size for the link and hop limits
- Address resolution – without using ARP, a router can discover the layer 2 address of connected devices
- Next-hop discovery – the layer 2 address of the next hop to get to a certain destination
- Neighbor unreachability – a router can determine that another host or router is no longer reachable

- Duplicate Address Detection – a router can determine whether an address it wants to use is already allocated
- Redirect – a router can notify another device of a better next-hop address for a destination

Some of the features above will be discussed shortly.

NDP packets are exchanged by connected links and therefore should be using either a link-local address or a multicast address with a link-local scope. The hop limit is set to 255 to prevent hackers from spoofing or attacking NDP. If a packet arrives with a hop limit of 254 or less, it is dropped (because it has passed through at least one router).

As previously mentioned, ICMPv6 has IPv6-specific messages built in to support IPv6 features. These messages include:

- Redirects – mentioned above
- Router Advertisement (RA) – allows routers to advertise their presence on the link and can include MTU, prefix, and hop limits; RA messages are periodically sent in response to Router Solicitations
- Router Solicitation (RS) – generated by a router to request an RA from another router
- Neighbor Solicitation (NS) – generated in order to request the data link address from another router and for Duplicate Address Detection (DAD)
- Neighbor Advertisement (NA) – these are sent in response to NS messages

2 0.061367	::	ff02::1:ff00:72b7	ICMPv6	78 Neighbor Solicitation for fe80::211:77ff:fe80:72b7
3 0.0921972	fe80::211:77ff:fe80:7ff02::1		ICMPv6	86 Neighbor Advertisement fe80::211:77ff:fe80:72b7 (r)
4 0.0932658	fe80::211:77ff:fe80:7ff02::1		ICMPv6	86 Router Advertisement from c2:00:06:f9:00:00
5 0.0942951	fe80::211:77ff:fe80:7ff02::16		ICMPv6	90 Multicast Listener Report Message v2
6 0.0953978	fe80::211:77ff:fe80:7ff02::16		ICMPv6	90 Multicast Listener Report Message v2
7 1.294567	fe80::211:77ff:fe80:7ff02::16		ICMPv6	90 Multicast Listener Report Message v2
8 1.305994	fe80::211:77ff:fe80:7ff02::16		ICMPv6	90 Multicast Listener Report Message v2
9 5.083457	c2:00:06:f9:00:00	CDP/VTTP/DTP/PMgP/UDLQ CDP		389 Device ID: R1.lab.local Port ID: FastEthernet0/0
10 6.078576	c2:00:06:f9:00:00	CDP/VTTP/DTP/PMgP/UDLQ CDP		389 Device ID: R1.lab.local Port ID: FastEthernet0/0
11 8.745130	c2:00:06:f9:00:00	c2:00:06:f9:00:00	LOOP	60 Reply

```

> Frame 2: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
> Ethernet II, Src: c2:00:06:f9:00:00 (c2:00:06:f9:00:00), Dst: IPv6icast_tt:00:72:b7 (33:33:tt:00:72:b7)
> Internet Protocol Version 6, Src: :: (::), Dst: ff02::1:ff00:72b7 (ff02::1:ff00:72b7)
▼ Internet Control Message Protocol, v6
  Type: Neighbor Solicitation (135)
  Code: 0
  Checksum: 0x1d97 [correct]
  Reserved: 00000000
  Target Address: fe80::211:77ff:fe80:72b7 (fe80::211:77ff:fe80:72b7)

```

FIG 4.17 – An IPv6 neighbor solicitation packet

R1#ping ipv6 FE80::C001:7FF:FE10:0

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FE80::C001:7FF:FE10:0, timeout is 2 seconds:

Packet sent with a source address of FE80::C000:7FF:FE10:0

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/24/40 ms

R1#

*Mar 1 01:41:47.103: ICMPv6-ND: DELETE -] INCMP: FE80::C001:7FF:FE10:0

*Mar 1 01:41:47.103: ICMPv6-ND: **Sending NS** for FE80::C001:7FF:FE10:0 on FastEthernet0/0

*Mar 1 01:41:47.119: ICMPv6-ND: **Received NA** for FE80::C001:7FF:FE10:0 on FastEthernet0/0 from FE80::C001:7FF:FE10:0

For the output above, I had the debug ipv6 nd and debug ipv6 icmp running.

Router Discovery

RA messages are periodically sent by routers running IPv6 to advertise their presence to any device on the same link. These messages will work on broadcast networks such as Ethernet, where multiple devices can receive the message. They are also sent in response to RS messages from local devices.

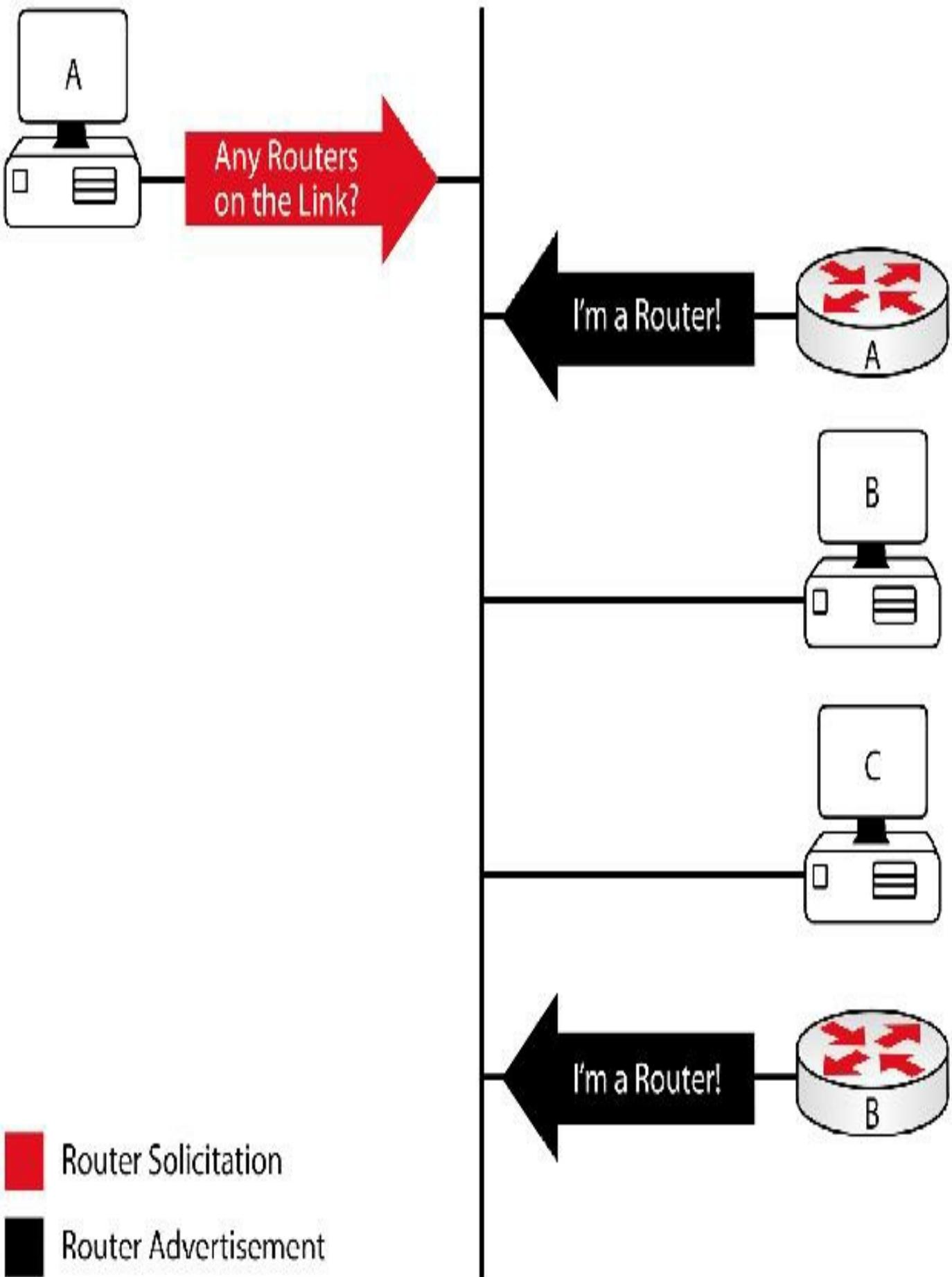


FIG 4.18 – RS and RA messages

Cisco set the interval to 200 seconds (with built-in jitter control) for RA messages but this value can be changed.

R1#show ipv6 interface fast0/0

FastEthernet0/0 is up, line protocol is up

IPv6 is enabled, link-local address is FE80::C000:7FF:FE10:0

No Virtual link-local address(es):

No global unicast address is configured

Joined group address(es):

FF02::1

FF02::2

FF02::1:FF10:0

MTU is 1500 bytes

ICMP error messages limited to one every 100 milliseconds

ICMP redirects are enabled

ICMP unreachables are sent

ND DAD is enabled, number of DAD attempts: 1

ND reachable time is 30000 milliseconds

ND advertised reachable time is 0 milliseconds

ND advertised retransmit interval is 0 milliseconds

ND router advertisements are sent every 200 seconds

ND router advertisements live for 1800 seconds

ND advertised default router preference is Medium

Hosts use stateless autoconfig for addresses.

RD messages are sent to the all-nodes multicast address FF02::1.

1 0.921972	fe80::211:77ff:fe80:7fff02::1	ICMPv6	66 Neighbor Advertisement (fc80::211:77ff:fe80:72b7) [r]
4 0.932858	fe80::211:77ff:fe80:7fff02::1	ICMPv6	66 Neighbor Advertisement (from c2:00:00:19:33:01)
5 0.942054	fc80::211:77ff:fe80:7fff02::16	ICMPv6	90 Multicast Listener Report Message v2
6 0.953378	fc80::211:77ff:fc80:7fff02::16	ICMPv6	90 Multicast Listener Report Message v2
7 1.294587	fc80::211:77ff:fc80:7fff02::16	ICMPv6	90 Multicast Listener Report Message v2
8 1.303394	fc80::211:77ff:fc80:7fff02::16	ICMPv6	90 Multicast Listener Report Message v2
9 5.683487	c2:00:00:f9:00:00	COP/VTP/DTI/PAgP/UDLD CDP	389 Device ID: R1.lab.local Port ID: FastEthernet0/0
10 6.678576	c2:00:00:f9:00:00	COP/VTP/DTI/PAgP/UDLD CDP	389 Device ID: R1.lab.local Port ID: FastEthernet0/0
11 8.745136	c2:00:00:f9:00:00	c2:00:00:f9:00:00	LOOP
			60 Reply

Frame 4: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
 Ethernet II, Src: c2:00:00:19:33:01 (c2:00:00:19:33:01), Dst: IPv6 Multicast 00:00:00:00:00:01 (33:33:00:00:00:01)
 Internet Protocol Version 6, Src: fc80::211:77ff:fc80:72b7 (fc80::211:77ff:fc80:72b7), Dst: ff02::1 (ff02::1)
 Internet Control Message Protocol v6
 Type: Router Advertisement (134)
 Code: 0
 Checksum: 0x71f7 (correct)
 Cur hop limit: 64
 Flags: 0x00
 Router lifetime (s): 1000



FIG 4.19 – RD messages sent to FF02::1

R1#debug ipv6 nd

ICMP Neighbor Discovery events debugging is on

*Mar 1 01:46:26.695: ICMPv6-ND: Sending RA from FE80::C000:7FF:FE10:0 to FF02::1 on FastEthernet0/0

*Mar 1 01:46:26.695: ICMPv6-ND: MTU = 1500

RAs are sent in response to RS messages, as mentioned.

Mar 1 01:49:47.199: ICMPv6-ND: Request to send RA for FE80::C000:7FF:FE10:0

Duplicate Address Detection

We discussed address autoconfiguration with EUI-64 addressing earlier, so you know that the chances of actually having a duplicate address are remote; however, the facility to detect any duplicate address is built into IPv6 just in case.

Whenever a device attempts to allocate an IPv6 address it does so tentatively. The address is never confirmed until DAD takes place (i.e., no other device replies to an NS packet generated by the router with that address in it as the target and the source address

as unspecified). If a response is received from the target address it wants to allocate, then it confirms that the address is in use.

Neighbor Address Resolution

A major difference between IPv4 and IPv6 is how they discover the data link layer address of a host they want to send a packet to. You've already learned that IPv4 uses ARP for this purpose but IPv6 uses NDP.

If the IPv6 address of the host is on the local link, then the router can simply examine its neighbor cache for the data link layer address. In the output below there is only one address in the neighbor cache:

```
R1#show ipv6 neighbors
```

IPv6 Address	Age	Link-layer Addr	State	Interface
FE80::C001:7FF:FE10:0	0	c201.0710.0000	REACH	Fa0/0

Here is the output from my Windows 7 command prompt:

```
C:\Users\owner>netsh
```

```
netsh>interface ipv6
```

```
netsh interface ipv6>show neighbors
```

```
Interface 14: VMware Network Adapter VMnet1
```

Internet Address	Physical Address	Type
ff02::1	33-33-00-00-00-01	Permanent
ff02::2	33-33-00-00-00-02	Permanent
ff02::c	33-33-00-00-00-0c	Permanent
ff02::16	33-33-00-00-00-16	Permanent
ff02::fb	33-33-00-00-00-fb	Permanent
ff02::1:2	33-33-00-01-00-02	Permanent
ff02::1:3	33-33-00-01-00-03	Permanent
ff02::1:fff2:6984	33-33-ff-f2-69-84	Permanent

Mini-lab – Neighbor Discovery Protocol in Action

The output below shows the NDP process taking place with a short configuration and debugging. Use the ipv6 enable command on both Fast Ethernet interfaces to create IPv6

addresses, and then make a note of them for your ipv6 route command. Also, make sure that you enable IPv6 globally on both routers with the ipv6 unicast-routing command. Add an IPv6 address manually to a Loopback interface on R2.

We'll kick off the configuration commands below presuming that you have already done the above. First, add the IPv6 address to a Loopback interface on R2.



FIG 4.20 – Mini-lab: Neighbor Discovery Protocol in Action

```
R2(config)#int lo0
```

```
R2(config-if)#ipv6 add 2001::1/64
```

```
R2(config-if)#end
```

On R1 start debugs for ND messages and add a static IPv6 route for the 2001::1 address (the static route below is for any network in fact). Check your local IPv6 addresses on your equipment when you copy my commands because they will differ (show ipv6 interface f0/0). You must add the next-hop address in this example.

```
R1#debug ipv6 nd
```

ICMP Neighbor Discovery events debugging is on

```
R1#conf t
```

```
R1(config)#ipv6 route ::/0 FastEthernet0/0 FE80::C001:7FF:FE10:0
```

```
R1(config)#end
```

Next, ping the Loopback address on R1. Because the address is not present in the neighbor cache, it is marked as incomplete (INCMP). An NS is then sent and an NA is received, allowing the Data Link field to be populated and the packet sent.

```
R1#ping ipv6 2001::1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2001::1, timeout is 2 seconds:

!

```
*Mar 1 06:39:08.622: ICMPv6-ND: STALE -] DELAY: FE80::C001:7FF:FE10:0
```

```
*Mar 1 06:39:08.650: ICMPv6-ND: DELETE -> INCMP: 2001::1
```

*Mar 1 06:39:08.650: **ICMPv6-ND: Sending NS for 2001::1 on FastEthernet0/0!**

*Mar 1 06:39:13.618: ICMPv6-ND: Sending NS for FE80::C001:7FF:FE10:0 on FastEthernet0/0

***Mar 1 06:39:13.654: ICMPv6-ND: Received NA for FE80::C001:7FF:FE10:0 on FastEthernet0/0 from FE80::C001:7FF:FE10:0**

*Mar 1 06:39:13.654: ICMPv6-ND: PROBE -> REACH: FE80::C001:7FF:FE10:0

*Mar 1 06:39:13.662: ICMPv6-ND: Received NS for FE80::C000:7FF:FE10:0 on FastEthernet0/0 from FE80::C001:7FF:FE10:0

*Mar 1 06:39:13.666: ICMPv6-ND: Sending NA for FE80::C000:7FF:FE10:0 on FastEthernet0/0

Success rate is 60 percent (3/5), round-trip min/avg/max = 8/22/36 ms

R1#

[END OF MINI-LAB]

Mini-lab – Configuring IPv6

Cisco IOS supports IPv6 commands in version 12.2(2)T or later. To implement IPv6 on a Cisco device, add the configuration below to the interface:

```
R1#config t
R1(config)#ipv6 unicast-routing
R1(config)#interface FastEthernet0/0
R1(config-if)#ipv6 address 2001:c001:b14:2::c12/125
R1(config-if)#end
```

```
R1#show ipv6 interface
FastEthernet0/0 is up, line protocol is down
IPv6 is enabled, link-local address is FE80::20E:83FF:FEF5:FD4F [TENTATIVE]
Global unicast address(es):
2001:C001:B14:2::C12, subnet is 2001:C001:B14:2::C10/125 [TENTATIVE]
```

You can also use automatic address configuration with the commands below:

```
R1(config)#int fast0/1
R1(config-if)#ipv6 address autoconfig
R1(config-if)#no shut
R1(config-if)#end

R1#show ipv6 int f0/1
```

FastEthernet0/1 is up, line protocol is up

IPv6 is enabled, link-local address is FE80::C000:6FF:FE95:1

No Virtual link-local address(es):

No global unicast address is configured

To enable IPv6 processing on an interface that has not been configured with an explicit IPv6 address, use the `ipv6 enable` command in interface configuration mode.

[END OF MINI-LAB]

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 4 exam.

Chapter 4 Labs

Lab 1: Simple IPv6

The physical topology is shown in Figure 4.21 below:



FIG 4.21 – Simple IPv6

Lab Exercise

Your task is to configure the network in Figure 4.21 above. Text in Courier New font indicates commands that can be entered on the router. You can ignore the clock rate command if you are using GNS3.

Purpose

IPv6 addressing is very easy to configure.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 4.21 above.
2. Enable IPv6 on both routers.
3. Add the designated IPv6 address on each router interface.
4. Ping across the Serial link.

Lab Walk-through

1. To set the IP addresses on a router interface, you will need to do the following:

```
Router#config t  
Router(config)#hostname Router1  
Router1(config)#ipv6 unicast routing  
Router1(config)#interface f0/0  
Router1(config-if)#ipv6 address 2001:c001:b14:2::c2/125  
Router1(config-if)#no shutdown  
Router1(config-if)#end
```

Router 2:

```
Router#config t
Router(config)#hostname Router2
Router2(config)#ipv6 unicast routing
Router2(config)#interface f0/0
Router2(config-if)#ipv6 address 2001:c001:b14:2::c1/125
Router2(config-if)#no shutdown
Router2(config-if)#^Z
```

2. Check your IPv6 addresses and interface details. Note how the MAC address has been used to automatically create a link-local EUI-64 address.

```
Router1#show ipv6 interface brief
FastEthernet0/0      [administratively down/down]
  FE80::C006:8FF:FE56:0
2001:C001:B14:2::C2
FastEthernet0/1      [administratively down/down]
```

Router1#

Router1#show ipv6 interface f0/0

FastEthernet0/0 is up, line protocol is down

IPv6 is enabled, link-local address is **FE80::C006:8FF:FE56:0** [TEN]

No Virtual link-local address(es):

Global unicast address(es):

2001:C001:B14:2::C2, subnet is 2001:C001:B14:2::C0/125 [TEN]

Joined group address(es):

FF02::1

FF02::2

MTU is 1500 bytes

ICMP error messages limited to one every 100 milliseconds

ICMP redirects are enabled

ICMP unreachables are sent

ND DAD is enabled, number of DAD attempts: 1

ND reachable time is 30000 milliseconds

Hosts use stateless autoconfig for addresses.

Router1#

3. Ping across the link now.

```
Router1#ping ipv6 2001:c001:b14:2::c1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2001:C001:B14:2::C1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/13/52 ms

Router1#

Show Runs

```
hostname Router1
!
ipv6 unicast-routing
!
interface FastEthernet0/0
ipv6 address 2001:C001:B14:2::C2/125
!
```

[output truncated]

```
hostname Router2
!
ipv6 unicast-routing
!
interface FastEthernet0/0
ipv6 address 2001:C001:B14:2::C1/125
!
```

[output truncated]

Chapter 5 — IP Routing Technologies

What You Will Learn in This Chapter

What Is Routing?

Static Routing

Gateway of Last Resort

Administrative Distances

Classful and Classless Routing

Dynamic Routing

Distance Vector Protocols

Link State Protocols

Cisco Express Forwarding

Syllabus Topics Covered

4.0 IP Routing Technologies

4.1 Describe basic routing concepts

4.1.a Packet forwarding

4.1.b Router lookup process

4.1.c Process switching/fast switching/CEF

4.5 Configure and verify routing configuration for a static or default route given specific routing requirements

4.6 Differentiate methods of routing and routing protocols

4.6.a Static vs. dynamic

4.6.b Link state vs. distance vector

4.6.c Next hop

4.6.d IP routing table

4.6.e Passive interfaces (how they work)

We will also cover the ICND2 subjects below in this chapter because they are a natural fit:

2.4 Differentiate methods of routing and routing protocols

2.4 a Administrative distance

2.4.b Split horizon

2.4.c Metric

2.4.d Next hop

Routing concepts are a core CCNA topic. As a network engineer, you will be expected to understand the key differences between distance vector and link state protocols and advise your clients on the benefits and challenges associated with each protocol. You must understand the topics in this chapter if you hope to become a good Cisco engineer.

What Is Routing?

At its most basic level, IP routing (IP forwarding) is the process of moving information in packet format across networks from point A to B. It could be down the road to a friend's house or across the world to a mail server. Routing occurs at the third layer of the OSI model. It is the router's job to deal with the process of routing (hence the name).

Routing involves two processes:

1. Optimal path determination – deciding the best path from A to B
2. Routing/packet switching – actually sending the information from A to B

Sending the information is the easy part; determining the best route to take is the difficult part. It's just like planning a journey across America using a map. Putting your foot on the gas and steering is the easy part; working out the route and dealing with detours and maybe getting lost en route is the hard part.

To determine how to get from A to B, the router has to perform a calculation to decide how many possible paths there are to get there. Just like making a phone call, a signal has many possible paths to choose from.

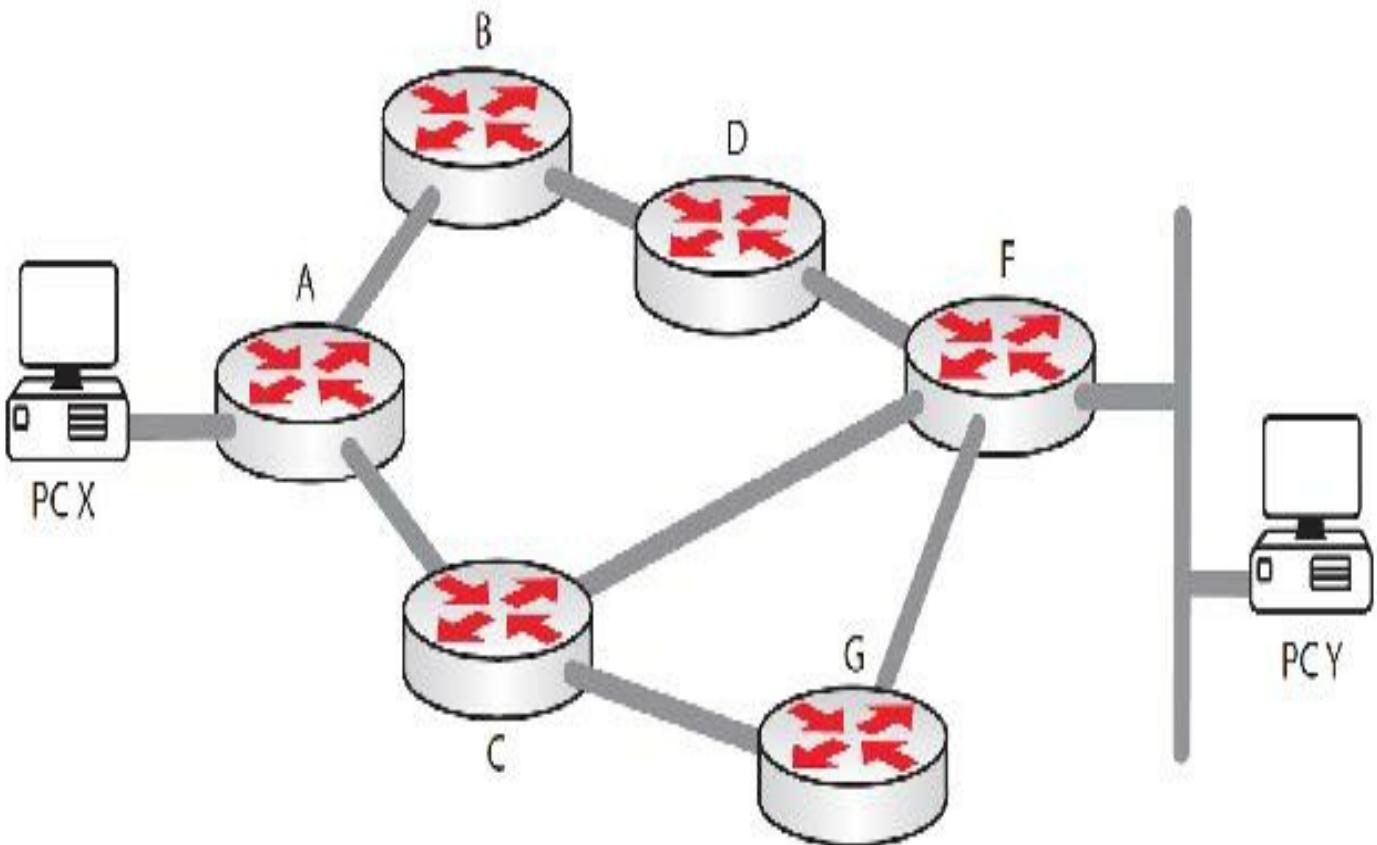


FIG 5.1 – How do you find the best path from PC X to PC Y?

In Figure 5.1 above, it may appear that PC X should use Routers A, C, and then F to get to PC Y. This is the shortest path after all. What if the link between Router C and Router F is only a 56 Kbps link? If PC X takes the path through Routers A, B, D, and then F, what if the link between Router D and Router F is not very reliable? What if it keeps going up and down (flapping) every few minutes? Should PC X take the slower path through Routers A, C, and F instead?

In finding the best path from PC X to PC Y, there are several choices to make, and each has pros and cons since the best path may not be the shortest path and it may not be the path with the highest bandwidth. Multiple factors are used in deciding which is the best path and these vary from protocol to protocol. Of course, as the network administrator, you can add commands so that one route is preferred over another.

In deciding which path to take, the router consults its routing table to determine the best path. This obviously requires that a routing protocol is in use, the interfaces are all correctly addressed, and static routing or at least a default gateway for the router to send outgoing traffic. We will look at gateways shortly. There should be both a route in the routing table AND a pointer in the form of a next-hop address or an exit interface.

If you are using a routing protocol (dynamic routing), then there must be several mechanisms in place to converge all the routes the networks learn. There must also be a

mechanism to decide how a router will propagate its learned routes. If Router F learns how to reach the network PC X is connected to, for example, it may advertise this fact to Routers D, G, and C, but this may cause an issue because for these routers, but surely they should use Routers C, B, or A to reach this network and not F which is further away.

There are other issues to consider, such as should the routers advertise only to the networks that are directly connected to them or should they also be able to advertise to networks learned from neighbor routers? How should these neighbor relationships form in the first place and how should these relationships be maintained?

Each routing protocol deals with these issues in a different way. We'll look into the protocols covered in the CCNA syllabus. If you enjoy learning about them, then I encourage you to progress to CCNP RS certification once you pass your CCNA exam.

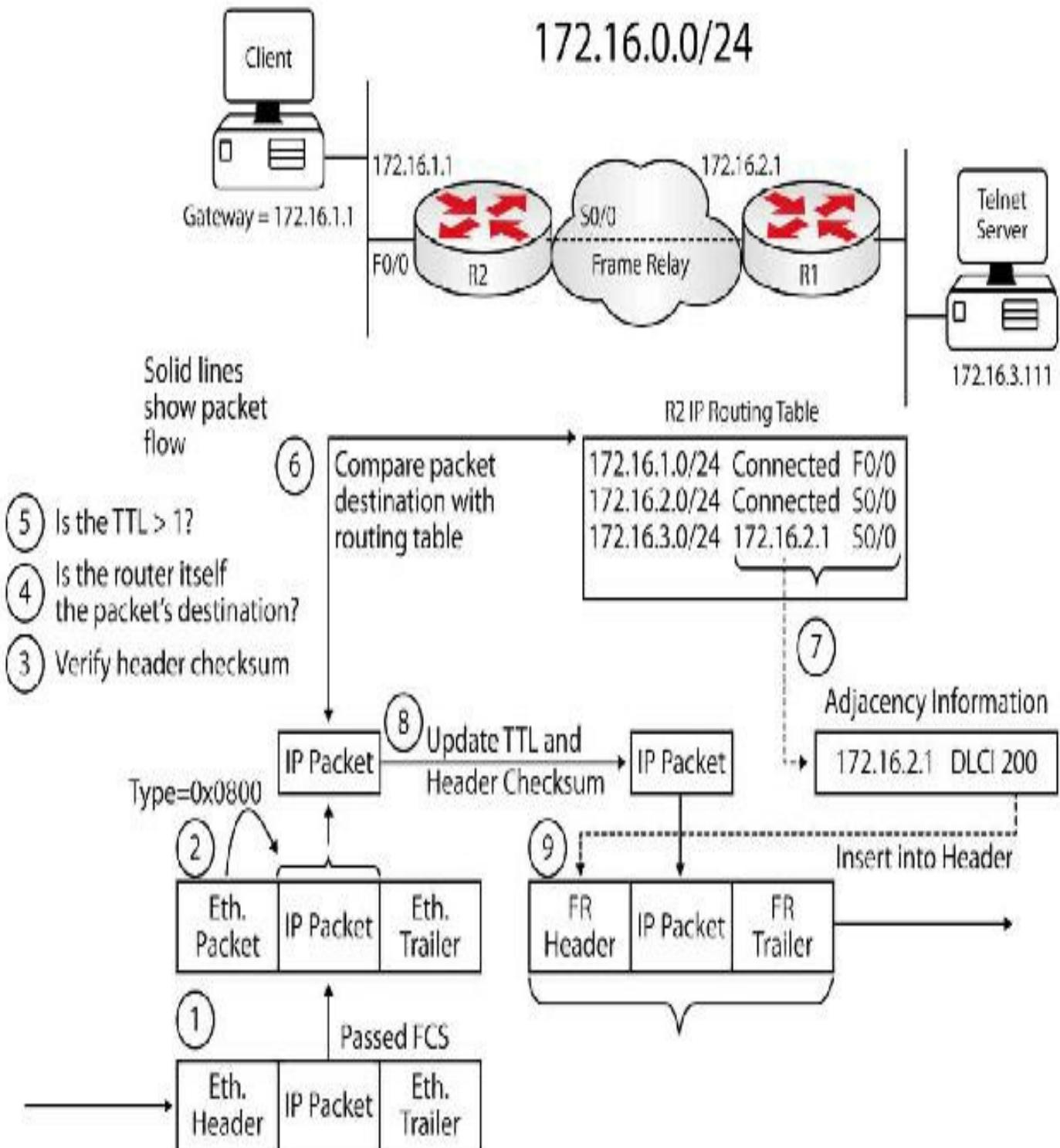


FIG 5.2 – The routing process

Referencing Figure 5.2 above, the routing table contains information about different networks, how to reach them, and how valid the path is. This information in the routing table can be entered manually by the network administrator or automatically learned from its neighbors via a routing protocol. As will be discussed shortly, the router will choose the best path to take based on the longest match rule and the administrative

distance.

1. When a router receives a frame, it first checks the FCS field. If there is an error found there is no point in moving to the next step, so the packet is discarded.
2. The router then checks the Type field and extracts the data, discarding the data link layer header and trailer.
3. The router verifies the checksum on the packet header.
4. If the destination of the packet is the router itself, there is no need to forward it.
5. If the TTL value is less than 1, then the packet is discarded.
6. The router checks its routing table to find the most specific match (there could be more than one path to reach the destination). If there is no entry for the destination in the routing table, the packet will be dropped.
7. The routing table should include the exit interface as well as the next-hop IP address. This information is required to correctly encapsulate the frame. Layer 2 and 3 addresses are also required.
8. The TTL field must then be decremented and a new header checksum calculated.
9. A new FCS is calculated and the frame is encapsulated with a new data link header and the destination address. The IP address will not change; however (as discussed in Proxy ARP), the data link source and destination will. This is presuming the packet is going over an Ethernet connection. A different data link address will be used for Frame Relay or PPP, for example.

Cisco offers a range of switching methods and enhancements, including Cisco Express Forwarding, which will be covered later.

Routing information entered by the administrator is known as static routing. If the router automatically learns the information from its neighbors, this is known as dynamic routing.

Prefix Matching

Routers use the longest prefix match rule when determining which of the routes placed into the routing table should be used to forward traffic to a destination network or node. Longer or more specific routing table entries are preferred over less specific entries, such as summary addresses, when determining which entry to use to route traffic.

The longest prefix or the most specific route will be used to route traffic to the destination network or node, regardless of the administrative distance of the route source or even the routing protocol metric assigned to the prefix if multiple overlapping

prefixes are learned via the same routing protocol. Table 5-1 below illustrates the order of route selection on a router sending packets to the address 1.1.1.1. This order is based on the longest prefix match lookup.

Table 5-1: Matching the longest prefix

Routing Table Entry	Order Used
1.1.1.1/32	1st
1.1.1.0/24	2nd
1.1.0.0/16	3rd
1.0.0.0/8	4th
0.0.0.0/0	5th

NOTE: Although the default route is listed last in the route selection order in Table 5-1, keep in mind that a default route is not always present in the routing table. If that is the case and no other entries to the address 1.1.1.1 exist, packets to that destination are simply discarded by the router. In most cases, the router will send the source host an ICMP message indicating that the destination is unreachable.

Building the IP Routing Table

Without a populated routing table or Routing Information Base (RIB) that contains entries for remote networks, routers will not be able to forward packets to those remote networks. The routing table may include specific network entries or simply a single default route. The information in the routing table is used by the forwarding process to forward traffic to the destination network or host. The routing table itself does not actually forward traffic.

Cisco routers use administrative distance, the routing protocol metric, and prefix length to determine which routes will actually be placed into the routing table, which allows the router to build the routing table. The routing table is built via the following general steps:

1. If the route entry does not currently exist in the routing table, add it to the routing table.
2. If the route entry is more specific than an existing route, add it to the routing table. It should also be noted that the less specific entry is still retained in the routing table.
3. If the route entry is the same as an existing one but is received from a more preferred route source, replace the old entry with the new entry.

4. If the route entry is the same as an existing one and is received from the same protocol:
 - a. Discard the new route if the metric is higher than the existing route; or
 - b. Replace the existing route if the metric of the new route is lower; or
 - c. If the metric for both routes is the same, use both routes for load balancing.

When building the RIB by default, the routing protocol with the lowest administrative distance value will always win when the router is determining which routes to place into the routing table. For example, if a router receives the 10.0.0.0/8 prefix via external EIGRP, OSPF, and internal BGP, the OSPF route will be placed into the routing table. If that route is removed or is no longer received, the external EIGRP route will be placed into the routing table. Finally, if both the OSPF and the external EIGRP routes are no longer present, the internal BGP route will be used.

Once routes have been placed into the routing table, by default the most specific or longest match prefix will always be preferred over less specific routes. This is illustrated in the following example, which shows a routing table that contains entries for the 80.0.0.0/8, 80.1.0.0/16, and 80.1.1.0/24 prefixes. These three route prefixes are received via the EIGRP, OSPF, and RIP routing protocols, respectively.

R1#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B – BGP,

D - EIGRP, EX - EIGRP external, O - OSPF,

IA - OSPF inter area, N1 - OSPF NSSA external type 1,

N2 - OSPF NSSA external type 2, E1 - OSPF external type 1,

E2 - OSPF external type 2, i - IS-IS, L1 - IS-IS level-1,

L2 - IS-IS level-2, ia - IS-IS inter area,

* - candidate default, U - per-user static route, o – ODR,

P - periodic downloaded static route

Gateway of last resort is not set

R 80.1.1.0/24 [120/1] via 10.1.1.2, 00:00:04, Ethernet0/0.1

D 80.0.0.0/8 [90/281600] via 10.1.1.2, 00:02:02, Ethernet0/0.1

O E2 80.1.0.0/16 [110/20] via 10.1.1.2, 00:00:14, Ethernet0/0.1

Referencing the output shown above, the first route is the 80.1.1.0/24 route. This route is learned via RIP and therefore has a default administrative distance value of 120. The

second route is the 80.0.0.0/8 route. This route is learned via internal EIGRP and therefore has a default administrative distance value of 90. The third route is the 80.1.0.0/16 route. This route is learned via external OSPF and therefore has an administrative distance value of 110.

NOTE: Because the routing protocol metrics are different, they are a non-factor in determining the best route to use when routes from multiple protocols are installed into the routing table.

Based on the contents of this routing table, if the router received a packet destined to 80.1.1.1, it would use the RIP route because this is the most specific entry, even though both EIGRP and OSPF have better administrative distance values and are therefore more preferred route sources. The show ip route 80.1.1.1 command illustrated below can be used to verify this statement:

```
R1#show ip route 80.1.1.1
```

Routing entry for 80.1.1.0/24

Known via “rip”, distance 120, metric 1

Redistributing via rip

Last update from 10.1.1.2 on Ethernet0/0.1, 00:00:15 ago

Routing Descriptor Blocks:

* 10.1.1.2, from 10.1.1.2, 00:00:15 ago, via Ethernet0/0.1

Route metric is 1, traffic share count is 1

Static Routing

Static routes are very common on networks both large and small. The administrator configures the router so that to get to a particular network, traffic either leaves through a certain router interface or goes to the next-hop IP address. Configuring the routing in this way gives you very precise control over which traffic goes where on your network.

Static routes are ideal when the network state does not change often. An example of this is a stub network. This is where there is only one way in and out of the network. Static routes also improve network security inasmuch that bogus dynamic routes can't be injected into the network by an attacker (if the network is running only static routes).

The disadvantage of using static routes is that an administrator is needed to update the route when there is a change to any routes in the network. For instance, if the next hop loses its connectivity to a route, the router will continue to send traffic to it until an administrator notices this and updates or removes the route. Also, once you have more

than a handful of static routes, they can become a time-consuming administrative task to change and update.

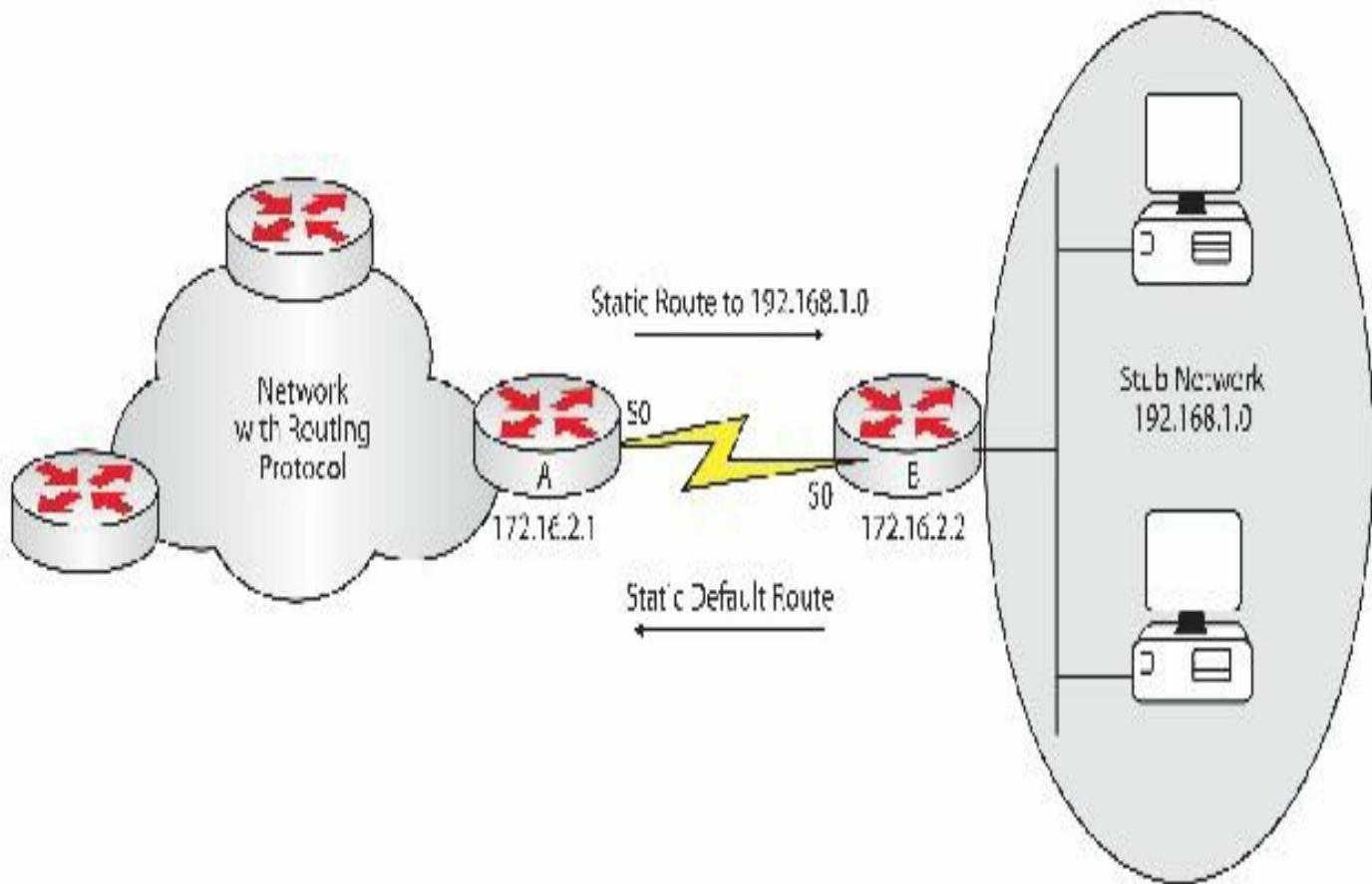


FIG 5.3 – Static routing into a stub network

In the network above, Router A will run a routing protocol to exchange routes with its neighbor routers. It has a static route to reach the 192.168.1.0 network either pointing out of the Serial interface or to the next hop of 172.16.2.2. There is no point in including Router B in the routing updates because it's a stub network (i.e., it doesn't lead to any other networks). All traffic that isn't bound for the attached network of 192.168.1.0 must go to Router A.

Mini-lab – Configuring a Static Route

In order to configure a static route the router has to be in global configuration mode:

```
RouterA(config)#ip route [network prefix_mask] [address | interface] [distance]
```

- **network** – the destination network
- **prefix_mask** – the subnet mask for that network
- **address** – the IP address of the next-hop router
- **interface** – the interface by which the traffic leaves
- **distance** – (optional) the administrative distance of the route; this is an indicator

of the validity of the routing protocol (source of the route), and lower distances are always preferred

- There are other parameters but these have been removed as they are not relevant to the CCNA exam. Bear in mind that in order for a static route to work, the next-hop address (if used) needs to be reachable and you must add an IP address to your exit interface. It always pays to test your configurations with a ping as we will do in our labs.

Look at Figure 5.4 below and configure the IP addressing as per the diagram. Next, add the static route below on Router A. You can replace PCs for Loopback interfaces. If you are using a PC you will need to add a default gateway.

```
RouterA(config)#ip route 192.168.1.0 255.255.255.0 172.16.1.2 110
```

192.068.1.0 is the destination network. 255.0.0.0 is the subnet mask for that network and 172.16.1.2 is the next hop for the router to use. 110 is the administrative distance (AD), which we will look at later. Static routes with an AD specified by the network administrator are known as floating static routes. They can serve as backup in case a dynamic route goes down by using an alternative link to reach the destination network.

Alternatively, you could have specified the interface by which the traffic leaves:

```
ip route 192.168.1.0 255.255.255.0 Serial0
```

This tells the router that to get to network 192.168.1.0, leave by interface Serial 0. A static route specifying an exit interface is given an AD of 0; this is the same as a connected network's AD. We will discuss administrative distances shortly.

```
ip route 192.168.1.0 255.255.255.0 s0
```

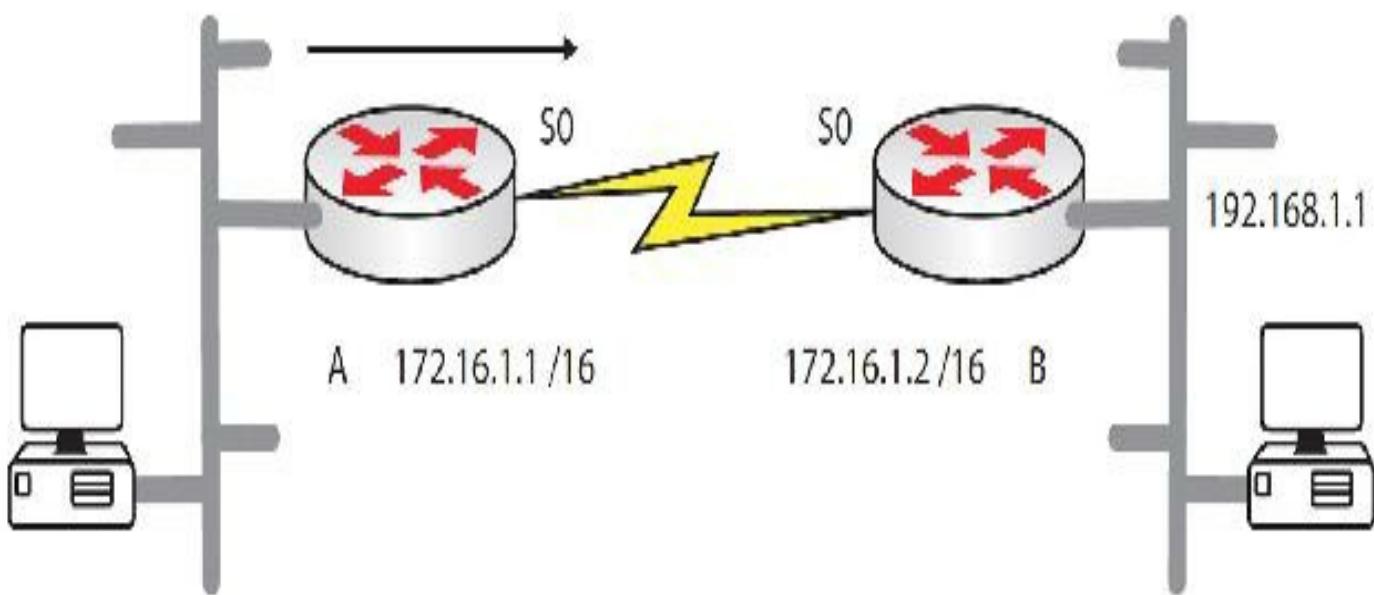


FIG 5.4 – A static route to get to Network B

[END OF MINI-LAB]

When you give the router a static route to use, it is vital that the router knows how to get to that next hop. Normally, the next hop would be part of a directly connected network. Routers are automatically aware of directly connected networks. If the router does not know how to get to the next hop, then the route will not be installed in the routing table.

One important fact to remember is that communication is a two-way process, so the router on the other side (destination) must have a route back to the source. If the router is not aware of the source network there will never be a response. Just as if you do not put a return address on an envelope, a lost packet will not be returned to the source.

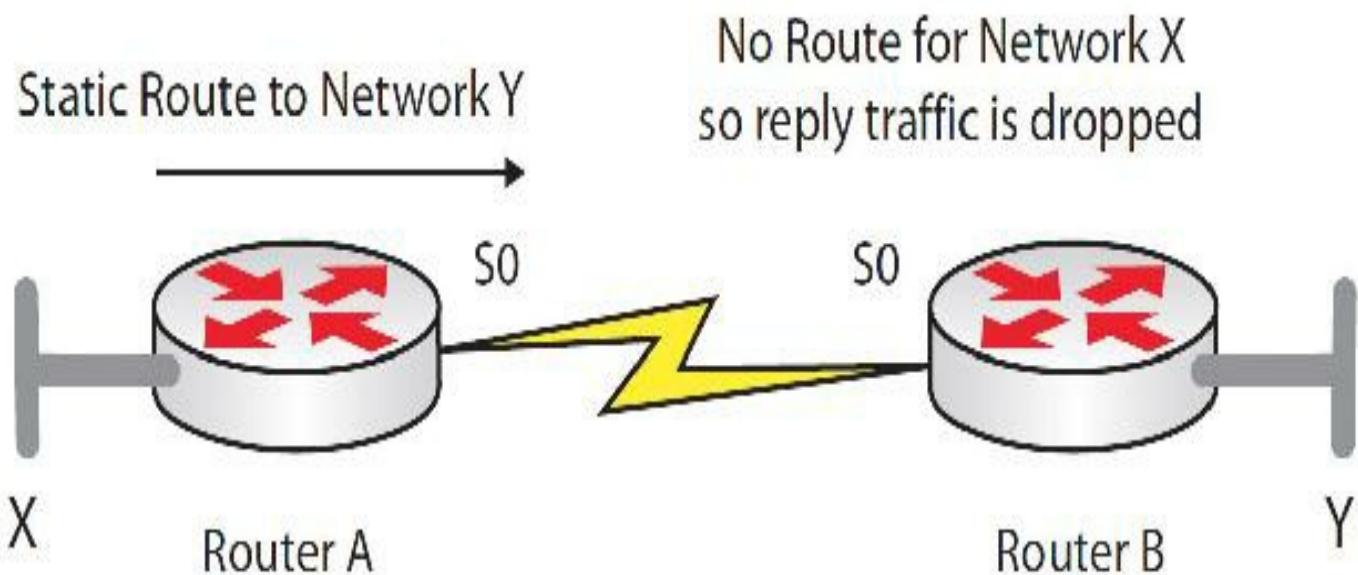


FIG 5.5 – Destination devices must know how to get back to the source network

You can also specify an exit interface along with a next-hop address to prevent constant ARP broadcasts on the segment. An example of this is:

```
ip route 192.168.1.0 255.255.255.0 Serial0 172.16.1.2
```

You can view the static-only routes in the routing table with the `show ip route static` command.

Mini-lab – Configuring an IPv6 Static Route

IPv6 static routing follows much the same process as IPv4; however, as you know, IPv6 uses a network prefix instead of a subnet mask. You must also enable IPv6 routing because it's disabled by default. You can add a next-hop address for any configured route or an exit interface.

We will use the simple topology in Figure 5.6 below, which uses EUI-64 addressing on

the Fast Ethernet interfaces and has a manually added address to Loopback 0 on Router 2.



FIG 5.6 – Mini-lab: Configuring an IPv6 Static Route

```
R1(config)#ipv6 unicast-routing
```

```
R1(config)#int f0/0
```

```
R1(config-if)#ipv6 enable
```

```
R1(config-if)#no shut
```

```
*Mar 1 00:06:05.047: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
```

```
R1(config-if)#end
```

```
R1#show ipv6 interface brief
```

```
FastEthernet0/0 [up/up]
```

```
    FE80::C000:8FF:FE01:0
```

```
FastEthernet0/1 [administratively down/down]
```

You can now configure the IPv6 addresses on R2:

```
R2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#ipv6 uni
```

```
R2(config)#int f0/0
```

```
R2(config-if)#ipv6 enable
```

```
R2(config-if)#no shut
```

```
R2(config-if)#int lo0
```

```
R2(config-if)#ipv6 address fec0::1/16
```

```
R2(config-if)#end
```

```
R2#show ipv6 int brief
FastEthernet0/0      [up/up]
    FE80::C001:8FF:FE01:0
FastEthernet0/1      [administratively down/down]
Loopback0            [up/up]
    FE80::C001:8FF:FE01:0
    FEC0::1
```

You can see that there is no route to the FEC0 network on R1:

```
R1#show ipv6 route
```

IPv6 Routing Table - 1 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6,
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea,
IS - ISIS summary, O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,
ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

L FF00::/8 [0/0]

via ::, Null0

A useful feature to determine the remote IPv6 address (if you don't have an accurate diagram or access to the device) is CDP, which was already discussed briefly. You will need to use the detail tag:

```
R1#show cdp nei detail
```

Device ID: R2.lab.local

Entry address(es):

 IPv6 address: **FE80::C001:8FF:FE01:0** (link-local)

Platform: Cisco 3725, Capabilities: Router Switch IGMP

Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/0

Holdtime : 176 sec

[output truncated]

You can ping this address to check connectivity:

R1#ping ipv6 FE80::C001:8FF:FE01:0

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FE80::C001:8FF:FE01:0, timeout is 2 seconds:

Packet sent with a source address of FE80::C000:8FF:FE01:0

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/22/40 ms

It should be no surprise that the next ping fails because there is no route to it:

R1#ping ipv6 FEC0::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FEC0::1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

You need to add a route to reach the network. If you are doing this on a broadcast network (such as Ethernet), you need to also specify an exit interface because R2 will not respond to a neighbor solicitation for a network or host it has a route to. If it was a Serial link, you could just specify the destination network and next-hop (link-local) address. Your link-local address may well differ from mine, so please check.

R1(config)#ipv6 route fec0::/64 FastEthernet0/0 FE80::C001:8FF:FE01:0

R1(config)#end

R1#ping ipv6 FEC0::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FEC0::1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/20/28 ms

R1#show ipv6 route

IPv6 Routing Table - 2 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6,
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea,
IS - ISIS summary, O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,
ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

S **FEC0::/64 [1/0]**

via **FE80::C001:8FF:FE01:0, FastEthernet0/0**

L FF00::/8 [0/0]

via ::, Null0

[END OF MINI-LAB]

Gateway of Last Resort

Normally, when a router is looking for a network that is not in its routing table, it will simply drop the packet. To ensure that a router has a last resort address or interface to send packets to, you can define a gateway of last resort. This process can be enabled with the ip default-gateway command, the ip route 0.0.0.0 0.0.0.0 command, or the ip default-network command.

IP Default-Gateway

The ip default-gateway command is only used on routers without IP routing enabled. Without IP routing enabled, a router is just another host that needs a default gateway. You should rarely or never need to use this command. Since it's unlikely that you would ever use this command, watch out for trick questions in the exam. This command is used on network switches so that it can be remotely managed via Telnet and traffic can be sent to the router in the case of interVLAN routing.

IP Route 0.0.0.0 0.0.0.0

In Figure 5.7 below, the stub network has only one way for the traffic to go to reach several different networks. To configure several static routes would be a long-winded way of achieving what could be done with one command (presuming that 192.168.1.2 is the next-hop IP address).

```
Router(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

The 0s indicate any network and any subnet mask. This means that any traffic to anywhere goes via the next-hop address 192.168.1.2.

You could have specified an exit interface instead of the next-hop address:

```
Router(config)#ip route 0.0.0.0 0.0.0.0 Serial0
```

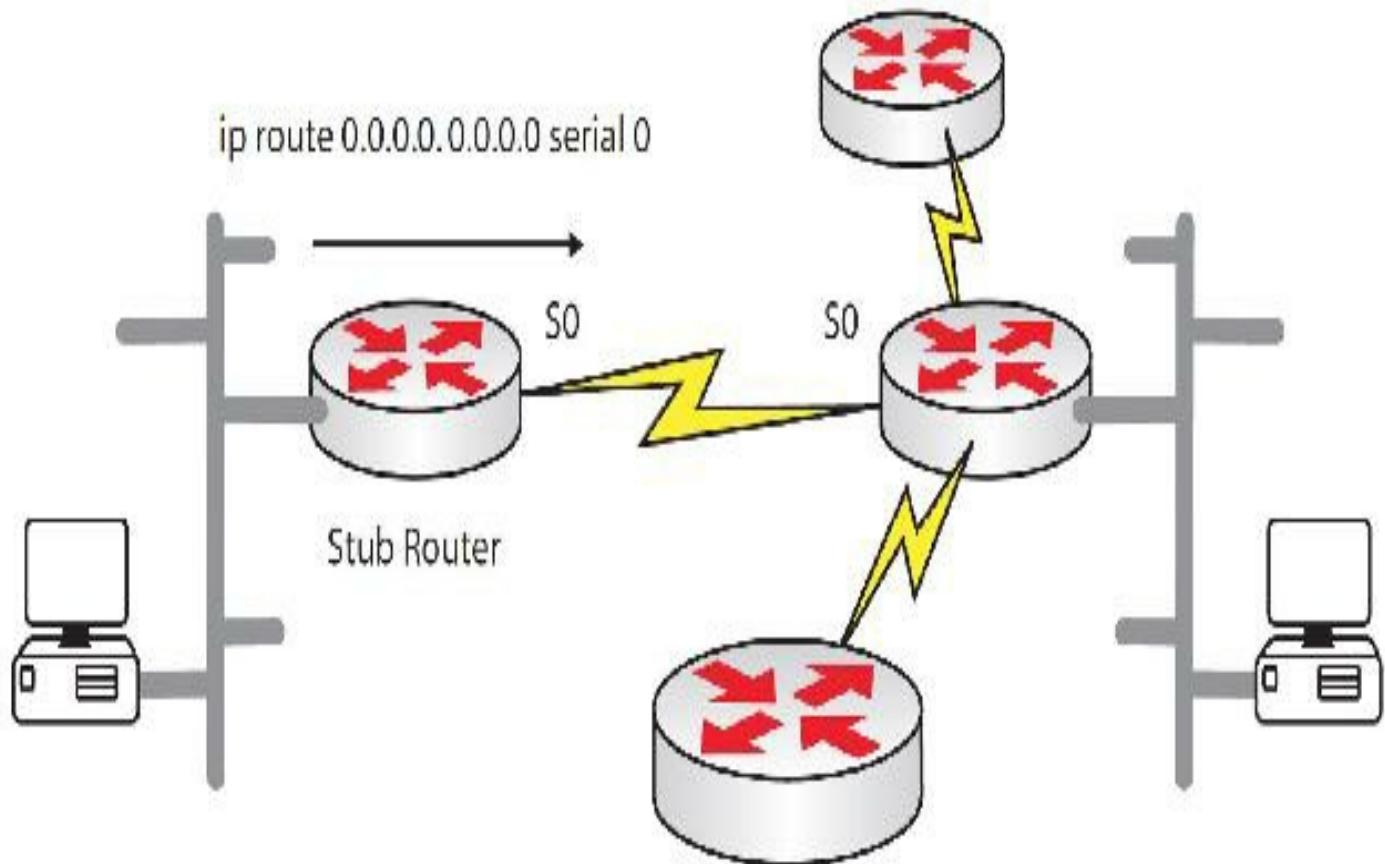


FIG 5.7 – All traffic will leave the stub router through Serial 0

You could even have some static routes configured and then a default route at the end. A practical use may be that you work in a small office and any unknown traffic will be passed to a larger router at your head office that can make all the routing decisions for your router.

IP Default-Network

The ip default-network command is used on routers that have IP routing enabled and works best alongside routing protocols. When this command is used, routes to that network are considered the gateway of last resort by the router. A static route will only affect traffic leaving the configured router. With the ip default-network command, each router will learn about the default network via a routing protocol and send traffic to that address if there is no other route in the routing table (as opposed to dropping it).

Administrative Distance

Administrative distance (AD) is used by the router to determine how believable a route learned from a routing protocol is. If a route is learned via more than one routing

protocol, then the protocol (route source) with the lower administrative distance is preferred. The most trusted routes start at 0 and go all the way up to an unknown route valued at 255, which will never be used.

Table 5-2: Administrative distances

Route Source	Default Distance
Connected interface	0
Static route (next-hop IP address)	1
Enhanced IGRP summary route	5
External BGP	20
Internal enhanced IGRP (EIGRP)	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
External enhanced IGRP	170
Internal BGP	200
Unknown	255

In the exam Cisco might try to catch you out by asking which route would be preferred from a list learned via OSPF, EIGRP, static, or connected. Looking at Table 5-2 above, you already know that the router will prefer a connected network over a static route, a static route over EIGRP, and so on.

Memorize the administrative distances! The ones most commonly referred to are in bold, but you need to know them all for the exam. These are easy marks in the exam that would be a shame for you to miss out on.

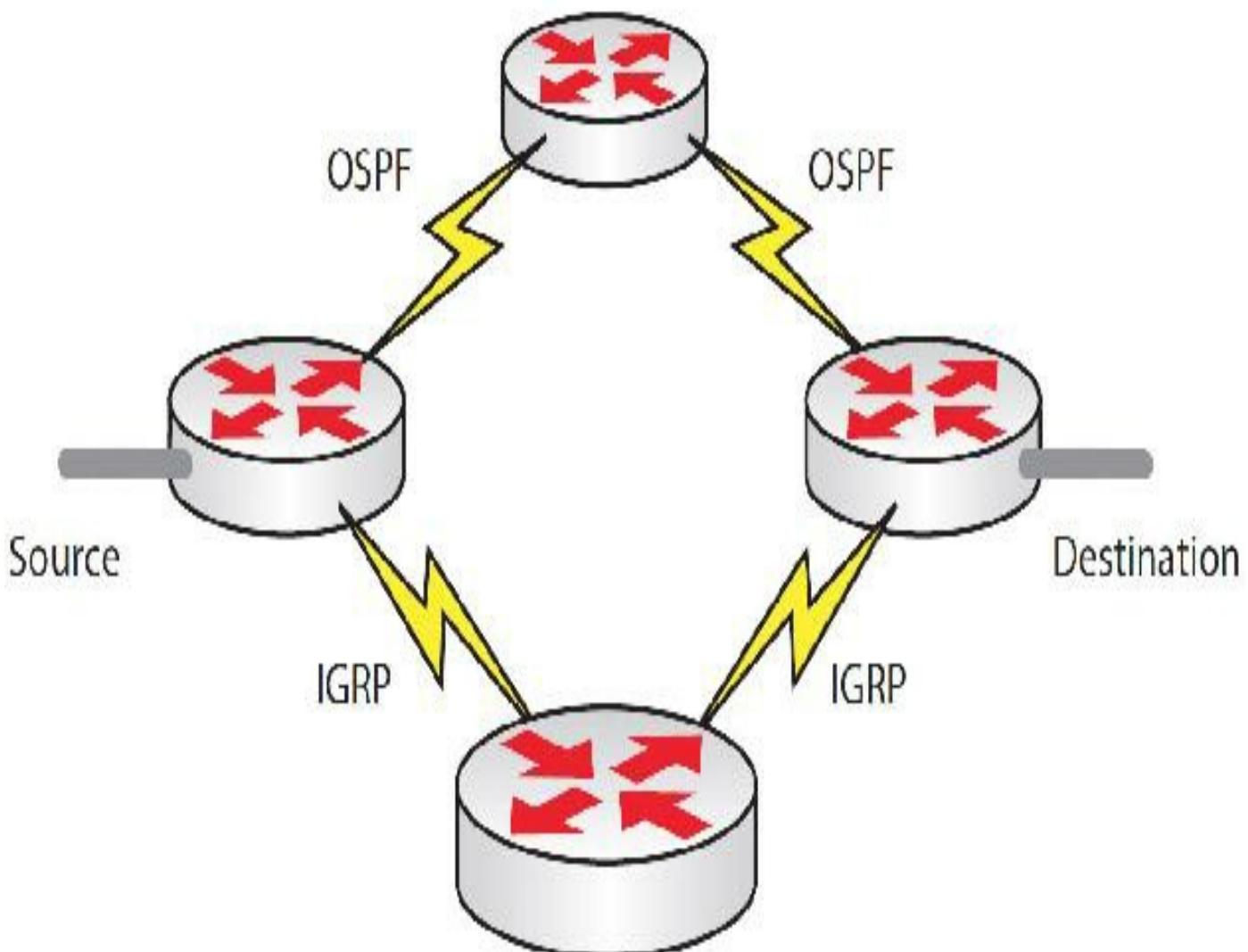


FIG 5.8 – Which path will the traffic take?

Classful/Classless

Remember from previous sections that you can use VLSM to break down your network into subnets? Some routing protocols do not understand VLSM and presume that you are using the default subnet mask. These types of protocols are known as classful. They do not send any subnet information with their routing updates because, as far as they are concerned, variable length subnet masking does not exist. Classful protocols do not have an option of sending the subnet mask when they advertise networks, for example, RIP and IGRP. Classful routing protocols are rarely used in networks today.

Mini-lab – Classful and Classless Routing Protocols

Configure the IP addresses as per the network in Figure 5.9 below. Add a Loopback interface to R1 and add an address from the 172.16.1.0/24 network to it.

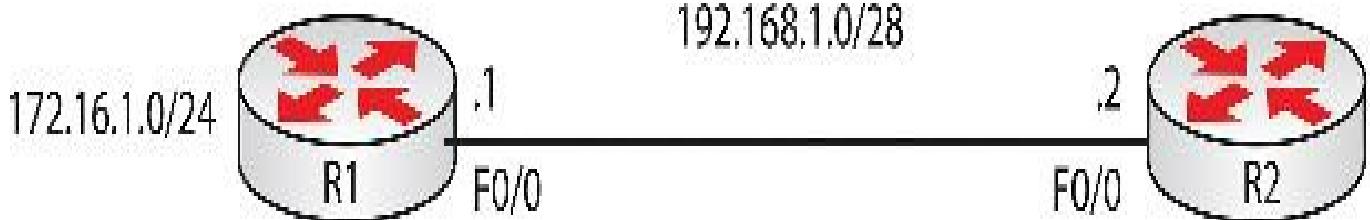


FIG 5.9 – Mini-lab: Classful and Classless Routing Protocols

We'll start with RIP, which is a classless routing protocol. Here is the configuration for R1. It will be the same for R2 but you need the .2 IP address and you won't need to advertise the 172 network:

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#int f0/0

R1(config-if)#ip add 192.168.1.1 255.255.255.240

R1(config-if)#no shut

R1(config-if)#int lo0

R1(config-if)#ip add 172.16.1.1 255.255.255.0

R1(config-if)#router rip

R1(config-router)#network 172.16.1.0

R1(config-router)#network 192.168.1.0

If you have configured R2 correctly you should be able to see the network on the Loopback of R1 advertised:

R2#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

[output truncated]

Gateway of last resort is not set

R 172.16.0.0/16 [120/1] via 192.168.1.1, 00:00:01, FastEthernet0/0

192.168.1.0/28 is subnetted, 1 subnets

C 192.168.1.0 is directly connected, FastEthernet0/0

Since RIP is classless, it will not be aware of VLSM; with this in mind it has no need to advertise any subnet mask information. The output below shows a debug ip rip command running on R2. Notice that the networks are advertised via broadcast and

there is no subnet mask information:

```
*Mar 1 00:05:27.819: RIP: sending v1 flash update to 255.255.255.255 via  
FastEthernet0/0 (192.168.1.1)
```

```
*Mar 1 00:05:27.819: RIP: build flash update entries
```

```
*Mar 1 00:05:27.819: network 172.16.0.0 metric 1
```

Classless protocols do not make any such presumptions and always advertise the subnet mask with the routing update. For this reason, you are free to use subnet masks other than the default mask of Class A, B, or C networks. Examples of classless protocols are RIPv2, OSPF, EIGRP, and ISIS.

Now add the command below to **both routers**:

```
R2(config)#router rip  
R2(config-router)#version 2  
R2(config-router)#end
```

Here is the debug again; however, I've updated the routing protocol to RIPv2. Now you can see that updates are multicast and subnet information is also included:

```
R2#  
*Mar 1 06:15:40.406: RIP: received v2 update from 192.168.1.1 on FastEthernet0/0  
*Mar 1 06:15:40.406: 172.16.0.0/16 via 0.0.0.0 in 1 hops  
R2#  
*Mar 1 06:15:42.906: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0  
(192.168.1.2)
```

The 172 network will still show as /16 in the routing table, and in the debug above you can see that the mask of /24 has been summarized to the default of 16. Some routing protocols will automatically do this and we will cover this in more detail in the EIGRP section. I'll disable the automatic summarization process on R1:

```
R1(config)#router rip  
R1(config-router)#no auto-summary  
R1(config-router)#end
```

Now R2 receives the /24 mask:

```
R2#  
*Mar 1 06:18:33.586: RIP: received v2 update from 192.168.1.1 on FastEthernet0/0  
*Mar 1 06:18:33.586: 172.16.1.0/24 via 0.0.0.0 in 1 hops
```

As I said, we'll cover the automatic summarization part in detail later. I wanted to demonstrate the difference between classful and classless protocols in this mini-lab.

[END OF MINI-LAB]

Dynamic Routing

Dynamic routing involves using a routing protocol on the router to discover which networks are present where and how best to get there. Rather than the time-consuming task of having to configure a static route to every network on every router, the network administrator configures the router with the networks to advertise for the chosen dynamic routing protocol. After that the routers automatically carry out the task of finding the networks and working out how to reach them.

Even for the network in Figure 5.10 below, which is fairly small and simple, each router will need to be configured with several static routes, possibly requiring regular manual updates. You will also have to configure different administrative distances for the alternative paths to reach the various networks.

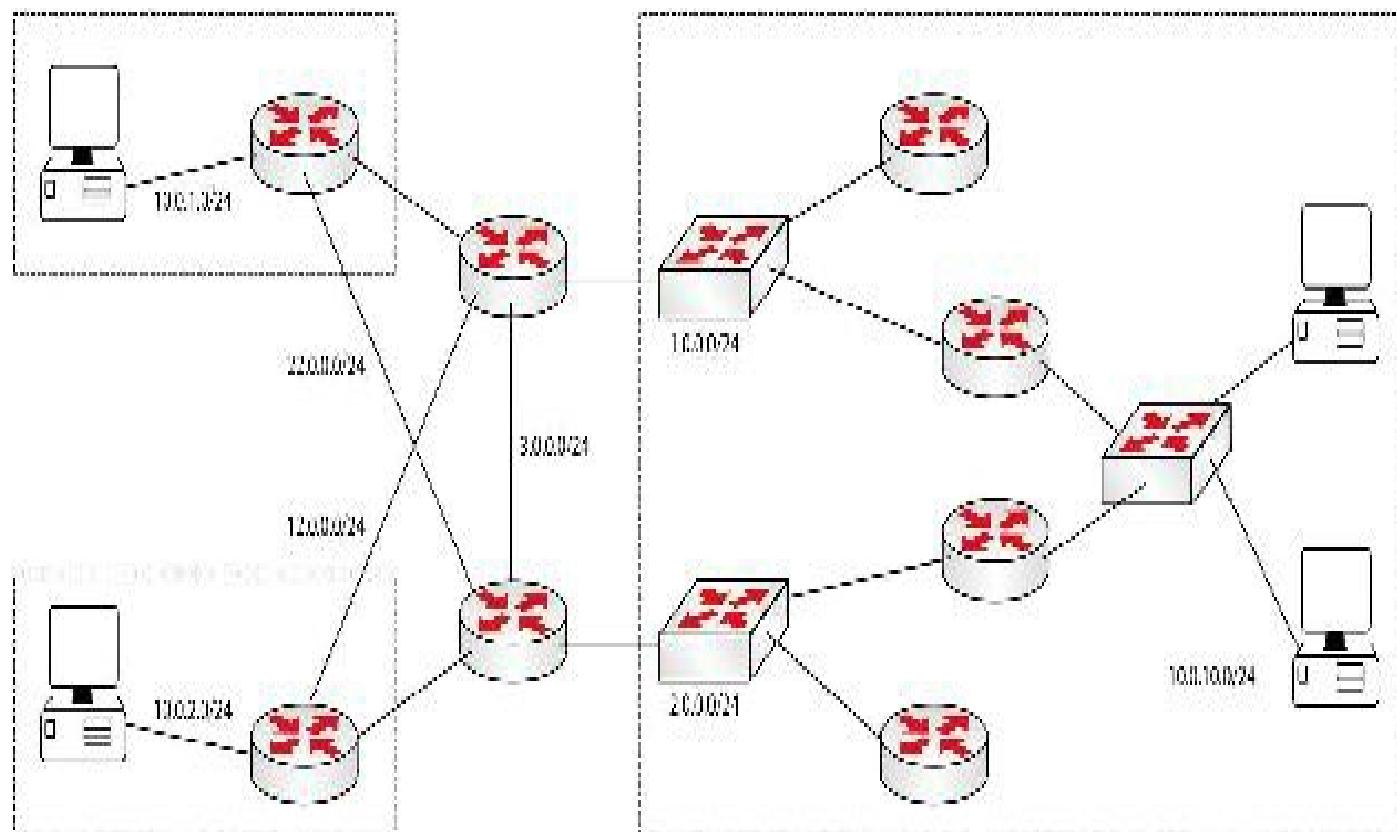


FIG 5.10 – Managing multiple static routes is challenging

Dynamic routing is a far easier way to manage a medium to large network; this is because the routing takes place automatically after the initial setup and configuration. Static routing is, however, still useful in small networks that change very little.

Dynamic routing protocols are divided into two types:

- Distance vector
- Link state

Both do the same job but achieve it in different ways. A hybrid of the two is called EIGRP, which we will address in the ICND2 section. All routing protocols have certain roles to perform, such as:

- Advertise connected networks
- Learn other networks
- Deal with faulty or lost routes
- Determine the best loop-free path for traffic to take

We will dig into the individual metrics in more detail when we cover specific routing protocols.

Metrics

Routing protocols feature a mechanism to deal with the case that multiple routes to the same destination network are learned. Metrics run on the learned routes to choose the best path to the network. There must be a ranking system of some sort so that the preferred path is placed into the routing table.

RIP, for example, uses hop count, while EIGRP uses the lowest bandwidth and the total delay; it also builds a table of the next-best routes should the main route fail. The metrics used by routing protocols include:

- Bandwidth – the fastest link speed is chosen
- Hop count – a hop is another router, so three routers away equals three hops
- Load – the amount of traffic using the link; the lowest loaded link is chosen
- Delay – the amount of time it takes a packet to traverse a path
- Reliability – the likelihood of a link failing based on errors or interface resets
- Cost – no fixed definition of this value; it reflects a less or more preferred route

Of course, this is a simplistic view of each metric. Some protocols use one or more of the metrics above by default but you can add others. Some protocols accept metrics such as bandwidth but they are displayed purely for cosmetic reasons. We will revisit some of the metrics above as we progress through the guide.

Distance Vector Protocols

Distance vector protocols (also referred to as Bellman-Ford algorithms) run an algorithm to calculate the best path to reach routes advertised by other routers. They then update their routing tables, allowing them to communicate changes in the network topology. Distance refers to the measurement used to calculate the distance to the route

in the table and vector is the direction the traffic to the route should take. Figure 5.11 below demonstrates a simplified view of how distance vector protocols operate:

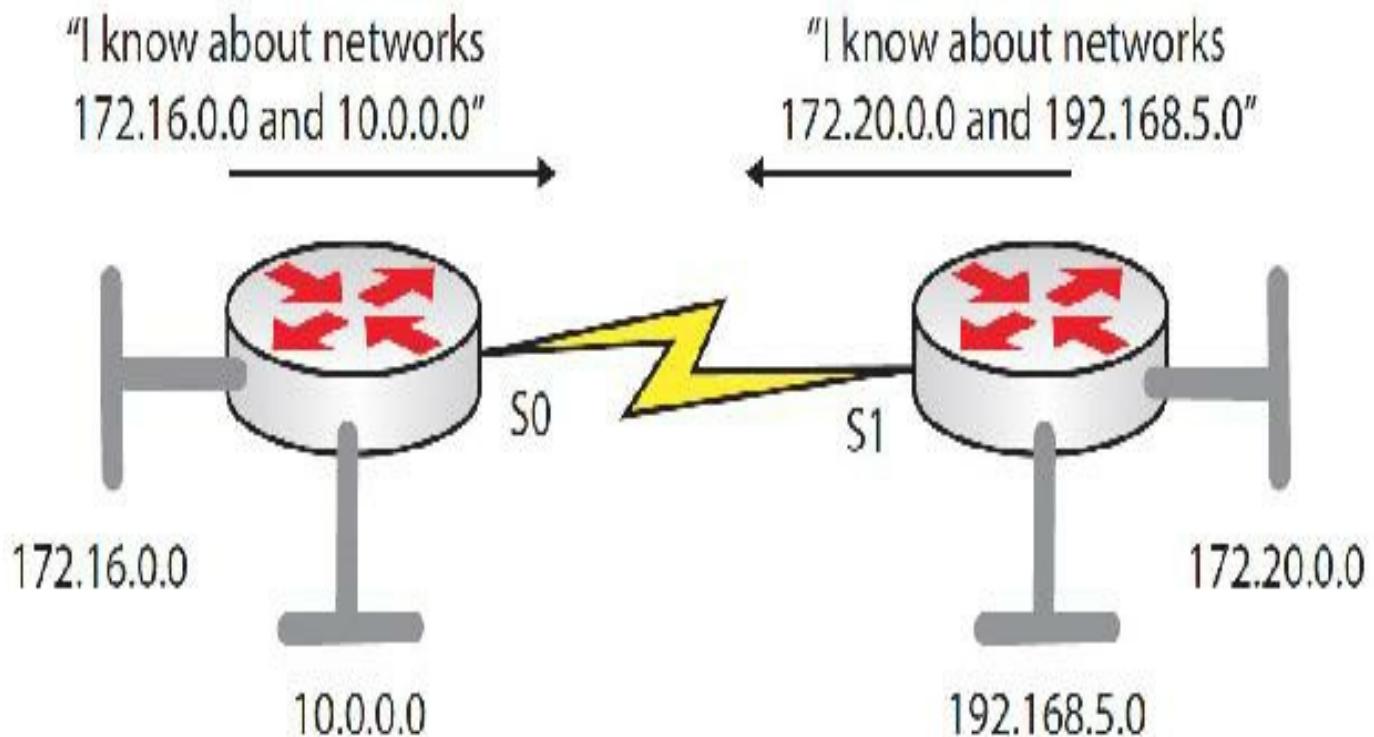


FIG 5.11 – Routers exchange routing tables

Distance vector routing is a property of certain routing protocols that build an internal picture of the topology by periodically exchanging full routing tables and metrics between neighbor devices. When the network first converges, routers first discover their attached networks before exchanging routing tables with all other routers in the domain. An illustration of how this occurs is shown in Figure 5.12 below:

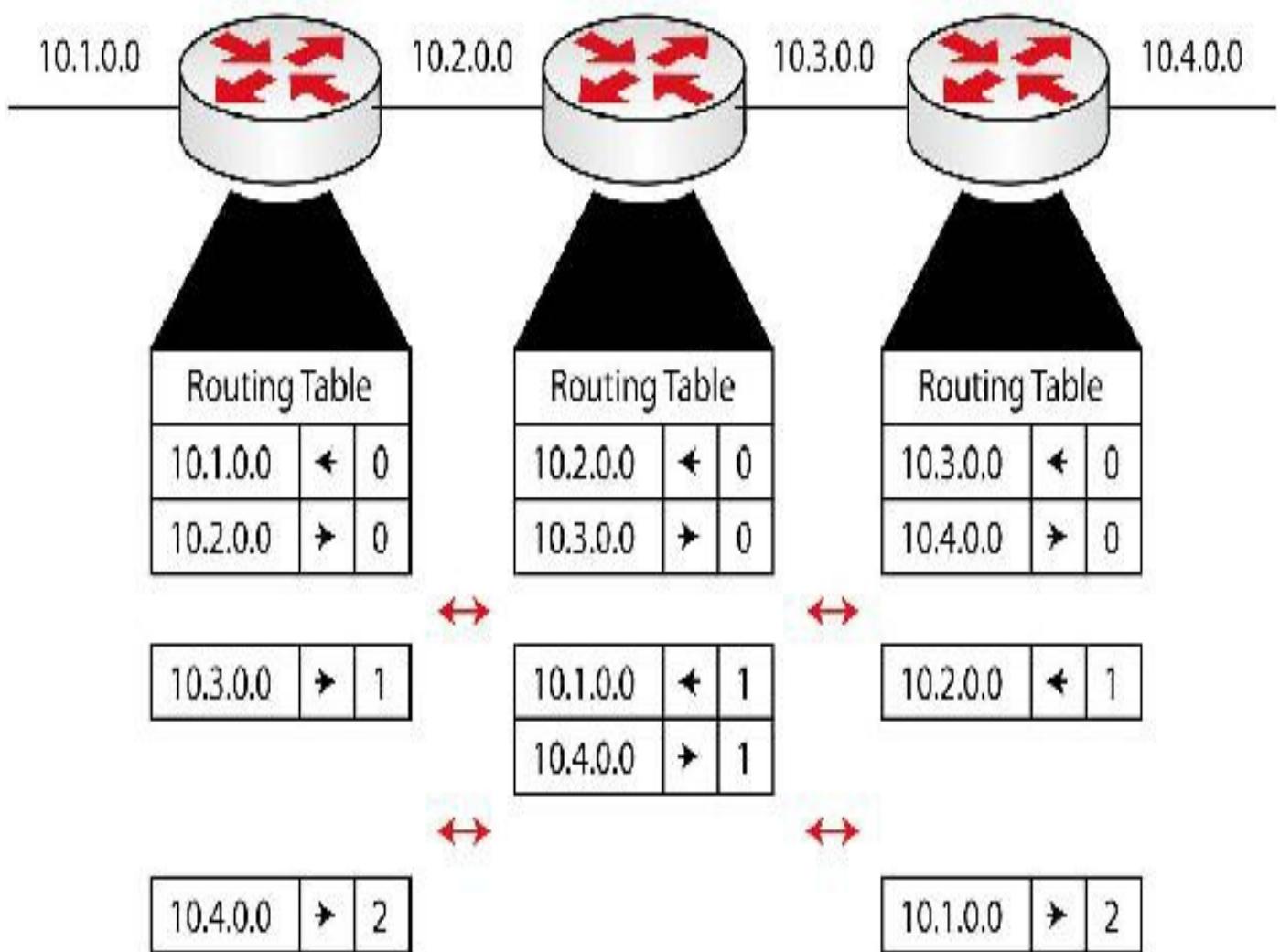


FIG 5.12 – Routing table exchanges

The main difference between distance vector routing protocols and link state protocols is the way they exchange routing updates. Distance vector protocols function using the routing-by-rumor technique, where every router relies on its neighbors to maintain correct routing information, meaning the entire routing table is sent periodically to neighbors. This is illustrated in Figure 5.13 below. The interval varies from protocol to protocol. You will see in the show ip protocols output below that RIP sends updates every 30 seconds. Of course, this would cause you issues in a very large enterprise network.

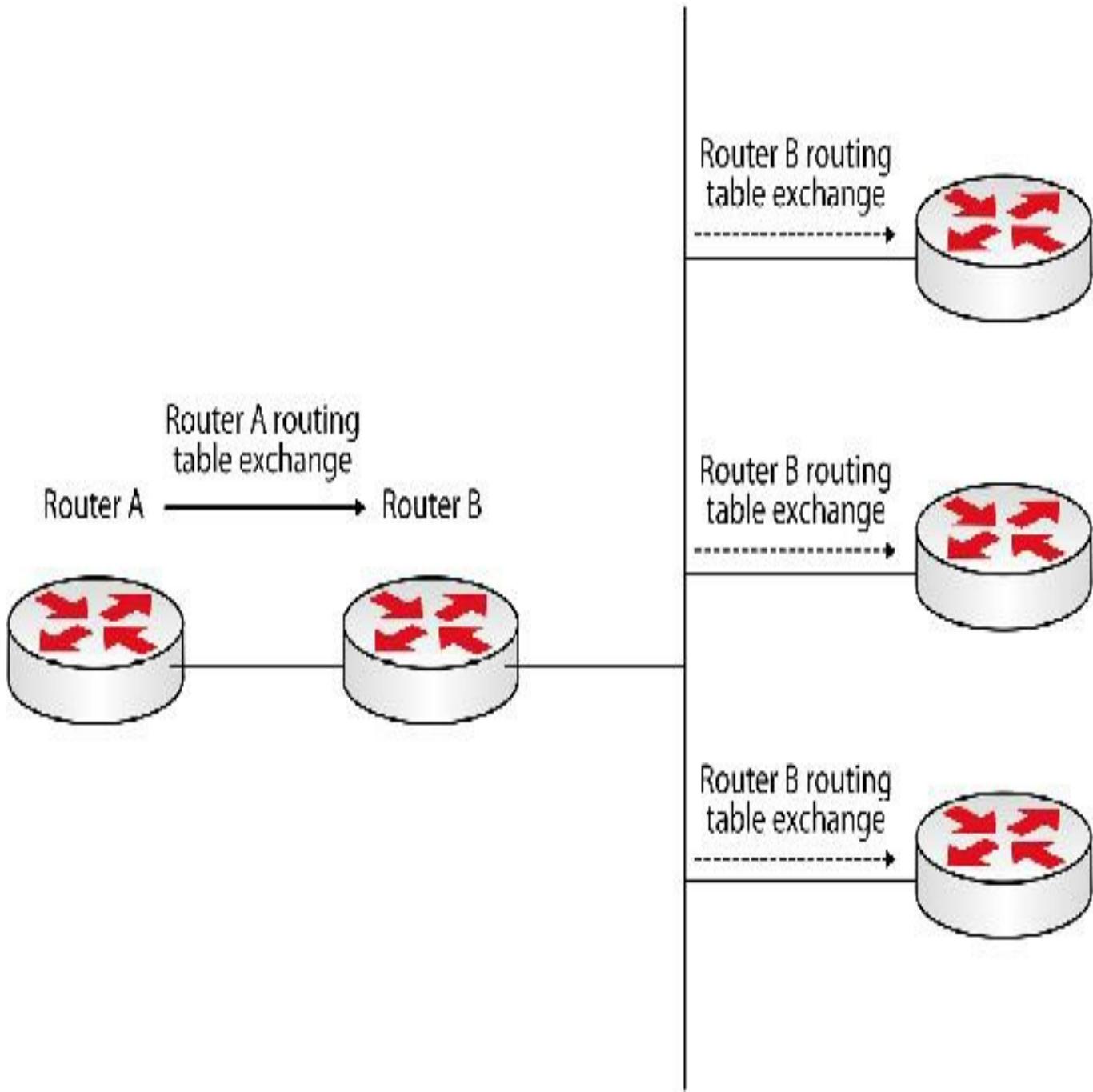


Figure 5.13 – Distance vector routing protocol behavior

The most important advantage of distance vector routing protocols is that they are easy to implement and maintain. The downside is mostly the convergence times. A converged network is a network in which every router has the same perspective of the topology. When a topology change occurs, the routers in the respective area propagate the new information to the rest of the network. Considering this is done on a hop-by-hop basis (every router passes its full updated routing information to each neighbor), network convergence won't be established until a significant amount of time has passed.

Besides the slow convergence downside, distance vector protocols are also bandwidth-

intensive. This happens especially in large networks, where routing tables can be of considerable size. Considering these aspects, distance vector protocols are recommended only in small enterprise network implementations.

An example of a distance vector routing protocol still used in modern networks is RIPv2 (Routing Information Protocol v2, described in RFC 2453). RIPv2 uses hop count as a metric for path selection, with a maximum hop count of 15. RIPv2 updates are sent using multicast by default, although they can be configured as unicast or broadcast, and, unlike its predecessor (RIPv1), RIPv2 permits VLSM in the network as you saw in the debugs in the previous mini-lab. Unfortunately, RIP has been dropped from the CCNA syllabus, but I do recommend that you take the time to learn how to configure it because you can learn a lot by doing so.

Devices receive routing information from their neighbors and pass it on to other neighbors. RIP repeats this process every 30 seconds. The downside in this scenario is that when the network is stable and there are no changes in the topology, RIP still sends its routing table every 30 seconds, which is not very efficient and is somewhat pointless if there have been no changes, and, of course, it wastes bandwidth.

R1#show ip protocols

Routing Protocol is “rip”

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Sending updates every 30 seconds, next due in 15 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

Redistributing: rip

Neighbors in the context of routing protocols means routers sharing a common data link or common configuration information, which enables them to share routing information.

If a router that uses a distance vector protocol has inaccurate information, that information will be propagated throughout the entire network. Moreover, distance vector routing protocols are prone to major problems, including routing loops.

Distance Vector Problems

Distance vector protocols usually use low amounts of network resources (e.g., router CPU and network bandwidth). They are also easy to maintain on smaller networks and simple to configure. However, there are some drawbacks to using these protocols, especially slow convergence on larger networks due to periodic updates of full routing

tables. These drawbacks, which include routing loops and counting to infinity, must be addressed by either the network administrator or the protocols themselves.

Routing Loops

Routing loops can prove catastrophic for a network. They consume network bandwidth and resources, cause loss of information, and can take time to locate and troubleshoot. Perhaps now you can see why static routes are so popular. Routing loops are caused when network convergence is slow and routers are advertising inaccurate routing tables. A router can lose connectivity to a network, but other routers are not yet aware of the problem.

In Figure 5.14 below, Router D advertises network 172.20.0.x to Router B. Router B tells Router C, which tells Router A that it knows how to get to network 172.20.0.x. For some reason, network 172.20.0.x goes down. Router D informs Router B and Router B stops routing traffic for this network to Router D.

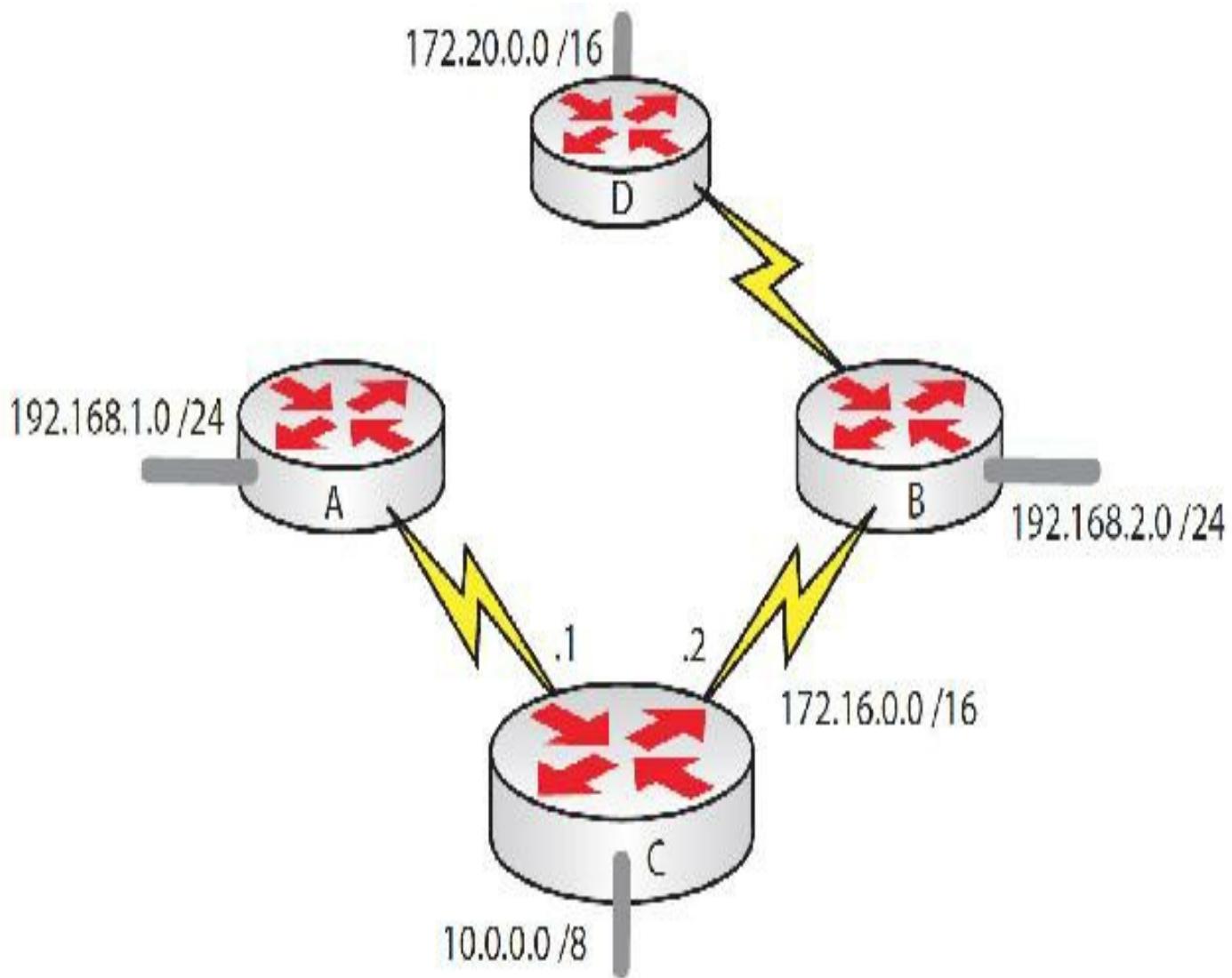


FIG 5.14 – Routing loops can happen quickly

Eventually, the message is passed to Router C but, unfortunately, during this time Router A thinks that 172.20.0.x is still up. Therefore, Router A advertises that it can reach 172.20.0.x to Router D (via Routers C and B) based on the previous entry it had learned before 172.20.0.x went down. Now Routers B, C, and D all believe that Router A has a path to network 172.20.0.x and route traffic to it.

When Router A receives the traffic, it routes it to Router C, thinking that it is the path to take based on the old routing entry in the routing table. When this happens the hop count keeps increasing every hop the packet takes. There is a Time to Live field in the IP packet, as you know, and this will count down from 255, but it still means that a looping packet will pass through 255 routers before being destroyed. We'll cover TTL shortly. The routing loop thus consumes bandwidth and router CPU cycles, degrades convergence, and causes issues with routing updates.

Counting to Infinity

Routing loops can lead to another problem known as counting to infinity, where a packet travels around the network looking for a particular IP address but never reaches its destination. This is normally because the network is down, but the sending router was not aware of this fact. Take the network in Figure 5.15 below as an example:

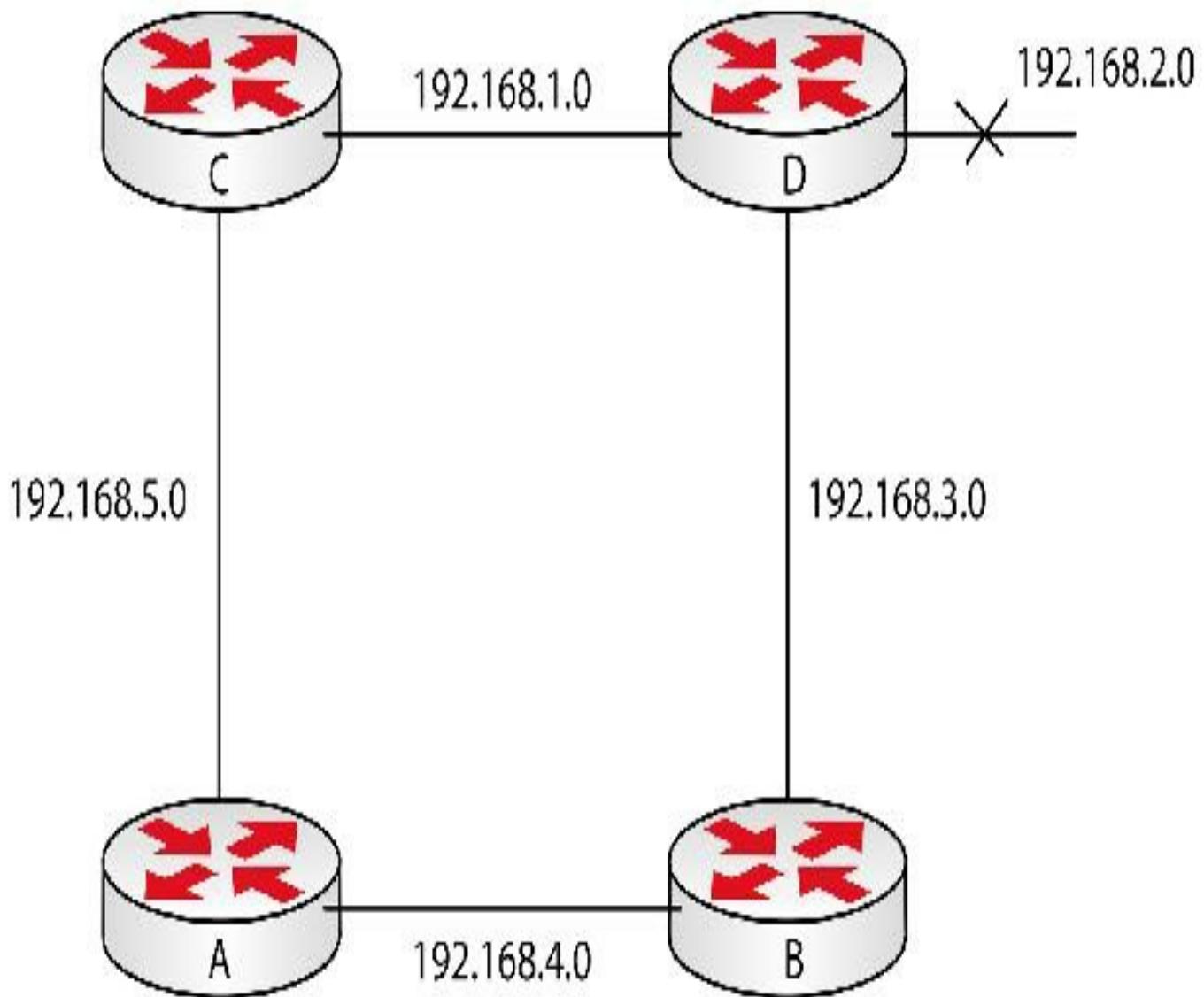


FIG 5.15 – Counting to infinity

Router D is advertising that it can reach the 192.168.2.0 network. When the link to that network fails, Router D will send the relevant routing update. Router A, however, has a three-hop route to 192.168.2.0 in its table and this information is forwarded to Router B, which in turn informs Router D that an alternative (four-hop) path is available. This alternative route is forwarded from Router D to Router C, which advertises that it has a five-hop route to 192.168.2.0. Router C passes this route to Router A, which notes that the path cost has increased but still adds the route to its routing table, and so on.

Solving Distance Vector Problems

Maximum Hop Count

As the packets traverse the network the information can become out of date, or sometimes a route that was valid when the packet was sent is now invalid. Certain safety measures have been put into place to prevent this from becoming a problem.

RIP has a maximum hop count of 15 hops; at the fifteenth hop the packet is dropped. This means that (for RIP) a packet can travel across a network with only 15 routers and on each router the Time to Live (TTL) value is incremented by one. This also means that RIP can only be used in a network with a maximum diameter of 15 routers.

```
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.9 (224.0.0.9)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0xc0 (DSCP 0x30; Class Selector 6; ECN: 0x00; Not-ECT (No))
Total Length: 52
Identification: 0x0000 (0)
Flags: 0x00
Fragment offset: 0
Time to live: 2
Protocol: UDP (17)
Header checksum: 0x1647 [correct]
```

FIG 5.16 – TTL packet capture

If there was a routing loop, at least the looping packet would travel only 15 hops before dying and the routing loop begins all over again. Note that the maximum hop count does NOT solve routing loop problems. It only ensures that the count to infinity problem is fixed.

You could run a packet capture and see the TTL field decrementing on each router. From RFC 1812, Requirements for IP Version 4 Routers, “When a router forwards a packet, it MUST reduce the TTL by at least one.” You would see the first router receive the packet with a TTL value of 255 and forward it with a value of 254, and so on.

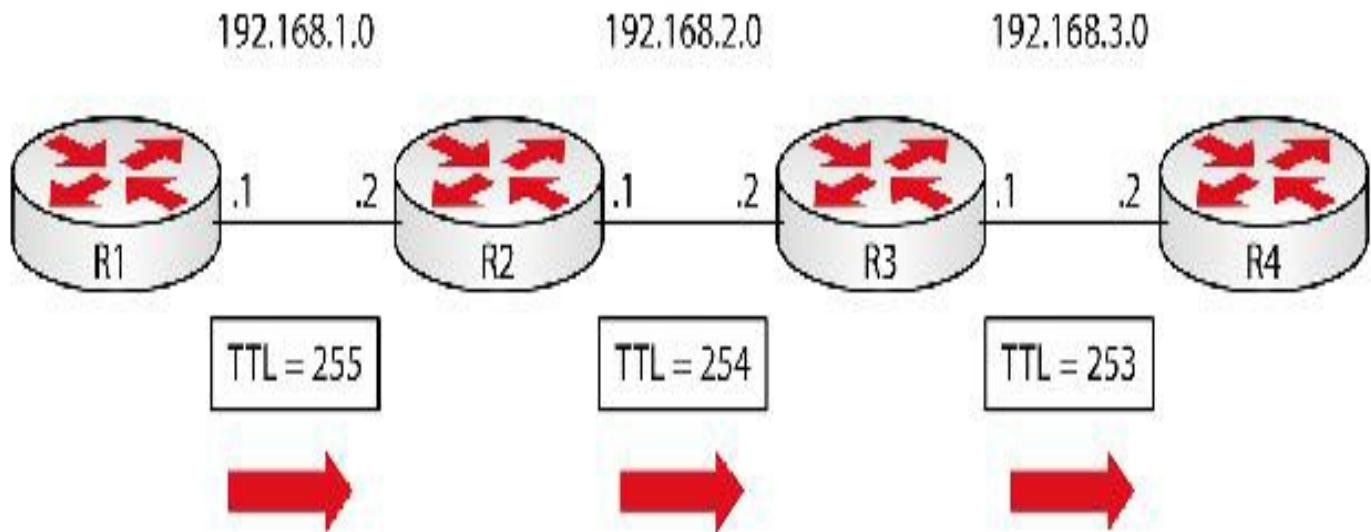


FIG 5.17 – TTL value decremented each hop

Here is a capture of a ping packet from R1 arriving at R4:

9	21.878936	192.168.1.1	192.168.3.2	ICMP
10	21.889813	192.168.3.2	192.168.1.1	ICMP
11	21.941310	192.168.1.1	192.168.3.2	ICMP
12	21.951862	192.168.3.2	192.168.1.1	ICMP

Header length: 20 bytes

- ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Total Length: 100)
- Identification: 0x0000 (0)
- ▷ Flags: 0x00
- Fragment offset: 0

Time to live: 253

Protocol: ICMP (1)

- ▷ Header checksum: 0x3845 [correct]
- Source: 192.168.1.1 (192.168.1.1)
- Destination: 192.168.3.2 (192.168.3.2)

FIG 5.18 – TTL packet decremented

Split Horizon

When a route is received by a router, it automatically adds this route to its routing table. The route is then advertised out of the router's interfaces so that other routers in the network hear about the route.

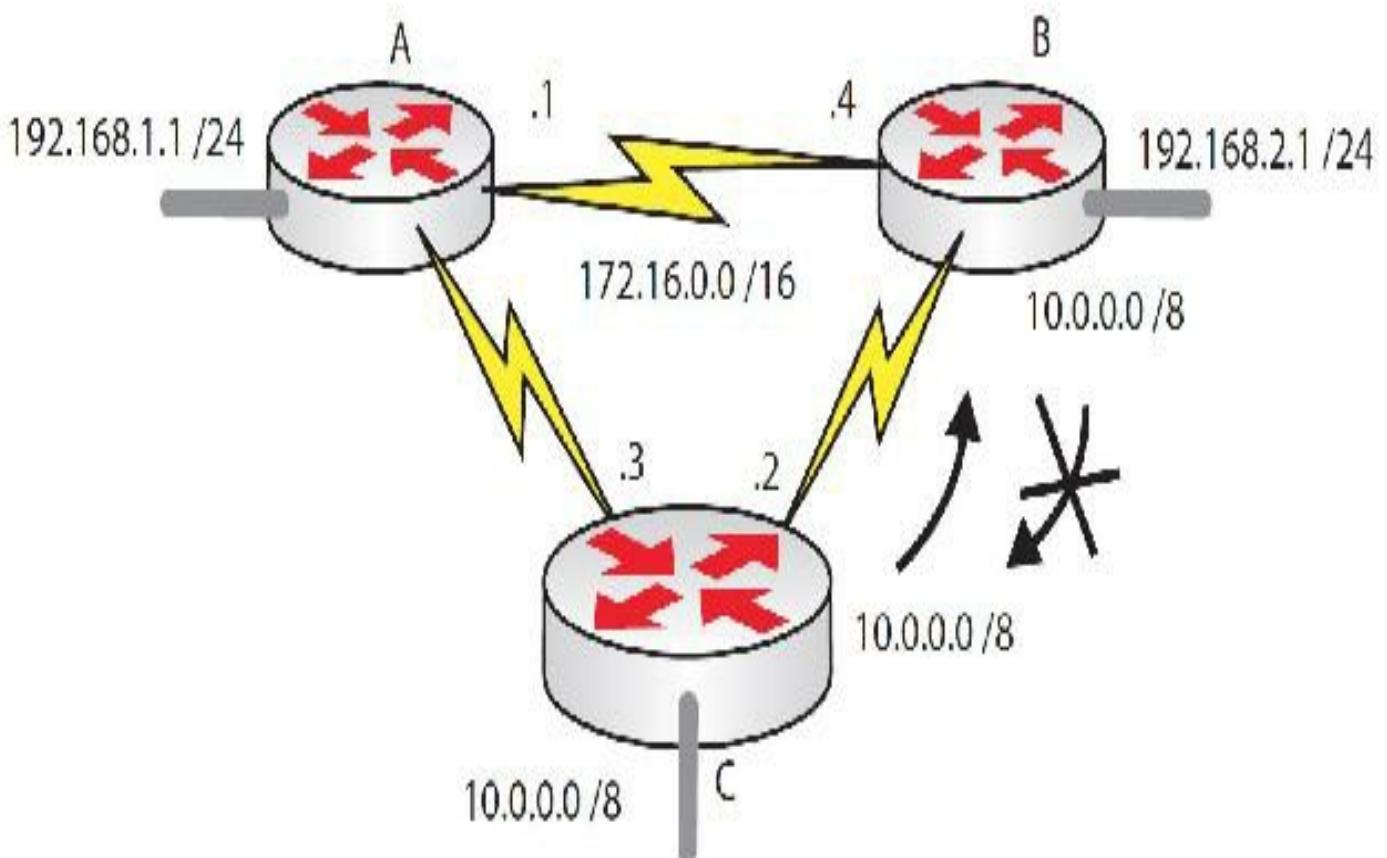


FIG 5.19 – Split horizon prevents sending a route out of the interface it was learned on

A problem occurs when the route is advertised out of the same interface that the router learned the route from. Now there is some confusion as to where that network actually is. In Figure 5.19 above, Router C tells Router B to come over the Serial link if it wants to reach 10.0.0.0/8. Router B would normally send this information out of every interface, but this would cause further confusion because Router C is where the network is directly attached. The split horizon rule prevents this scenario from happening because any information or route learned on an interface from a router will not be advertised back on the same interface to the same router. In this case, Router B will not advertise network 10.0.0.0 to Router C via the Serial interface (when it learns the route via Router A). Thus, split horizon prevents the possibility of routing loops.

Hold-Down Timers

Hold-down timers are used to prevent route update messages from being sent or received (for a period of time known as the hold-down time) and from reinstalling a route that may actually be down. Sometimes, before an interface fails completely it goes up and down rapidly. This is known as link flapping, and every time the interface comes up the router advertises the network as up, and when it goes down it advertises that it is no longer available.

When hold-down timers are in use, the router receives the message that the network is

down and begins a timer. If it receives a message that the network is up before the hold-down timer expires, it will ignore the update. The theory is that by the time the timer expires, the flapping interface will have stabilized.

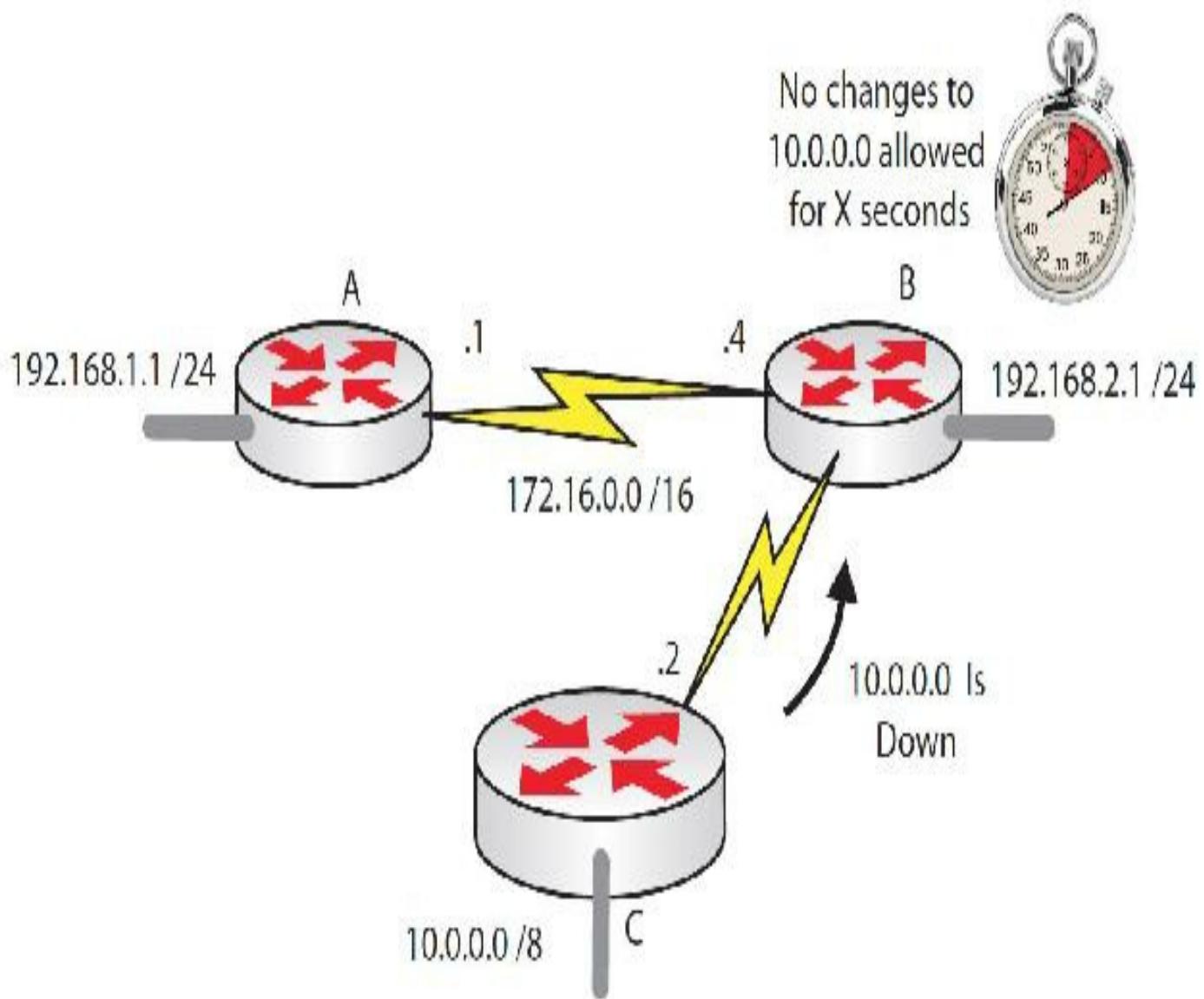


FIG 5.20 – Hold-down timers prevent routing table updates for X seconds

What can happen is that while the hold-down timer is active, an update with a better metric is received from a different router. This route will then be accepted at the end of the hold-down timer and the hold-down timer ends. Hold-down timers can be tuned or modified but this should not be done without proper guidance and expertise. The downside of very high hold-down timer values is that network convergence takes longer, while low values can lead to fast convergence but this becomes ineffective in cases of links flapping.

R1#show ip protocols

Routing Protocol is “rip”

Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Sending updates every 30 seconds, next due in 15 seconds
Invalid after 180 seconds, **hold down 180**, flushed after 240
Redistributing: rip

Route Poisoning

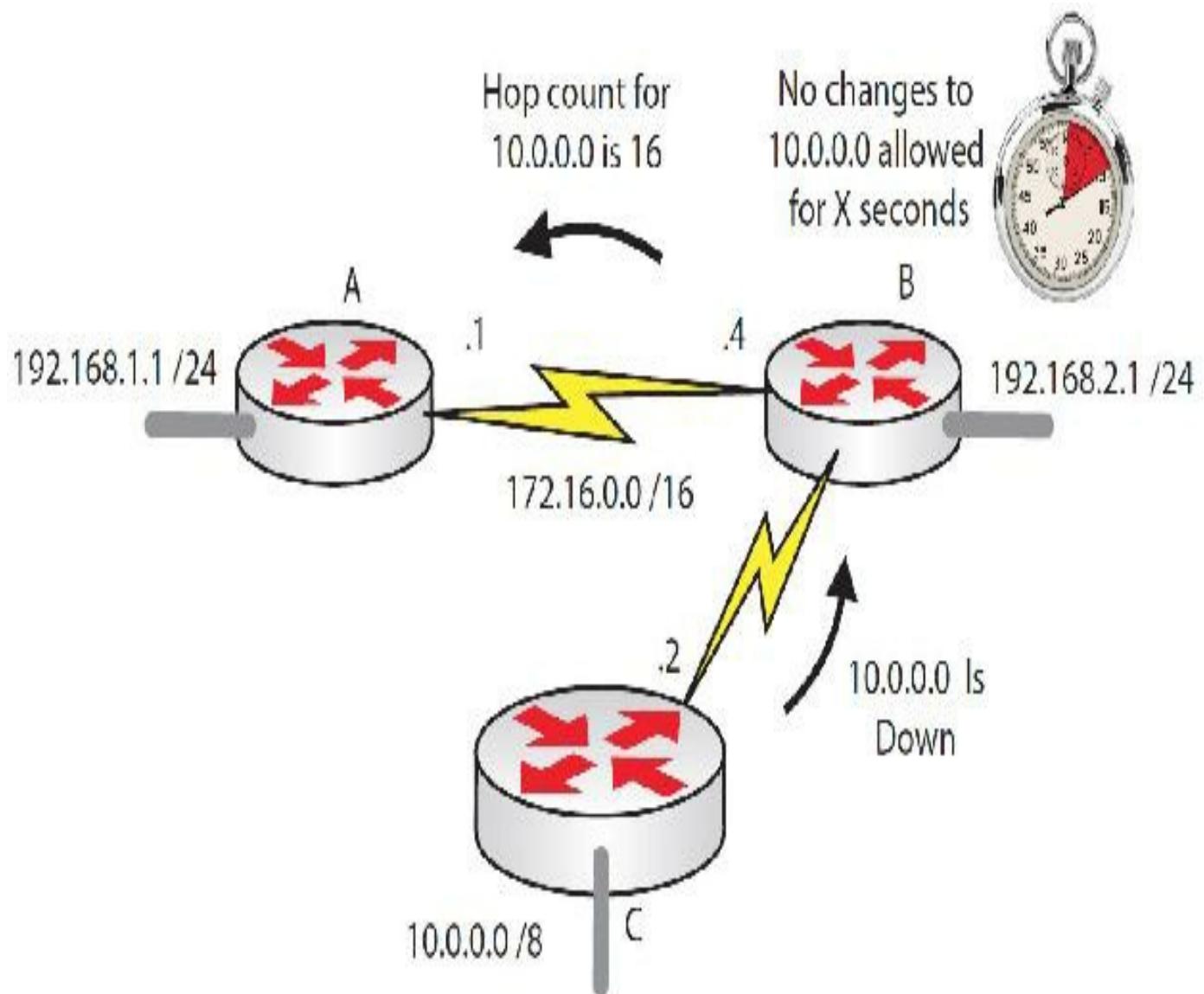


FIG 5.21 – Route poisoning sets the hop count as unreachable on this RIP network

A form of split horizon uses a method known as route poisoning (or poison reverse). Route poisoning allows a router to set the distance to a network as infinity to allow the rest of the network to converge without receiving inaccurate updates. When used with hold-down timers, route poisoning can prove to be a reliable solution to preventing loops.

In Figure 5.21 above, RIP is the protocol in use, so Router B sends an update to Router A telling it that 10.0.0.0 is 16 hops away (i.e., unreachable).

Triggered Updates

Distance vector routing protocols rely on regular updates from neighbor routers; these updates occur after the expiration of update timers. Sometimes there are changes to the network topology, but the network will not learn about these changes until the next routing update is due. This can cause serious problems for the network in terms of slow convergence, resulting in lost information and lost time and resources for the business.

A triggered update is an unscheduled routing update sent out due to a change in the network topology, including an interface coming up or going down, a route becoming unreachable, or a new route being placed into a routing table. Each router that receives the triggered update passes it on to its neighbor routers.

Using triggered updates along with hold-down timers can help prevent bad routing information from being passed around the network, as well as reduce the likelihood of counting to infinity. Triggered updates happen along with regular updates.

Distance vector protocols are best suited for smaller networks with very few or, preferably, no redundant links. The two most popular distance vector protocols are RIP and IGRP, which were retired from the CCNA exam some time ago.

Autonomous Systems

Some protocols, such as EIGRP, use the concept of an autonomous system (AS). An AS number (ASN) is used to specify a routing domain that is independent from other routing domains. In EIGRP, routers must be in the same AS if they are to exchange routing information. Routers in different autonomous systems can also communicate but only with protocol redistribution (which is outside the scope of the CCNA exam).

Autonomous systems are self-contained networks normally administered by one or a single group of network administrators. They contain their own addressing scheme and normally feature just one routing protocol. Autonomous systems can be joined together by a border router, which will take care of any protocol translation and routing issues. These have to be configured manually by the network administrator.

When one company buys out another company, they also acquire their network. A common method used to join the networks together is to make two autonomous systems and join them with a border router. The border router is responsible for routing between the two different autonomous systems.

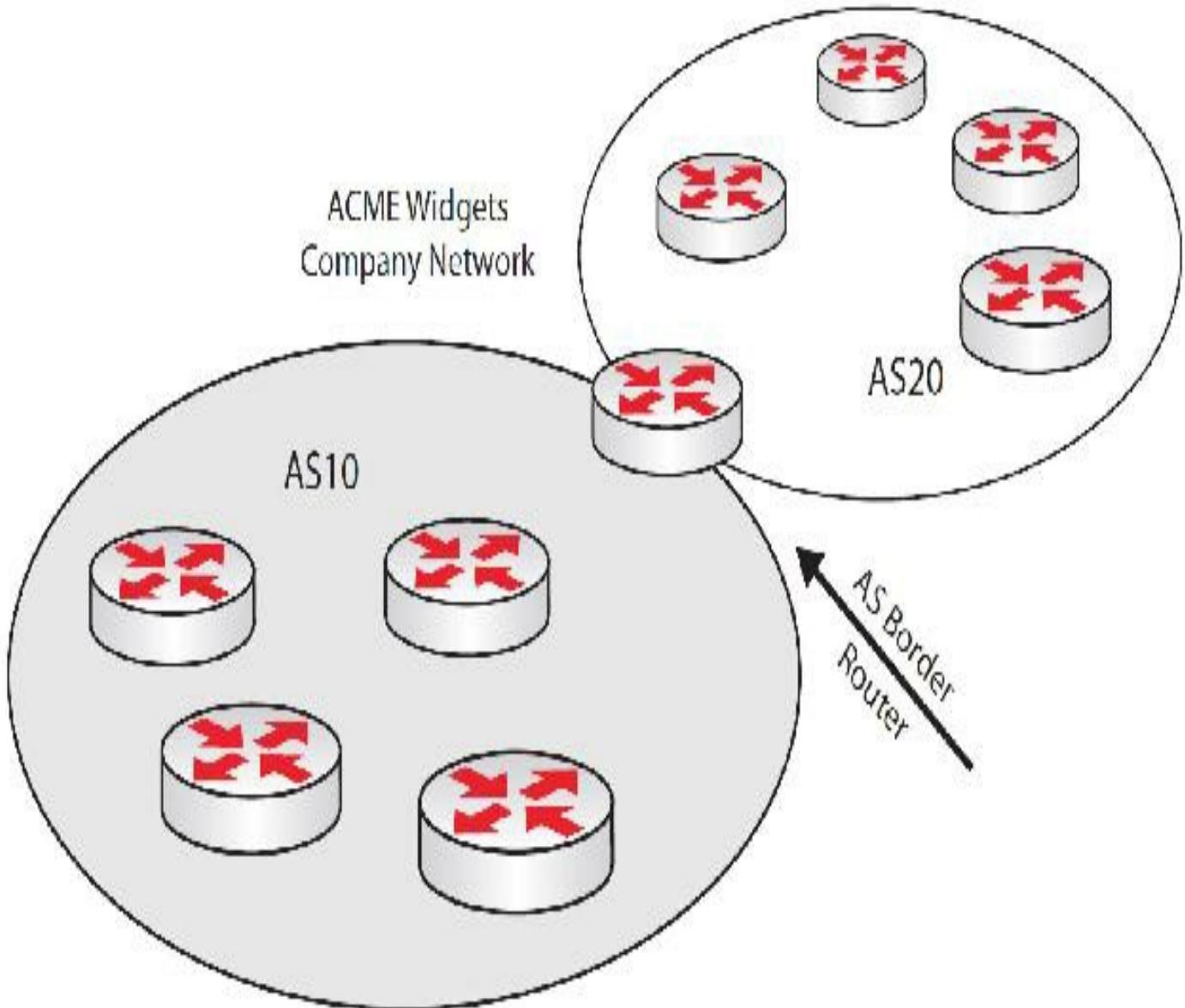


FIG 5.22 – Autonomous systems joining two networks

Passive Interface

When dynamic routing protocols (such as EIGRP, RIP, or OSPF) are configured, routing updates are sent and received over a router interface configured for that protocol. If you do not want an interface to participate actively in a routing protocol, you can make that interface passive. The `passive-interface` command is used to make an interface passive. In RIP, this means that the interface is still able to receive updates but is unable to send any routing updates. In EIGRP and OSPF, passive interface suppresses Hellos so neighbor relationships cannot be formed. This means that routers can neither receive nor advertise any routing updates.

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

```
R1(config)#router eigrp 10  
R1(config-router)#network 192.168.1.0  
R1(config-router)#passive-interface fast0/0
```

If you want routing updates to be sent to a certain host that is connected to Fast Ethernet 0/0, but do not want updates to be broadcast (or multicast to 244.0.0.9) out of it, you can use the neighbor command. This will send a unicast packet to the neighbor.

```
R1(config-router)#neighbor 192.168.1.10
```

The neighbor command can also be used on non-broadcast networks, such as Frame Relay (since multicast and broadcast is not supported), to allow unicast updates to pass across the media.

We will revisit passive interface again when reviewing routing protocols.

IP Unnumbered

There is a way for you to borrow an IP address from another interface and use it on a second interface. You may want to use this to save addresses or to make sure that the interface stays up. The ip unnumbered command is often used with Loopback interfaces to increase reliability because the Loopback interface cannot go down (it exists only in software).

```
R1#config t  
R1#(config)#interface FastEthernet0/0  
R1#(config-if)ip address 192.168.1.1 255.255.255.0  
R1#(config-if)no shutdown  
R1#(config-if)#interface Serial0/0  
R1#(config-if)#ip unnumbered FastEthernet0/0  
R1#(config-if)#no shutdown
```

Link State Protocols

Link state routing protocols do not route by rumor. The routing devices exchange information between them about their link states. Each device independently builds a map of the network (and does not rely on a map of a particular node) based on the link state information each router generates and propagates to the other routers.

Unlike distance vector routing protocols, where only the best routes are exchanged between neighbors, link state protocols flood information about their links to a specific area or to all the routers in the network. This way every router in the topology has detailed knowledge of the entire network. The routing decisions are made by applying

Shortest Path First (SPF), or Dijkstra's algorithm, to the information received from various sources. The result of this calculation consists of the shortest path to each destination in the network.

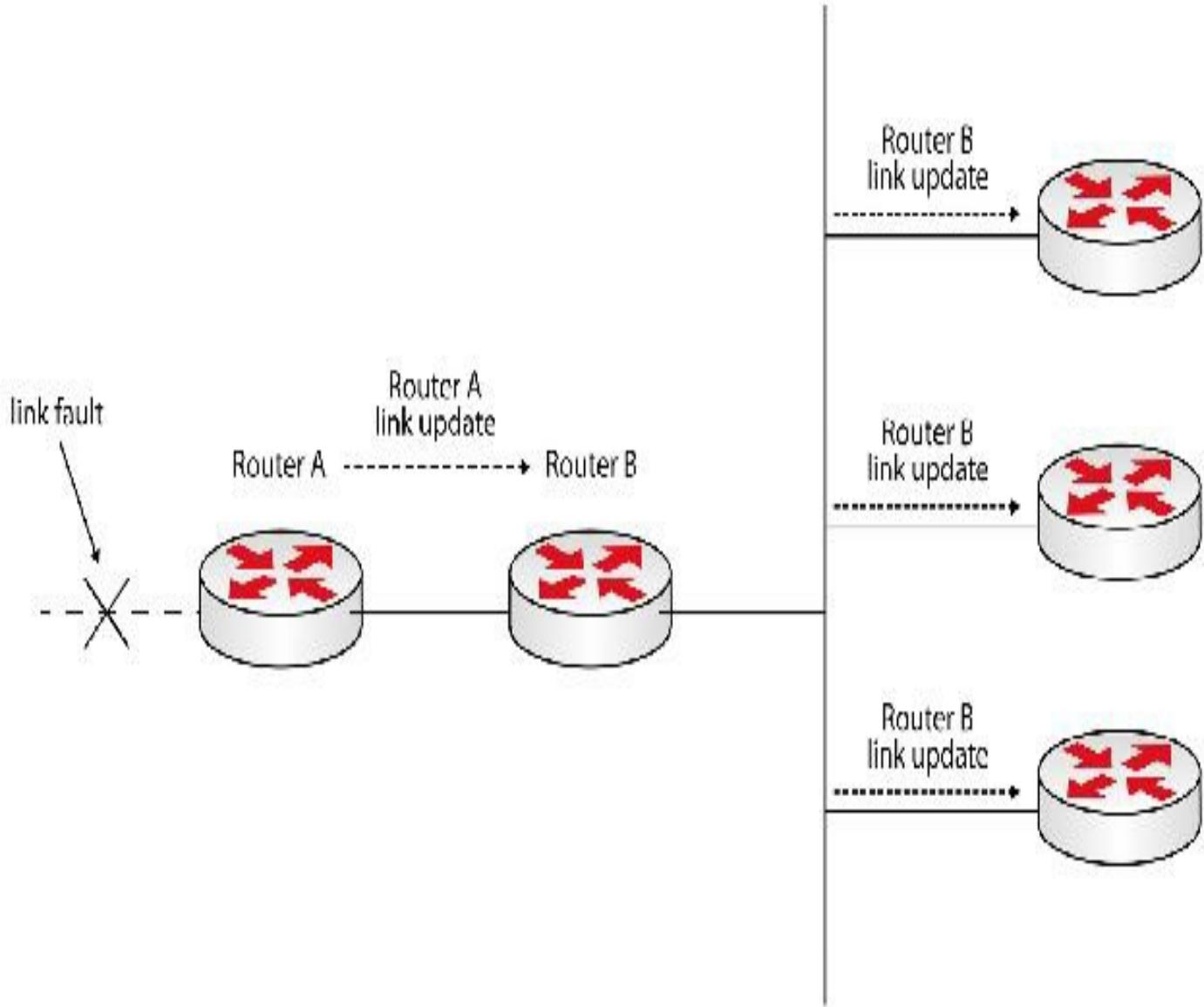


FIG 5.23 – Link state routing protocol behavior

This is a much more efficient approach to building routing databases, and there is no fixed updated timer as in distance vector technologies. Link state protocols reflood their entire routing information every 30 minutes in order to ensure that the network is properly converged (this is the case for OSPF at least).

Link state protocols offer a series of important advantages compared with distance vector protocols. The most important advantage relates to the convergence factor. Convergence happens a lot faster because as soon as a network topology changes, only that specific information is sent to the routers in a given area.

The routing updates stop after all the routers learn about the specific change, thus

decreasing the need for bandwidth, unlike distance vector protocols that periodically exchange routing tables, even if no topology change occurs. In link state routing, updates are triggered only when a link state changes somewhere in the network. Depending on the routing protocol in use, this can mean a link going up/down or changing some of its parameters (e.g., bandwidth).

Examples of link state routing protocols are OSPF (Open Shortest Path First), described in RFC 2328, and IS-IS (Intermediate System to Intermediate System), described in RFC 1142.

Planes of Operation

Before we discuss CEF (Cisco Express Forwarding), it will be worth covering the planes of operation. This isn't actually a CCNA exam subject; however, you will see references to terms such as management plane in most IT manuals and will certainly be expected to know about them in your day-to-day role as a network engineer.

A router is typically divided into three planes of operation and each plane serves a specific role. The three planes are the control plane, the data plane, and the management plane.

You can consider the control plane the router's brain. It consists of dynamic IP routing protocols such as OSPF, the routing table (list of known routes, also known as the Routing Information Base or RIB), routing updates, and other protocols such as ICMP, ARP, and others. The role of the control plane is to maintain sessions and exchange protocol information with other routers or network devices. All of the protocols mentioned will run on the CPU; however, high-end models can accommodate line cards with separate CPUs, which can take on some of the processing load.

The data plane is also referred to as the forwarding plane. It is responsible for switching packets through the router (e.g., processing and CEF switching). Features that can affect packet forwarding on the data plane include Quality of Service (QoS) and access control lists (ACLs).

The management plane, as the name suggests, is used to manage a device. Unlike direct cable access via the console port, it is managed over the network through its connection to the network. Protocols used to manage the device include Simple Network Management Protocol (SNMP), Telnet, File Transfer Protocol (FTP), Secure FTP, and Secure Shell (SSH). These protocols provide the means to monitor and manage the device remotely, as well as provide command line interface (CLI) access.

MANAGEMENT, CONTROL AND DATA PLANES

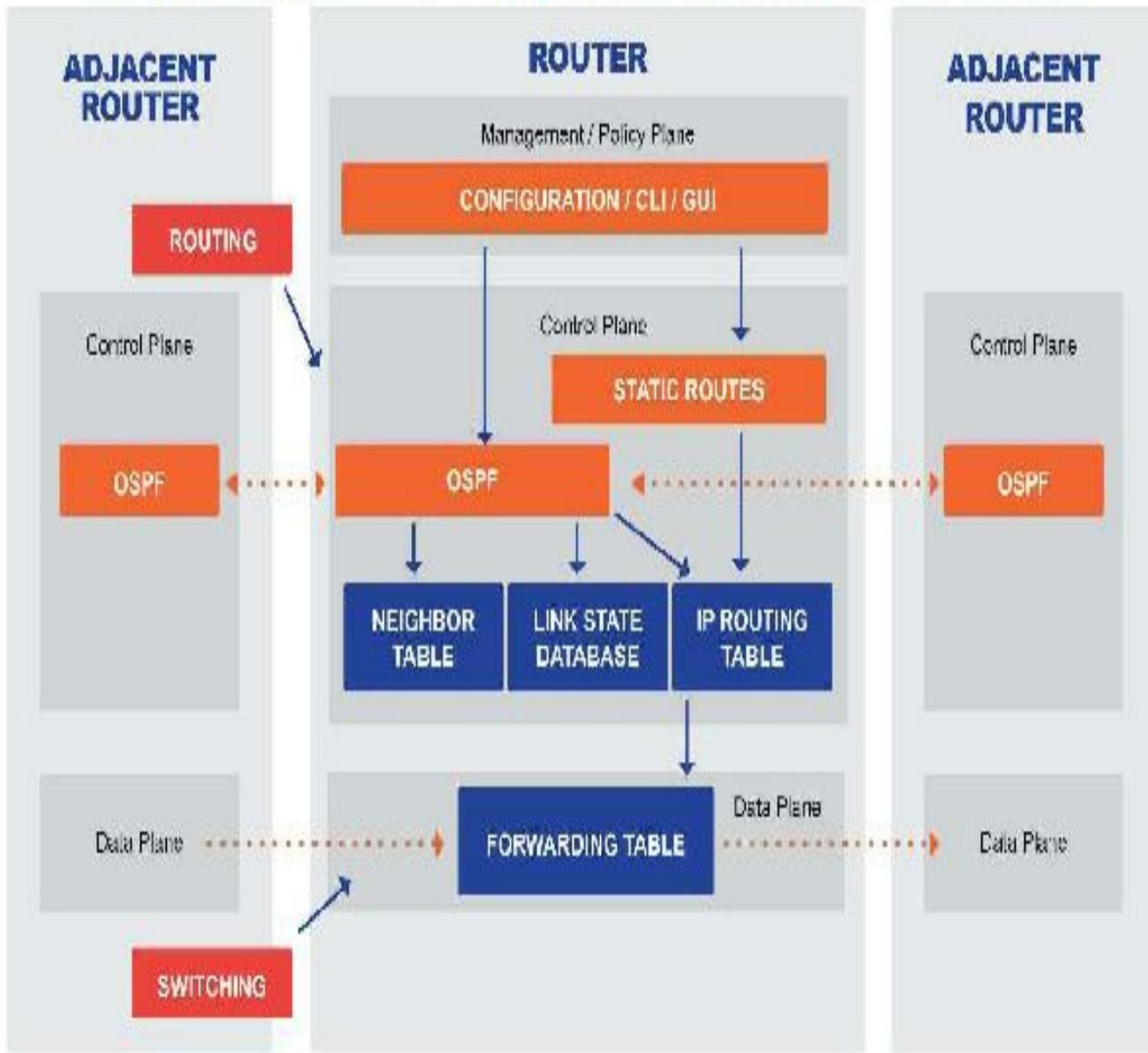


FIG 5.24 – Planes of operation

Topology-based (CEF) Switching

Traditionally, before a router could forward a packet, it would strip off the layer 2 information, check that a valid route existed, and rewrite the packet. Technology has advanced to the point that this information can simply be rewritten without removing the layer 2 information. Packet switching or forwarding is a faster process of forwarding or dropping packets (if there is no destination route).

An earlier method was referred to as process switching, but Cisco improved on this

with fast switching and then Cisco Express Forwarding (CEF). Fast switching has been deprecated.

A route lookup is the process of selecting the best route in the routing table based on the destination address of a packet. This process can be processor-intensive since large routers may have to match against thousands of routes in the routing table.

This process of matching the layer 2-to-layer 3 addresses and then forwarding the frame or packet is known as switching. Each interface will have a default switching method and this differs with router model, IOS, and interface type.

You have already learned that the earlier method was known as process switching. This method is inefficient since it is quite CPU-intensive. Fast switching involves creating a route cache of recently forwarded packets and their layer 2 information for faster forwarding of packets. Fast switching is enabled with the ip route-cache interface command:

```
R1(config-if)#ip route-cache ?
```

cef	Enable Cisco Express Forwarding
flow	Enable Flow fast-switching cache
policy	Enable fast-switching policy cache for outgoing packets
same-interface	Enable fast-switching on the same interface

[cr]

To further improve the process, Cisco created Cisco Express Forwarding.

Cisco Express Forwarding

CEF is a proprietary switching mechanism that creates a forwarding table from the information in the routing table. CEF eliminates the performance problem with process switching by storing information in hardware, rather than software. CEF stores information in two tables, the forwarding information base (FIB) and the adjacency table. All Cisco routers have CEF turned on by default.

The forwarding information base contains networks and their next hop, a streamlined version of the routing table in hardware. It has a one-to-one mapping with the routing table and, as such, this eliminates the need for a route cache, which was required in fast switching.

You should note that when a routing table change occurs, the CEF forwarding table is also updated. During this time, packets will be process switched.

The adjacency table takes the information in the FIB a step further by keeping the data

link information of the next hops. The adjacency table stores information about the layer 2 adjacency of the next-hop addresses. The adjacency table stores the MAC address information of the next hop in its table. This eliminates the need for consulting the ARP table for every packet.

Furthermore, once the next hop is determined from the FIB and an adjacency is determined from the adjacency table, a pointer to the right adjacency is stored in the FIB element so that subsequent FIB lookups already have the right adjacency stored. This process reduces lookup and significantly increases the speed of packet switching.

Accelerated and Distributed CEF

By default, Cisco switches use a central layer 3 switching engine. In large networks, having one hardware to perform all the layer 3 switching functions can cause a bottleneck in the network. To address this, Cisco has provided some CEF optimization features using specialized hardware in its high-end switches. These are Accelerated CEF (aCEF) and Distributed CEF (dCEF).

Accelerated CEF involves distributing a portion of the FIB to capable line card modules in a high-end switch. This allows forwarding decisions to be made directly on the line card using a locally stored, scaled-down version of the CEF table. If the FIB entry is not found on the line card, then it is sent to the larger FIB on the layer 3 engine.

Distributed CEF is a similar concept but refers to the use of multiple CEF tables distributed across multiple line cards installed in the chassis. In dCEF, the layer 3 engine maintains the routing table (RIB) and generates the FIB, and the FIB is then downloaded to all the line cards. This allows for multiple FIB lookups to be performed simultaneously.

Generally, aCEF and dCEF are used to enhance the overall system performance by allowing CEF operations to be performed in parallel. CEF technology offers the following benefits:

- Increased performance – CEF switching in hardware is less CPU-intensive than process switching or fast switching route caching. This allows more CPU processing power to be dedicated to other layer 3 services, such as QoS and encryption, for example.
- Scalability – CEF offers full switching capacity at each line card in high-end platforms, such as the Catalyst 6500 Series Switches, when dCEF mode is active.
- Resilience – CEF provides more resiliency in large dynamic networks since a route cache does not have to be managed in software. The routing changes in

large networks cause the routing cache used in fast switching to be invalidated and this causes traffic to be process-switched.

Mini-lab – Configuring Cisco Express Forwarding

CEF can be enabled using a single command, which is the ip cef [distributed] global configuration command. The [distributed] keyword is only applicable to high-end switches that support dCEF, such as the Catalyst 6500 Series Switches. The following output shows how to configure CEF on a lower-end platform, such as the Catalyst 3750 Series Switch:

```
Switch1(config)#ip cef  
Switch1(config)#exit
```

The following output illustrates how to enable dCEF on the Catalyst 6500 Series Switches:

```
Switch1(config)#ip cef distributed  
Switch1(config)#exit
```

Use the show ip cef command to check settings. If you add an IP address to an interface, you can also check it with the show ip interface f0/0 command:

```
R1#show ip interface f0/0  
  
FastEthernet0/0 is up, line protocol is up  
    Internet address is 172.16.1.1/16  
    Broadcast address is 255.255.255.255  
    Address determined by setup command  
  
IP fast switching is enabled  
    IP fast switching on the same interface is disabled  
    IP Flow switching is disabled  
    IP CEF switching is enabled  
    IP CEF Fast switching turbo vector  
  
[END OF MINI-LAB]
```

NOTE: There is no explicit command to configure or enable aCEF.

CEF is actually one of several methods to improve the efficiency of the internal forwarding process routers use. Two other common methods are process switching and fast switching.

With process switching, the router's CPU is involved in every forwarding decision. Fast switching uses the CPU for the first packet, but after that the path is stored in a fast switching cache and every packet matching the same source and destination is fast-switched (i.e., no need for the CPU or a route lookup).

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 5 exam.

Chapter 5 Labs

Lab 1: Static Routes

The physical topology is shown in Figure 5.25 below:

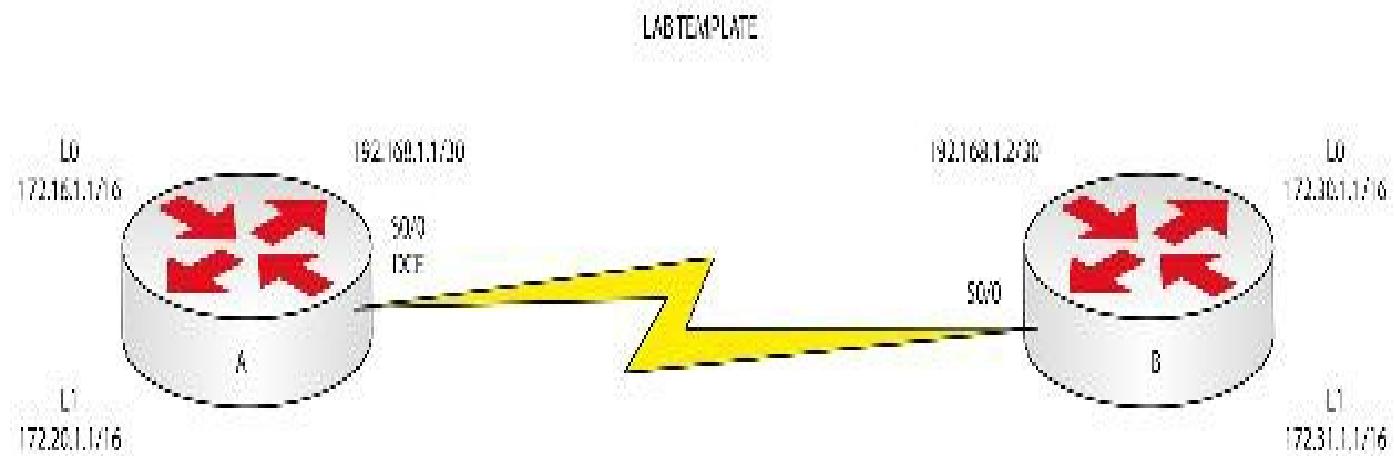


FIG 5.25 – Static Routes Lab

Lab Exercise

Your task is to configure the network in Figure 5.25 to allow full connectivity using static routes. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

The majority of small businesses have just one router that connects to another router provided by the service provider. These routers will only need to be configured with a very basic configuration, including IP addresses and a static route to reach the ISP. This lab will show you just how to do that.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 5.25. Router A needs to have a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Configure the static routes to provide connectivity to all networks attached to the neighbor router, except the network used for the Serial connection.
5. Ensure that routing information is correct by checking the routing table for entries to the neighbor's networks.

- Finally, try to ping all the Loopback interfaces of the neighbor's networks, and then try to access the neighbor router via Telnet.

Lab Walk-through

- To set the IP addresses on an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

To set the clock rate on a Serial interface (DCE connection only), you need to use the **clock rate #** command on the Serial interface, where # indicates the speed:

```
RouterA(config-if)#clock rate 64000
```

- To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration
RouterA(config-line)#login local i This will use local usernames and passwords for Telnet access
RouterA(config-line)#exit i Exits the VTY config mode
RouterA(config)#username banbury password ccna i Creates username and password for Telnet access (login local)
```

- To set the enable password, do the following:

```
RouterA(config)#enable secret cisco i Sets the enable password (encrypted)
```

- To configure static routes on a router, there is only one step:

```
RouterA(config)#ip route 172.30.0.0 255.255.0.0 192.168.1.2
RouterA(config)#ip route 172.31.0.0 255.255.0.0 192.168.1.2
```

The command above will configure a static route on Router A. To get to the destination networks 172.30.0.0 and 172.31.0.0, use the next-hop address

192.168.1.2. Instead of using the commands above, you can enter the ones below. This time the router is told to use an exit interface instead of a next hop. **DO NOT USE BOTH THE ABOVE AND BELOW TOGETHER.**

```
RouterA(config)#ip route 172.30.0.0 255.255.0.0 Serial0/0  
RouterA(config)#ip route 172.31.0.0 255.255.0.0 Serial0/0
```

5. Next, configure the same commands on Router B. Set the IP addresses:

```
Router#config t  
Router(config)#hostname RouterB  
RouterB(config)#interface Serial0/0  
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252  
RouterB(config-if)#no shutdown  
RouterB(config-if)#interface Loopback0  
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0  
RouterB(config-if)#interface Loopback1  
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0  
RouterB(config-if)#^Z  
RouterB#
```

6. Now make sure that you can ping across the Serial link. If you cannot, then check the configurations again.

```
RouterA#ping 192.168.1.2  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms
```

Configure Telnet access:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit  
RouterB(config)#username banbury password ccna
```

Configure the enable secret password:

```
RouterB(config)#enable secret cisco
```

Set the static route:

```
RouterB(config)#ip route 172.16.0.0 255.255.0.0 192.168.1.1
```

```
RouterB(config)#ip route 172.20.0.0 255.255.0.0 192.168.1.1
```

The command above will configure a static route on Router B. To get to the destination networks 172.20.0.0 and 172.16.0.0, use the next-hop address 192.168.1.1. Instead of using the commands above, you can enter the ones below. This time the router is told to use an exit interface instead of a next hop. **DO NOT USE BOTH THE ABOVE AND BELOW TOGETHER.**

```
RouterB(config)#ip route 172.16.0.0 255.255.0.0 Serial0/0  
RouterB(config)#ip route 172.20.0.0 255.255.0.0 Serial0/0
```

This command will configure a static route to the 172.20.0.0 or 172.16.0.0 network, but instead of having a next-hop address, you have specified an exit interface to use.

7. Use the show ip route command to check that the static routes are in the routing table and that the next-hop address is correct:

```
RouterA#show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP,

M - mobile, B – BGP, D - EIGRP, EX - EIGRP external,
O – OSPF, IA - OSPF inter area,
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2,
E1 – OSPF external type 1, E2 - OSPF external type 2,
E – EGP, i - IS-IS, L1 - IS-IS level-1,
L2 - IS-IS level-2, ia – IS-IS inter area,
* - candidate default, U - per-user static route,
o – ODR, P - periodic downloaded static route

Gateway of last resort is not set

C 172.16.0.0/16 is directly connected, Loopback0
C 172.20.0.0/16 is directly connected, Loopback1
S 172.31.0.0/16 [1/0] via 192.168.1.2
S 172.30.0.0/16 [1/0] via 192.168.1.2
192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Serial0/0

8. To test connectivity, you will need to use the ping command, and to log in to the neighbor router, you need to use the telnet command:

```
RouterA#ping 172.30.1.1 ! This will send a ping packet to the address  
specified; there should be five replies if everything is OK.
```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max = 1/2/4 ms
RouterA#ping 172.31.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.31.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
RouterA#
RouterA#telnet 172.30.1.1 i This will open a Telnet connection to the neighbor router. If Telnet access has been set up correctly, you should be presented with a login message. Type exit to quit a Telnet session or ctrl+shift+6 + exit.
RouterA#telnet 172.30.1.1
Trying 172.30.1.1 ... Open
User Access Verification
Username: banbury
Password: i **Password will not show as you type it**
RouterB>exit
[Connection to 172.30.1.1 closed by foreign host]
RouterA#

Do the same on Router B:

RouterB#ping 172.16.1.1
RouterB#ping 172.20.1.1
RouterB#telnet 172.16.1.1

Show Runs

RouterA#show run
Building configuration...

Current configuration : 704 bytes
!
version 15.1
!
hostname RouterA
!
username banbury password 0 ccna
!

```
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Loopback1
ip address 172.20.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
clockrate 64000
!
ip classless
ip route 172.30.0.0 255.255.0.0 192.168.1.2
ip route 172.31.0.0 255.255.0.0 192.168.1.2
no ip http server
!
line con 0
line 1 8
line aux 0
line vty 0 4
login local
!
end
```

```
RouterB# show run
Building configuration...

Current configuration : 678 bytes
!
version 15.1
!
hostname RouterB
!
username banbury password 0 ccna
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
```

```
ip address 172.31.1.1 255.255.0.0
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
ip classless
ip route 172.16.0.0 255.255.0.0 192.168.1.1
ip route 172.20.0.0 255.255.0.0 192.168.1.1
no ip http server
!
line con 0
line aux 0
line vty 0 4
login local
!
end
RouterB#
```

Chapter 6 — OSPF

What You Will Learn in This Chapter

Overview of OSPF

OSPFv3

Syllabus Topics Covered

4.7 Configure and verify single-area OSPF

 4.7.a Benefit of a single area

 4.7.b Configure single-area OSPFv2

 4.7.c Configure single-area OSPFv3

 4.7.d Router ID

 4.7.e Passive interface

Open Shortest Path First (OSPF) is an open standards protocol that uses a link state algorithm to calculate the best path to a particular network. It was developed in 1988 by the Internet Engineering Task Force (IETF) to meet the needs of modern networks whose purposes could no longer be served by RIP. It is far more robust and flexible than its distance vector predecessors and is ideally suited for use in modern enterprise networks.

OSPF employs the use of areas to simplify network administration and confine network instability to a specific location. It also allows extensive control of routing updates through several methods that we will examine in this guide.

OSPF is the standard open routing protocol in use on networks today. Open refers to the fact that it is driven by Dijkstra's Shortest Path First (SPF) algorithm, which isn't proprietary to any organization or vendor.

OSPF improves on older protocols by adding features such as:

- No hop-count limitation
- Rapid convergence
- Classless (allows the use of VLSM)
- Password authentication
- Advanced path selection capabilities
- Tagging of external routes
- Better use of bandwidth via multicasts and periodic routing updates
- Allows networks to be divided into smaller logical areas for efficiency

- Uses multicast addresses for efficient and reliable routing update process
- Uses equal-cost load balancing over multiple paths for efficient bandwidth usage
- Supports MD5 authentication for secure route exchange
- No split horizon issues

Overview of OSPF

We will cover some OSPF fundamentals in this section, and then some more advanced concepts in the ICND2 section, which is how Cisco addresses the subject in the syllabus.

OSPF is a classless routing protocol and it maps to IP protocol 89. It discovers and maintains a relationship with neighbor routers by multicasting Hello packets and Dead timers. Updates take the form of link state advertisements (LSAs). The link state database is OSPF's topology table. LSAs flood OSPF areas until each router has a consistent map of the network (i.e., the link state databases all match).

Once the map is consistent, the SPF algorithm is run on the database and a loop-free path to each network destination is built. This is referred to as the SPF tree. These can be seen in the routing table. An arbitrary metric of cost is used by OSPF when determining the shortest path from A to B.

A simplified view of OSPF is as follows:

1. Routers configured with OSPF send Hello packets out of all OSPF-configured interfaces. If the router on the shared link agrees on certain values in the Hello packets, they become neighbors.
2. Some neighbors form adjacencies. This depends on the type of OSPF router and network, such as point-to-point or broadcast.
3. Link state advertisements are sent over every adjacency. These describe the router links, neighbors, and the link state.
4. The received LSA is stored in the router's link state database and a copy is sent to all neighbors.
5. Each router will build an identical map or link state database.
6. Once all routers share the same database, they run the SPF algorithm to calculate the best loop-free path to every destination in the network. Each router considers itself to be the SPF root.
7. Each router can now build the routing table.

OSPF Terminology

OSPF was traditionally considered to be a CCNP topic, and even then only a reasonable working knowledge was expected. For the CCNA exam you will be expected to have a good working knowledge of OSPF configuration and troubleshooting for IPv4 and IPv6, both single- and multi-area.

A few terms you will need to be familiar with in relation to OSPF are listed below:

- Cost – This is the value OSPF assigns to a link to another router. Cost is used as opposed to hops because it offers far more granularity. The cost is based on the bandwidth of the link. (You can see the default bandwidths in Table 6-1 below.) Cisco routers calculate cost at $10^8/\text{Bandwidth}$, rounded down. The bandwidth is either the configured or the default bandwidth of the link.

Table 6-1: OSPF costs per interface

Interface	Cost ($10^8/\text{Bandwidth}$)
ATM, Fast Ethernet, Gigabit Ethernet, 10/100 Gbps	1
HSSI (45 Mbps)	2
16 Mbps Token Ring	6
10 Mbps Ethernet	10
4 Mbps Token Ring	25
T1 (1.544 Mbps)	64
DS-0 (64 K)	1562
56 K	1785

Here is the output of a `show ip ospf interface X` command for an Ethernet interface running at 10 Mbps:

```
R1#show ip ospf int f0/0
```

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.1/24, Area 0

Process ID 1, Router ID 10.0.0.1, Network Type BROADCAST, **Cost: 10**

Transmit Delay is 1 sec, State DR, Priority 1

Here is an interface running at 100 Mbps:

```
Router#show ip ospf int f0/0
```

FastEthernet0/0 is up, line protocol is up

Internet address is 192.168.1.1/24, Area 0

Process ID 1, Router ID 10.0.0.1, Network Type BROADCAST, **Cost: 1**

Transmit Delay is 1 sec, State BDR, Priority 1

You can manually override the cost on the interface with the `ip ospf cost [1-65535]` interface command. Remember though that the cost is cumulative, so each interface cost is added across the network. The OSPF cost can be seen with the `show ip route` command. The cost for the route below is 11, while 110 is the administrative distance for OSPF.

R1#`show ip route`

172.16.1.1 [110/11] via 192.168.1.2, 00:01:21, FastEthernet0/0

You can check each OSPF interface along the path. The output below shows a Loopback interface with a cost of 1 and a Fast Ethernet interface with a cost of 10, resulting in a total cost of 11:

R2#`show ip ospf interface`

Loopback0 is up, line protocol is up

Internet Address 172.16.1.1/16, Area 0

Process ID 1, Router ID 192.168.1.2, Network Type LOOPBACK, **Cost: 1**

Loopback interface is treated as a stub Host

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.2/24, Area 0

Process ID 1, Router ID 192.168.1.2, Network Type BROADCAST, **Cost: 10**

Transmit Delay is 1 sec, State BDR, Priority 1

- **Area ID** – An OSPF area is a group of routers divided into a subdomain based on an area ID. Every router within the same area shares the same link state information. Areas are identified by a 32-bit area ID. They can be represented either as decimal numbers or as dotted decimals (like IP addresses). Area 0 and Area 0.0.0.0 mean the same thing.
- **Link** – A link is a connection to another router. The OSPF topology table is referred to as the link state database.
- **Link state** – This is the state of the link to another router. The link state database consists of a list of the status of the links between routers in the same area.
- **Link state database** – The LSDB is a list of the link states for other routers in the network. The link state database is essentially the network topology. The

database is built from the exchange of LSAs.

- Link state advertisement – LSAs are OSPF data packets that contain routing and link state information that is shared between OSPF routers.
- Process ID – This is designated by the router ospf process [id] command, such as router ospf 20. The process ID is needed to identify a unique instance of an OSPF database and is locally significant to the router. The process ID can be any number from 1 to 65,535. The process ID does NOT need to match on neighbor routers (it isn't the same thing as the EIGRP ASN).
- Router ID – This will be covered in detail shortly.
- Network type – This is the type of network the OSPF interface connects to. This will be discussed in more detail later.
- Neighbors – Two routers that have interfaces on a common network are considered neighbors. The neighbors are discovered and maintained using the Hello protocol.
- Designated router – The DR is the central point for the exchange of routing information on a broadcast network. The principle here is to reduce the amount of traffic passing across the shared interface (e.g., an Ethernet interface with five routers). The DR is elected via Hello packets being passed across the area. The router with the highest priority wins; If all the routers have the same priority, then the router with the highest Router ID (usually the highest IP address) wins. A backup designated router (BDR) can also be elected to take over if the DR fails.
- Internal router – A router with all directly connected networks belonging to the same area (does not have to be area 0).
- Area border router (ABR) – A router with networks in more than one area.
- Backbone router – A router with an interface in the backbone (area 0).
- AS boundary router (ASBR) – A router that exchanges updates with routers in other autonomous systems.

OSPF routing table entries can be either internal to that area, represented by a O in the routing table, or from an ABR, represented by a O IA. A third type of entry—O E2—represents routes from an ASBR.

OSPF Router ID

The OSPF Router ID is actually an ICND2 topic; however, it makes more sense to bring it up here.

Every router running OSPF has to have a separate identity from the other routers. The router will select its own 32-bit unique ID based on the router's interface. The unique

ID is required in order to easily identify duplicate LSAs and endpoints of virtual links and to determine any tie-breakers between the DR and BDR (primary and secondary update sources, which will be covered later).

The router will choose the Router ID from the highest IP address on the router or the router's Loopback address if it has one. The Loopback address will be chosen above any other IP address. You can see this in the output below, where the Loopback address is chosen despite the fact that the address is lower than the Serial interface. If multiple Loopback interfaces are present, the one with the highest IP address is chosen.

The interface from which the Router ID is taken doesn't have to be running OSPF or taking part in the OSPF process. Network administrators usually assign a Loopback address because they can then assign a predictable Router ID and, of course, as a virtual interface it can never go down.

Please note that when the router boots, OSPF will only consider an address for the Router ID if it is active (that is, up/up). Once the Router ID is set it cannot be changed unless the router is reloaded or the OSPF process is reset (with the clear ip ospf process command).

Router#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0	unassigned	YES	unset	administratively	down down
Loopback0	172.16.1.1	YES	manual	up	up
Serial0	192.168.1.1	YES	manual	up	up
Serial1	unassigned	YES	unset	administratively	down down

Router#show ip ospf 20

Routing Process ospf 20 with ID **172.16.1.1** is **OSPF router ID**

You can set the Router ID by configuring the router with a high Loopback address (such as 192.168.100.100) or by using the router-id command, which is a best practice.

```
Router#config t
Router(config)#router ospf 20
Router(config-router)#router-id 192.168.100.100
```

OSPF Timers

OSPF uses several timers to control broadcasts, link state propagation, and several other operational factors. The default timers are different, depending on the type of network OSPF is configured on—point-to-point and non-broadcast are two examples.

The timers must match if a neighbor relationship is to form between routers running OSPF in the same area. On Routers 1 and 2 below, a neighbor relationship will not form due to mismatched timer values. OSPF timers can be seen with the `show ip ospf interface x` command, where x is the relevant interface.

```
Router1#show ip ospf interface Serial0/0
Serial0/0 is down, line protocol is down
  Internet Address 192.168.1.1/24, Area 0
  Process ID 20, Router ID 192.168.1.2, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State DOWN,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
```

```
Router2#show ip ospf interface Serial0/0
Serial0/0 is down, line protocol is down
  Internet Address 192.168.1.2/24, Area 0
  Process ID 20, Router ID 192.168.1.2, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State DOWN,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
```

- **Hello interval** – The default parameter for this timer depends entirely on what type of interface OSPF is operating on. For broadcast interfaces (such as Ethernet), the timer value is 10 seconds, and for Non-Broadcast Multi-Access interfaces (such as Frame Relay), it is 30 seconds. OSPF timers need to match on an interface before a neighbor relationship can be established.

```
RouterA(config-if)#ip ospf Hello-interval 40
```

The command above will change the Hello timer to 40 seconds. The Dead and Wait timers will automatically be changed when you change this timer.

- **Dead interval** – The Dead interval is the time it takes to declare the neighbor dead if there is no Hello. The Dead interval is four times the Hello interval.

```
RouterA(config-if)#ip ospf dead-interval 240
```

The command above will change the Dead interval to 240 seconds.

- **Retransmit interval** – This changes the retransmission interval between neighbors. When OSPF sends an update to a neighbor router, it expects to receive an acknowledgment. If no acknowledgment is heard, a retransmit takes place. The default is 5 seconds.

```
RouterA(config-if)#ip ospf retransmit-interval 10
```

The Wait timer is the interval that breaks the wait period and causes the designated

router to be selected in the network. This timer is always the same as the Dead timer. If you simply wanted to remove a timer configuration from an interface so it returns to the default setting, you would add the same configuration line but with a no in front:

```
RouterA(config-if)#no ip ospf hello-interval 40
```

OSPF Routes

OSPF operates in different ways depending on the type of link it is configured on, including broadcast, non-broadcast, and point-to-point. Depending on the link type, there may not be a DR/BDR election and Hello packets may be sent using a different multicast address.

When OSPF is configured on a router it begins to send Hello packets out of all OSPF interfaces using the multicast address of 224.0.0.5, which is known as the AllSPFRouter address in broadcast and point-to-point networks. In NBMA networks such as Frame Relay, the Hello packet is unicast to specific neighbors.

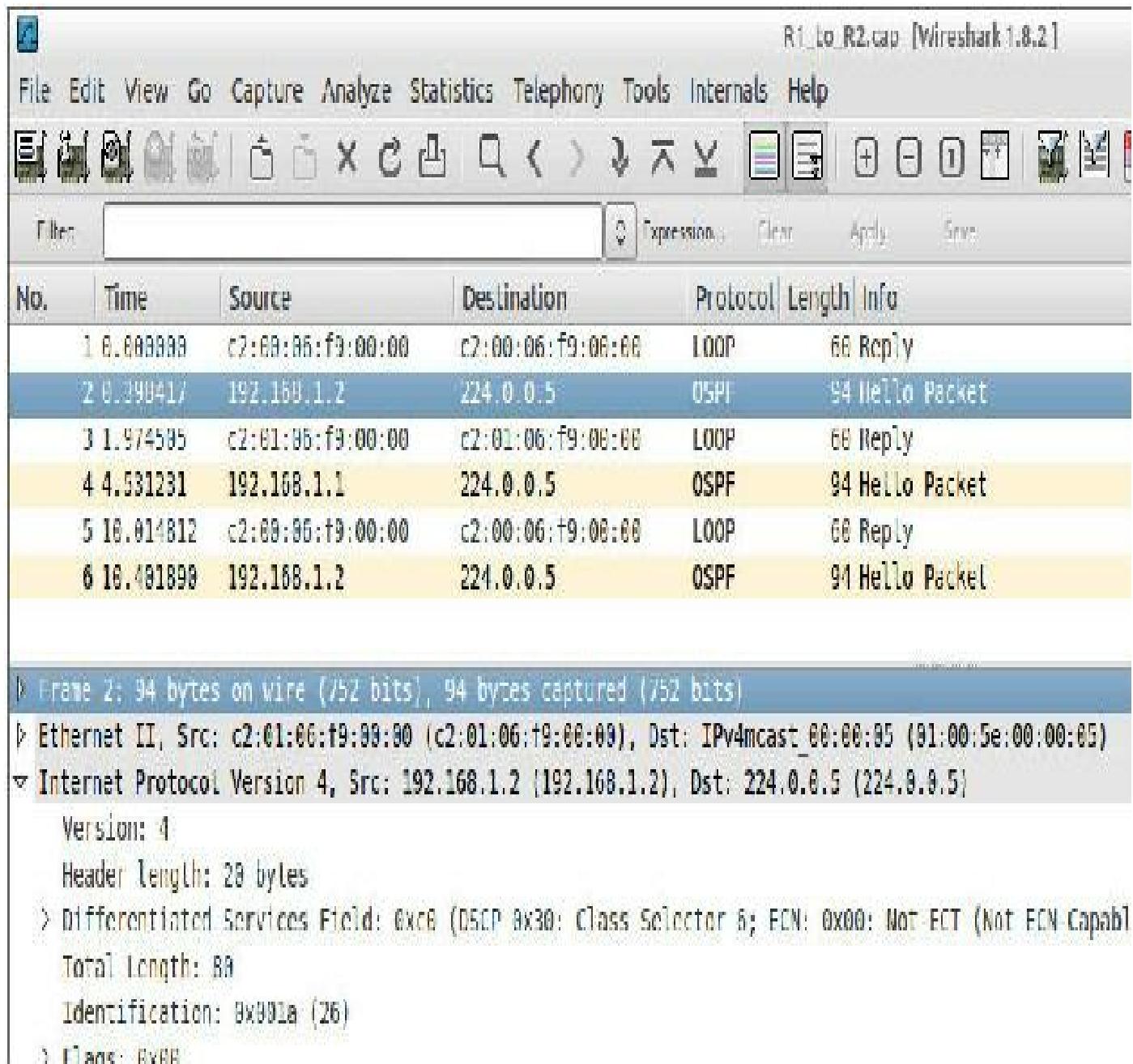


FIG 6.1 – OSPF multicast packet

The Hello packet differs from the EIGRP Hello in that it actually contains data and is used for data exchange, as you can see in the packet capture below. Information includes sending Router ID, area ID, address mask of the sender, authentication type, Hello and Dead intervals of originating interface, DR and BDR, and other information.

Message Type: Hello Packet (1)
Packet Length: 48
Source OSPF Router: 10.0.0.1 (10.0.0.1)
Area ID: 0.0.0.0 (Backbone)
Packet Checksum: 0x9d99 [correct]
Auth Type: Null
Auth Data (none)

▽ OSPF Hello Packet

 Network Mask: 255.255.255.0
 Hello Interval: 10 seconds
 ▷ Options: 0x12 (L, E)
 Router Priority: 1
 Router Dead Interval: 40 seconds
 Designated Router: 192.168.1.2
 Backup Designated Router: 192.168.1.2
 Active Neighbor: 192.168.1.2

▷ OSPF LLS Data Block

FIG 6.2 – OSPF Hello packet fields

Once the packet has been verified the two routers form a neighbor relationship. This does not mean that the neighbors will form an adjacency (a virtual link between routers used to send routes). In order for an adjacency to form, the neighbors must agree on parameters such as Hello interval, Dead interval, area ID, password (if used), and authentication type.

Each router will send its link state information to its neighbor, which records it and floods it onward to its current neighbors. All routers then build an identical link state database. A loop-free path to every known route is built and the local router is known as the root.

OSPF Virtual Links

Cisco warns that the use of virtual links indicates a bad OSPF network design. Virtual links are used to extend area 0 across another area and caters to the rule that all non-zero areas should directly connect to area 0 (the backbone). The virtual link is used to

tunnel LSAs through a non-zero area. The area used to transit cannot be a stub area (this will be covered later in ICND2).

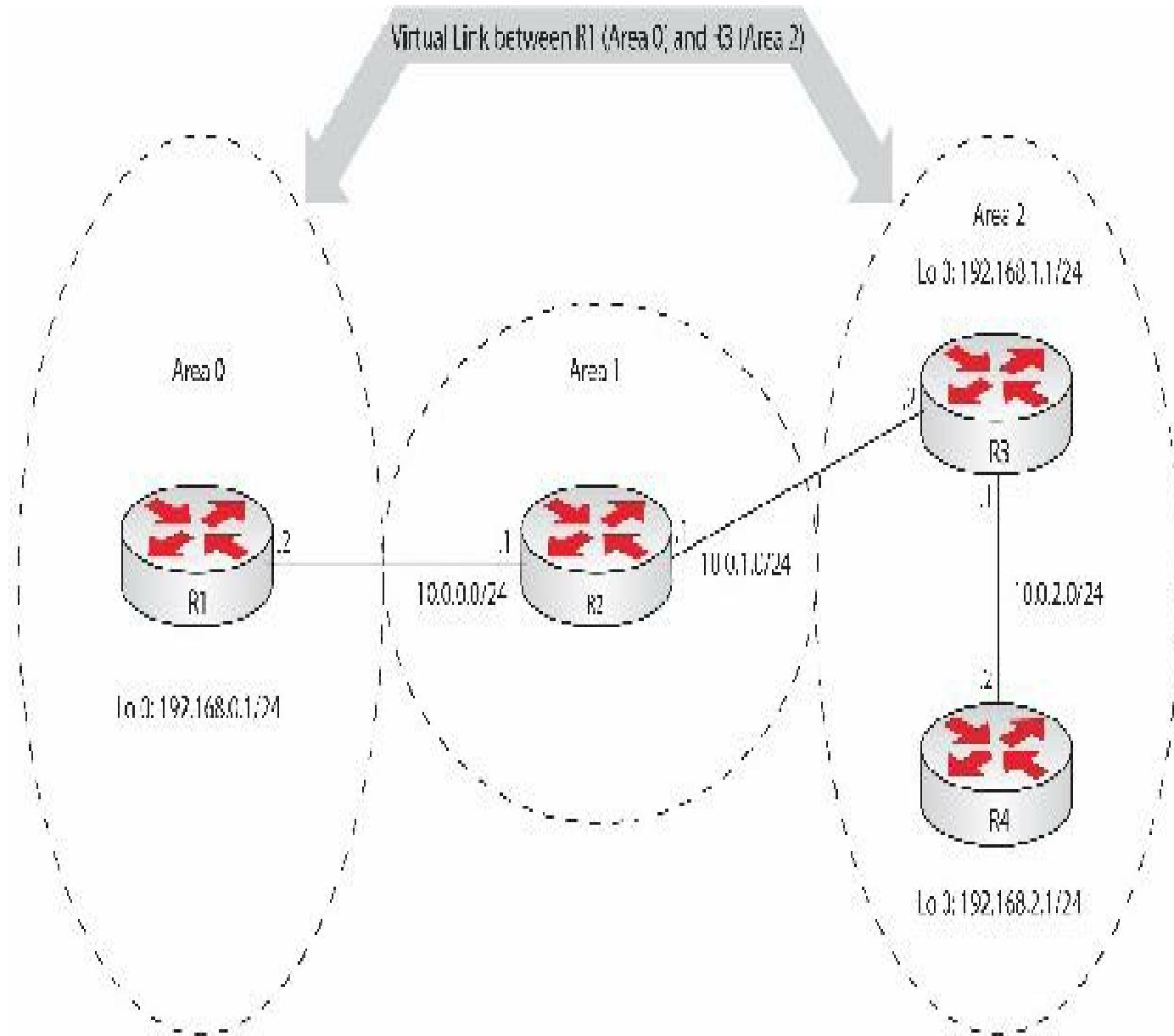


FIG 6.3 – OSPF virtual links

Virtual links rely on fixed Router IDs because the RID value is used in the virtual link configuration.

OSPF Load Balancing

If there is more than one equal path (same cost and administrative distance) to reach a route, then, by default, OSPF will load balance traffic over four paths.

R1#show ip protocols

Routing Protocol is “ospf 1”

Router ID 10.0.0.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

10.0.0.0 0.0.0.255 area 0

192.168.1.0 0.0.0.255 area 0

[output truncated]

This value can be increased but doing so requires careful planning.

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#router ospf 1

R1(config-router)#maximum-paths ?

<1-16> Number of paths

OSPF Network Types and Neighbors

OSPF treats its neighbors differently depending on the link types that join them. There are five types for OSPF, some of which will be covered in detail later in this guide (such as NBMA).

- Point-to-point – An example would be a simple T1 link between two routers or a Frame Relay point-to-point connection. Hellos are multicast to 224.0.0.5 (AllSPFRouters). There is no DR/BDR election on this network type.



FIG 6.4 – OSPF point-to-point network

- Broadcast – Ethernet is the only modern example of this network type. It makes more sense to refer to it as a broadcast multi-access network because several devices can be connected so they can receive the same packet. A DR/BDR election takes place on this network type and Hellos are multicast on 224.0.0.5, as are all OSPF packets originated by the DR/BDR. All other routers will multicast updates and acknowledgment packets on 224.0.0.6, also known as

All DRouters.

- NBMA – These include Frame Relay natural or multipoint connections. On this network type (as you will discover later), multicast packets are not forwarded correctly to neighbors because there is no broadcast capability. OSPF neighbors must be configured by the administrator on this type of network with the neighbor command under Router(config-router)# mode on DR/BDR. A DR/BDR is elected; however, it should be the hub router (i.e., the one with a circuit to all the other routers).
- Point-to-multipoint – This network type must be defined on NBMA networks by the network administrator. There is no DR/BDR, and OSPF packets are multicast. Although it's probably outside the CCNA syllabus, if you are configuring this network type, you need to map the remote IP address to the layer 2 address (DLCI for Frame Relay) and add the broadcast keyword so that OSPF can multicast their Hello packets. Although it's a seemingly insignificant tag, OSPF will not work without it on this network type.

```
interface Serial0/1
ip address 10.0.0.1 255.255.0.0
encapsulation frame-relay
ip ospf network point-to-multipoint
frame-relay map ip 10.0.0.2 20 broadcast
no frame-relay inverse-arp
```

Don't worry about the configuration above. I'm just laying the groundwork for those who want to do more advanced Cisco studies after the CCNA exam. But please do bear in mind that Cisco has a reputation for dropping some fairly advanced questions into the CCNA exam without warning!

- Virtual links – These are used to link areas not directly connected to area 0. These were described briefly above.

Figure 6.5 below shows a simplified version of the network types above. Bear in mind though that some of the interface types are defined by configuration commands rather than physical topology.

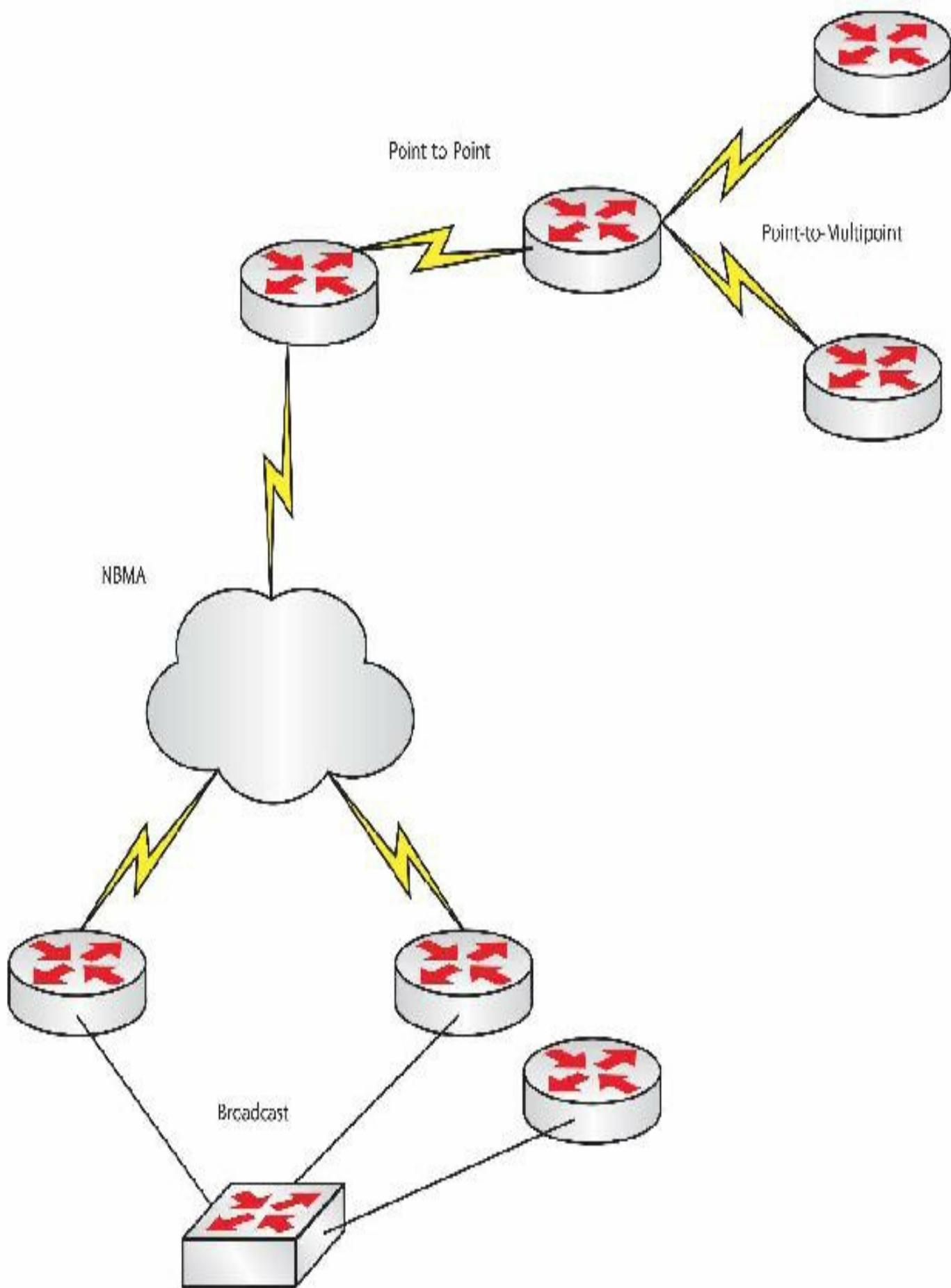


FIG 6.5 – OSPF network types

Although this is outside the CCNA RS syllabus, it's important to note that unlike EIGRP (for example), OSPF will not tolerate an arbitrary network topology. Implementing OSPF on your network requires very careful topology planning and a well thought out hierarchical IP address scheme.

“...about network topology and route summarization, adopting a hierarchical addressing environment and a structured address assignment are the most important factors in determining the scalability of your internetwork” (©Cisco Press).

Mini-lab – Configuring Single-area OSPF

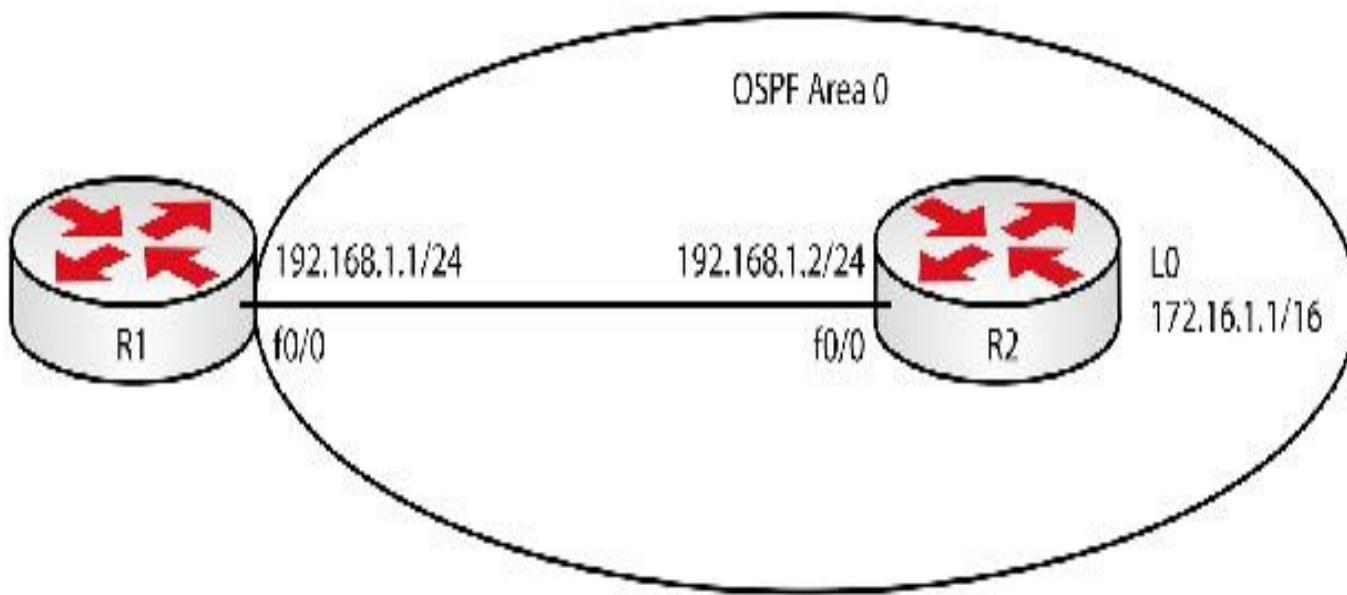


FIG 6.6 – Mini-lab: Configuring Single-area OSPF

Configuring OSPF can be a fairly complicated process due to issues with the protocol and the type of interface you are configuring the protocol on. Generally, you need to enable the OSPF process on the router, and then specify which interfaces will run OSPF and which areas those interfaces belong to.

The OSPF process ID is locally significant and need not be the same on all routers within an area or entire network. You can even have more than one OSPF process running on a single router. In real-world networks, many times the OSPF process IDs are kept unique to make troubleshooting an easier task.

OSPF can be configured in two steps for a very simple network (usually with only one area):

1. Define OSPF on the router:

```
Router(config)#router ospf [process-id]
```

The process ID is an internal number used to identify multiple instances of OSPF running on one router. It is only locally significant so it doesn't need to match other routers and it can be reused on other routers.

2. Assign networks to the relevant OSPF area:

```
Router(config-router)#network address wildcard-mask area area ID
```

- Address – the network address
- Wildcard mask – the inverse of a subnet mask (This will be discussed in detail in Chapter 7.)
- Area/area ID – the OSPF area you want the interface/network to be in; if you are using more than one area, one of them must be area 0.

For the network above, put the two Fast Ethernet interfaces into area 0, as well as the Loopback 0 interface on R2. I've used a ? in some parts so that you can see the options available.

```
R1(config)#int f0/0
```

```
R1(config-if)#ip add 192.168.1.1 255.255.255.0
```

```
R1(config-if)#no shut
```

```
R1(config)#router ospf ?
```

[1-65535] Process ID **i Note that the process ID can't be 0**

```
R1(config)#router ospf 1
```

```
R1(config-router)#network 192.168.1.0 0.0.0.255 area ?
```

[0-4294967295] OSPF area ID as a decimal value

A.B.C.D OSPF area ID in IP address format

```
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

With the commands above, you have put the Fast Ethernet 0/0 interface into area 0. You will need to do the same for R2.

```
R2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#int f0/0
```

```
R2(config-if)#ip add 192.168.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#router ospf 1
R2(config-router)#net 192.168.1.0 0.0.0.255 area 0
R2(config-router)#end
R2#
```

*Mar 1 00:04:28.207: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on FastEthernet0/0 from LOADING to FULL, Loading Done

```
R2(config)#int lo0
R2(config-if)#ip add 172.16.1.1 255.255.0.0
R2(config-if)#router ospf 1
R2(config-router)#net 172.16.0.0 0.0.255.255 area 0
```

Finally, issue a show ip route command. I've truncated the output. You won't see the O in front of the directly connected network because the route was learned from interface 192.168.1.2.

```
R1#show ip route
```

Gateway of last resort is not set

```
    172.16.0.0/32 is subnetted, 1 subnets
O 172.16.1.1 [110/11] via 192.168.1.2, 00:01:21, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

[END OF MINI-LAB]

It is important to note that the network command is not used to advertise subnets in OSPF. Instead, it is used to determine which interfaces are participating in OSPF, and the subnet mask on those particular interfaces determines which subnets are advertised. Furthermore, the wildcard mask doesn't have to be the inverse of the subnet mask configured on a particular interface. Once you have completed the configuration you should try out some show commands to determine what information is presented to you. We will revisit these commands throughout this guide but it's good to start using them now.

show ip route

```
show ip ospf neighbor  
show ip ospf interface brief  
show ip ospf interface f0/0
```

If you wanted to add all router interfaces into area 0, you would use the configuration lines below:

```
Router(config)#router ospf 1
```

```
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

```
Router(config-router)#end
```

```
Router#show ip prot
```

Routing Protocol is “ospf 1”

[output truncated]

Maximum path: 4

Routing for Networks:

```
0.0.0.0 255.255.255.255 area 0
```

Any IP address on any enabled interface will be added to OSPF area 0.

Mini-lab – Configuring OSPF Interfaces

You can alternatively configure OSPF areas per interface and you should be familiar with both methods for the exam.

Figure 6.7 below shows a simple topology where you want to connect R1 and R2 via OSPF area 0. Add the IP addresses and router names yourself.

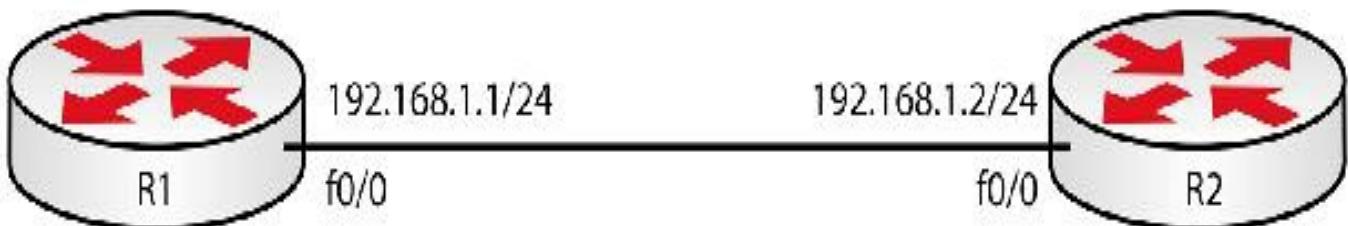


FIG 6.7 – Mini-lab: Configuring OSPF Interfaces

```
R1(config)#int f0/0
```

```
R1(config-if)#ip ospf 1 area 0
```

```
R1(config-if)#end
```

*Mar 1 01:19:37.127: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on FastEthernet0/0 from LOADING to FULL, Loading Done

R2(config)#int f0/0

R2(config-if)#ip ospf 2 area 0

R2(config-if)#end

*Mar 1 01:19:35.879: %OSPF-5-ADJCHG: Process 2, Nbr 192.168.1.1 on FastEthernet0/0 from LOADING to FULL, Loading Done

R1#show ip protocols

Routing Protocol is “ospf 1”

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.1.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

Routing on Interfaces Configured Explicitly (Area 0):

FastEthernet0/0

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

192.168.1.1	110	00:01:11
-------------	-----	----------

Distance: (default is 110)

R1#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
-------------	-----	-------	-----------	---------	-----------

192.168.1.2	1	FULL/DR	00:00:31	192.168.1.2	F0/0
-------------	---	---------	----------	-------------	------

R1#show ip ospf int f0/0

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.1/24, Area 0

Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost:10

Enabled by interface config, including secondary ip addresses

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 192.168.1.2, Interface address 192.168.1.2

Backup Designated Router (ID) 192.168.1.1, Interface address 192.168.1.1

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:09

[output truncated]

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 192.168.1.2 (Designated Router)

Suppress Hello for 0 neighbor(s)

[END OF MINI-LAB]

Mini-lab – OSPF Passive Interfaces

We have already covered passive interfaces for routing protocols, as well as the configuration commands, so please review that section. I've truncated some of the output below. Bear in mind that you can also use the `passive-interface default` command. Use any of the labs above or any IP addressing you want. After you configure and verify OSPF, you can create the passive interface.

R2#show ip ospf interface

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.2/24, Area 0

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:07

R2(config)#router ospf 1

R2(config-router)#passive-interface fast0/0

R2(config-router)#^z

R2#show ip ospf interface f0/0

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.2/24, Area 0

Process ID 1, Router ID 192.168.1.2, Network Type BROADCAST, Cost: 10

No Hellos (Passive interface)

[END OF MINI-LAB]

OSPFv3

OSPFv3 has been developed to be used with IPv6 addressing and it shares almost the same functionality principles with OSPFv2 for IPv4. This includes the way packets are formatted, adjacencies are built, timers, network types, and LSA flooding mechanisms.

Most of the concepts and rules you learned for OSPFv2 still apply when using OSPFv3 and this includes areas, backbone area, neighbor relationships, etc. Like OSPFv2, OSPFv3 functions by advertising LSA packets between nodes in order to share a common view of the network. In the next phases, the LSDB is computed and the SPF algorithms run in order for each router to compute the shortest path to every reachable IPv6 destination within a network. The shortest paths are then installed in the routing table and used to forward packets.

However, the new protocol also has some particularities specific to IPv6, including:

- Slightly different configuration process (detailed in the next sections)
- OSPFv3 uses link-local IP addresses to form adjacencies
- The authentication mechanism is different
- OSPFv3 uses different multicast addresses
- Hello packets contain different fields

Figure 6.8 below shows a side-by-side packet capture of OSPFv2 and v3. You shouldn't be expected to know the differences between the fields in the packets for the CCNA exam. Details on these differences can be found in the following paragraphs.

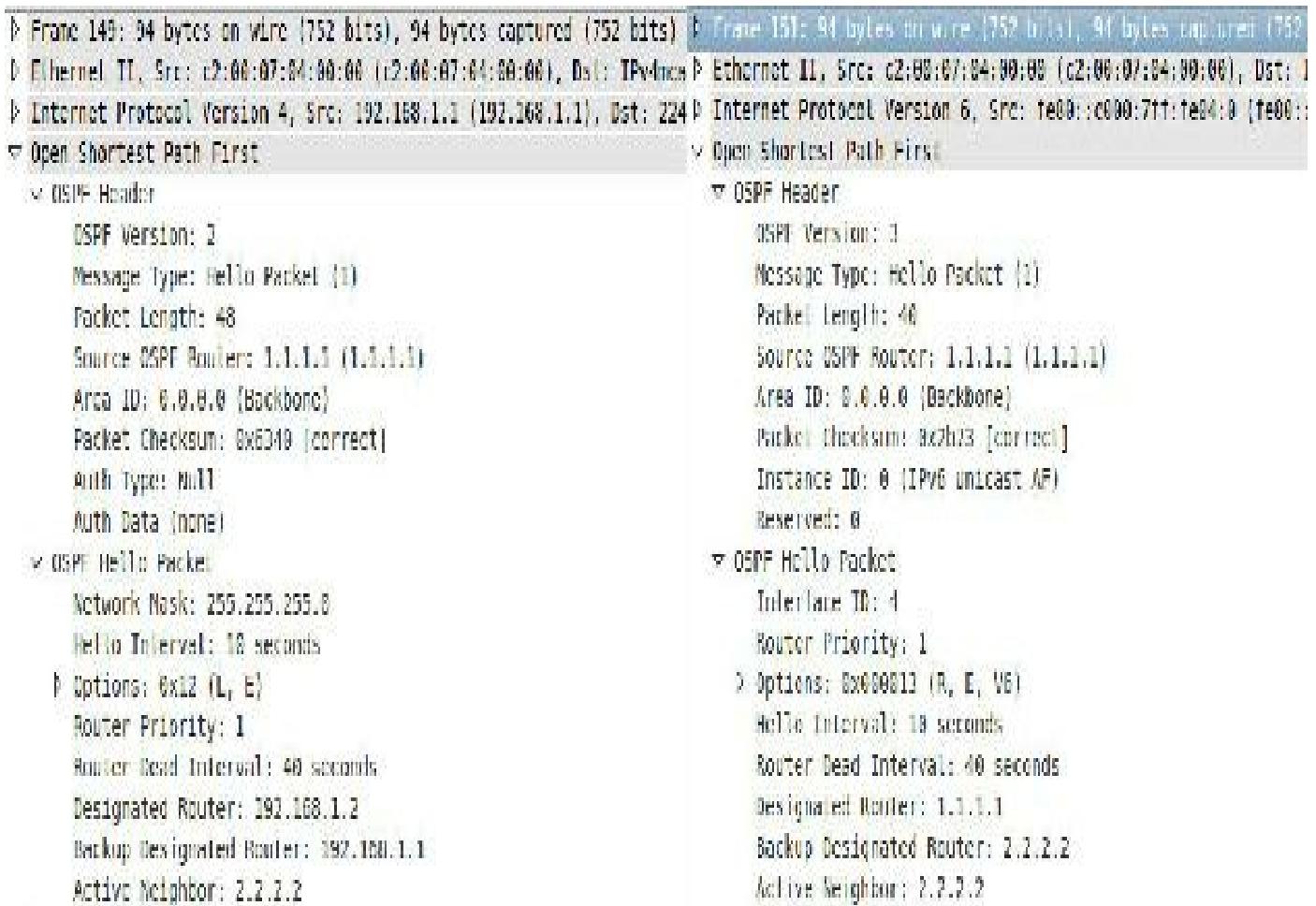


FIG 6.8 – OSPFv2 and v3 Hello packets compared

The configuration process for OSPFv3 changes slightly from the OSPFv2 implementation. Instead of using the network command in OSPF router mode to enable OSPF routing on specific interfaces and advertise the associated prefixes, this is done in interface configuration mode using the `ipv6 ospf [process_id] area [area_id]` command. For example:

```
Router(config)#interface GigabitEthernet1/1
```

```
Router(config-if)#ipv6 ospf 10 area 0
```

Note that recent versions of the Cisco IOS also support this method of configuring OSPFv2 on interfaces. However, the network command is still the most common way of enabling OSPFv2 on IPv4 interfaces.

After enabling OSPFv3 on the relevant interfaces, you still need to go into OSPFv3 router configuration mode using the `ipv6 router ospf [process id]` command to define additional parameters, for example, configuring a router ID as shown below:

```
Router(config)#ipv6 router ospf 10
```

```
Router(config-rtr)#router-id 10.10.10.1
```

One interesting feature in OSPFv3 is the router ID. You would normally expect OSPF for IPv6 to use an IPv6 address as the router ID. But the router ID is still a 32-bit value expressed in dotted decimal. Although it looks similar to an IPv4 address, it is really just a 32-bit number, and this hasn't changed in OSPFv3.

Like OSPFv2, If the router is assigned IPv4 addresses, the OSPFv3 process will use the highest IPv4 address assigned to a logical or connected physical interface as the router ID. However, if the router does not have any IPv4 addresses configured, you will have to manually configure the OSPFv3 router ID.

Since OSPFv3 uses IPv6 link-local addresses to establish adjacencies and these addresses are automatically generated on Cisco router interfaces, you can have two routers running a healthy OSPF adjacency, even if they don't have any sort of IPv4 or IPv6 addresses manually configured on their connecting interface.

Mini-lab – Configuring Single-area OSPFv3

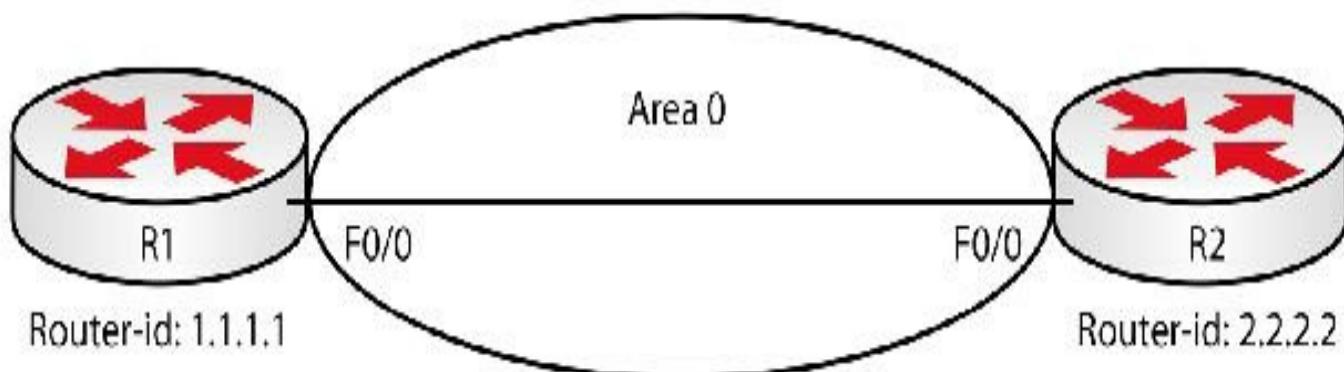


FIG 6.9 – Mini-lab: Configuring Single-area OSPFv3

Configuration on R1:

```
R1(config)#ipv6 unicast-routing  
R1(config)#int fa0/0  
R1(config-if)#ipv6 enable  
R1(config-if)#ipv6 ospf 1 area 0  
R1(config-if)#exit  
R1(config)#ipv6 router ospf 1  
R1(config-rtr)#router-id 1.1.1.1
```

Configuration on R2:

```
R2(config)#ipv6 unicast-routing  
R2(config)#int fa0/0
```

```
R2(config-if)#ipv6 enable  
R2(config-if)#ipv6 ospf 1 area 0  
R1(config-if)#exit  
R2(config)#ipv6 router ospf 1  
R2(config-rtr)#router-id 2.2.2.2
```

OSPFv3 configuration verification uses show commands similar to the ones used with OSPFv2. Don't forget to replace show ip with show ipv6:

```
R1#show ipv6 ospf 1 neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/DR	00:00:36	4	FE0/0

```
R2#show ipv6 ospf 1 nei
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/BDR	00:00:32	4	FE0/0

You can see whether the adjacency has formed using the autogenerated link-local IPv6 address on interface Fast Ethernet 0/0. I've removed some of the output to save space.

```
R1#show ipv6 ospf 1 interface
```

FastEthernet0/0 is up, line protocol is up

Link Local Address FE80::C001:FFF:FE00:0, Interface ID 4

Area 0, Process ID 1, Instance ID 0, Router ID 1.1.1.1

Network Type BROADCAST, Cost: 10

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 2.2.2.2, local address FE80::C002:2DFF:FE6C:0

Backup Designated router (ID) 1.1.1.1, local address FE80::C001:FFF:FE00:0

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 2.2.2.2 (Designated Router)

```
R2#show ipv6 ospf 1 interface
```

FastEthernet0/0 is up, line protocol is up

Link Local Address FE80::C002:2DFF:FE6C:0, Interface ID 4

Area 0, Process ID 1, Instance ID 0, Router ID 2.2.2.2

Network Type BROADCAST, Cost: 10

Transmit Delay is 1 sec, State DR, Priority 1

Designated Router (ID) 2.2.2.2, local address FE80::C002:2DFF:FE6C:0

Backup Designated router (ID) 1.1.1.1, local address FE80::C001:FFF:FE00:0

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 1.1.1.1 (Backup Designated Router)

[END OF MINI-LAB]

Link State Problems

When you are deciding on which routing protocol is best for your network, you will have to make a choice based on certain advantages and disadvantages of the protocols you are considering. Link state protocols are no different, as there are pros and cons to using them. Whether you do use a link state protocol will depend on the benefits outweighing the costs for your particular network and requirements.

The two main drawbacks of using OSPF are:

1. High CPU utilization
2. Bandwidth utilization

CPU and Memory Requirements

In order to run the OSPF protocol, a large number of CPU cycles will be consumed. While this is happening, other processes on the router will slow down because they have limited CPU time available to them. A router running a link state protocol will require more memory and CPU power than a distance vector algorithm.

Link state protocols are extremely popular in modern networks. Although they use more CPU cycles and bandwidth, they can have faster convergence and they offer more power and granularity when configuring various parameters, so they make a very attractive choice.

You can check on the CPU load on a router with the show processes command. You need to execute it every 60 seconds to get a picture of usage.

RouterA#show processes

CPU utilization for five seconds:4%/0%; one minute: 2%;five minutes: 1%

PID Runtime(ms)Invoked uSecs 5Sec 1Min 5Min TTY Process

```
1 820 898      913  0.00% 0.03% 0.05% 0 Load Meter
```

```
2 16236 1672    9710 4.33% 1.18% 0.61% 0 Exec
```

There will be many more processes than the shortened example above.

You can also execute the show processes cpu command to get the five-second, one-minute, and five-minute display of CPU utilization for each process running.

The solution to the high-CPU issue is to spend more money on a higher-end router capable of running the protocol. Although low-end routers can run OSPF, this is not recommended in real-world environments. Users should check Cisco documentation to confirm the CPU load and whether the router is suitable to run OSPF.

Bandwidth Utilization

Link state protocols do not require a lot of bandwidth after the network is converged. When the protocol is first enabled, the network is flooded with LSA packets that last until convergence is complete. If you have limited bandwidth it can be completely consumed for some time, severely impacting your network.

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 6 exam.

Chapter 6 Labs

Lab 1: Single-area OSPF

The physical topology is shown in Figure 6.10 below:

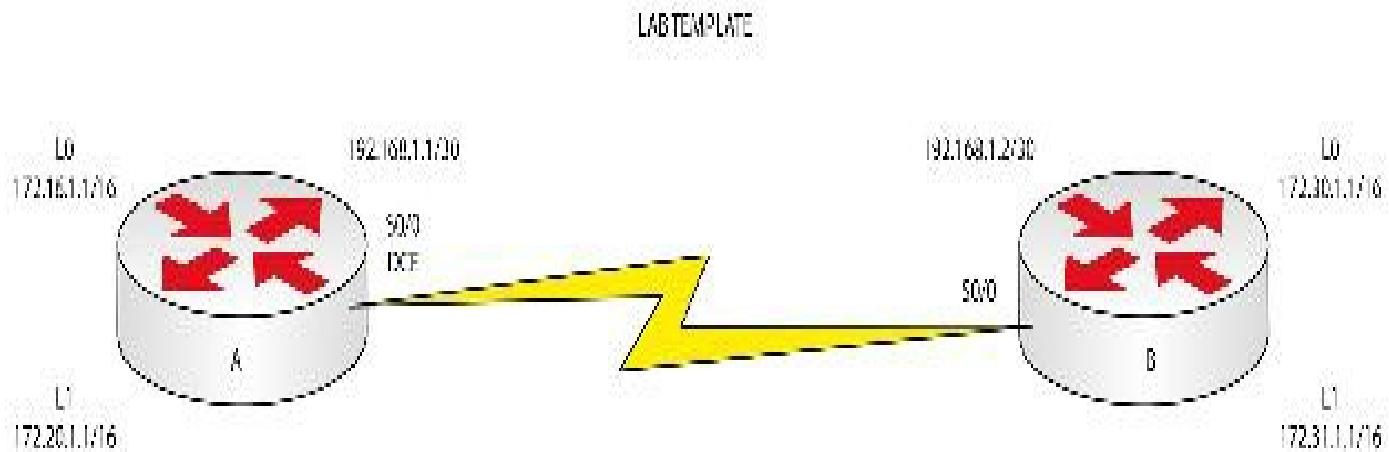


FIG 6.10 – Single-area OSPF Lab (all interfaces in area 0)

Lab Exercise

Your task is to configure the network in Figure 6.10 to allow full connectivity using the OSPF routing protocol. Please consider all interfaces in area 0, including Loopback interfaces. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

OSPF is a highly robust and scalable protocol and by far the most popular in medium to large companies. A good working knowledge of the protocol is vital to your success in the exam and as a Cisco engineer.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 6.10. On Router A, you need to configure a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Configure the OSPF routing protocol to advertise all networks attached to the router.
5. Ensure that the routing information is correct by checking the routing table for entries of the neighbor's addresses.

- Finally, try to ping all the neighbor Loopback interfaces, and then try to access the neighbor router via Telnet.

Lab Walk-through

- To set the IP addresses on an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000 i If this is the DCE side
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#
```

- To set the clock rate on a Serial interface (DCE connection only), you need to use the `clock rate #` command on the Serial interface, where # indicates the speed:

```
RouterA(config-if)#clock rate 64000
```

Ping across the Serial link now.

3. To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this type (in configuration mode):

RouterA(config)#line vty 0 4 **i Enters the VTY line configuration**

RouterA(config-line)#login local **i This will use local usernames and passwords for Telnet access**

RouterA(config-line)#exit **i Exits the VTY config mode**

RouterA(config)#username banbury password ccna **i Creates username and password for Telnet access (login local)**

Router B:

RouterB(config)#line vty 0 4

RouterB(config-line)#login local

RouterB(config-line)#exit

RouterB(config)#username banbury password ccna

4. To set the enable password, do the following:

RouterA(config)#enable secret cisco **i Sets the enable password (encrypted)**

Router B:

RouterB(config)#enable secret cisco

5. To configure OSPF on a router, there are two steps: first, enable the routing protocol; and second, specify the networks to be advertised by OSPF:

RouterA(config)#router ospf 20 **i Enables the OSPF routing process**

RouterA(config-router)#network 172.20.0.0 0.0.255.255 area 0

RouterA(config-router)#network 192.168.1.0 0.0.0.3 area 0

RouterA(config-router)#network 172.16.0.0 0.0.255.255 area 0

i Specifies the networks for OSPF to advertise; one network statement is needed for every network advertised.

Router B:

RouterB(config)#router ospf 20

RouterB(config-router)#network 192.168.1.0 0.0.0.3 area 0

RouterB(config-router)#network 172.30.0.0 0.0.255.255 area 0

RouterB(config-router)#network 172.31.0.0 0.0.255.255 area 0

You should see a console message telling you that the OSPF adjacencies have been formed. For the command to take, you should exit configuration mode with the ^Z or type exit twice.

RouterB#

02:38:57: %SYS-5-CONFIG_I: Configured from console by console
02:38:59: %OSPF-5-ADJCHG: Process 20, Nbr 172.20.1.1 on Serial0 from
LOADING to FULL, Loading Done

Use the show ip route command to determine whether the networks being advertised by the neighbor's OSPF process are in your routing table.

RouterA#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B – BGP, D - EIGRP, EX - EIGRP external,
O – OSPF, IA - OSPF inter area,
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2,
E1 - OSPF external type 1, E2 - OSPF external type 2,
E – EGP, i - IS-IS, L1 - IS-IS level-1,
L2 - IS-IS level-2, ia - IS-IS interarea,
* - candidate default, U - per-user static route,
o – ODR, P - periodic downloaded static route

Gateway of last resort is not set

C 172.16.0.0/16 is directly connected, Loopback0
C 172.20.0.0/16 is directly connected, Loopback1
172.31.0.0/32 is subnetted, 1 subnets
O 172.31.1.1 [110/65] via 192.168.1.2, 00:01:33, Serial0/0
172.30.0.0/32 is subnetted, 1 subnets
O 172.30.1.1 [110/65] via 192.168.1.2, 00:01:33, Serial0/0
192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Serial0

RouterA#

You can issue a show ip protocols command to check on the OSPF configuration:

RouterA#show ip protocols

Routing Protocol is ospf 20

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 172.20.1.1

Maximum path: 4

Routing for Networks:

172.16.0.0 0.0.255.255 area 0

172.20.0.0 0.0.255.255 area 0

192.168.1.0 0.0.0.3 area 0

Routing Information Sources:

Gateway	Distance	Last Update
172.31.1.1	110	00:05:48
172.20.1.1	110	00:05:48

Distance: (default is 110)

6. To test connectivity you will need to use the ping command, and to log in to the neighbor's router you will need to use the telnet command:

RouterA#ping 172.30.1.1 **This will send a ping packet to the address specified; there should be five replies if everything is OK.**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms

RouterA#

RouterA#telnet 172.31.1.1 **This will open a Telnet connection to the neighbor's router. If Telnet access has been set up correctly you will be presented with a login message.**

RouterA#telnet 172.31.1.1

Trying 172.31.1.1 ... Open

User Access Verification

Username: banbury

Password:

RouterB>exit

[Connection to 172.31.1.1 closed by foreign host]

RouterA#

RouterA#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.31.1.1	1	FULL/ -	00:00:29	192.168.1.2	Serial0

Test the following commands also:

show ip ospf database

show ip ospf interface

debug ip ospf packet

Do the same with Router B:

RouterB#ping 172.16.1.1

```
RouterB#ping 172.20.1.1  
RouterB#telnet 172.16.1.1
```

7. Now please enter reload at the Router# prompt and type yes.

Show Runs

```
RouterA#show run
```

```
Building configuration...
```

```
!  
version 15.1  
!  
hostname RouterA  
!  
enable secret 5 $1$rujI$BJ8GgiK8U9p5cdfXyApPr/  
!  
username banbury password 0 ccna  
!  
interface Loopback0  
ip address 172.16.1.1 255.255.0.0  
!  
interface Loopback1  
ip address 172.20.1.1 255.255.0.0  
!  
interface Serial0/0  
ip address 192.168.1.1 255.255.255.252  
clockrate 64000  
!  
router ospf 20  
log-adjacency-changes  
network 172.16.0.0 0.0.255.255 area 0  
network 172.20.0.0 0.0.255.255 area 0  
network 192.168.1.0 0.0.0.3 area 0  
!  
ip classless  
no ip http server  
!  
line con 0  
password letmein  
login
```

```
line aux 0
line vty 0 4
login local
!
end
```

```
RouterB#show run
Building configuration...

Current configuration : 853 bytes
!
version 15.1
!
hostname RouterB
!
enable secret 5 $1$ydeA$MyfRKeV0ckjm7w/0ornnB1
!
username banbury password 0 ccna
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
router ospf 20
log-adjacency-changes
network 172.30.0.0 0.0.255.255 area 0
network 172.31.0.0 0.0.255.255 area 0
network 192.168.1.0 0.0.0.3 area 0
!
ip classless
no ip http server
!
line con 0
```

```
password letmein
login
line aux 0
line vty 0 4
login local
!
end
```

Chapter 7 — IP Services

What You Will Learn in This Chapter

DHCP Functionality

Access Control Lists

Network Address Translation

Network Time Protocol

Syllabus Topics Covered

5.0 IP Services

5.1 Configure and verify DHCP (IOS router)

5.1.a Configuring router interfaces to use DHCP

5.1.b DHCP options (basic overview and functionality)

5.1.c Excluded addresses

5.1.d Lease time

5.2 Describe the types, features, and applications of ACLs

5.2.a Standard (editing and sequence numbers)

5.2.b Extended

5.2.c Named

5.2.d Numbered

5.2.e Log option

5.3 Configure and verify ACLs in a network environment

5.3.a Named

5.3.b Numbered

5.3.c Log option

5.4 Identify the basic operation of NAT

5.4.a Purpose

5.4.b Pool

5.4.c Static

5.4.d 1-to-1

5.4.e Overloading

5.4.f Source addressing

5.4.g One-way NAT (PAT)

5.5 Configure and verify NAT for given network requirements

5.6 Configure and verify NTP as a client

6.0 Network Device Security

6.3 Configure and verify ACLs to filter network traffic

This chapter focuses on some of the Internet Protocol services that can be provided by a router. You will learn about how a router can be used to enable automatic assignment of IP addresses via DHCP, how to increase security in the network using IP access control lists, how the Network Address Translation (NAT) feature has been used to preserve IPv4 addressing until now, and how to configure Network Time Protocol.

DHCP Functionality

Dynamic Host Configuration Protocol (DHCP) is used to automatically assign IP information to hosts in a network. It reduces the burden for network administrators, speeds up device configuration, and reduces the chances of a configuration mistake, especially allocating the same IP address to more than one machine, which will cause network issues. The information provided by the DHCP servers includes IP addresses, subnet masks, default router, DNS servers, and other parameters.

DHCP was created to address the many shortfalls of protocols, such as BOOTP and RARP. DHCP is designed to work with IPv4. DHCPv6 for IPv6 was first described in RFC 3315 in 2003; however, it has been updated many times subsequently in other RFCs. We will look at DHCPv6 briefly later but it isn't specifically mentioned in the CCNA RS syllabus.

When the host first boots up, if it has been enabled to use DHCP (which is the default in most operating systems), it will send a broadcast message asking for its IP information. This broadcast message is heard by all the DHCP servers on its subnet and the DHCP servers respond with some configuration information.

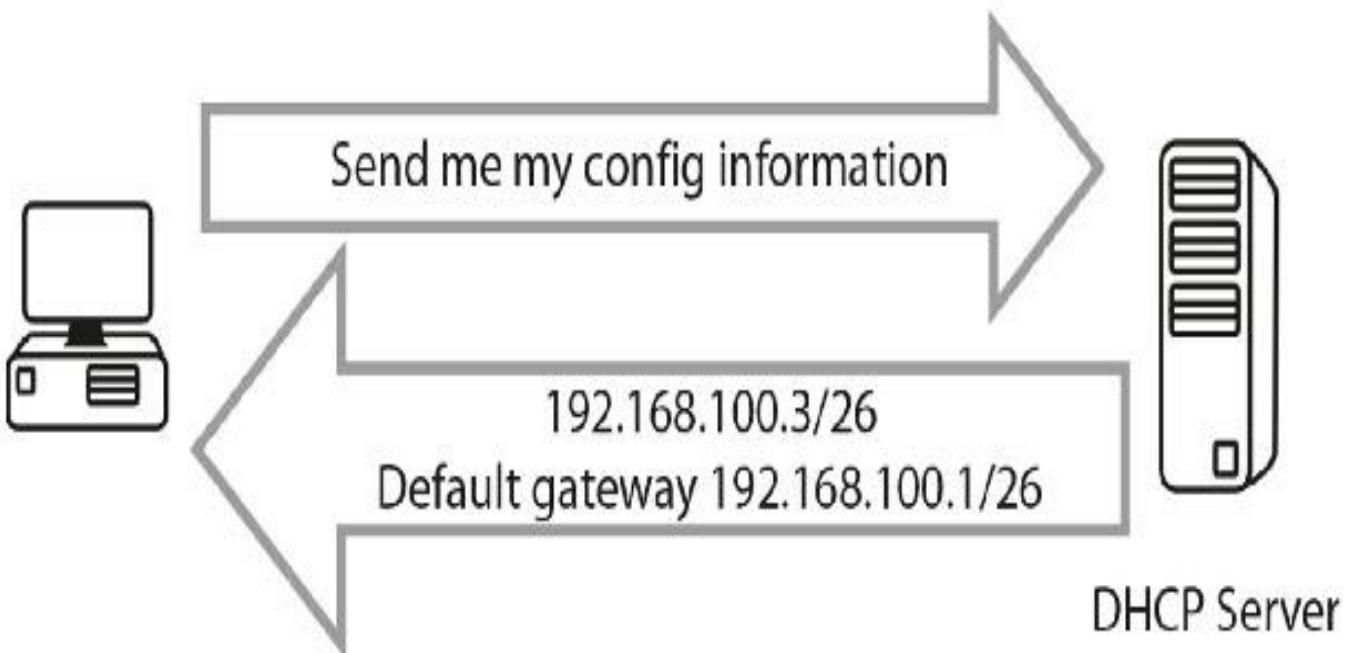


FIG 7.1 – Host requests IP configuration information

DHCP uses UDP ports 67 and 68 to communicate over the network. The device that needs an IP address is the DHCP client, while the device that provides the IP address is the DHCP server. In most networks, the DHCP service is run on a server, but a router can also be configured to provide this service. In fact, a router can be configured as a DHCP client or a DHCP server. To configure a router as a DHCP client (to obtain its IP address from a server), the interface-level command to use is `Router(config-if)#ip address dhcp`.

DHCP clients go through the following states:

1. Initializing
2. Selecting
3. Requesting
4. Bound
5. Renewing
6. Rebinding

IP addresses are allocated to a specific host by the DHCP server for a period of time. This period is called the lease time and it can be expressed in hours or days. DHCP servers should also exclude addresses from the DHCP pool that they don't want to assign to the host. Those IP addresses are assigned statically and are used on routers and server interfaces. Failure to exclude reserved IP addresses from the DHCP pool can lead to duplicate IP addresses being assigned to hosts.

The process for obtaining an IP address via DHCP is described in Figure 7.2 below:

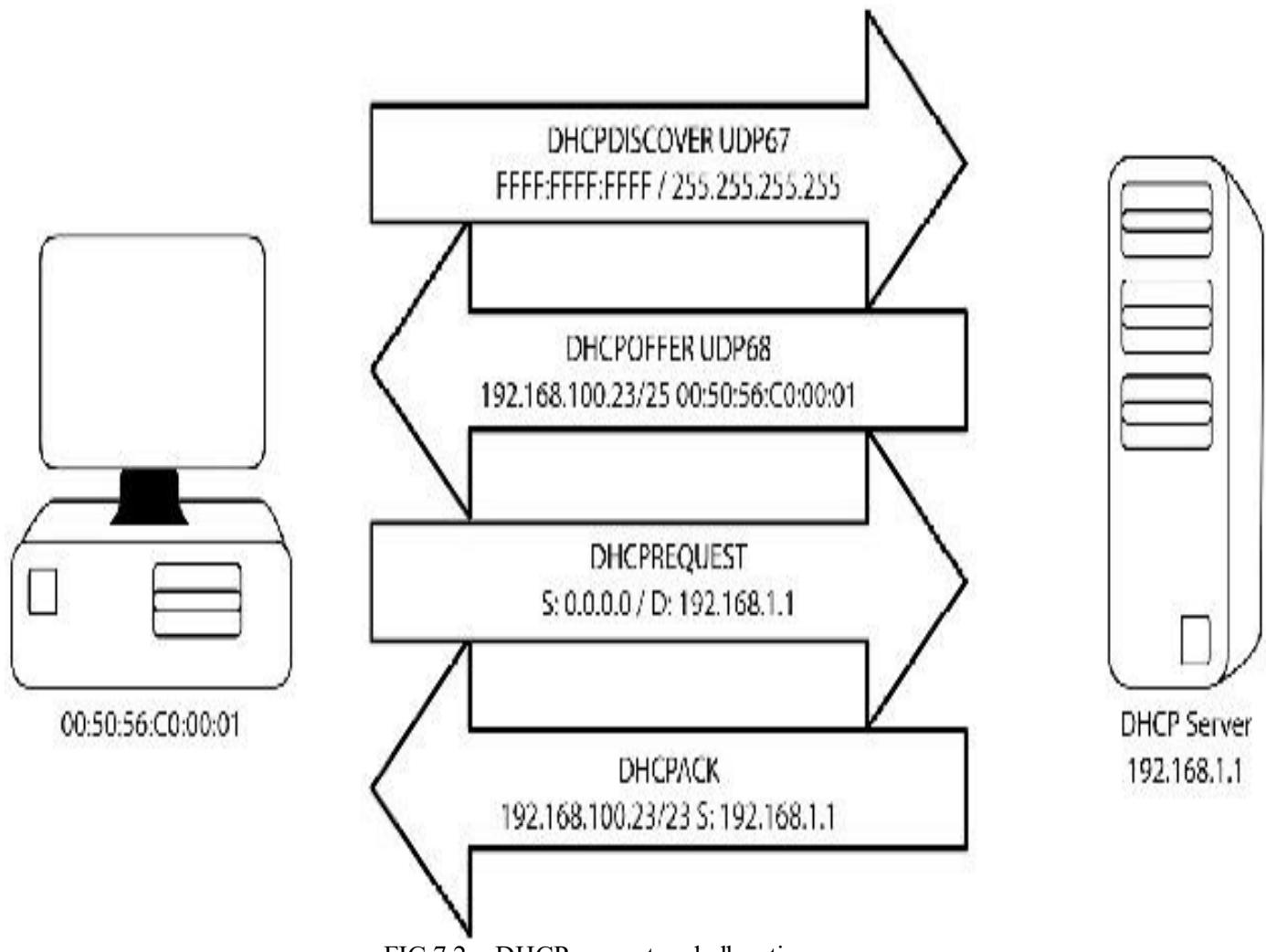


FIG 7.2 – DHCP request and allocation process

1. **DHCP Discover** – When a DHCP-enabled device boots up, it sends a broadcast out to UDP port 67. The broadcast packet will reach every device on its subnet, including any possible DHCP servers located there.
2. **DHCP Offer** – The DHCP servers on the subnet see the DHCP Discover message sent by the client and sends back a response (DHCP Offer packet). This is also a broadcast packet since the client still doesn't have an IP address yet, so it cannot receive unicast packets. The DHCP Offer packet contains the IP Information (address, subnet mask, gateway, etc.) that the client can use.
3. **DHCP Request** – Once the client workstation receives the offer made by the DHCP server, it will send a broadcast DHCP Request message to a specific DHCP server. The client might have received offers from multiple DHCP servers but it only needs a single IP address, so it must choose a DCHP server (based on an identifier) and this is usually done on a first-come-first-served basis. This broadcast also lets all the other DHCP servers know that it has selected an offer so they can stop sending offers.

4. DHCP Ack – The DHCP server sends another broadcast message, DHCP Ack, to confirm the IP address allocation to that specific client.

Figure 7.3 below shows a DHCP packet capture. For this capture I configured a router as a DHCP server with the address 192.168.1.1 and an address pool of 192.168.1.0/24, but I excluded IP address 192.168.1.1 because it was in use on the Fast Ethernet interface. The requesting router interface is offered the first available address from the pool, which is 192.168.1.2.

0 28.759187	0.0.0.0	255.255.255.255	DHCP	618 DHCP DISCOVER - Transaction ID 0x63c
10 28.769548	c2:03:08:f7:00:00	Broadcast	ARP	60 Who has 192.168.1.2? tell 192.168.1.1
11 30.004606	c2:03:08:f7:00:00	c2:03:08:f7:00:00	LOOP	60 Reply
12 30.773551	192.168.1.1	255.255.255.255	DHCP	342 DHCP Offer - Transaction ID 0x63c
13 30.797786	0.0.0.0	255.255.255.255	DHCP	618 DHCP Request - Transaction ID 0x63e
14 30.009255	192.168.1.1	255.255.255.255	DHCP	342 DHCP ACK - Transaction ID 0x63e
15 30.818861	c2:04:08:17:00:00	Broadcast	ARP	60 Gratuitous ARP for 192.168.1.2 (Reply)
16 31.371044	c2:04:08:17:00:00	c2:04:08:17:00:00	LOOP	60 Reply

P	LINKSUM: 0x3284 (VAL130J.LDN 0193D1C0)
⇒ Bootstrap Protocol	
Message type: Doot Reply (2)	
Hardware type:	Ethernet
Hardware address length:	6
Hops:	0
Transaction ID:	0x6D90C96e
Seconds elapsed:	0
⇒ Bootp flags: 0x8000 (Broadcast)	
Client IP address:	0.0.0.0 (0.0.0.0)
Your (client) IP address:	192.168.1.2 (192.168.1.2)
Next server IP address:	0.0.0.0 (0.0.0.0)
Relay agent IP address:	0.0.0.0 (0.0.0.0)
Client MAC address:	c2:04:08:f7:00:00 (c2:04:08:f7:00:00)
Client hardware address padding:	00000000000000000000
Server host name not given	
Doot file name not given	
Magic cookie:	DHCP
⇒ Option: {53} DHCP Message Type	

FIG 7.3 – DHCP Offer capture

It's worth noting that on the host router (R2 below), you will see a message telling you that you have been allocated an IP address via DHCP, and you can issue the show ip interface brief command to see that the method column is set to DHCP:

```
R2#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R2(config)#int f0/0
```

```
R2(config-if)#ip address dhcp
```

```
R2(config-if)#no shut
```

```
R2(config-if)#end
```

```
*Mar 1 00:03:45.387: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0  
assigned DHCP address 192.168.1.2, mask 255.255.255.0, hostname R2
```

```
R2#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Fa0/0	192.168.1.2	YES	DHCP	up	up
Fa0/1	unassigned	YES	unset	administratively down	down

```
R2#
```

We will cover the full IOS commands shortly.

Besides the IP address, the server can also supply other parameters in the DHCP Offer package, including:

- Subnet mask
- Lease duration
- Default gateway
- Domain Name Server

The lease duration in the DHCP Offer specifies the amount of time a host can use its assigned IP address before sending back a request to the DHCP server (IP address refresh). The DHCP server will usually allocate the same IP address to the client. When a DHCP client leaves the network or its lease period expires, the associated IP address is handed back to the DHCP server, which can assign it to another host.

DHCP servers can also be configured to assign specific addresses to certain hosts based on their MAC address. This feature is especially useful when configuring servers to use DHCP addresses (although the recommendation in this case is to manually assign IP addresses). This is because users usually identify a server based on a fixed IP address and we don't want that IP address to change over time.

Cisco IOS-based routers can be configured as either DHCP servers or DHCP clients (per interface).

DHCP client functionality for routers (as opposed to host computers) is not very common. This is because routers are critical infrastructure devices and the recommendation is to address them manually instead of dynamically so their IP addresses are always known and fixed.

DHCP server functionality on Cisco IOS routers involves a few steps:

- Enable the DHCP service (usually enabled by default)
- Assign a static IP on the LAN-facing interface you want to enable the DHCP service on
- Create a DHCP pool
- Define DHCP parameters within the DHCP pool
- Define excluded DHCP addresses
- Monitor correct DHCP address assignment

Mini-lab – DHCP Configuration on Cisco IOS Routers

Let's work on an example based on the topology in Figure 7.4 below:

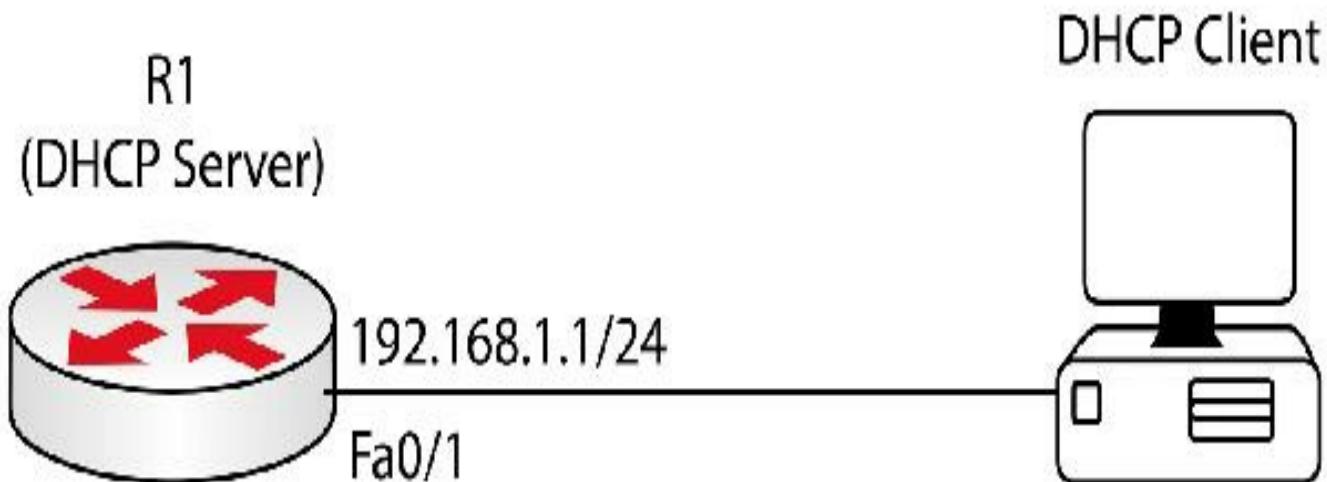


FIG 7.4 – Mini-lab: DHCP Configuration on Cisco IOS Routers

Start by enabling the DHCP service on Router 1:

```
R1(config)#service dhcp
```

Next, assign the 192.168.1.1/24 address on interface Fast Ethernet 0/1, facing the DHCP clients:

```
R1(config)#interface FastEthernet0/1
```

```
R1(config-if)#ip add 192.168.1.1 255.255.255.0
```

All the DHCP parameters that you want to transmit to the clients will be configured in DHCP configuration mode:

```
R1(config)#ip dhcp pool CCNA
```

```
R1(dhcp-config)#network 192.168.1.0 255.255.255.0
```

```
R1(dhcp-config)#default-router 192.168.1.1
```

```
R1(dhcp-config)#lease 1
```

```
R1(dhcp-config)#dns-server 8.8.8.8
```

Only the first two commands are compulsory. You have configured the DHCP pool so it will offer addresses from the 192.168.1.0/24 subnet, with a default gateway of 192.168.1.1 (the router interface). You have also configured a lease time of one day and the DNS server IP address.

Finally, configure the router so that it will not offer all the IP addresses from the 192.168.1.0/24 range. The DHCP client allocations in this example should start at 192.168.1.11 and finish at 192.168.1.250. There might be multiple reasons behind this logic but the most obvious one is that you want to statically assign the first and last addresses from the range to servers and network devices. The configuration to accomplish this is as follows:

```
R1(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

```
R1(config)#ip dhcp excluded-address 192.168.1.251 192.168.1.254
```

Please note that the excluded addresses are configured in configuration mode, not in DHCP configuration mode.

After the DHCP client broadcasts a DHCP Discover on the link, the router sees it come in on interface Fast Ethernet 0/1 and associates the request with DHCP pool CCNA (based on the configured network statement). It will then offer the DHCP client the first available IP address in the pool, which is 192.168.1.11/24.

Check the PC to ensure that the address has been allocated. Alternatively, if you have another router on the end of the connection, use the ip address dhcp command on the interface and then show ip interface brief. Remember to issue the no shut command on the interface.

[END OF MINI-LAB]

It's worth noting that DHCP Discover is used by the router to check whether the next IP address in the pool can be used. The router ARPs for the address and if for some reason it has been allocated (say by a user who didn't know DHCP was in use), the address would be removed from the pool and placed into a conflict table until the issue was

resolved.

```
Router#sh ip dhcp conflict
```

IP address	Detection method	Detection time	VRF
10.10.12.2	Ping	July 01 2015 12:01 AM	

You can verify the correct DHCP address assignment using the show ip dhcp binding command.

You can expect to be asked to configure a router with DHCP in the exam, or to be presented with various configuration commands and asked to choose which are necessary in order to achieve the desired result, so make sure that you can configure the commands above from memory. You also need to know how to configure a router to be a DHCP server as well as a client.

IP Helper Address

Routers do not forward broadcast traffic, by default. There are instances where you may want a broadcast packet to pass through the router, for example, when a client sends a DHCP Request to obtain an IP address and the DHCP server is on the other side of your router on another network. In this instance, you can use the ip helper-address command, which will turn the broadcast packet into a unicast packet sent to the IP address configured:

```
RouterA#config t  
RouterA(config)#interface FastEthernet0/0  
RouterA(config-if)#ip helper-address 192.168.1.1
```

The ip helper-address command will automatically forward eight common UDP ports that use broadcasts:

- Time (37)
- TACACS (49)
- DNS (53)
- BOOTP server (67)
- BOOTP client (68)
- TFTP (69)
- NetBIOS name service (137)
- NetBIOS datagram service (138)

Figure 7.5 below demonstrates how you would use the command:

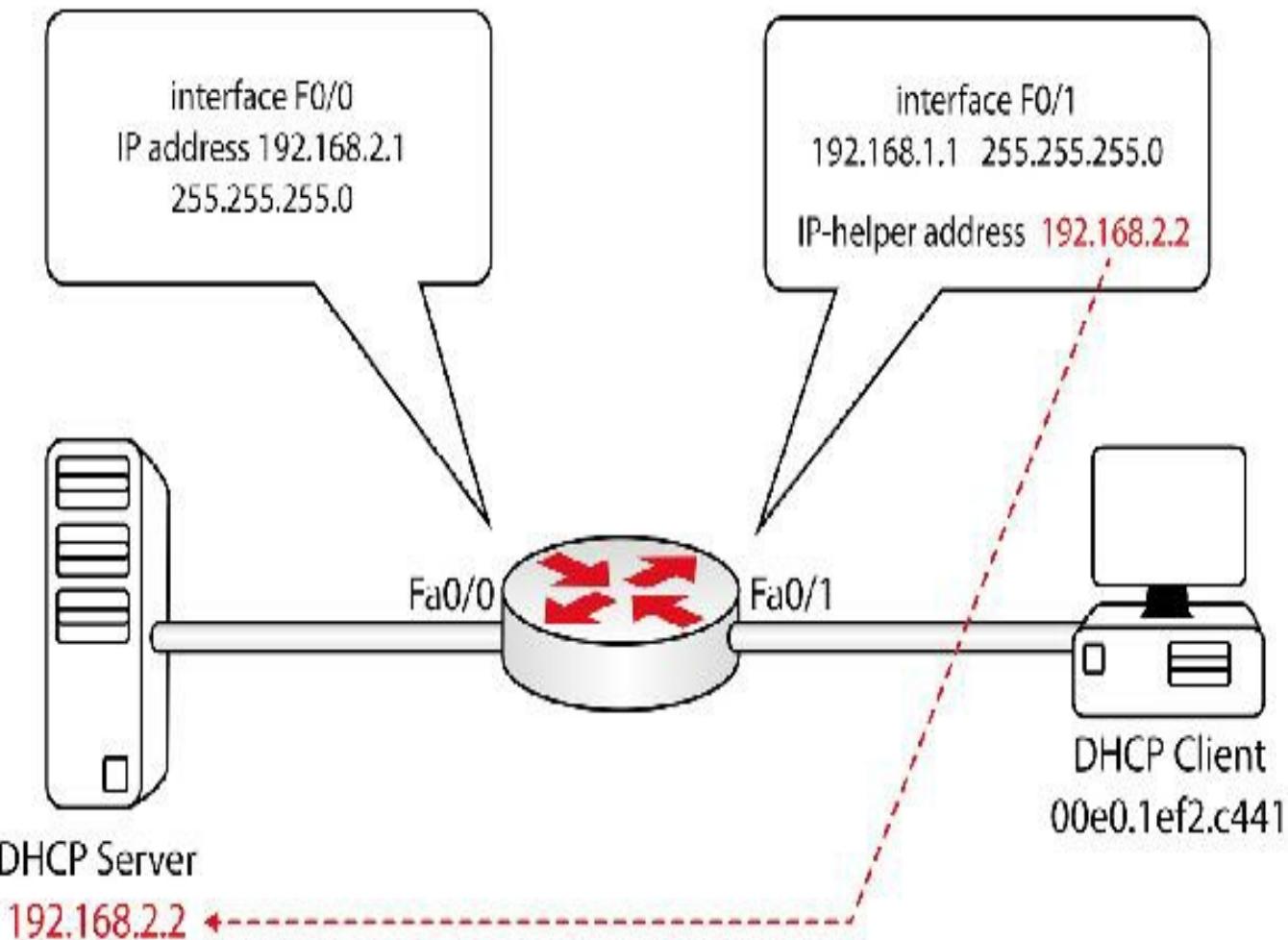


FIG 7.5 – IP helper address in operation

In the exam you might be presented with an issue about DHCP Requests being stopped at the router and then asked how you would fix the issue. Well now you know!

IP Forward Protocol

What if you wanted to forward a protocol not on the list above? The ip forward-protocol command allows you to specify a number of protocols and ports that the router will forward. The output below specifies an IP helper address along with which protocol should be forwarded:

```
RouterA#config
RouterA(config)#interface FastEthernet0/0
RouterA(config-if)#ip helper-address 192.168.1.1
RouterA(config-if)#ip forward-protocol udp
```

As with all Cisco IOS commands, you can Google them to do further research because many of them have multiple permutations. There are so many, in fact, that in the CCIE lab exam, Cisco gives you access to Command References and Configuration Guides on

CD ROM. Bear in mind that some commands were introduced in later versions of IOS and these may not be available to you. Some have also been deprecated over time for business or security reasons.

You can access Cisco's feature navigator tool for IOS to check which services and commands are available for your model of devices as well as your IOS release. If your company has a contract with a Cisco TAC (Technical Assistance Center), they can do this for you or your presales advisor can.

Access Control Lists

The most fundamental method to protect your network (after passwords) is to decide which traffic can enter and leave your router. Access control lists (ACLs) are a set of filters that the traffic is checked against. When the traffic matches the access list it can be either permitted or denied, depending on what you have configured.

Although access lists are relatively straightforward to configure, they can cause problems for many junior network engineers. This is because there are some commands at the bottom of any access list that exist but cannot be seen (these are known as implicit commands). Also, access lists operate by a certain set of special rules, and if you are not aware of these rules your access list will not work or will work only part of the time.

Cisco has made changes to ACL rules and features as IOS versions have been released so you will find that what applies to 12.3, for example, may not apply to 12.4. I'll explain the main upgrades as we progress in this lesson. You will be tested on IOS version 15.X in the exam.

As with any skill, the more you configure commands the better you will understand them. I strongly advise you to type all of the commands discussed onto a router (not all of them will work on Packet Tracer). You will make mistakes and get frustrated but eventually it will become second nature to you.

As traffic reaches the router interface, it's checked against the access list and if it is permitted it is routed and then sent to the outbound interface. To see the full list of the order in which the router processes incoming and outgoing traffic, please Google "Cisco router order of operations"; however, for the purposes of the CCNA exam you don't need to know it.

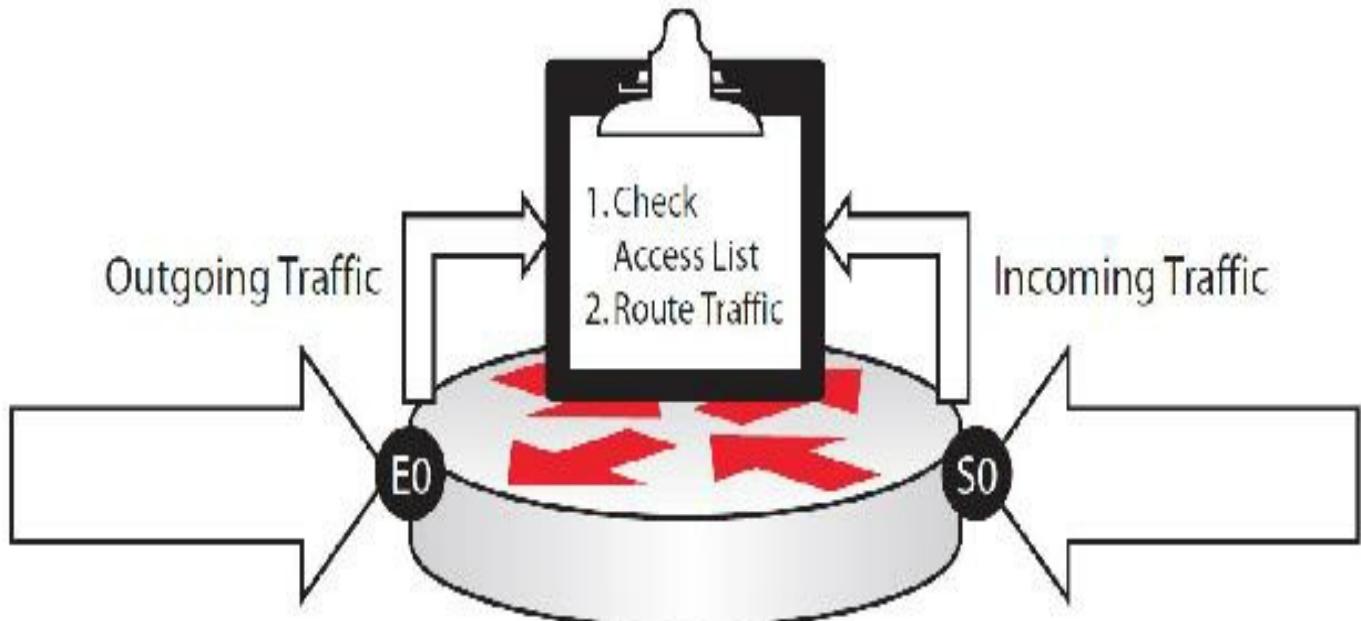


FIG 7.6 – Access list order of operations

IP access lists come in two main varieties—standard and extended. Standard access lists are very simple to configure but do not allow any granularity (they only match based on source IP addresses). Extended access lists are far more flexible but they are more complex to configure. Both standard and extended ACLs can be configured with names instead of numbers to make them easier to manage.

Access List Numbers

There are many ranges of access list numbers based on protocol type. In order for the router to understand which one you want to configure, you have to use the correct range. For example, to configure IP standard access lists, use any number from 1 to 99 (or 1300 to 1999). For the sake of brevity, I've only included the IP ranges (leaving out Appletalk, IPX, etc., which are now defunct).

Table 7-1: Access list ranges

Protocol	Range
Standard IP	1 to 99
Standard IP (expanded range)	1300 to 1999
Extended IP	100 to 199
Extended IP (expanded range)	2000 to 2699

Standard IP Access Lists

Standard IP access lists filter traffic based on the IP source address of the traffic only.

They do not filter web, e-mail, or any other variety of traffic or filter based on destination.

The access list is given a number to identify the type of list it is (see the access list ranges above), configured to permit or deny traffic and then configured with the parameters of which traffic to permit or deny. Many people then think this is all they need to do. You actually have to apply this access list to an interface. Otherwise, the access list will just sit there doing nothing in particular and all traffic will be allowed through the router.

All traffic not included in the access list is denied. Even if you do not enter a deny any command at the end of the list, it will do this automatically (unless you configure it to permit everything). The rationale is that if you are permitting certain traffic, surely you want to deny everything else! This is known as the implicit deny statement and you need to **remember this ALWAYS!**

The command syntax for a standard access list is:

```
access-list number [deny | permit] source-prefix source-wildcard log
```

Wildcard Masks

The wildcard mask can cause confusion for anybody new to access lists. Just remember that the router is working in binary instead of decimal. The wildcard mask is there to tell the access list which parts of the address to look at. A 1 in binary means that part of the address can be ignored, while a 0 means that it must match.

Example

If you want to match all traffic from 172.16.2.x, then you would add the wildcard mask 0.0.0.255, or in binary:

10101100.00010000.00000010.00000000 = 172.16.2.0

00000000.00000000.00000000.11111111 = 0.0.0.255

Match Match Match Ignore

In action, this would mean that any host from the network starting with 172.16.2.x would match the access list, but 172.16.3.x would not match the access list.

Example

The output below shows an access list permitting any traffic from the 10.x.x.x network:

```
access-list 9 permit 10.0.0.0 0.255.255.255
```

Remember that there is an implicit deny any at the end of the access list, meaning that

when you apply this to an interface the only permitted network will be 10.x.x.x.

access-list 9 permit 10.0.0.0 0.255.255.255

access-list 9 deny 0.0.0.0 255.255.255.255 **i This is present, but you will not see it.
This is the implicit deny any.**

Example

access-list 12 permit 172.16.2.0 0.0.0.255 **i 172.16.2.x allowed**

access-list 12 permit 192.168.1.0 0.0.0.255 **i 192.168.1.x allowed**

access-list 12 permit 10.4.0.0 0.0.255.255 **i 10.4.x.x allowed**

The access list above permits three networks and implicitly denies traffic from any other network.

Example

access-list 15 deny 172.16.0.0 0.0.255.255

access-list 15 deny 192.168.2.1 **i You can specify a host address**

access-list 15 permit any

When you want to deny a few networks, subnets, or hosts and permit the rest, use the logic shown above. You are denying anything from the 172.16.x.x network but you can also specify a single host without using a wildcard mask. Just enter the host number and the router will add an automatic 0.0.0.0 to it. In the example above you can see that host 192.168.2.1 is denied.

Finally, you can add permit any to the end of the list to prevent the implicit deny rule from denying any other traffic. If you forget this line all traffic will be denied anyway.

You can break down wildcard masks from the default subnet boundaries just as you can use VLSM to change the default subnet mask for an IP address.

Example

If you wanted to deny the 192.168.100.96 255.255.255.224 subnet, you would use the following wildcard mask:

0.0.0.31

This would make more sense written out in binary:

1.1.1.1.1.1.1 1.1.1.1.1.1.1 1.1.1.1.1.1.1 1.1.1.0.0.0.0.0 = 255.255.255.224

0.0.0.0.0.0.0 0.0.0.0.0.0.0 0.0.0.0.0.0.0 0.0.0.1.1.1.1 = 0.0.0.31

As you can see, the access list matches the first 27 bits, and the last five can be any bits. The simplest way to look at it is to swap each “on” bit to an “off” bit when you are

writing out the wildcard mask.

Example

10.1.64.0 255.255.192.0

If you wanted to permit or deny this subnet, you would need to create a specific wildcard mask to match that subnet. The wildcard mask needs to be the reverse of the subnet mask to permit or deny this subnet.

1.1.1.1.1.1.1.1 1.1.1.1.1.1.1.1 1.1.0.0.0.0.0.0 0.0.0.0.0.0.0 = 255.255.192.0

0.0.0.0.0.0.0 0.0.0.0.0.0.0 0.0.1.1.1.1.1.1 1.1.1.1.1.1.1 = 0.0.63.255

If you add the two columns together below and get 255, you know that the wildcard mask is correct (as also shown in the examples above).

Subnet mask	255	255	192	0
Wildcard mask	0	0	63	255
Equals	255	255	255	255

Example

Which wildcard mask would deny the subnet 172.16.32.0 255.255.240.0?

Subnet mask	255	255	240	0
Wildcard mask	0	0	15	255
Equals	255	255	255	255

As you can see above, the wildcard mask 0.0.15.255 would deny subnet 172.16.32.0.

Example

I actually came across one very tricky example during my CCIE lab exam attempt back in 2002 that threw me off at the time. In this example, you have to match both 10.0.0.0/24 and 10.0.1.0/24. The last octet won't matter so you can use 255 for the wildcard mask; however, the third octet has two subnets to contend with.

You need to use the wildcard mask 0.0.1.255. In order for this to make any sense, write down all three addresses in binary:

00001010.00000000.00000000.00000000 = 10.0.0.0

00001010.00000000.00000001.00000000 = 10.0.1.0

0000000.0000000.0000001.1111111 = 0.0.1.255

From the output above, you can see that only the first 23 bits have to match. This means that all addresses in the range of 10.0.0.0 to 10.0.1.255 will match, which matches both subnets 10.0.0.0 and 10.0.1.0. This was achieved simply by adding the “don’t care” bit to the end of the third octet.

Wildcard masks are commonly used with OSPF to advertise specific subnets. For the exam it would be time well spent to practice your wildcard masks for various subnets, such as 192, 224, 240, etc.

Access List Logging

There is an optional log command that you can add to the end of an access list. This allows any matches to the access list to be logged to the console session or router memory in case you want to check for hacking attacks or configuration problems.

```
access-list 15 deny 172.16.0.0 0.0.255.255 log
```

If you added the logging buffered command to the configuration, the output would be added to the router’s memory buffer. This could be interrogated later to investigate access list violations.

Logging access lists consumes CPU cycles so use the command with caution.



Extended IP Access Lists

Many administrators find that they need a lot more flexibility when locking down their networks. Filtering based on only traffic source addresses is very limited. Extended access lists can filter based on source addresses, destination addresses, protocols, port numbers, and other features (see the explanation of port numbers in the following section).

Please do type out these commands on a router as you see them. Your output and options may differ slightly due to IOS versions but the core function will remain the same.

The syntax for a TCP access list is shown below. As you will see, there are also IP, UDP, and ICMP access lists and the syntax is slightly different for each.

```
access-list access-list-number [dynamic dynamic-name [timeout minutes]]  
[deny | permit] tcp source source-wildcard [operator [port]] destination destination-  
wildcard [operator [port]] [established][precedence precedence] [tos] [log | log-  
input]
```

- number – extended access lists can be from 100 to 199
- dynamic – creates temporary entries in the access list
- timeout – how long the dynamic list is activated for
- deny/permit – the list denying or permitting traffic
- tcp – you can deny a tcp port or all tcp traffic
- source – source host or network
- destination – destination host or network
- operator – can be greater than, less than, equal to
- port – port number
- established – only permitted traffic instigated from inside the network
- precedence – filters based on the precedence levels 0 to 7
- tos – filters based on service levels 0 to 5 or by name
- log – logs the access list violations

The access list looks at the source address going to the destination address (i.e., the first address is from and the second address is to).

It is worth briefly mentioning the established command in more detail. Perhaps you want to allow certain types of traffic into your network but only if your internal users have started the session or requested the service. The established keyword checks for the ACK (acknowledged) bit in the packet. If the ACK or RST (reset) bit is set, then this indicates that the packet is a response packet to a TCP request made from your network and thus it is safe. This, obviously, only works with TCP packets. Figure 7.7 below shows a capture revealing the ACK and RST fields:

762 3140.156659 192.168.1.2	192.168.1.1	TCP	00 telnet > 48203 [SYN, ACK] Seq: 0 Ack: 1 Win: 128 Len: 8
763 3140.167788 192.168.1.1	192.168.1.2	TCP	00 48203 > telnet [ACK] Seq: 1 Ack: 1 Win: 128 Len: 8
764 3140.177915 192.168.1.1	192.168.1.2	TELNET	03 Telnet Data ...
765 3140.178001 192.168.1.2	192.168.1.1	TELNET	00 Telnet Data ...
766 3140.188156 192.168.1.1	192.168.1.2	TCP	00 [TCP Dup ACK 761v1] 48203 > telnet [ACK] Seq: 1B Ack: 1
767 3140.188217 192.168.1.2	192.168.1.1	TELNET	00 Telnet Data ...
768 3140.200965 192.168.1.1	192.168.1.2	TELNET	00 Telnet Data ...
769 3140.201079 192.168.1.2	192.168.1.1	TELNET	00 Telnet Data ...
770 3140.214485 192.168.1.1	192.168.1.2	TELNET	00 Telnet Data ...
771 3140.214490 192.168.1.2	192.168.1.1	TELNET	00 Telnet Data ...

Transmission Control Protocol, Src Port: Telnet (23), Dst Port: 48203 (48203), Seq: 0, Ack: 1, Len: 8

Source port: Telnet (23)

Destination port: 48203 (48203)

[Stream index: 0]

Sequence number: 0 (relative sequence number)

Acknowledgment number: 1 (relative ack number)

Header length: 24 bytes

Flags: 0x012 (SYN, ACK)

- 000... = Reserved: Not set
- ...0 = Nonce: Not set
- ...0.... = Congestion Window Reduced (CWR): Not set
- ...0.... = ECN-Echo: Not set
- ...0.... = Urgent: Not set
- ...0...1 = Acknowledgment: Set ← Red arrow
- ...0....0... = Push: Not set
- ...0....0... = Reset: Not set ← Red arrow

FIG 7.7 – ACK and RST fields

It would probably help to have some examples now.

access-list 101 permit tcp any any

The command above permits TCP traffic from any network to any network.

access-list 102 permit udp host 20.0.2.1 any eq 53

The command above permits UDP traffic from host 20.0.2.1 to anywhere, provided that the port is equal to 53 (DNS).

access-list 101 permit tcp host 11.1.2.1 host 172.16.1.1 eq telnet

access-list 101 permit tcp host 11.1.1.2 host 172.16.1.1

access-list 101 permit udp host 15.2.1.5 host 172.16.1.1

access-list 101 permit ip 15.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255

The commands above permits Telnet traffic from host 11.1.2.1 to host 172.16.1.1 (you

(could have put 23 at the end instead of telnet), any TCP traffic from host 11.1.1.2 to host 172.16.1.1, any UDP traffic from host 15.2.1.5 to host 172.16.1.1, and any IP traffic from network 15.1.1.x to network 172.16.1.x. The implicit deny any at the end of the access list will deny any other traffic.

Port Numbers

Traffic flowing from network to network uses IP addressing to access information or a certain service or to carry out a certain task. In order for the network to understand what type of traffic is contained within the packet, a port number is used.

A port number identifies whether the service is web, e-mail, Telnet, name resolution, etc. There are literally thousands of port numbers available. The port numbers 0 to 1023 are called well-known numbers and they are reserved. There are 65,535 available port numbers in total.

Table 7.2: Common port numbers

Port Number	Service	Protocol
20	File Transfer Protocol-Data (FTP-Data)	TCP
21	File Transfer Protocol (FTP)	TCP
22	Secure Shell (SSH)	TCP
23	Telnet	TCP
25	Simple Mail Transfer Protocol (SMTP)	TCP
53	Domain Name System (DNS)	TCP/UDP
69	Trivial File Transfer Protocol (TFTP)	UDP
80	HyperText Transfer Protocol (HTTP)	TCP/UDP
110	Post Office Protocol (POP3)	TCP
119	Network News Transfer Protocol (NNTP)	TCP
123	Network Time Protocol (NTP)	UDP
161, 162	Simple Network Management Protocol (SNMP)	UDP
443	HTTP Secure (HTTPS)	TCP

You need to be aware that some ports are exclusive to TCP, some to UDP, and some are shared (such as DNS). You can either type in the port number or (depending on your IOS release) use the service name. I've removed most of the output to save space.

```
Router(config)#access-list 100 permit tcp any any eq ?
```

[0-65535] Port number
bgp Border Gateway Protocol (179)
domain Domain Name Service (53)
echo Echo (7)
ftp File Transfer Protocol (21)
ftp-data FTP data connections (used infrequently, 20)
pop2 Post Office Protocol v2 (109)
pop3 Post Office Protocol v3 (110)
smtp Simple Mail Transport Protocol (25)
telnet Telnet (23)
www World Wide Web (HTTP, 80)

Router(config)#access-list 100 permit udp any any eq ?

[0-65535] Port number
biff Biff (mail notification, comsat, 512)
bootpc Bootstrap Protocol (BOOTP) client (68)
bootps Bootstrap Protocol (BOOTP) server (67)
domain Domain Name Service (DNS, 53)
pim-auto-rp PIM Auto-RP (496)
rip Routing Information Protocol (router, in.routed, 520)
snmp Simple Network Management Protocol (161)
tftp Trivial File Transfer Protocol (69)
time Time (37)

You can find all the port numbers listed at:

<http://www.iana.org/assignments/port-numbers>

Access lists can also filter ICMP traffic (output truncated):

Router(config)#access-list 100 permit icmp any any ?
[0-255] ICMP message type
echo Echo (ping)
echo-reply Echo reply
log Log matches against this entry
log-input Log matches against this entry,
no-room-for-option Parameter required but no room
port-unreachable Port unreachable
precedence Match packets with given precedence
source-quench Source quenches
source-route-failed Source route failed

tos	Match packets with given TOS value
traceroute	Traceroute
ttl-exceeded	TTL exceeded
unreachable	All unreachables

You will only need to remember ICMP echo and echo-reply, which is the ping request and response using ICMP.

Access Lists and Routing Protocols

Access lists will permit the traffic you specify. It is vital to remember that if you are using routing protocols and access lists on your router that you permit the routing protocol as well.

To permit IGRP, specify:

```
access-list 101 permit igrp any any
```

To permit RIP, specify:

```
access-list 101 permit udp any any eq rip
```

To permit OSPF, specify:

```
access-list 101 permit ospf any any
```

To permit EIGRP, specify:

```
access-list 101 permit eigrp any any
```

Access List Rules

There are a few ACL rules that you need to be aware of for the exam and the real world. If you aren't aware of them then you will quickly run into configuration and troubleshooting issues.

ACL Rule 1 – Use Only One ACL Per Interface, Per Direction, and Per Protocol.

The Cisco IOS will not let you have more than one ACL per interface per direction, and, frankly, you don't need more than one. Each interface can have an ingress and egress policy, per layer 3 protocol. Since most modern networks use IP, you can only have a single IP ACL per direction. Think about it: you can compress the entries of multiple ACLs into one ACL, so you should never need more than one!

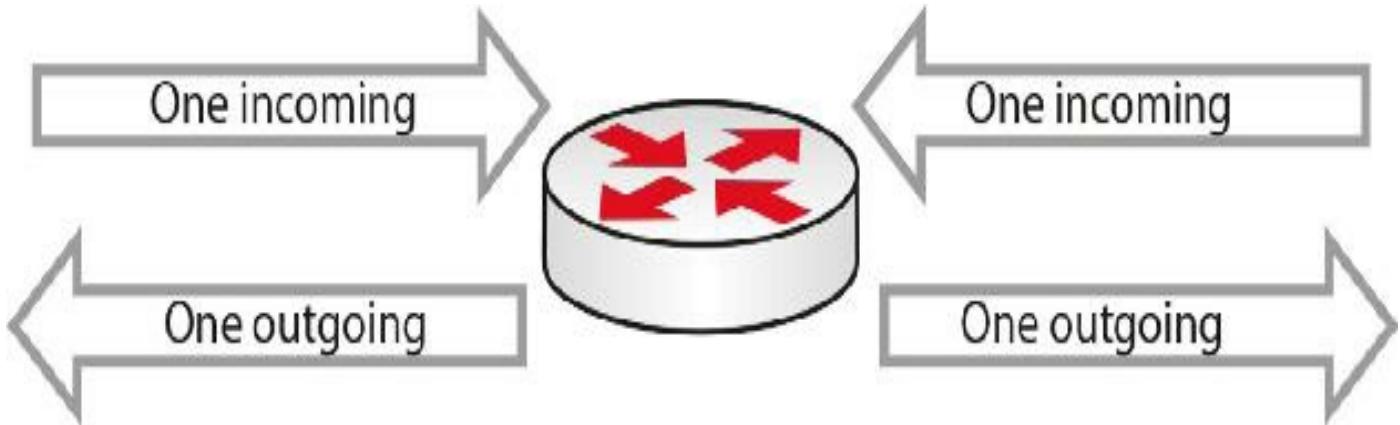


FIG 7.8 – One ACL per interface and per direction

ACL Rule 2 – The Lines Are Processed Top-Down.

It is really important to remember this. Always put more specific entries higher in the ACL. Since the router processes ACLs top-down, it will stop once it reaches a match. This can make some ACL entries redundant if another policy matching the same parameters exist above.

For example, take the ACL blocking host 172.16.1.5. Here is what will happen as the router checks each line:

1. Permit 10.0.0.0 – No match (move to next line)
2. Permit 192.168.1.5 – No match (move to next line)
3. Permit 172.16.0.0 – Match (permit packet and do not move to next line)
4. Permit 172.16.1.0 – Not processed
5. Deny 172.16.1.5 – Not processed

In the example above, the Deny 172.16.1.5 entry needs to be above the Permit 172.16.0.0 statement for it to have any effect.

ACL Rule 3 – There Is an Implicit Deny All at the Bottom of Every ACL.

If there is no explicit rule permitting a packet, then it is denied. The only way to override this is to configure a permit any entry at the bottom of the ACL.

ACL Rule 4 – The Router Can't Filter Self-Generated Traffic.

It is important to remember this rule when testing ACLs. Traffic generated from the router will not be affected by an ACL. This is demonstrated in Figure 7.9 below, which shows that a packet passing through the router is checked.

ACL - Deny 172.16.1.1

BLOCKED

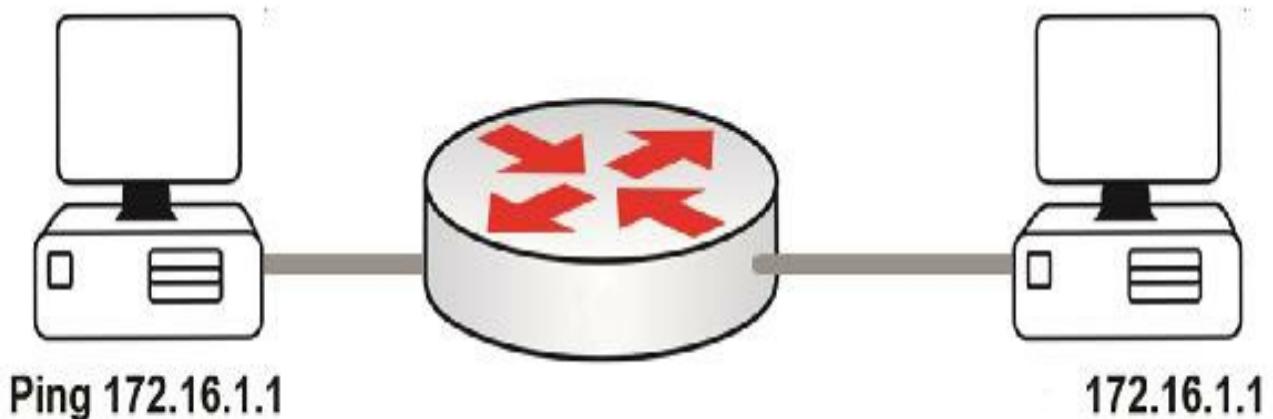


FIG 7.9 – Traffic passing through the router is checked

The next packet is generated by the router and not checked by the ACL:

ACL - Deny 172.16.1.1

UNCHECKED

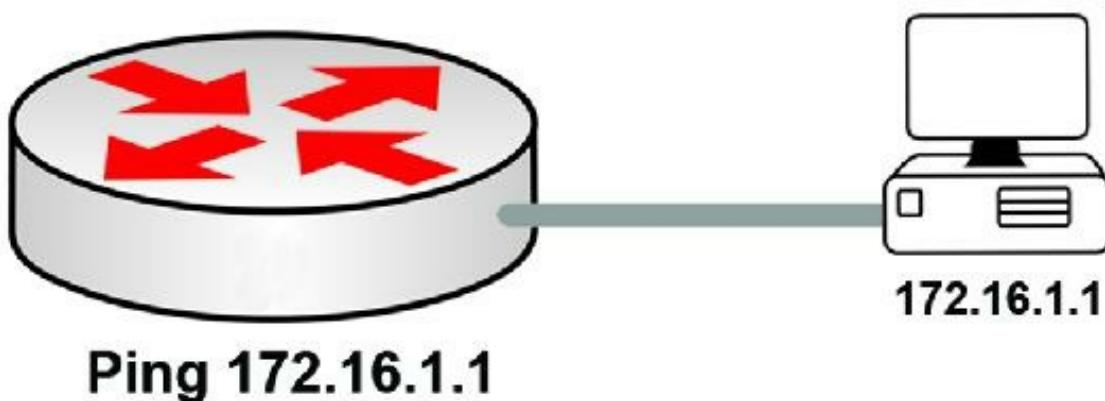


FIG 7.10 – Traffic generated by the router is not checked

ACL Rule 5 – You Can Edit a Live ACL.

Prior to IOS 12.4, the safe way to edit an ACL was to remove it, edit it in a text editor such as Notepad, and then reapply the ACL, as removing a line in the ACL would delete the entire ACL. After IOS 12.4, you can now edit a live ACL without first removing it.

ACL Rule 6 – You Can Disable the ACL on the Interface.

ACLs need to be applied on an interface to take effect. Similarly, removing the ACL from the interface disables it. You do not need to delete the entire ACL to disable it. This is useful since multiple interfaces might have the same ACL applied and removing the ACL disables it on all the interfaces. The configuration snippet below shows how to disable an ACL that was applied to interface Fast Ethernet 0/0:

```
Router(config)#int fast0/0  
Router(config-if)#no ip access-group 1 in  
Router(config-if)#^Z
```

ACL Rule 7 – You Can Reuse the Same ACL.

The same ACL can be applied to multiple interfaces. If you have the same ACL policy, there is no point in creating multiple ACLs; just reapply the same one to the multiple interfaces. This is common in large ISPs where a group policy on which traffic should be permitted into and out of the network is agreed upon. It is then applied to all gateway routers or firewalls. This is demonstrated in Figure 7.11 below:

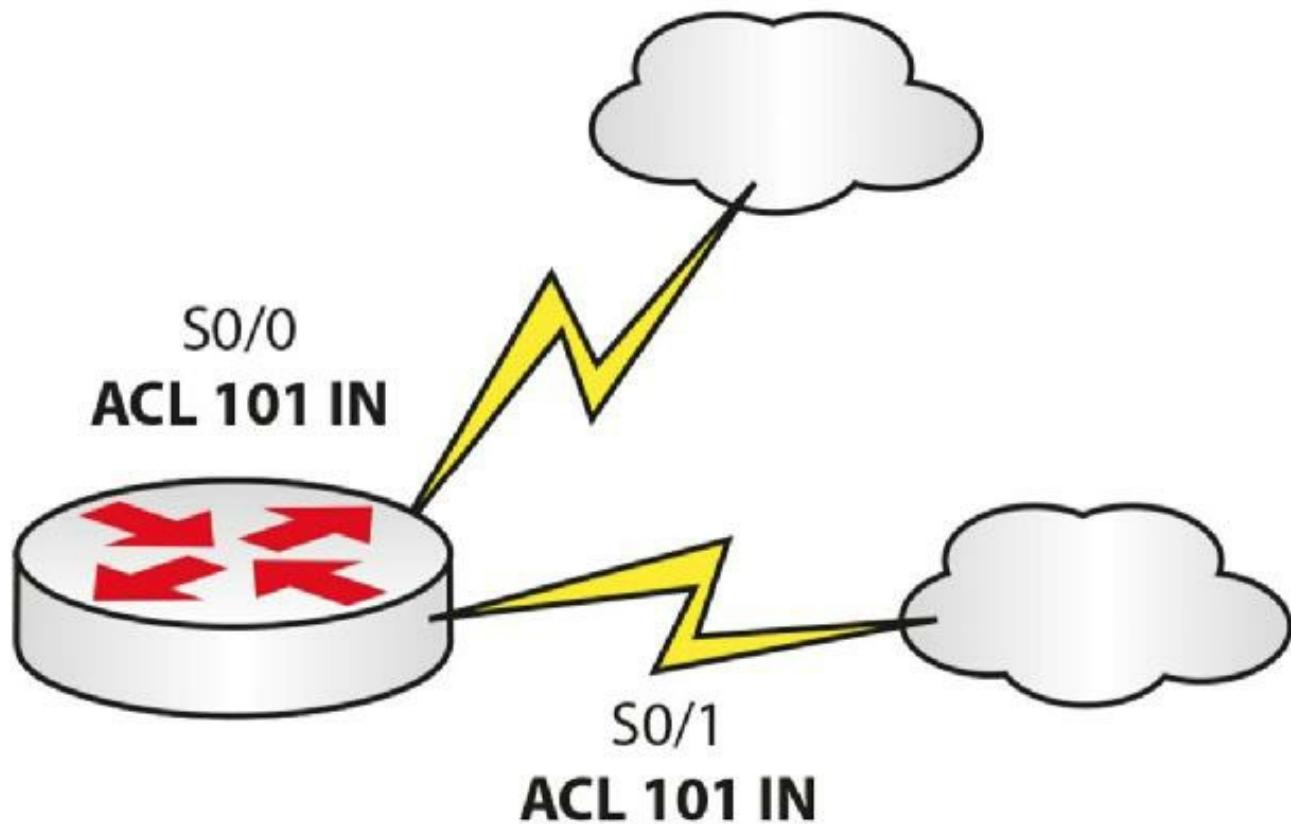


FIG 7.11 – You can reuse an ACL

ACL Rule 8 – Keep Them Short!

With proper subnetting and wildcard knowledge, multiple ACL lines can be shortened into a few lines. This is generally a good practice that conserves CPU cycles. If you

plan out your ACLs in advance, you should be able to see where excess lines can be reduced, for example:

```
access-list 2 deny 192.168.10.4  
access-list 2 deny 192.168.10.5  
access-list 2 deny 192.168.10.6  
access-list 2 deny 192.168.10.7
```

Four networks are being denied so you should be able to reduce this with a wildcard mask, which will match all four networks:

```
access-list 2 deny 192.168.10.4 0.0.0.3
```

ACL Rule 9 – Put Your ACL as Close to the Source as Possible.

To conserve resources, ACLs should be applied as close to the source as possible. There is no point in spending bandwidth and router resources in routing a packet across the network only to have the packet dropped by an ACL before it reaches its destination. This might not be possible all the time but you should always do so whenever possible.

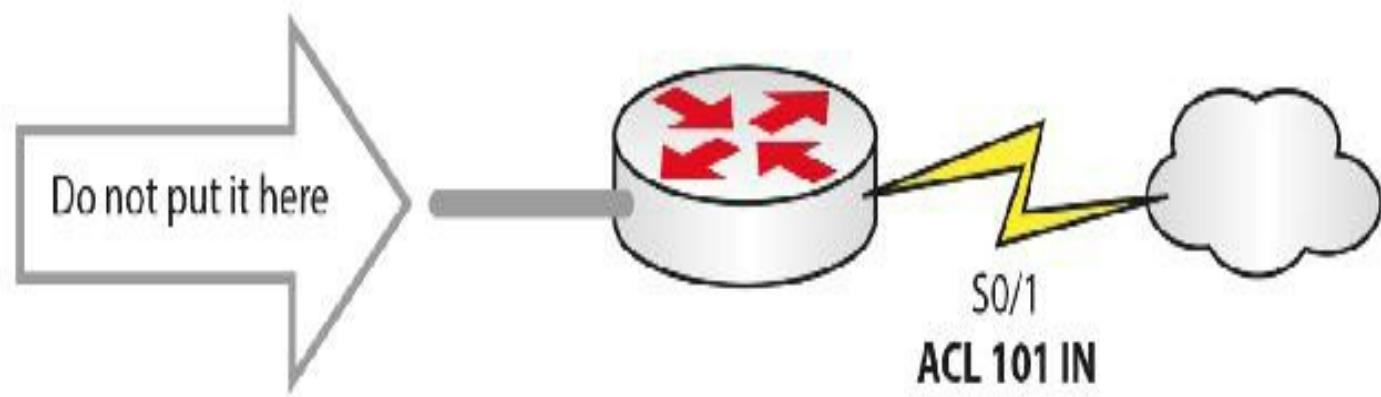


FIG 7.12 – Put your ACL close to the source

Cisco has a general rule to place standard access lists toward the destination and extended access lists closer to the source. In my experience, this isn't always the best thing to do. Just bear the general rule in mind in case you get an exam question on it.

Configuring Access Lists

You can configure three types of ACLs on a router: standard ACLs, extended ACLs, and named ACLs, although named ACLs will still be either standard or extended. You need to be familiar with all three methods for the CCNA exam. You should spend some time practicing the examples here in the labs and then make up your own examples. Before long, you will become an ACL master.

Standard ACLs

Standard ACLs are easy to configure since they only filter based on the packet's source IP address or network.

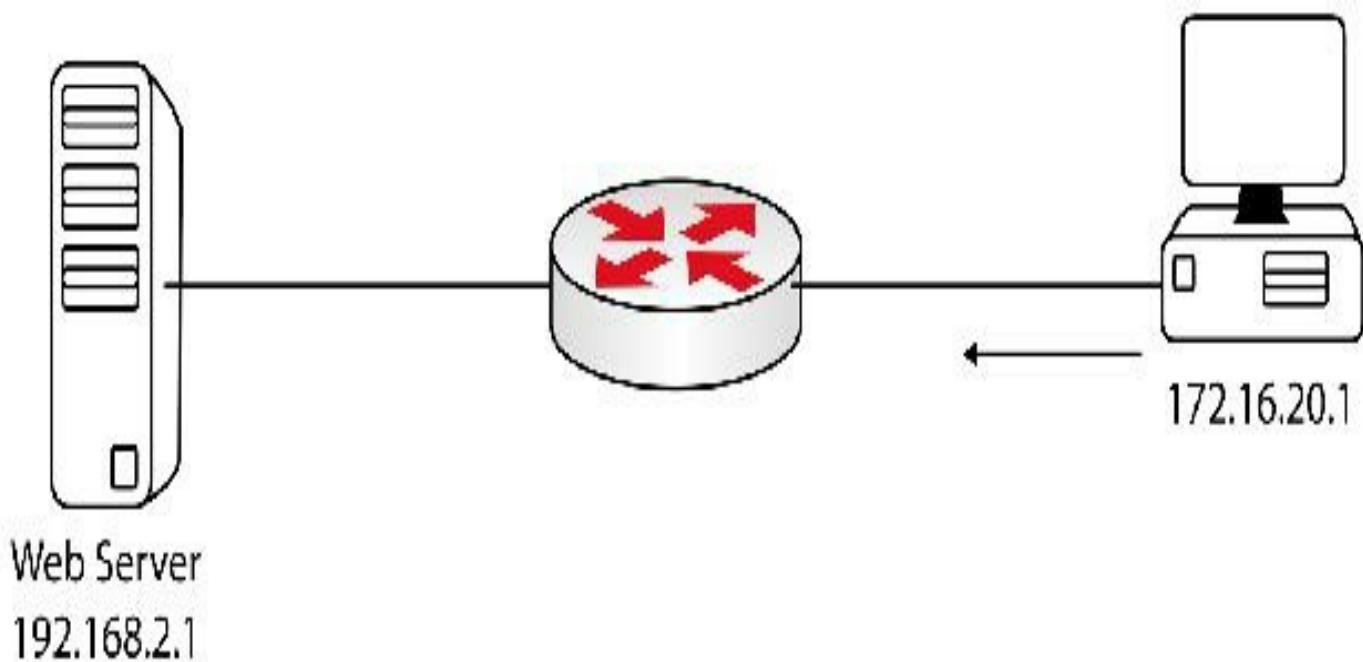


FIG 7.13 – Incoming packet with source and destination addresses

The incoming packet in Figure 7.13 above has a source address, a destination address, and a destination port number. In a standard ACL, the only parameter that matters is the source address (172.16.20.1). The ACL to permit this packet would be (as you saw earlier, you can miss off the host keyword):

```
Router(config)#access-list 1 permit host 172.16.20.1
```

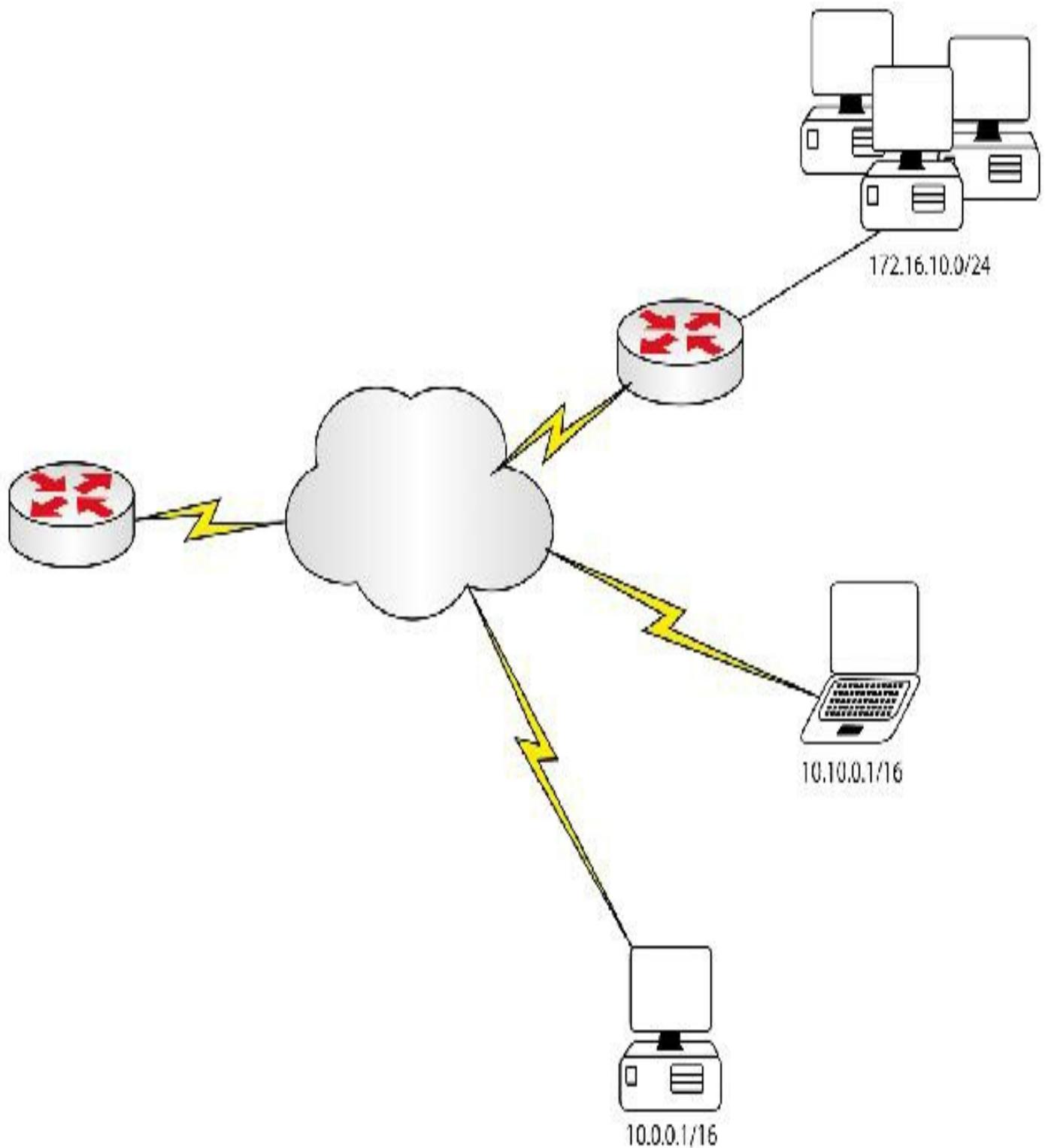


FIG 7.14 – Network with multiple hosts/networks

```
Router(config)#access-list 1 permit host 10.0.0.1
Router(config)#access-list 1 permit host 10.10.0.1
Router(config)#access-list 1 permit 172.16.10.0 0.0.0.255
```

This would be applied to the Internet-facing router interface. Remember that there will be an implicit deny all at the end of this list, so all other traffic will be blocked.

Extended ACLs

Extended ACLs have more flexibility since you can filter them based on source and destination networks, protocols, and application port numbers. The general syntax for extended access lists is as follows (simplified version):

```
access list# permit/deny [service/protocol] [source network/IP] [destination network/IP] [port#]
```

For example:

```
access-list 100 deny tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq 23  
access-list 100 permit tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq 21  
access-list 100 permit icmp any any
```

Figure 7.15 below shows a sample network that needs to be configured with an ACL on the server-side router:

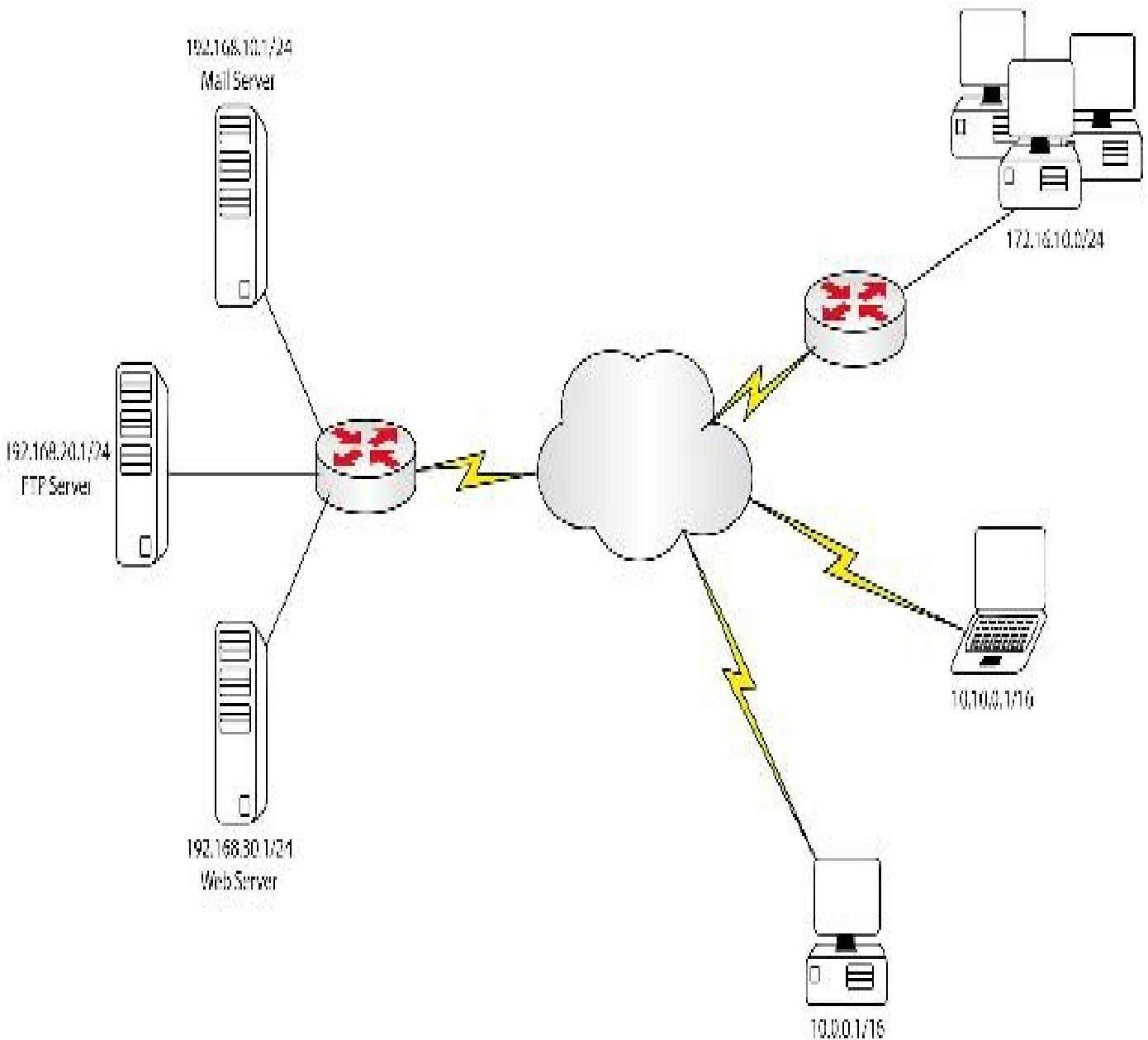


FIG 7.15 – Example of blocking server access

An example of an access list you could configure for the network above, featuring e-mail, web, and file servers, would be as follows:

```
access-list 101 permit tcp host 10.0.0.1 host 192.168.10.1 eq smtp
access-list 101 permit tcp 172.16.10.0 0.0.0.255 host 192.168.20.1 eq ftp
access-list 101 permit tcp host 10.10.0.1 host 192.168.30.1 eq www
```

If you break down each line you can see that the services are all being permitted and that they are all using TCP, but the source and destination host or network differs. Also, each port differs because each server is using a different service. You can also write out the port number or the service (i.e., www or 80).

Named ACLs

As well as using numbers for access lists, you can also use names. Named access lists were available beginning with IOS release 11.2. Using a name instead of a number makes the access list easier to identify when looking at the configuration. Because you are using names, you have to specify whether the access list is standard or extended:

```
ip access-list [standard | extended] name
```

The following syntax will depend on whether you are configuring a standard or extended access list. The access list below has been named inbound_access:

```
Router(config)#ip access-list extended inbound_access
Router(config-ext-nacl)#permit tcp any 172.16.0.0 0.0.255.255 eq 80
Router(config-ext-nacl)#exit
Router(config)#int s0
Router(config-if)#ip access-group inbound_access in
Router(config-if)#exit
Router(config)#+
```

Editing named access lists works in the same way as for standard and extended access lists after they were modified (for standard and extended access lists, you used to have to delete the old access list and create a new one). For named, standard, and extended access lists, you are able to delete a specific entry.

```
Router#conf t
Router(config)#ip access-list standard lan_traffic
Router(config-std-nacl)#permit 172.16.0.0
Router(config-std-nacl)#permit 172.30.0.0
Router(config-std-nacl)#permit 192.168.2.0
Router(config-std-nacl)#permit 10.0.0.0
Router(config-std-nacl)#exit
Router(config)#int fast0
Router(config-if)#ip access-group lan_traffic in
Router(config-if)#+Z
```

```
Router#show ip access-lists
Standard IP access list lan_traffic
 40 permit 10.0.0.0
 30 permit 192.168.2.0
 20 permit 172.30.0.0
```

```
10 permit 172.16.0.0
```

```
Router#conf t
```

```
Router(config)#ip access-list standard lan_traffic
```

```
Router(config-std-nacl)#no permit 10.0.0.0
```

```
Router(config-std-nacl)#exit
```

```
Router(config)#exit
```

```
Router#
```

```
Router#show ip access-lists
```

```
Standard IP access list lan_traffic
```

```
30 permit 192.168.2.0
```

```
20 permit 172.30.0.0
```

```
10 permit 172.16.0.0 i The 10.0.0.0 entry has been deleted
```

Or for numbered ACLs:

```
Router(config)#access-list 10 permit 172.16.0.0
```

```
Router(config)#access-list 10 permit 172.30.0.0
```

```
Router(config)#access-list 10 permit 192.168.2.0
```

```
Router(config)#access-list 10 permit 10.0.0.0
```

```
Router(config)#exit
```

```
Router#sh ip access-list 10
```

```
Standard IP access list 10
```

```
40 permit 10.0.0.0
```

```
30 permit 192.168.2.0
```

```
20 permit 172.30.0.0
```

```
10 permit 172.16.0.0
```

```
Router#conf t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)#ip access-list standard 10 i Edit the numbered ACL just like the named ACL
```

```
Router(config-std-nacl)#no permit 10.0.0.0
```

```
Router(config-std-nacl)#end
```

```
Router#sh ip access-list 10
```

```
Standard IP access list 10
```

```
30 permit 192.168.2.0
```

```
20 permit 172.30.0.0
```

```
10 permit 172.16.0.0
```

Applying ACLs

ACLs need to be applied on an interface or a terminal line for the ACLs to come into effect. It is a common mistake for junior Cisco engineers to create an ACL and then wonder why it isn't working! Configuring an ACL without applying it to an interface has NO effect! Remember also that you can use the same ACL multiple times.

ACLs can be applied to Telnet or console lines using the access-class command; similarly, it can be applied to an interface using the ip access-group command. It's very important to note where you would apply an access class or access group for the exam and the real world.

Some examples of access lists being applied to a terminal line or an interface are shown below. Please do try these commands and note whether you are at the configuration line or the configuration interface prompt.

Interface:

```
Router(config)#int Gig0/0  
Router(config-if)#ip access-group 102 in
```

Terminal line for VTY (Telnet/SSH) access:

```
Router(config)#line vty 0 15  
Router(config-line)#access-class 10 in
```

Interface:

```
Router(config)#int fast0/0  
Router(config-if)#ip access-group BlockWEB in
```

ACL Sequence Numbers

In IOS versions 12.2(14)S and later, Cisco introduced a very useful ACL manipulation feature—the capability to assign sequence numbers to ACL entries. This offers a series of advantages in regard to ACL editing, including:

- The possibility to add an ACL entry anywhere in the list
- The possibility to remove any ACL entry
- The possibility to reorder ACL entries

Let's start by creating a standard access list with one entry, permitting traffic from network 10.0.1.0:

```
R1(config)#ip access-list standard CCNA  
R1(config-std-nacl)#permit 10.0.1.0
```

You can see that the ACL entries are numbered even if you do not specifically configure a sequence number. The ACL entry created here was assigned a sequence number of 10:

```
R1#show access-list CCNA
```

```
Standard IP access list CCNA
```

```
    10 permit 10.0.1.0
```

Next, insert two more entries for networks 10.0.2.0 and 10.0.3.0, one after the initial sequence number and one before, and assign them numbers 5 and 15:

```
R1(config)#ip access-list standard CCNA
```

```
R1(config-std-nacl)#5 permit 10.0.2.0
```

```
R1(config-std-nacl)#15 permit 10.0.3.0
```

```
R1#show access-list CCNA
```

```
Standard IP access list CCNA
```

```
    5 permit 10.0.2.0
```

```
    15 permit 10.0.3.0
```

```
    10 permit 10.0.1.0
```

As you can see from the show command output above, the two entries were correctly assigned sequence numbers 5 and 15. Now, let's assume you want to delete the first entry created for network 10.0.1.0:

```
R1(config)#ip access-list standard CCNA
```

```
R1(config-std-nacl)#no 10 permit 10.0.1.0
```

```
R1#sho access-list CCNA
```

```
Standard IP access list CCNA
```

```
    5 permit 10.0.2.0
```

```
    15 permit 10.0.3.0
```

You can see that ACL entry number 10 has been deleted. You can also resequence ACL entries by shifting their sequence numbers based on a new sequence start number and a predefined step. Let's assume that you want to resequence the CCNA access list so the sequence numbers start at 200 and are incremented at 15 for each line. The command used to accomplish this is:

```
R1(config)#ip access-list resequence CCNA 200 15
```

```
R1#sho access-list CCNA
```

Standard IP access list CCNA

```
200 permit 10.0.2.0
```

```
215 permit 10.0.3.0
```

There are many other types of access lists, such as turbo, time-based, lock and key, reflexive, and dynamic. You can use a dynamic access list to authenticate remote users with a unique username and password. The authentication process is done by the router or a central access server, for example, a TACACS+ or RADIUS server, which then grants access to the network for a period determined by the person configuring the ACL. Cisco has documentation on this; however, configuration is beyond the scope of the CCNA RS exam.

It is well worth having a very good working knowledge of access lists, both for the CCNA exam and as a Cisco engineer.

An Alternative to Access Lists

An alternative to using access lists, which can become complicated and take up valuable CPU cycles, is to install a route to a Null interface. A Null interface, much like a Loopback interface, exists in software only. Any traffic routed to the Null interface is automatically dropped by the router. One key difference to note here is that while ACLs match based on different parameters, depending on whether it is a standard or an extended ACL, Null routing only matches based on the destination address (like every other type of routing).

Say, for example, that you want to prevent any traffic leaving the router destined for network 10.2.4.x 255.255.255.0. You would install a static route and send traffic destined for that network to the Null 0 interface:

```
Router(config)#ip route 10.2.4.0 255.255.255.0 Null0
```

Network Address Translation

No discussion about IP addressing would be complete without including Network Address Translation (NAT), which was implemented under RFC 1631. RFC 1918 addressed the shortage of IP addresses and allocated the ranges of private addresses 10.x.x.x, 172.16.x.x to 172.31.x.x, and 192.168.x.x. As you already know, private IP addresses allocated under RFC 1918 are not routable on the Internet. NAT allows these private internal IP addresses to be translated into addresses that are routable on the Internet.

One of the benefits of using NAT is that it helps prevent the depletion of public IP addresses. You can use private IP addressing on your LAN safe in the knowledge that

you will still be able to connect out to the Internet using NAT. It also prevents you from having to manually readdress internal hosts using private addressing that requires Internet access.

Another benefit is that hosts inside your LAN are protected from advertising their addresses out to the Internet. Your Internet-facing router will translate the private address to a public address and back again, so the NATing process is invisible to the hosts even if they are on the other side of a WAN link.

A network using NAT is split into two logical halves—an inside half, which (usually) uses private addresses, and an outside half, which uses one or more public addresses. When the inside hosts attempt to contact another device on the Internet, the router swaps the private address for a public address and maintains a record of which private address was swapped for which public address.

The inside and outside parts of the network are defined with the following commands in interface configuration mode:

```
RouterA(config-if)#ip nat inside
```

```
RouterA(config-if)#ip nat outside
```

A router only needs one valid public IP address to perform NAT. This one IP address can be used many times over by assigning a port number to the inside hosts for the connection to the outside (using Port Address Translation).

There are three ways to configure NAT and your choice will depend on different factors, including how many public addresses you have and what you want to achieve with the NAT configuration. Options include:

- **Static NAT** – maps a private to a public address on a one-to-one basis
- **Dynamic NAT** – maps the private (inside) address to a group or pool of public addresses; after a predetermined period of time the translation times out
- **NAT Overload** – maps private addresses to one public address; also known as Port Address Translation (PAT) or One-way NAT

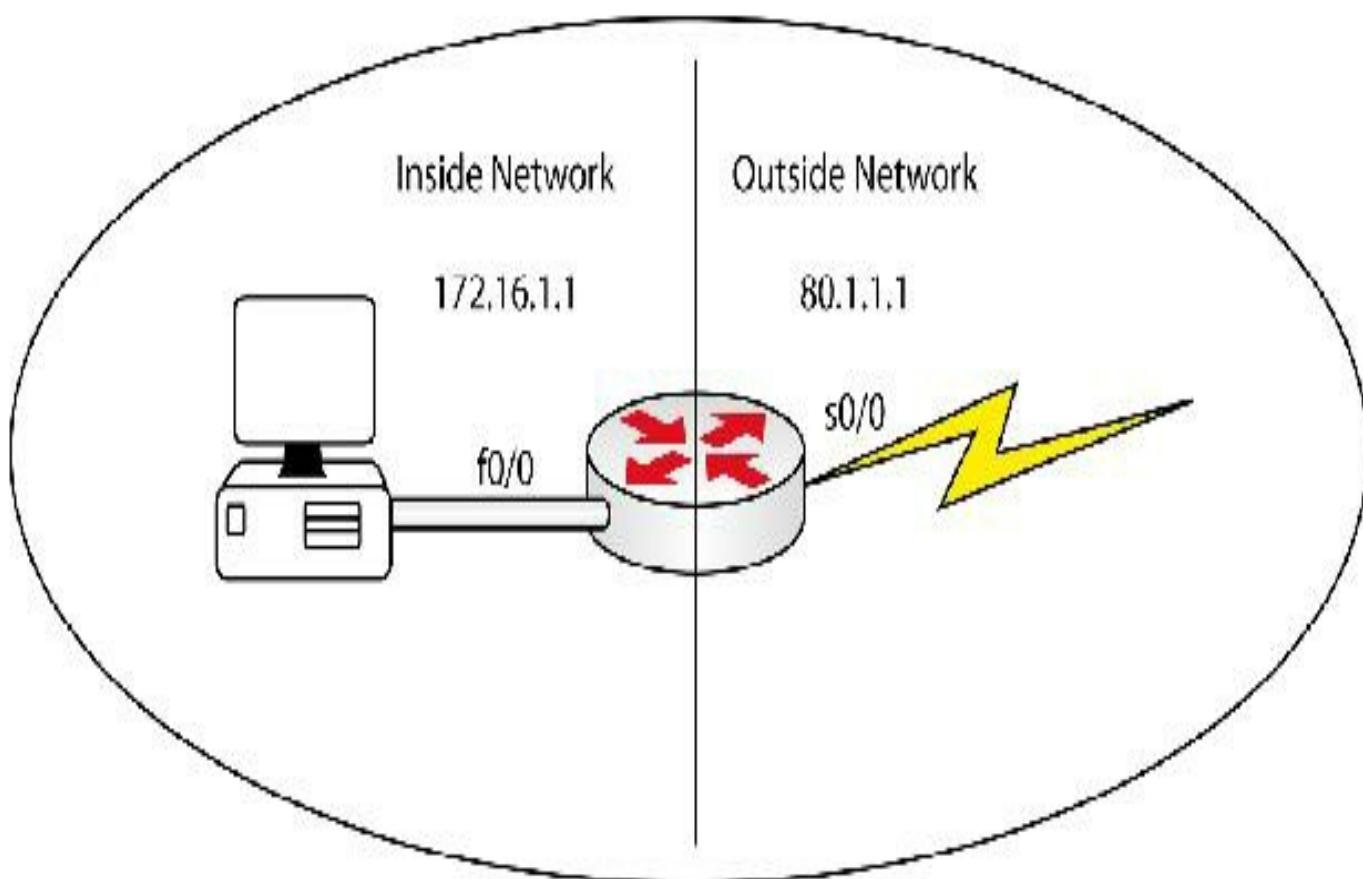


FIG 7.16 – NAT connection to the Internet

Local and Global Addresses

The terms used to describe the types of NATed addresses can cause some confusion but it is worth knowing which is which:

- **Inside local** – the address assigned to a host inside the network; likely to be a private address not routable (RFC 1918) on the Internet
- **Inside global** – a routable IP address usually assigned by an ISP; represents one or more hosts on the LAN to the outside world
- **Outside local** – the address of an outside host as it appears to the internal network (LAN)
- **Outside global** – an outside address assigned to a host by the administrator; a public (routable) address

A local address is an address on the inside part of the network and a global address is an address on the outside part of the network (usually the Internet).

Referring to Figure 7.17 below, the output below shows the static NAT configuration for the host translated to 80.1.1.1 (some of the configuration has been omitted for clarity):

```
RouterA(config)#ip nat inside source static 172.16.1.1 80.1.1.1
```

```

RouterA(config)#interface Serial0/0
RouterA(config-if)#ip nat outside
RouterA(config-if)#interface FastEthernet0/0
RouterA(config-if)#ip nat inside

```

Notice in the configuration example above that the public address 80.1.1.1 is not applied to an interface. The public address is kept inside the router configuration and is used only when the inside host wants to go out to the Internet. If the inside host wanted to contact another host inside the same LAN, then a NAT translation would not take place.

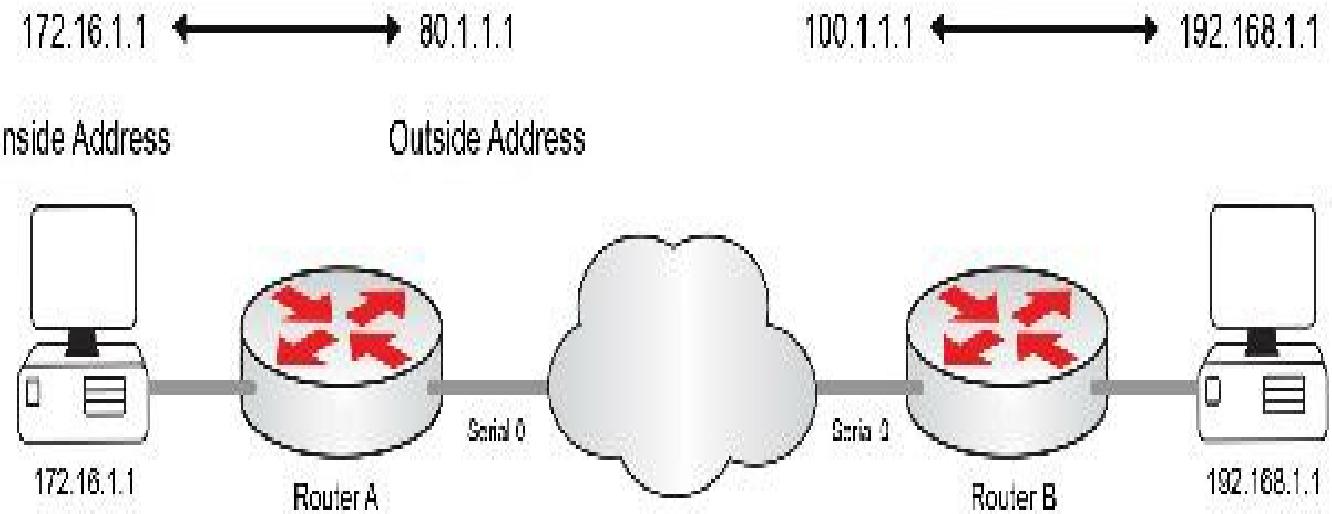


FIG 7.17 – NAT example

When the host behind Router A goes out to the Internet to connect to the host behind Router B, you will see the following addresses in Router A's NAT table (they may differ depending on how you configured NAT on your router):

Inside Global	Inside Local	Outside Global	Outside Local
80.1.1.1	172.16.1.1	100.1.1.1	100.1.1.1

NOTE: The public addresses above belong to a company somewhere on the Internet and should not be used as part of your NAT configurations or for testing in your labs if your network has a connection out to the Internet.

Running a debug ip nat command as the host on the LAN goes through the router shows the translation taking place for host 172.16.1.1, pinging address 192.168.1.1, and being NATed to 80.1.1.1. The output below may not work or look the same in Packet Tracer:

```
RouterA#debug ip nat
03:38:28: NAT: s=172.16.1.1-[80.1.1.1, d=192.168.1.1 [30]
03:38:29: NAT: s=192.168.1.1, d=80.1.1.1-[172.16.1.1 [30]
03:38:29: NAT: s=172.16.1.1-[80.1.1.1, d=192.168.1.1 [31]
03:38:29: NAT: s=192.168.1.1, d=80.1.1.1-[172.16.1.1 [31]
03:38:29: NAT: s=172.16.1.1-[80.1.1.1, d=192.168.1.1 [32]
03:38:29: NAT: s=192.168.1.1, d=80.1.1.1-[172.16.1.1 [32]
03:38:29: NAT: s=172.16.1.1-[80.1.1.1, d=192.168.1.1 [33]
03:38:29: NAT: s=192.168.1.1, d=80.1.1.1-[172.16.1.1 [33]
03:38:29: NAT: s=172.16.1.1-[80.1.1.1, d=192.168.1.1 [34]
03:38:29: NAT: s=192.168.1.1, d=80.1.1.1-[172.16.1.1 [34]
```

The source (s=) address is the host 172.16.1.1 on the LAN, which is translated to 80.1.1.1. The destination (d=) address 192.168.1.1 is another device on an outside network. The numbers in brackets [] indicate the IP identification number of the packet, which is useful for debugging.

```
RouterA#show ip nat translation
```

Pro	Inside global	Inside local	Outside local	Outside global
---	80.1.1.1	172.16.1.1	---	---

There are several ways to configure NAT depending on the network requirements. It is advisable to have a good working knowledge of NAT and how to configure static, dynamic, and PAT. The NAT labs will help you gain a good working knowledge of NAT.

Static NAT

Static NATs (or 1-to-1 NAT) are manually defined mappings and are always in the NAT translation table (i.e., they are not dynamically created and torn down). This is useful in allowing an internal device to be reachable from the Internet. When configuring static NAT, you do not need an access list to specify inside addresses or a NAT pool to specify global addresses. Let's take a look at Figure 7.18 below:

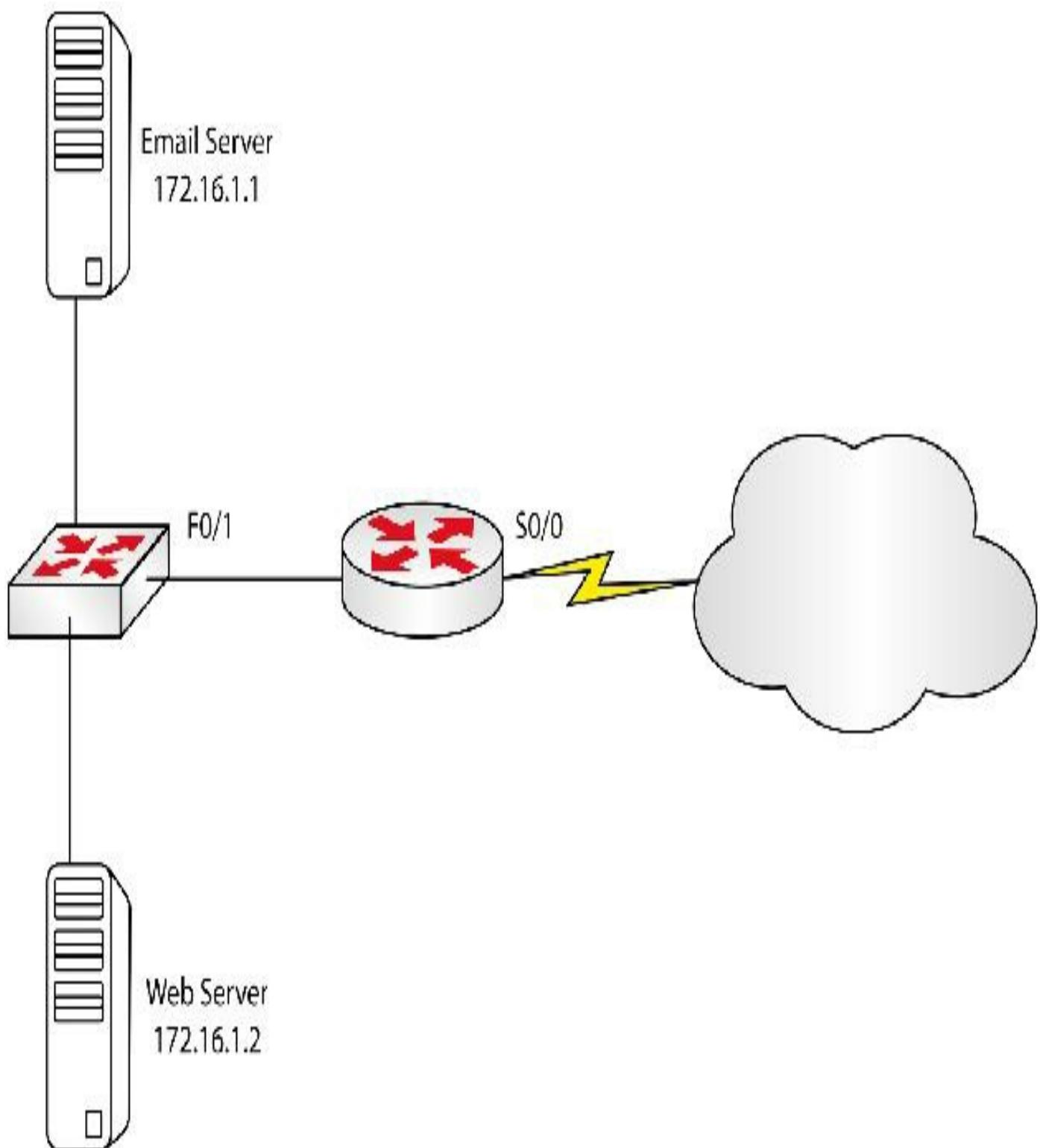


FIG 7.18 – Static NAT in use

Inside Addresses	Outside NAT Addresses
172.16.1.1	80.1.1.1
172.16.1.2	80.1.1.2

For the network above, your configuration would be as follows:

```
Router(config)#interface f0/1
Router(config-if)#ip address 172.16.1.3 255.255.0.0
Router(config-if)#ip nat inside
Router(config)#interface s0/0
Router(config-if)#ip add 100.1.1.1 255.255.0.0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip nat inside source static 172.16.1.1 80.1.1.1
Router(config)#ip nat inside source static 172.16.1.2 80.1.1.2
```

The ip nat inside source static command defines the NAT as a static NAT that translates the source address to a global public address when traffic hits the inside interface.

Many network engineers run into configuration mistakes when they apply the wrong ip nat inside and ip nat outside statements! You should be very familiar with these kinds of errors and be able to spot them in the CCNA exam.

The IP addresses 80.1.1.1, 80.1.1.2, and 100.1.1.1 are allocated to hosts on the Internet, so don't use them on any equipment you have connected to the Internet.

Dynamic NAT and Port Address Translation

We will cover dynamic NAT (NAT pool) and Port Address Translation (PAT/NAT overload) at the same time. The reason is that to turn dynamic NAT into PAT you simply add one tag command to the end of your configuration line. PAT is also known as One-way NAT because it can only be initiated from the inside of your network to the outside; it is also referred to as NAT Overload.

Static NAT and dynamic NAT help us translate addresses, but they do not help us conserve IP address space, since the translations are 1:1. This is where NAT Overload (PAT) comes in. With port overloading, you not only translate the IP source address but also the port numbers. This way, each port number of the global IP address can be used for a different connection. This allows you to have up to 65,000 connections using the same global IP address!

NAT Overload is configured the same way as dynamic NAT. The only difference is that the overload keyword is added to the configuration. First, we will configure dynamic NAT for the network below using a pool of addresses from 200.1.1.1 to 100.

In the real world, NAT Overload uses the interface IP address more often than an address pool.

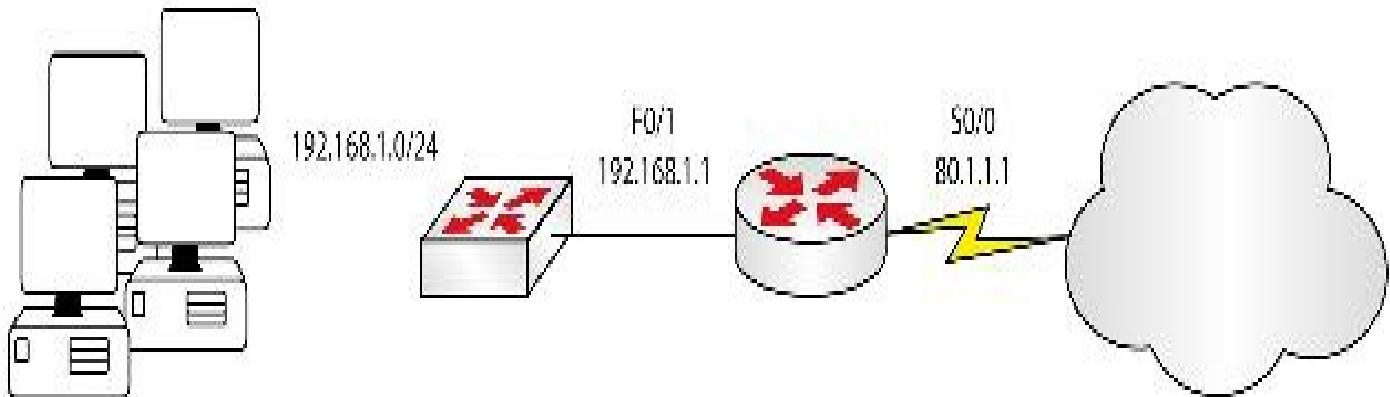


FIG 7.19 – Dynamic NAT example

```
Router(config)#interface f0/0
Router(config-if)#ip nat inside
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#interface s0/0
Router(config-if)#ip address 80.1.1.1 255.0.0.0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip nat pool NAT_Pool 200.1.1.1 200.1.1.100 netmask 255.255.255.0
Router(config)#ip nat inside source list 1 pool NAT_Pool
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

I didn't include the routing commands but we will cover this in the NAT labs later on. The output below shows the NAT translation table on the router after a host on the inside is NATed as it goes out of the router to host 80.1.1.2 on the Internet. Because of the configuration, dynamic NAT is usually used for internal hosts to reach hosts on the Internet, not the other way around.

```
Router#show ip nat translations
```

Protocol	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.1:0	192.168.1.2:0	80.1.1.2:0	80.1.1.2:0

--- 200.1.1.1 192.168.1.2 --- ---

Address 192.168.1.2 was allocated the routable address 200.1.1.1 from the NAT pool.

To configure PAT, you would carry out the exact same configuration as for dynamic NAT, but you would add the keyword **overload** to the end of the pool. With PAT you only need one routable IP address for your pool. It's still called a NAT pool even though you are using only one IP address. I didn't include the interface-level commands because they are the same.

```
Router(config)#ip nat pool PAT_Pool 200.1.1.1 200.1.1.1 netmask 255.255.255.0  
Router(config)#ip nat inside source list 1 pool PAT_Pool overload  
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

This time you can see that PAT has utilized a port number, in this case port number 1, which is appended to the end of the translation addresses.

```
Router#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.1:1	192.168.1.2:1	80.1.1.2:1	80.1.1.2:1

As Dario mentioned above, you are more likely to overload your outside interface IP address, thus avoiding having to purchase a pool of routable IP addresses.

```
Router(config)#interface f0/0  
Router(config-if)#ip nat inside  
Router(config-if)#ip address 192.168.1.1 255.255.255.0  
Router(config-if)#interface s0/0  
Router(config-if)#ip address 80.1.1.1 255.0.0.0  
Router(config-if)#ip nat outside  
Router(config-if)#exit  
Router(config)#ip nat inside source list 1 interface s0/0 overload  
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

Configuring and Verifying NAT

NAT configuration using Cisco IOS involves a few configuration steps. For dynamic NAT configuration, here are the steps required:

1. Select the inside NAT interface using the **ip nat inside** interface configuration command.
2. Select the outside NAT interface using the **ip nat outside** interface configuration command.

3. Identify traffic that would be translated using an access list. This can be a standard/extended and named or numbered access list. I recommend named access lists since they are more intuitive.
4. Configure a pool of global addresses to which the local addresses would be translated. This step is optional because you can translate directly to an interface. The syntax for that is ip nat pool [name [start-ip]] end-ip [netmask [mask] | prefix-length [length]] global configuration command.
5. Tie all the commands together: Configure dynamic nat using the syntax ip nat inside source list [ACL] [interface|pool] [name] [overload] global configuration command.

The output below shows how to configure dynamic NAT on the Cisco IOS. The description and remark features available in the Cisco IOS have been used to explain the configuration. This is quite useful on live networks to provide some context on why the network was configured in a particular manner.

```
R1(config)#interface GigabitEthernet0/0
R1(config-if)#description "Connected To The Inside Network"
R1(config-if)#ip address 10.1.1.1 255.255.255.248
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface Serial0/1
R1(config-if)#description "Connected To The Internet"
R1(config-if)#ip address 161.1.1.1 255.255.255.248
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#access-list 101 remark "Translate Inside Addresses Only"
R1(config)#access-list 101 permit ip 10.1.1.0 0.0.0.7 any
R1(config)#ip nat pool INSIDE-POOL 161.1.1.3 161.1.1.6 prefix-length 24
R1(config)#ip nat inside source list 101 pool INSIDE-POOL
R1(config)#exit
```

The show ip nat translations command can be used to verify that the inside networks are actually getting translated to the NAT pool, as shown below:

```
R1#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
icmp	161.1.1.4:4	10.1.1.1:4	200.1.1.1:4	200.1.1.1:4
	icmp	161.1.1.3:1	10.1.1.2:1	200.1.1.1:1

```
tcp 161.1.1.5:159 10.1.1.3:159 200.1.1.1:23 200.1.1.1:23
```

Troubleshooting NAT

There is no mention of troubleshooting NAT in the CCNA syllabus but as I've learned over years of taking Cisco exams, the fact that it isn't mentioned doesn't mean that you won't be asked about it!

You have already learned all you need to know to troubleshoot NAT because (apart from IOS bugs), NAT issues are almost always a configuration error, in particular:

- NAT inside and outside statements missing from the interface
- Name calling the NAT pool doesn't match the NAT pool name (case sensitive)
- ACL incorrectly configured to match wrong addresses or subnets
- ACL blocking the traffic before it is NATed
- Pool contains too few addresses and fills quickly
- No routing configured to reach the NAT pool addresses

You should be familiar with the `show ip nat translations` and `debug ip nat` commands to see real-time translations taking place. Don't rush into troubleshooting NAT before checking the obvious, such as interface status, IP addressing, and IP connectivity. The network must have a route to reach all addresses, including those you have added to the NAT pool.

If you need to clear your NAT translations table, then use the `clear ip nat translation *` command in privileged mode. Bear in mind that all dynamic translations will expire when the timeout value is reached, so if you wait too long to check the translations table, you will find it isn't there. This doesn't mean that NAT isn't working. You can change NAT timeout values but its best to do this after consulting with a Cisco TAC.

Network Time Protocol

Having an automated way to ensure that consistent and accurate time is held by your network devices is vital to the efficient operation of your network infrastructure. Network Time Protocol (NTP) is a protocol that allows automatic time configuration on distributed network devices based on a client/server model. The device peers with an NTP server, which provides the correct clock when clients request this. NTP functions on UDP port 123. A Cisco device can be configured to pull the exact time information from an NTP server (192.168.1.2 in the output below) using the following command:

```
R1(config)#ntp server 192.168.1.2
```

```
R1#show ntp status
```

```
Clock is unsynchronized, stratum 16, no reference clock
```

nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)

clock offset is 0.0000 msec, root delay is 0.00 msec

root dispersion is 0.00 msec, peer dispersion is 0.00 msec

R1#show ntp associations

address	ref clock	st	when	poll	reach	delay	offset	disp
---------	-----------	----	------	------	-------	-------	--------	------

~192.168.1.2	0.0.0.0	16	-	64	0	0.0	0.00	6000
--------------	---------	----	---	----	---	-----	------	------

* master (synced), # master (unsynced), + selected, - candidate, ~ configured

You can see that the output above includes the term stratum. The stratum grades the reliability of time sources. A stratum 1 device is directly connected to a reliable source such as an atomic clock. A stratum 2 device obtains its time from a stratum 1 device. A stratum 3 device obtains its time from a stratum 2 device, and so on.

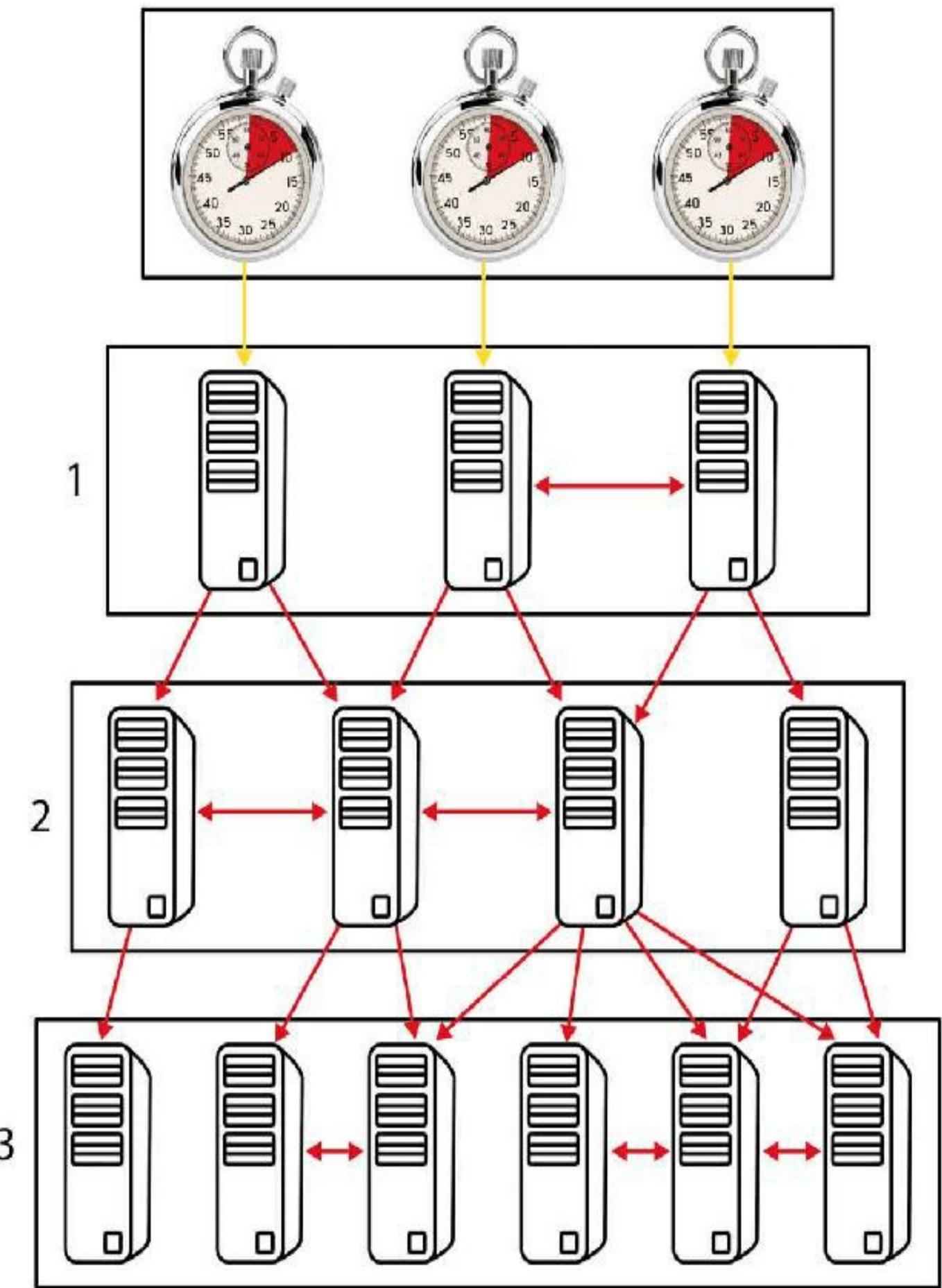


FIG 7.20 – NTP stratum sources

NTP offers a number of configuration options, including authentication and peering, but going into depth on these will take us off track for the purposes of the CCNA exam.

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 7 exam.

Chapter 7 Labs

Lab 1: Configuring a Router as a DHCP Server

The physical topology is shown in Figure 7.21 below:

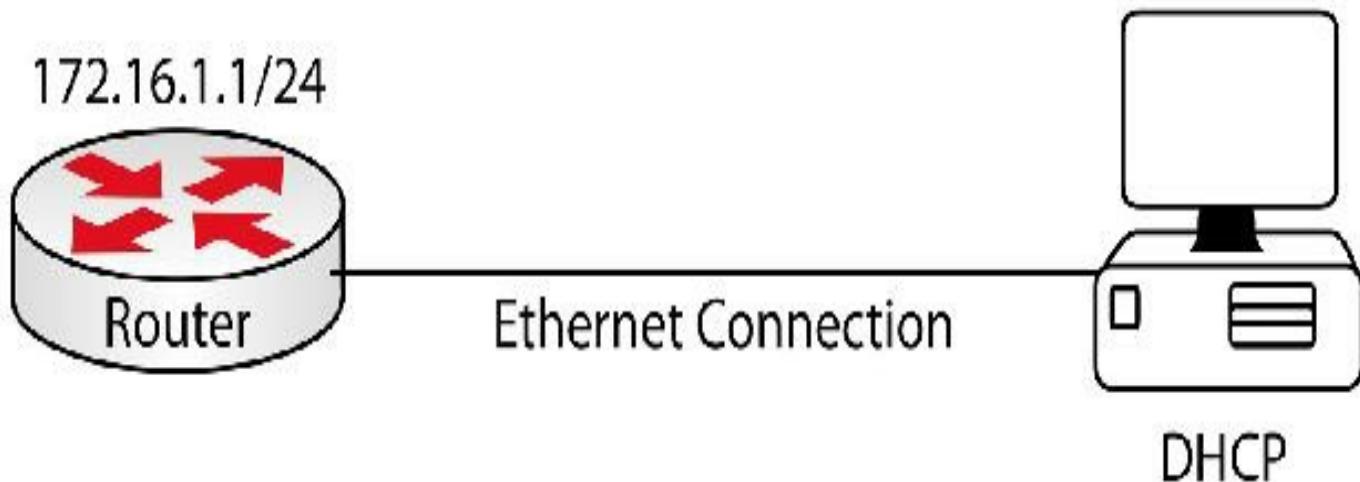


FIG 7.21 – DHCP server

Lab Exercise

Your task is to configure the router to issue an IP address to the host via DHCP. Please feel free to try the lab without following the Lab Walk-through section.

Purpose

Configuring a router to act as a DHCP server is now included in the CCNA exam. You may also have to carry out this task as a network engineer while on site.

Lab Objectives

1. Use the IP addressing scheme depicted in the diagram above. You will need to connect the router to the PC via a crossover cable unless you have a switch you can use.
2. Set the IP address on the router's Fast Ethernet interface as 172.16.1.1/24.
3. Set the PC to search for an IP address via DHCP.
4. Configure a DHCP pool on the router for network 172.16.1.0/24.
5. Add an excluded address on the router and add TCP settings (optional).
6. Finally, to test that DHCP is working, renew the IP address on the PC.

Lab Walk-through

1. To set the IP addresses to an interface, you will need to do the following:
Router#conf t

```
Router(config)#interface fast0/0
```

```
Router(config)#ip address 172.16.1.1 255.255.255.0
```

2. Enable DHCP on your router and set the address pool (in configuration mode):

```
Router(config)#service dhcp i Turn DHCP on
```

```
Router(config)#ip dhcp pool pool1 i Name your pool pool1
```

```
Router(dhcp-config)#network 172.16.1.0 255.255.255.0 i This is your DHCP pool
```

```
Router(dhcp-config)#lease 3 i 3-day lease on the IP address
```

```
Router(dhcp-config)#ip dhcp excluded-address 172.16.1.1
```

```
Router(config)#exit i Drops back to config mode
```

3. If you are using your home PC or laptop, set the network adaptor to obtain the IP address automatically:

Internet Protocol (TCP/IP) Properties



General Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically

Use the following IP address:

IP address:

Subnet mask:

Default gateway:

4. Issue an ipconfig /all command to check whether an IP address has been assigned to your PC. You may need to issue an ipconfig /renew command if an old IP address is still in use:

PC]ipconfig /all

Physical Address.....: 0001.C7DD.CB19

IP Address.....: 172.16.1.4

Subnet Mask.....: 255.255.255.0

Default Gateway.....: 0.0.0.0

DNS Servers.....: 0.0.0.0

5. If you want, you can go back into the DHCP pool and add a default gateway and a DNS server address, which will also be set on the host PC:

Router(config)#ip dhcp pool pool1

```
Router(dhcp-config)#default-router 172.16.1.2  
Router(dhcp-config)#dns-server 172.16.1.3
```

PC]ipconfig /renew

```
IP Address.....: 172.16.1.4  
Subnet Mask.....: 255.255.255.0  
Default Gateway.....: 172.16.1.2  
DNS Server.....: 172.16.1.3
```

Show Run

```
Current configuration : 860 bytes  
!  
! Last configuration change at 22:19:11 UTC Wed Dec 6 2006  
!  
version 15.1  
!  
hostname Router  
!  
ip dhcp excluded-address 172.16.1.1  
!  
ip dhcp pool pool1  
network 172.16.1.0 255.255.255.0  
dns-server 172.16.1.3  
default-router 172.16.1.2  
lease 3  
!  
ip cef  
ip audit po max-events 100  
!  
interface FastEthernet0/0  
ip address 172.16.1.1 255.255.255.0  
!  
end
```

Lab 2: Access Lists (Standard)

The physical topology is shown in Figure 7.22 below:

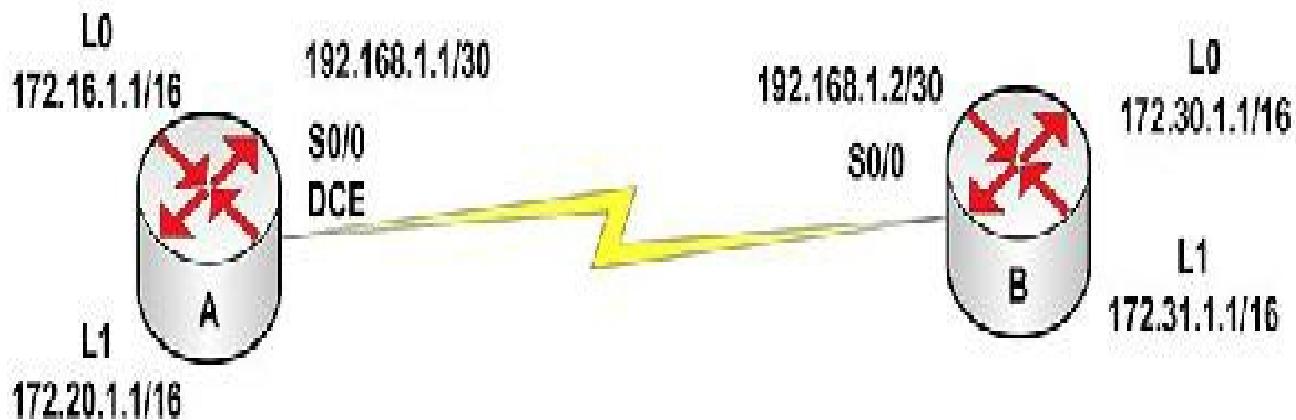


FIG 7.22 – Standard access lists

Lab Exercise

Your task is to configure the network in Figure 7.22 to allow full connectivity using a default route. Then you will need to configure an access list to deny connections from the neighbor network on their Loopback 0. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Access lists are a fundamental way of protecting the router and are also a very useful troubleshooting tool. Standard access lists allow you to filter traffic based on a source address or a network and are a great introduction before moving on to the more sophisticated extended access list.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.22. Router A (if it is the DCE) needs to have a clock rate on interface Serial 0/0: set this to be 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable secret password to be cisco.
4. Configure a default route to allow full connectivity.
5. Configure an access list to deny any connection from the neighbor router's Loopback 0 interface, while still allowing all other traffic through.
6. Finally, to test that the access list is working, you will need to use the extended ping command.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
```

Ping across the Serial interface now. If you wait until the access list is in place, you will struggle to troubleshoot the problem. It could be a Serial link or access list issue.

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration
RouterA(config-line)#login local i This will use local usernames and passwords for Telnet access
RouterA(config-line)#exit i Exits the VTY config mode
RouterA(config)#username banbury password ccna i Creates username and password for Telnet access (login local)
```

Router B:

```
RouterB(config)#line vty 0 4
RouterB(config-line)#login local
RouterB(config-line)#exit
```

```
RouterB(config)#username banbury password ccna
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco i Sets the enable password (encrypted)
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. To configure a default route, there is one simple step (in configuration mode):

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0 i For all unknown addresses, send packet out of Serial0/0
```

Router B:

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

5. To configure an access list, there are two steps: first, specify the networks to permit or deny; and second, apply the access list to an interface:

```
RouterA(config)#access-list 1 deny 172.30.0.0 0.0.255.255 i Denies the network specified; remember to use a wildcard mask.
```

```
RouterA(config)#access-list 1 permit any i Permit everything else
```

```
RouterA(config)#interface Serial0/0
```

```
RouterA(config-if)#ip access-group 1 in i Assigns the access list to the interface and the direction of traffic to be checked
```

Router B:

```
RouterB(config)#access-list 1 deny 172.16.0.0 0.0.255.255
```

```
RouterB(config)#access-list 1 permit any
```

```
RouterB(config)#interface Serial0/0
```

```
RouterB(config-if)#ip access-group 1 in
```

6. To test the access list, you need to use an extended ping. The extended ping command allows you to specify a different source address for the ping instead of using the IP address assigned to the exiting interface:

```
RouterA#ping i Press Enter here
```

```
Protocol [ip]: i Press Enter here
```

```
Target IP address: 192.168.1.2
```

```
Repeat count [5]
```

```
Datagram size [100]
```

```
Timeout in seconds [2]
```

Extended commands [n]: y
Source address or interface: 172.16.1.1
Type of service [0]
Set DF bit in IP header? [no]
Validate reply data? [no]
Data pattern [0xABCD]
Loose, Strict, Record, Timestamp, Verbose[none]
Sweep range of sizes [n]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
U.U.U i Traffic from 172.16.0.0 network blocked by acl on Router B
Success rate is 0 percent (0/5)

NOTE: Your response may be ... instead of U. U. U.

RouterA#ping
Protocol [ip]
Target IP address: 192.168.1.2
Repeat count [5]
Datagram size [100]
Timeout in seconds [2]
Extended commands [n]: y
Source address or interface: 172.20.1.1
Type of service [0]
Set DF bit in IP header? [no]
Validate reply data? [no]
Data pattern [0xABCD]
Loose, Strict, Record, Timestamp, Verbose[none]
Sweep range of sizes [n]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.20.1.1, timeout is 2 seconds:
!!!! i Traffic from 172.20.0.0 network permitted by acl
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

Router B:

RouterB#ping
Protocol [ip]:
Target IP address: 192.168.1.1
Repeat count [5]:

Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.30.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:
U.U.U i Traffic from 172.30.0.0 network denied by acl
Success rate is 0 percent (0/5)#[br/>RouterB#ping
Protocol [ip]:
Target IP address: 192.168.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.31.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.31.1.1, timeout is 2 seconds:
!!!! i Traffic from 172.31.0.0 network permitted by acl
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

Try the following commands:

RouterA#show ip access-lists
RouterA#show access-lists
RouterA#show run interface Serial0/0 **i This command may not work in the exam (so use show run instead)**

Show Runs

```
RouterA#show run
Building configuration...

Current configuration : 810 bytes
!
version 15.1
!
hostname RouterA
!
enable secret 5 $1$jjQo$YJXxLo.EZm9t6Sq4UYeCv0
!
username banbury password 0 ccna
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Loopback1
ip address 172.20.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
ip access-group 1 in
clockrate 64000
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
no ip http server
!
access-list 1 deny 172.30.0.0 0.0.255.255
access-list 1 permit any
!
line con 0
line aux 0
line vty 0 4
login local
!
end
```

RouterA#

RouterB#show run

Building configuration...

Current configuration : 791 bytes

!

version 15.1

!

hostname RouterB

!

enable secret 5 \$1\$HrXN\$ThplDHEZdnCbbeA/Ie67E1

!

username banbury password 0 ccna

!

interface Loopback0

ip address 172.30.1.1 255.255.0.0

!

interface Loopback1

ip address 172.31.1.1 255.255.0.0

!

interface Serial0/0

ip address 192.168.1.2 255.255.255.252

ip access-group 1 in

!

ip classless

ip route 0.0.0.0 0.0.0.0 Serial0/0

no ip http server

!

access-list 1 deny 172.16.0.0 0.0.255.255

access-list 1 permit any

!

end

RouterB#

Lab 3: Access Lists (Extended)

The physical topology is shown in Figure 7.23 below:

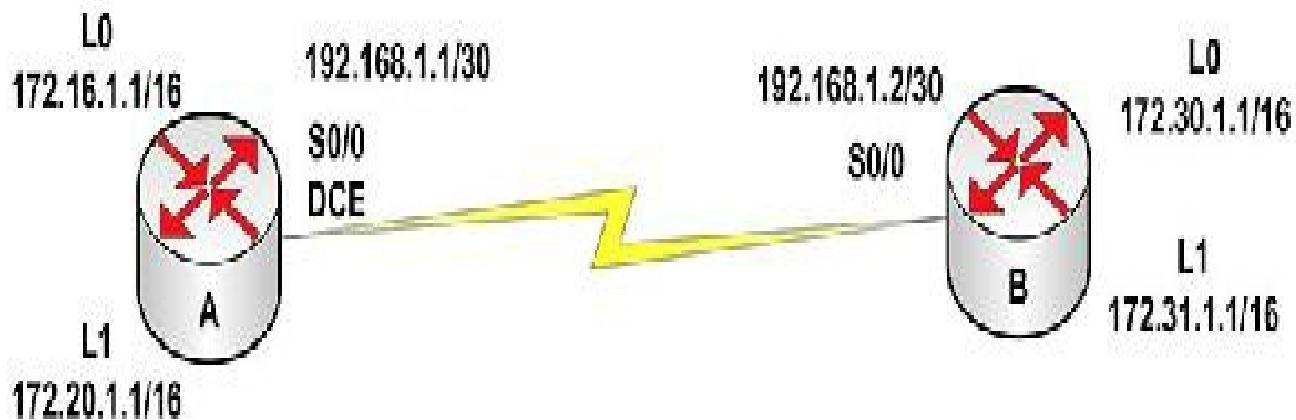


FIG 7.23 – Extended access lists

Lab Exercise

Your task is to configure the network in Figure 7.23 to allow full connectivity using a default route. Then you will need to configure an access list to deny Telnet connections into Router A and Web (HTTP) traffic into Router B. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Extended access lists are one of the foundation skills of any competent CCNA. You will be expected to be able to configure one to protect a client's network from certain types of traffic. The number one tip for access lists is to practice over and over again; the number two tip is to write them out on paper before you configure them

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.23. Router A needs a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Configure a default route to allow full connectivity.
5. Configure an access list to deny any Telnet connections from the neighbor router, while still allowing all other traffic through.
6. Finally, to test that the access list is working, you will need to Telnet to the neighbor router.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#
```

2. To set the clock rate on a Serial interface (DCE connection only), you need to use the `clock rate #` command on the Serial interface, where # indicates the speed:

```
RouterA(config-if)#clock rate 64000
```

Ping across the Serial link now.

3. To set Telnet access you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration
RouterA(config-line)#login local i This will use local usernames and passwords for Telnet access
```

RouterA(config-line)#exit **i Exits the VTY config mode**
RouterA(config)#username banbury password ccna **i Creates username and password for Telnet access (login local)**

Router B:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit  
RouterB(config)#username banbury password ccna
```

4. To set the enable password, do the following:

RouterA(config)#enable secret cisco **i Sets the enable password (encrypted)**

Router B:

```
RouterB(config)#enable secret cisco
```

To configure a default route, there is one simple step (in configuration mode):

RouterA(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0 **i For all unknown addresses, send the packet out of Serial0/0**

Router B:

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

5. To configure an access list, there are two steps: first, specify the networks to permit or deny; and second, apply the access list to an interface:

RouterA(config)#access-list 100 deny tcp any any eq 23 **i Denies any TCP connection using Telnet**

RouterA(config)#access-list 100 permit ip any any **i Permits everything else**

RouterA(config)#interface Serial0/0

RouterA(config-if)#ip access-group 100 in **i Assigns the access-list to the interface and the direction of traffic to be checked**

Router B:

```
RouterB(config)#access-list 100 deny tcp any any eq 80
```

```
RouterB(config)#access-list 100 permit ip any any
```

```
RouterB(config)#interface Serial0/0
```

```
RouterB(config-if)#ip access-group 100 in
```

```
RouterB(config-if)#exit
```

RouterB(config)#ip http server **i Will permit Telnet on port 80**

6. To test this access list, you will need to Telnet to the neighbor router. If the access list is working, the connection will be denied:

```
RouterA#telnet 192.168.1.2 80 i Telnet using port 80 to test
Trying 192.168.1.2 ...
% Destination unreachable; gateway or host down
```

Router B:

```
RouterB#telnet 192.168.1.1
Trying 192.168.1.1 ...
% Destination unreachable; gateway or host down
```

7. To make sure the access list is doing its job, remove the access list from the Serial interface and try the Telnet connection again. Router B must have the ip http server command added so that you can test the Telnet connection on port 80.

Router B:

```
RouterB#config t
RouterB(config)#interface Serial0/0
RouterB(config-if)#no ip access-group 100 in i Removes the access list from the interface on Router B
RouterB(config-if)#exit
```

Router A:

```
RouterA#telnet 192.168.1.2 80 i Telnets from Router A to B
Trying 192.168.1.2, 80 ... Open
exit Type exit
HTTP/1.0 501 Not Implemented
Date: Mon, 01 Mar 1993 00:17:40 UTC
Content-type: text/html
Expires: Thu, 16 Feb 1989 00:00:00 GMT
[H1]501 Not Implemented[/H1]
[Connection to 192.168.1.2 closed by foreign host]
RouterA#
```

Router A:

```
RouterA#config t
RouterA(config)#interface Serial0/0
RouterA(config-if)#no ip access-group 100 in i Removes the access list from
```

the interface on Router A

RouterB#telnet 192.168.1.1 **i Telnets from Router B to A**

02:03:55: %SYS-5-CONFIG_I: Configured from console by console 192.168.1.1

Trying 192.168.1.1 ... Open

User Access Verification

Username: banbury

Password:

RouterA>

The Telnet connection should be successful now because the access list is no longer in use.

Show Runs

RouterA#show run

Building configuration...

Current configuration : 799 bytes

!

version 15.1

!

hostname RouterA

!

enable secret 5 \$1\$jjQo\$YJXxLo.EZm9t6Sq4UYeCv0

!

username banbury password 0 ccna

!

ip subnet-zero

!

interface Loopback0

ip address 172.16.1.1 255.255.0.0

!

interface Loopback1

ip address 172.20.1.1 255.255.0.0

!

interface Serial0/0

ip address 192.168.1.1 255.255.255.252

ip access-group 100 in

clockrate 64000

!

ip classless

```
ip route 0.0.0.0 0.0.0.0 Serial0/0
ip http server
!
access-list 100 deny tcp any any eq telnet
access-list 100 permit ip any any
!
end
```

```
RouterB#show run
Building configuration...

Current configuration : 781 bytes
!
version 15.1
!
hostname RouterB
!
enable secret 5 $1$HrXN$ThplDHEZdnCbbeA/Ie67E1
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
ip access-group 100 in
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
ip http server
!
access-list 100 deny tcp any any eq www
```

```
access-list 100 permit ip any any  
!  
end
```

RouterB#

Lab 4: Access Lists (Named)

The physical topology is shown in Figure 7.24 below:

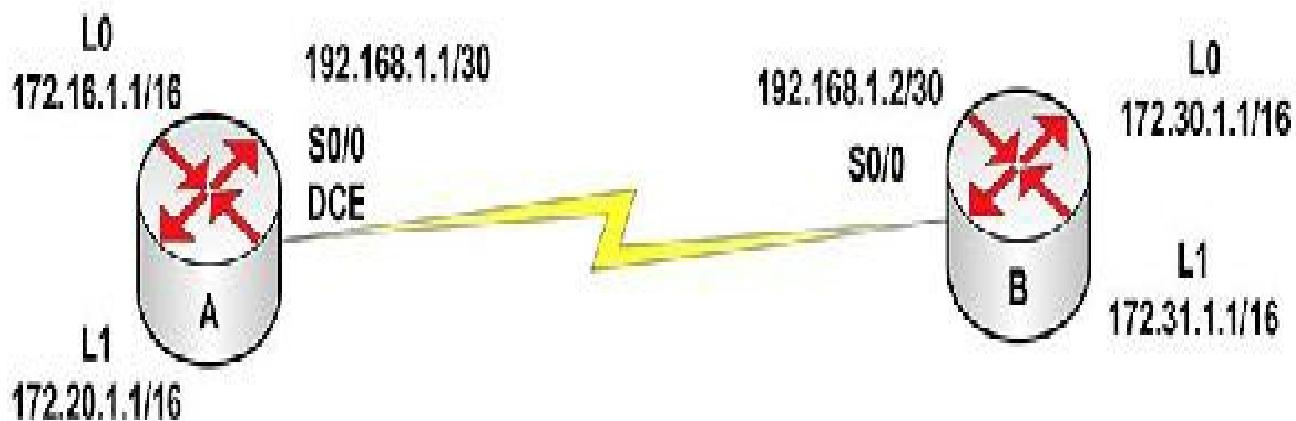


FIG 7.24 – Named access lists

Lab Exercise

Your task is to configure the network in Figure 7.24 to allow full connectivity using a default route. Then you will need to configure a named access list to permit pings from Loopback 0 on Router B to Loopback 0 on Router A and Telnet traffic to Loopback 1 on Router A only. Any other traffic will be denied (which is done by default). Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Named access lists are one of the foundation skills of any competent CCNA. You will be expected to be able to configure one to protect a client's network from certain types of traffic. Practice them over and over again and write them out on paper before you configure them.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.24. Router A needs a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.

3. Configure the enable password to be cisco.
4. Configure a default route to allow full connectivity.
5. Configure an access list on Router A to permit ICMP from 172.30.1.1 to 172.16.1.1 and Telnet to 172.20.1.1 only.
6. Finally, to test that the access list is working, you will need to telnet to the neighbor router.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```

Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#

```

Router B:

```

Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#

```

2. To set the clock rate on a Serial interface (DCE connection only), you need to use the `clock rate #` command on the Serial interface, where # indicates the speed:

```
RouterA(config-if)#clock rate 64000
```

Ping across the Serial link now.

3. To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration
```

```
RouterA(config-line)#login local i This will use local usernames and passwords  
for Telnet access
```

```
RouterA(config-line)#exit i Exits the VTY config mode
```

```
RouterA(config)#username banbury password ccna i Creates username and  
password for Telnet access (login local)
```

Router B:

```
RouterB(config)#line vty 0 4
```

```
RouterB(config-line)#login local
```

```
RouterB(config-line)#exit
```

```
RouterB(config)#username banbury password ccna
```

4. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco i Sets the enable password (encrypted)
```

Router B:

```
RouterB(config)#enable secret cisco
```

To configure a default route, there is one simple step (in configuration mode):

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0 i For all unknown addresses,  
send the packet out of Serial0/0
```

Router B:

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

5. To configure an access list, there are two steps: first, specify the networks and traffic to permit or deny; and second, apply the access list to an interface:

```
RouterA#config t
```

```
RouterA(config)#ip access-list extended secure_LAN
```

```
RouterA(config-ext-nacl)#permit icmp host 172.30.1.1 host 172.16.1.1 i Goes  
into Named mode
```

```
RouterA(config-ext-nacl)#permit tcp any host 172.20.1.1 eq telnet
```

```
RouterA(config-ext-nacl)#exit
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip access-group secure_LAN in i Assigns the access-list to the interface and the direction of traffic to be checked
```

Router B:

To test this access list, you will need to telnet to the neighbor router; if the access list is working, the connection will be denied:

RouterB#telnet 192.168.1.1 i Telnets to the Serial interface

Trying 192.168.1.1 ...

% Destination unreachable; gateway or host down

RouterB#telnet 172.20.1.1 i Telnet to Loopback1 will work

Trying 172.20.1.1 ... Open

User Access Verification

Password:

Test the ICMP deny statement by pinging Loopback 0 from the Serial interface on Router B:

RouterB#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

RouterB#

Now, ping from source interface 172.30.1.1, which should be permitted:

RouterB#ping

Protocol [ip]:

Target IP address: 172.16.1.1

Repeat count [5]:

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]: y

Source address or interface: 172.30.1.1

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

Show Runs

```
RouterA#show run
Building configuration...

Current configuration : 831 bytes
!
version 15.1
!
hostname RouterA
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Loopback1
ip address 172.20.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
clockrate 64000
ip access-group secure_LAN in
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
!
ip access-list extended secure_LAN
permit icmp host 172.30.1.1 host 172.16.1.1
permit tcp any host 172.20.1.1 eq telnet
!
line con 0
line aux 0
line vty 0 4
password cisco
```

```
login
!
end

---
RouterB#show run
Building configuration...

Current configuration : 574 bytes
!
version 15.1
!
hostname RouterB
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
!
end
```

Lab 5: Static NAT

The physical topology is shown in Figure 7.25 below:

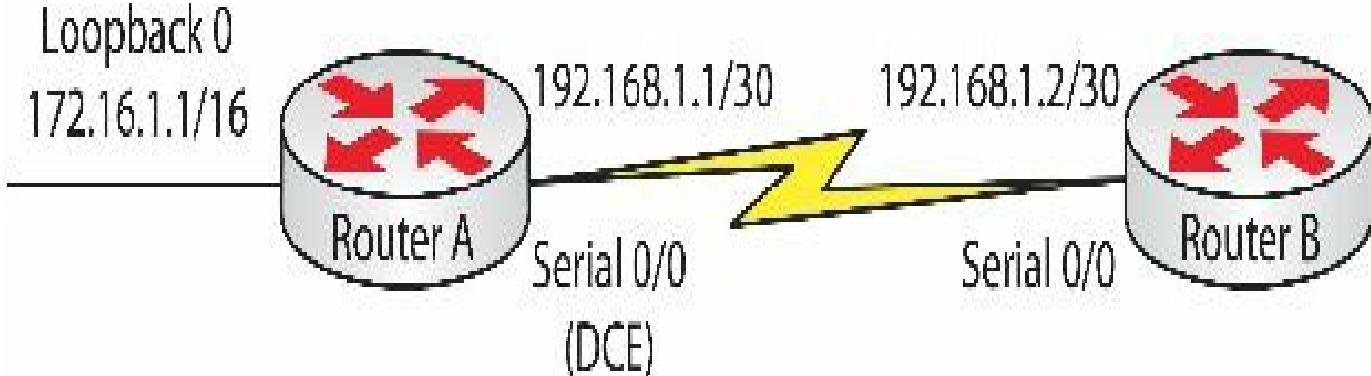


FIG 7.25 – Static NAT

Lab Exercise

Your task is to configure the network in Figure 7.25 to allow the 172.16.1.1 host on the 172.16.1.0 LAN to access the Internet using address 10.0.0.1. The 10.0.0.1 address would normally be a routable public address provided by your ISP (such as 80.1.1.1), but for this lab, we will use a private non-routable address. Please feel free to try the lab without following the Lab Walk-through section and then just check the notes for the NAT configuration.

Purpose

Being able to configure NAT is a fundamental CCNA skill. Any client who needs to access the Internet will want to use NAT. The key is to understand the clients' requirements and then design a solution to fit their needs.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.25. Router A needs a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna (not needed for NAT but added for practice purposes).
3. Put a static route on the router.
4. Configure the inside and outside NAT interfaces on the router.
5. Configure a static NAT translation.
6. Test the NAT translation.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```
Router#config t  
Router(config)#hostname RouterA
```

```
RouterA(config)#  
RouterA(config)#interface Serial0/0  
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252  
RouterA(config-if)#clock rate 64000 i If this is the DCE side  
RouterA(config-if)#no shutdown  
RouterA(config-if)#ip nat outside i The outside NAT network  
RouterA(config-if)#interface Loopback0 i No need for no shutdown on Loopback interfaces  
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0  
RouterA(config-if)#ip nat inside i The inside NAT network  
RouterA(config-if)#^Z  
RouterA#
```

Router B:

```
Router#config t  
Router(config)#hostname RouterB  
RouterB(config)#interface Serial0/0  
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252  
RouterB(config-if)#no shutdown  
RouterB(config-if)#exit  
RouterB(config)#ip route 0.0.0.0 0.0.0.0 s0/0 i Static route  
RouterB(config)#^Z  
RouterB#
```

Ping across the Serial link now.

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access; to do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4 i Enters the VTY line configuration  
RouterA(config-line)#login local i This will use local usernames and passwords for Telnet access  
RouterA(config-line)#exit i Exist the VTY config mode  
RouterA(config)#username banbury password ccna i Creates username and password for Telnet access (login local)
```

Router B:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit
```

```
RouterB(config)#username banbury password ccna
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco i Sets the enable password (encrypted)
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. Configure a static NAT translation:

```
RouterA(config)#ip nat inside source static 172.16.1.1 10.0.0.1
```

```
RouterA(config)#^Z
```

5. To see if NAT is working, you need to turn on a debug with the debug ip nat command. Now imagine that the Loopback address 172.16.1.1 is a host in the LAN that wants to get out to the Internet. When the packet from the NATed LAN passes through the router, it will match the access list and be statically translated to 10.0.0.1.

```
RouterA#debug ip nat i Turns on the NAT debug
```

```
RouterA#ping
```

Protocol [ip]:

Target IP address: 192.168.1.2 **i Pings Router B**

Repeat count [5]:

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]: y

Source address or interface: Loopback0 **i Source is the LAN**

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [n]:

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/40 ms

```
03:48:01: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [35]
```

```
03:48:01: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [35]
```

```
03:48:01: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [36]
```

```
03:48:01: NAT*: s=192.168.1.2, d=10.0.0.1->172.16.1.1 [36]
03:48:01: NAT: s=172.16.1.1->10.0.0.1, d=192.168.1.2 [37]
03:48:01: NAT*: s=192.168.1.2, d=10.0.0.1->172.16.1.1 [37]
03:48:01: NAT: s=172.16.1.1->10.0.0.1, d=192.168.1.2 [38]
03:48:01: NAT*: s=192.168.1.2, d=10.0.0.1->172.16.1.1 [38]
03:48:01: NAT: s=172.16.1.1->10.0.0.1, d=192.168.1.2 [39]
03:48:01: NAT*: s=192.168.1.2, d=10.0.0.1->172.16.1.1 [39]
```

You can see that the NAT debug shows the source (s=) as the Loopback interface address 172.16.1.1, which is translated to 10.0.0.1. The destination (d=) is the Serial address 192.168.1.2 for Router B. The * shows the returning packet that is translated back. The numbers in brackets (e.g., [35]) are the IP identification numbers of the packets.

```
RouterA#show ip nat translations
Pro Inside global Inside local Outside local Outside global
-- 10.0.0.1    172.16.1.1  ---      ---
```

You can see from the NAT translation table above that the router is doing a 1-to-1 translation of the address.

6. Now please enter reload at the Router# prompt and type yes to confirm.

Show Runs

```
RouterA#show run
Building configuration...
Current configuration : 757 bytes
!
version 15.1
!
hostname RouterA
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
ip nat inside
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
clock rate 64000
ip nat outside
```

```
!
ip nat inside source static 172.16.1.1 10.0.0.1
!
end
```

RouterA#

```
-----
RouterB#show run
Building configuration...
```

Current configuration : 456 bytes

```
!
version 15.1
!
hostname RouterB
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
!
end
```

Lab 6: NAT Pool

The physical topology is shown in Figure 7.26 below:

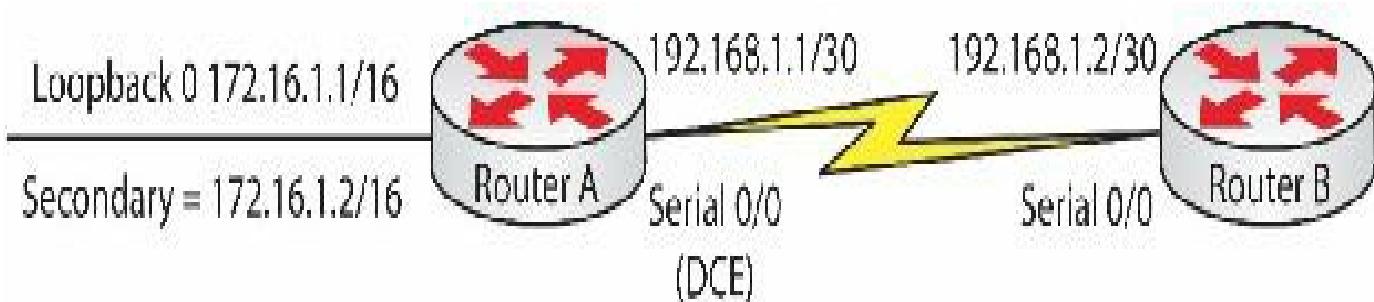


FIG 7.26 – NAT pool

Lab Exercise

Your task is to configure the network in Figure 7.26 to allow the hosts on the 172.16.0.0 LAN (we will simulate this with the Loopback address and secondary address) to

access the Internet using the NAT pool 10.0.0.1 to 10.0.0.10. Please feel free to try the lab without following the Lab Walk-through section.

Purpose

Again, being able to configure NAT is a fundamental CCNA skill. Any client who needs to access the Internet will want to use NAT. The key is to understand the clients' requirements and then design a solution to fit their needs.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.26. Router A needs a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna (optional).
3. Put a static route on the router.
4. Configure the inside and outside NAT interfaces on the router.
5. Configure a pool of addresses the router will use as a NAT pool.
6. Test the NAT configuration with a ping and debug.

Lab Walk-through

1. To set the IP addresses for an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000 If this is the DCE side
RouterA(config-if)#no shutdown
RouterA(config-if)#ip nat outside
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#ip address 172.16.1.2 255.255.0.0 secondary
```

The secondary address will act as a second host on the LAN.

```
RouterA(config-if)#ip nat inside
RouterA(config-if)^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#exit
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
RouterB(config)#^Z
RouterB#
```

Ping across the Serial link now.

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4
RouterA(config-line)#login local
RouterA(config-line)#exit
RouterA(config)#username banbury password ccna
```

Router B:

```
RouterB(config)#line vty 0 4
RouterB(config-line)#login local
RouterB(config-line)#exit
RouterB(config)#username banbury password ccna
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. You need to configure a NAT pool and then tell the pool which access list to access to determine which traffic you want to be NATed:

```
RouterA(config)#ip nat pool internet_out 10.0.0.1 10.0.0.10 prefix-length 24
(or you could have entered ip nat pool internet_out 10.0.0.1 10.0.0.10 netmask
255.255.255.0)
RouterA(config)#ip nat inside source list 1 pool internet_out
RouterA(config)#access-list 1 permit 172.16.0.0 0.0.255.255
RouterA(config)#^Z
```

5. To see if NAT is working, you need to turn on a debug with the debug ip nat command. Now imagine that the Loopback address 172.16.1.1 is a host in the LAN that wants to get out to the Internet. When the packet from the NATed LAN passes through the router, it will match the access list and be translated to an address from the NAT pool.

RouterA#debug ip nat i **Turns on the NAT debug**

RouterA#ping

Protocol [ip]:

Target IP address: 192.168.1.2 i **Pings Router B**

Repeat count [5]:

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]: y

Source address or interface: Loopback0 i **Source is the LAN**

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [n]:

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/40 ms

RouterA#

02:12:37: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [20]

02:12:37: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [20]

02:12:37: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [21]

02:12:37: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [21]

02:12:37: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [22]

02:12:37: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [22]

02:12:37: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [23]

02:12:37: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [23]

02:12:37: NAT: s=172.16.1.1-[10.0.0.1], d=192.168.1.2 [24]

02:12:37: NAT*: s=192.168.1.2, d=10.0.0.1-[172.16.1.1] [24]

RouterA#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
-----	---------------	--------------	---------------	----------------

---	10.0.0.1	172.16.1.1	---	---
-----	----------	------------	-----	-----

RouterA#

You can see that the NAT debug shows the source (s=) as the Loopback interface, which is translated to 10.0.0.1. The destination (d=) is the Serial address 192.168.1.2 for router B. The * shows the returning packet that is translated back.

The numbers in brackets (e.g., [20]) are the IP identification numbers of the packets. Feel free to also issue the show ip nat statistics command.

If you want to check that the pool is allocating addresses correctly, you can source a second ping—this time from the secondary address. There should be another address allocated from the NAT pool.

RouterA#ping

Protocol [ip]:

Target IP address: 192.168.1.2

Repeat count [5]:

Datagram size [100]:

Timeout in seconds [2]:

Extended commands [n]: y

Source address or interface: 172.16.1.2

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [0xABCD]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [n]:

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/46/68 ms

RouterA#

04:09:23: NAT: s=172.16.1.2-[10.0.0.2, d=192.168.1.2 [45]

04:09:23: NAT*: s=192.168.1.2, d=10.0.0.2-[172.16.1.2 [45]

04:09:23: NAT: s=172.16.1.2-[10.0.0.2, d=192.168.1.2 [46]

04:09:23: NAT*: s=192.168.1.2, d=10.0.0.2-[172.16.1.2 [46]

04:09:23: NAT: s=172.16.1.2-[10.0.0.2, d=192.168.1.2 [47]

04:09:23: NAT*: s=192.168.1.2, d=10.0.0.2-[172.16.1.2 [47]

04:09:23: NAT: s=172.16.1.2-[10.0.0.2, d=192.168.1.2 [48]

04:09:23: NAT*: s=192.168.1.2, d=10.0.0.2-[172.16.1.2 [48]

```
04:09:23: NAT: s=172.16.1.2-[10.0.0.2, d=192.168.1.2 [49]
04:09:23: NAT*: s=192.168.1.2, d=10.0.0.2-[172.16.1.2 [49]
RouterA#show ip nat translations
Pro Inside global Inside local Outside local Outside global
--- 10.0.0.1    172.16.1.1 --- ---
--- 10.0.0.2    172.16.1.2 --- ---
```

6. Now please enter reload at the Router# prompt and type yes to confirm.

Show Runs

```
RouterA#show run
Building configuration...
Current configuration : 749 bytes
!
version 15.1
!
hostname RouterA
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
ip address 172.16.1.2 255.255.0.0 secondary
ip nat inside
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
clockrate 64000
ip nat outside
!
ip nat pool internet_out 10.0.0.1 10.0.0.10 prefix-length 24
ip nat inside source list 1 pool internet_out
!
access-list 1 permit 172.16.0.0 0.0.255.255
!
end
```

```
RouterA#
```

```
- - -
```

```

RouterB#show run
Building configuration...

Current configuration : 456 bytes
!
version 15.1
!
hostname RouterB
!
ip subnet-zero
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
ip route 0.0.0.0 0.0.0.0 Serial 0/0
!
end

```

Lab 7: NAT Overload

The physical topology is shown in Figure 7.27 below:

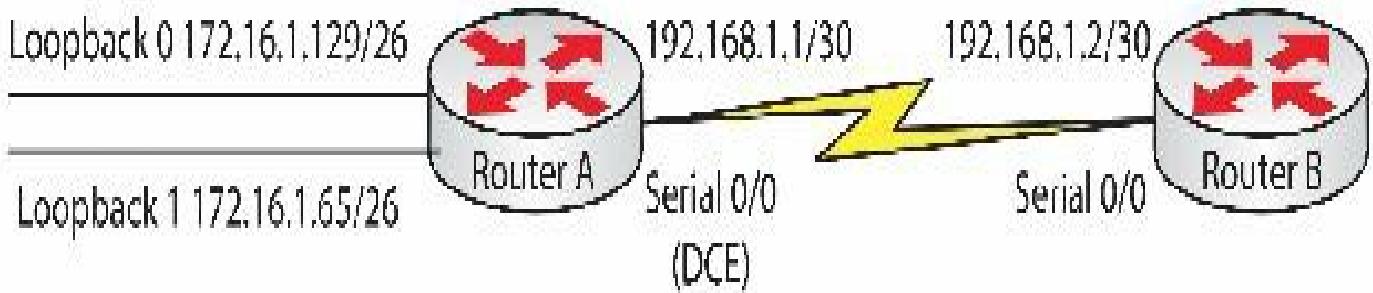


FIG 7.27 – NAT Overload

Lab Exercise

Your task is to configure the network in Figure 7.27 to allow the hosts in the 172.16.1.128 subnet to access the Internet using the NAT Overload address 10.0.0.1. Hosts in the 172.16.1.64 subnet should not be NATed. Please feel free to try the lab without following the Lab Walk-through section.

Purpose

This lab will help you to understand NAT Overload.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 7.27. Router A needs a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Put a static route on the router.
4. Configure the inside and outside NAT interfaces on the router.
5. Configure NAT Overload.
6. Finally, test the NAT Overload from Loopback 0 and Loopback 1.

Lab Walk-through

1. To set the IP addresses for an interface, you will need to do the following:

```

Router#config t
Router(config)#hostname RouterA
RouterA(config)#
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#ip nat outside
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.129 255.255.255.192
RouterA(config-if)#ip nat inside
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.16.1.65 255.255.255.192
RouterA(config-if)#ip nat inside
RouterA(config-if)^Z
RouterA#

```

Router B:

```

Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#exit
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
RouterB(config)^Z
RouterB#

```

Ping across the Serial link now.

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access; to do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4  
RouterA(config-line)#login local  
RouterA(config-line)#exit  
RouterA(config)#username banbury password ccna
```

Router B:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit  
RouterB(config)#username banbury password ccna
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. You need to configure a NAT pool and then tell the pool which access list to access to determine which traffic you want to be NATed:

```
RouterA(config)#ip nat pool internet_out 10.0.0.1 10.0.0.1 prefix-length 24  
RouterA(config)#ip nat inside source list 1 pool internet_out overload  
RouterA(config)#access-list 1 permit 172.16.1.128 0.0.0.63
```

5. To see if NAT is working, you need to turn on a debug with the debug ip nat command. Now imagine that the Loopback address 172.16.1.129 is a host in the LAN that wants to get out to the Internet. When the packet from the NATed LAN passes through the router, it will match the access list and be translated to the NAT Overload address.

```
RouterA#debug ip nat  
IP NAT debugging is on  
RouterA#ping  
Protocol [ip]:  
Target IP address: 192.168.1.2  
Repeat count[5]:  
Datagram size [100]:  
Timeout in seconds [2]:
```

Extended commands [n]: y
Source address or interface: Loopback0
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/41/40ms
RouterA#
00:43:59: NAT: s=172.16.1.129-[10.0.0.1, d=192.168.1.2 [20]
00:43:59: NAT: s=192.168.1.2, d=10.0.0.1-[172.16.1.129 [20]
00:43:59: NAT: s=172.16.1.129-[10.0.0.1 d=192.168.1.2 [21]
00:43:59: NAT: s=192.168.1.2, d=10.0.0.1-[172.16.1.129 [21]
00:43:59: NAT: s=172.16.1.129-[10.0.0.1, d=192.168.1.2 [22]
00:43:59: NAT: s=192.168.1.2, d=10.0.0.1-[172.16.1.129 [22]
00:43:59: NAT: s=172.16.1.129-[10.0.0.1, d=192.168.1.2 [23]
00:43:59: NAT: s=192.168.1.2, d=10.0.0.1-[172.16.1.129 [23]
00:43:59: NAT: s=172.16.1.129-[10.0.0.1, d=192.168.1.2 [24]
00:43:59: NAT: s=192.168.1.2, d=10.0.0.1-[172.16.1.129 [24]

RouterA#show ip nat tran
Pro Inside global Inside local Outside local
Outside global
icmp 10.0.0.1:8759 172.16.1.129:8759 192.168.1.2:8759 192.168.1.2:8759
icmp 10.0.0.1:8760 172.16.1.129:8760 192.168.1.2:8760 192.168.1.2:8760
icmp 10.0.0.1:8761 172.16.1.129:8761 192.168.1.2:8761 192.168.1.2:8761
icmp 10.0.0.1:8762 172.16.1.129:8762 192.168.1.2:8762 192.168.1.2:8762
icmp 10.0.0.1:8763 172.16.1.129:8763 192.168.1.2:8763 192.168.1.2:8763

RouterA#
00:44:59: NAT: expiring 10.0.0.1 (172.16.1.129) icmp 8759 (8759)
00:44:59: NAT: expiring 10.0.0.1 (172.16.1.129) icmp 8760 (8760)
00:44:59: NAT: expiring 10.0.0.1 (172.16.1.129) icmp 8761 (8761)
00:44:59: NAT: expiring 10.0.0.1 (172.16.1.129) icmp 8762 (8762)
00:44:59: NAT: expiring 10.0.0.1 (172.16.1.129) icmp 8763 (8763)

You can see from the NAT translation output above that the router is allocating

ports for the translations (i.e., ports 8759 to 8763).

Now ping from Loopback 1, which does not match the access list because it is in a different subnet. The address should not be NATed.

```
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: Loopback 1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/37/60 ms
RouterA#show ip nat tran
RouterA#
```

6. Now please enter reload at the Router# prompt and type yes to confirm.

Show Runs

```
RouterA#show run
Building configuration...
Current configuration : 757 bytes
!
version 15.1
!
hostname RouterA
!
ip subnet-zero
!
```

```
interface Loopback0
ip address 172.16.1.129 255.255.255.192
ip nat inside
!
Interface Loopback1
ip add 172.16.1.65 255.255.255.192
ip nat inside
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
ip nat outside
clock rate 64000
!
ip nat pool internet_out 10.0.0.1 10.0.0.1 prefix-length 24
ip nat inside source list 1 pool internet_out overload
ip classless
no ip http server
!
access-list 1 permit 172.16.1.128 0.0.0.63
!
!
```

end

RouterA#

- - -

```
RouterB#show run
Building configuration...
```

Current configuration : 456 bytes

!

version 15.1

!

hostname RouterB

!

ip subnet-zero

!

interface Serial0/0

ip address 192.168.1.2 255.255.255.252

```
!  
ip classless  
!  
ip route 0.0.0.0 0.0.0.0 Serial0/0
```

Chapter 8 — Network Device Security

What You Will Learn in This Chapter

Network Security Devices

Network Device Passwords

Securing Network Devices

Switch Port Security

Syllabus Topics Covered

2.0 LAN Switching Technologies

2.4 Verify network status and switch operation using basic utilities such as:

 2.4.c SSH

4.0 IP Routing Technologies

4.2 Configure and verify utilizing the CLI to set basic router configuration

 4.2.a Hostname

 4.2.b Local user and password

 4.2.c Enable secret password

 4.2.d Console and VTY logins

 4.2.e Exec-timeout

 4.2.f Service password encryption

 4.2.g Interface IP address

 4.2.g (i) Loopback

 4.2.h Banner

 4.2.i motd

 4.2.j Copy run start

4.4 Verify router configuration and network connectivity using:

 4.4.d SSH

 4.4.e Show cdp neighbors

6.0 Network Device Security

6.1 Configure and verify network device security features

 6.1.a Device password security

 6.1.b Enable secret versus enable

6.1.c Transport

6.1.c (i) Disable Telnet

6.1.c (ii) SSH

6.1.d VTYs

6.1.e Physical security

6.1.f Service password

6.1.g Describe external authentication methods

6.2 Configure and verify switch port security

6.2.a Sticky mac

6.2.b MAC address limitation

6.2.c Static/dynamic

6.2.d Violation modes

6.2.d (i) Err-disable

6.2.d (ii) Shutdown

6.2.d (iii) Protect restrict

6.2.e Shutdown unused ports

6.2.f Err-disable recovery

6.2.g Assign unused ports in unused VLANs

6.2.h Putting native VLAN in area other than VLAN 1

6.4 Configure and verify ACLs to limit Telnet and SSH access to the router

Network security is an important topic in the world of internetworking and will remain so as long as the threat of intrusion, espionage, theft, or hacking exists. Barely a single day passes without a story breaking about a major company that has suffered from an embarrassing hacking, industrial espionage, or internal security breach by a careless or disgruntled employee.

While the CCNA RS is not a security course, you will be tested on some areas of network security. You should be able to configure routers and switches to allow only certain people to access them and allow only certain traffic to pass through them from your network to the Internet, or vice versa. You should also be able to restrict access to certain applications and services within the network.

If you really enjoy learning about network security, then consider studying for the Cisco CCNA Security exam after passing the CCNA RS exam.

Network Security Devices

As well as the security measures mentioned here, you should have the following devices and services operational on your network:

- Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) – IDS and IPS devices function by performing traffic inspection to detect unauthorized traffic that tries to enter the enterprise network. Their main role is to monitor networks for intrusions or other malicious activity. The actions taken by a device in a promiscuous mode include sending alerts, alarms, log messages, or SNMP traps. The major difference between IPS and IDS is that IPS devices operate in line with the traffic (meaning they are placed in the middle of the traffic flow and all the packets pass through the inspection device), while IDS devices only retrieve a copy of the traffic so they can analyze it.
- Firewall – A firewall is a hardware- or software-based security device that filters traffic that is not allowed in the organization (while allowing legitimate traffic). Firewalls are positioned at the entry point in an organization or between critical network modules to create various security access policies. Most firewalls filter at layer 4 based on the source or destination address and TCP or UDP port number.
- Antivirus – These programs were originally developed to remove malicious computer code, and today they can protect you from browser hijacking, worms, Trojans, adware, and spyware. Big players in this field include AVG and Symantec.
- Antispyware – These programs are similar to antivirus programs inasmuch as they help to block and prevent spyware and other malware types on computers. This type of software can be used to capture sensitive data and transmit it to a target device. Antispyware programs such as Ad-Aware and Spybot monitor incoming data from e-mail, websites, or file downloads and prevent spyware from taking root in your operating system.

Network Device Passwords

Do you want just anybody to access your network devices? Perhaps you want only a handful of people to be able to log on to the router and a few others to be able to remotely connect to the router and administer it. Network device access needs to be protected from internal staff and external intruders.

Passwords on Cisco devices must contain from 1 to 25 uppercase and lowercase alphanumeric characters. Passwords are case sensitive; spaces can be used but not as the first character. Cisco recommends that the best way to handle passwords is to maintain them on a TACACS+ or RADIUS authentication server. Most routers, however, have a locally stored (in the router's configuration) privilege-level password.

Enable Password

Protecting privileged mode (or enable mode) on your router is very important and the process is very simple. When any person attempts to enter privileged mode from user exec mode, they will be prompted for a password.

```
Router>enable  
Router#config t  
Router(config)#enable password cisco i Passwords ARE case sensitive  
Router(config)#exit  
Router#disable  
Router>enable  
Password: i The password will not show as you type it  
Router#
```

By default, the enable password command can be seen when any user looks at the running or startup configuration of the router. You probably do not want this to happen (see the service password-encryption command below).

```
Router#show run  
Building configuration...  
  
hostname Router  
!  
enable password cisco
```

You can disable the enable password command by entering no in front of the command.

```
Router(config)#no enable password
```

You do not need to enter the password again.

Enable Secret

```
Router#conf t  
Router(config)#enable secret cisco  
Router(config)#exit  
Router#disable  
Router>enable  
Password: i The password will not show as you type it  
Router#
```

You can see that when a show running-configuration command is issued, the enable

secret command is encrypted. Only the relevant part of the configuration is shown below:

```
Router#show running-configuration  
Building configuration...
```

```
hostname Router
```

```
!
```

```
enable secret 5 $1$F3Dy$w0mwxVmJ79Ug9pK/snpRe/ i Hashed using the MD5 algorithm
```

The number 5 after enable secret stands for level 5 encryption. This uses a hashed value of the MD5 algorithm and it is harder to crack than level 7, which uses a weaker algorithm. If you forget your password, you will have to do password recovery using the console port (check Google for “[router model] password recovery” because each router and switch model has a slightly different recovery process).

Newer IOS releases also offer the SHA256 encryption algorithm (number 4).



Service Password Encryption

You can actually encrypt all of the passwords on the router with the service password-encryption command. This command will encrypt all current and future passwords added to the router.

```
Router(config)#enable password cisco  
Router(config)#service password-encryption  
Router(config)#exit  
Router#show run  
!  
service password-encryption  
hostname Router  
enable password 7 070724404206 i Weaker reversible algorithm
```

The service password-encryption command does not provide a high level of security. Use this command with additional security measures.

Auxiliary Password

In order to protect connections through the AUX port, you will need to assign a password to it. Note that when you configure the AUX port, the router drops into config-line mode as shown below:

```
Router#config t
Router(config)#line aux ?
[0-0] First Line number
Router(config)#line aux 0
RouterA(config-line)#password cisco i Config-line mode
Router(config-line)#login
Router(config-line)#^Z
Router#
```

The login command is very important as it tells the router to ask the user for a password. The command login local (see the Configuring Local Usernames and User-Specific Passwords section) tells the router to check a username and password you have configured on the router itself (the local database). You can instead put a server in the network, which does the job of authenticating all the users. These servers are known as TACACS or RADIUS servers, which are outside the scope of the CCNA exam.

The login and login local commands are covered comprehensively in the labs throughout this book.

Mini-lab – Adding a Telnet Password

In order to connect to your router over the Internet or remotely, you may want to telnet to it. To allow Telnet sessions, you need to have a password set on the VTY line. Terminal lines are logical (i.e., not physically attached to the router), so you will normally telnet via the Serial port or Ethernet port and a virtual terminal (known as VTY) will be opened. The number of available VTY ports depends on your router model; mine below has five. Please configure the IP addressing and hostnames as per Figure 8.1 below. Ping across the link to ensure that the network is working.

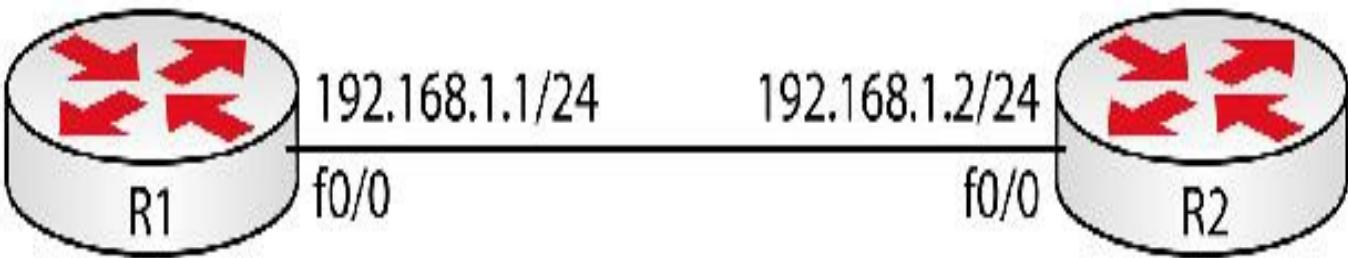


FIG 8.1 – Mini-lab: Adding a Telnet Password

```
R1#config t
R1(config)#enable secret howtonetwork
R1(config)#line vty 0 4 i Use ? to see your available VTY port numbers
R1(config-line)#password cisco
R1(config-line)#login
R1(config-line)#^Z
```

Now you can telnet to R1 from R2:

```
R2#telnet 192.168.1.1
Trying 192.168.1.1 ... Open
User Access Verification
Password:
R1>enable
Password:
R1# i You are now connected to Router A from Router B
R1#exit i You can use Ctrl+Shift+6 and x to exit
```

[Connection to 192.168.1.1 closed by foreign host]

R2#

[END OF MINI-LAB]

Configuring the router for Telnet access alone is not sufficient. The enable or enable secret command must also be configured to allow for privileged access once Telnet access has been allowed. Try it for yourself with and without an enable command on the device you are telnetting to. Remember also that you can protect the VTY lines with an access list, but you must apply it using the access-class command. We covered this in the ACL sections and hands-on labs in Chapter 7.

If there is no VTY password on the remote router, you will see:

```
R2#telnet 192.168.1.1
Trying 192.168.1.1 ... Open
```

Password required, but none set

[Connection to 192.168.1.1 closed by foreign host]

R2#

You can test this yourself by adding the command below to the R1 VTY lines:

R1(config-line)#no password

There must be either a login local password or a username and password configured on the router and the login local command issued under the VTY lines. If a VTY password is set but there is no enable command, you will still be able to telnet across, but once you try to enter privileged mode will you see:

R2#telnet 192.168.1.1

Trying 192.168.1.1 ... Open

User Access Verification

Password:

R1>enable

% No password set

R1>

If you are connected to R1, you can see other connections to the router with the show line command:

R1#show line

Tty	Typ	Tx/Rx	A Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
*	0 CTY	- - - - -	0	0	0/0	-				
	1 AUX	9600/9600	- - - - -	0	0	0/0	-			
*	2 VTY	- - - - -	2	0	0/0	-				
	3 VTY	- - - - -	0	0	0/0	-				
	4 VTY	- - - - -	0	0	0/0	-				
	5 VTY	- - - - -	0	0	0/0	-				
	6 VTY	- - - - -	0	0	0/0	-				

The * indicates that there is an active connection on that line. The CTY is the console port, and as you can see the console connection is active. The AUX is for connections to the auxiliary port. Finally, the VTY is for the virtual terminal lines that are used for inbound Telnet connections.

You may want to clear a Telnet session coming into your router either to throw the user

off or to free up a VTY line that should have cleared but has not. To do this, you would use the clear line # command:

```
R1#clear line 2  
[confirm]  
[OK]
```

In the output below, on my own network I have created the username paul and have allowed incoming Telnet sessions. When a user telnets to my router, I can check the connection that is in use with the show users command.

```
R2#show users
```

Line	User	Host(s)	Idle	Location
* 0 con 0		idle	00:00:00	
98 vty 0	paul	idle	00:00:17	192.168.1.1
Interface	User	Mode	Idle	Peer Address

I have VTY lines 0 to 933 available on this router because I'm using GNS3. You would think that the lines would be used in ascending order from 0 upward but that is not so. I've found that a random number is used from the range available. In the output above it's 98. You could technically protect your VTY lines by allowing only one incoming connection with the configuration below. Test it for yourself; however, I couldn't find a way to control which line you actually telnetted on. Also, note that the incoming session below came in on VTY 194.

```
R2(config)#line vty 0  
R2(config-line)#login local  
R2(config-line)#end
```

Now I will telnet in from R1 (IP address 192.168.1.1):

```
R2#show users
```

Line	User	Host(s)	Idle	Location
* 0 con 0		idle	00:00:00	
194 vty 0	paul	idle	00:00:14	192.168.1.1
Interface	User	Mode	Idle	Peer Address

```
R2#show line
```

Tty	Line	Typ	Tx/Rx	A Modem	Roty	AccO	AccI	Uses	Noise	Overruns	Int
*	0	0	CTY	-	-	-	-	0	0	0/0	-

1	1	AUX	9600/9600	-	-	-	-	0	0	0/0	-
*	194	194	VTY	-	-	-	-	1	0	0/0	-
195	195	VTY		-	-	-	-	0	0	0/0	-
196	196	VTY		-	-	-	-	0	0	0/0	-
197	197	VTY		-	-	-	-	0	0	0/0	-
198	198	VTY		-	-	-	-	0	0	0/0	-

The show users command will display incoming connections to your router and the show sessions command will display outgoing connections from your router to another device. There is a fourth type of connection known as a TTY. These are asynchronous lines used for modem and terminal connections.

Console Password

It is very important to protect your console port on the router. If you do not, any person who can get physical access to the router will be able to reconfigure and reboot it.

```
Router#config t
Router(config)#line console ?
[0-0] First Line number
Router(config)#line console 0
Router(config-line)#password hello
Router(config-line)#login
Router(config-line)#exit
Router(config)#exit
```

For added security you can specify a timeout value to lock the console connection if there is no activity for a specified amount of minutes. This will also work on the VTY and AUX ports.

```
Router(config)#line console 0
Router(config-line)#exec-timeout 5 i Sets timeout for 5 minutes
```

Timeout values can be set on AUX, console, and VTY lines. The default timeout value is 10 minutes. If you want to set it to never timeout, then the value must be 0. This does represent a security issue though as the lines will always be open.

Configuring Local Usernames and User-Specific Passwords

You may not want to have a generic password on your connections to the router. You can configure specific username and password combinations on a per-user basis.

```
RouterA#config term
RouterA(config)#username paul password cisco
RouterA(config)#username stuart password ccna
RouterA(config)#username davie password rugby
RouterA(config)#line vty 0 4
RouterA(config-line)#login local
RouterA(config-line)#exit
RouterA(config)#exit
```

I can now telnet from Router B to Router A providing I know my username and password:

```
RouterB#telnet 192.168.1.1
Trying 192.168.1.1 ... Open
```

User Access Verification

Username: paul

Password:

Securing Network Devices

Privilege Levels

You can assign different levels of access to different users based on their local user accounts on the router. For example, you might restrict junior members of the network team to use only basic show commands. This is done using privilege levels. Cisco has 16 privilege levels ranging from 0 to 15, where 15 is full access.

You can assign a specific privilege level to a user and assign some commands to that level. This is shown in the output below:

```
RouterA#conf t
RouterA(config)#username juniortech privilege 4 password support
RouterA(config)#privilege exec level 4 ping
RouterA(config)#privilege exec level 4 traceroute
RouterA(config)#privilege exec level 4 show ip interface brief
RouterA(config)#line console 0
RouterA(config-line)#password basketball
RouterA(config-line)#login local i Password is needed
RouterA(config-line)#^z
```

If a junior technician tries to log in using the juniortech username, and tries to make a

configuration change, access is denied because the command is not allowed in privilege level 4.

RouterA con0 is now available

Press RETURN to get started.

User Access Verification

Username: juniortech

Password:

RouterA#config t i **Not allowed to use this command**

^

% Invalid input detected at “^” marker.

Login

The login local password on the console, AUX, or VTY line overrides the console, AUX, or VTY password so that any person who telnets to Router A will be asked for his/her username and password from the local database.

You can use the enable secret [level] password command to define a password for a specific level of access and then give the password only to those who you want to have that level of access. You would then use the privilege exec level command to specify the commands available at the various access levels.

Logging Router Access

As a network administrator, it is very likely that you will want to be aware of who is attempting to log in to your network, as well as aware of any other network events.

Local Logging

There are several features included in the Cisco IOS to monitor events locally on the router:

- logging console [level] – This command monitors connections via the console port. Levels can be from 0 to 7. You can use the no logging console command to turn off console output if you do not want it to appear on your screen constantly.

Table 8-1: Logging levels

Level	Logging Message
0	Emergencies
1	Alerts
2	Critical

3	Errors
4	Warnings
5	Notifications
6	Informational
7	Debugging

- terminal monitor – This command will allow debug and system messages to appear on your terminal connection to a router. If you are telnetting to a remote router (via SSH or Telnet) you will need to use this command if you want to see the debug commands. The console connection already monitors the terminal.
- logging buffered [size in bytes | level] – This command will allow log messages to be kept in the router's memory.
- access list [specify action] log – This command enables the logging of packets that match the ACL configuration line criteria (e.g., access list 10 permit 192.168.1.1 log).
- service timestamps – This command allows the router to timestamp logging or debug messages.
- logging host address – This command will send logging messages to a syslog server (e.g., Router(config)#logging 172.16.1.5).

You can view the current logging levels with the show logging command:

```
RouterB#show logging
```

```
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
```

```
Console logging: level debugging, 20 messages logged
```

```
Monitor logging: level debugging, 0 messages logged
```

```
Buffer logging: level debugging, 20 messages logged
```

```
Trap logging: level informational, 24 message lines logged
```

```
Log Buffer (4096 bytes):
```

```
00:00:09: %LINK-3-UPDOWN: Interface Serial0, changed state to down
```

```
00:00:09: %LINK-3-UPDOWN: Interface Serial1, changed state to down
```

```
00:01:43: %LINK-5-CHANGED: Interface BRI0, changed state to administratively down
```

In summary, logging options include:

- logging buffered – logs messages to the router buffer
- logging host – logs messages to a syslog server
- logging console – logs messages to the console (you need to enable the terminal

monitor command to see the logs on non-console connections)

You can have logging and/or debug messages timestamped with the service timestamps debug datetime msec localtime and/or service timestamps log datetime msec localtime commands. If you want, you can clear the logging buffer with the clear logging command. If you have requested assistance from a Cisco TAC, they may ask you to enable timestamps on your debug messages to assist them with troubleshooting your issue. Below are two debug outputs, the first without and the second with a timestamp:

%LINK-3-UPDOWN: Interface Serial0, changed state to down

00:00:09: %LINK-3-UPDOWN: Interface Serial0, changed state to down

Please do type the commands above onto a router so they stick in your mind. We will look at syslog again in Chapter 15, Advanced IP Services.

Prevent Telnet Access

All traffic sent using Telnet (including network configuration commands and passwords) are sent in clear text, which means that the configuration commands being sent over a Telnet session can easily be captured by a network sniffer if it is attached to the network. This makes Telnet inherently insecure.

Figure 8.2 below is a packet capture of an incoming Telnet session. You can clearly see that the sent password is cisco.

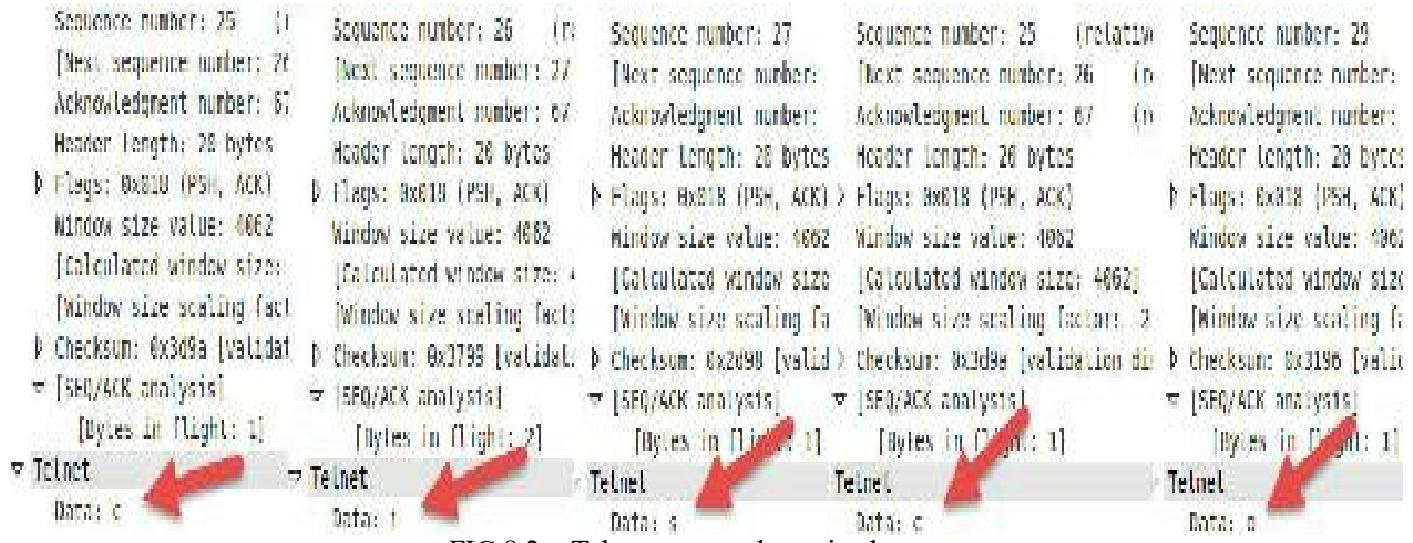


FIG 8.2 – Telnet password sent in clear text

Telnet is disabled by default since you need to set a password (and an optional username) to enable it. For secure remote management access to the router or switch, you can enable SSH (Secure Shell), which is described below. Later we will cover how to use ACLs to protect Telnet access if you need to enable it.

Enable SSH

It is highly recommended that you use SSH rather than Telnet to access your network devices whenever possible. Unlike Telnet, which sends traffic in clear text, SSH creates a secure encryption channel where all traffic sent to the switch is encrypted. This secures the traffic from packet sniffing attacks.

SSH requires Cisco IOS versions that support cryptography. These are the security versions of the IOS. An easy way to check whether your IOS supports cryptography is using the show version command. Security IOS versions usually have **K9** or **security** in their names.

Switch#sh version

Cisco IOS Software, C3560 Software (C3560-ADVIPSERVICES **K9-M**), Version 15.2(35)SE1, RELEASE SOFTWARE (fc1)

[output truncated]

System image file is flash:/c3560-advpipservices**k9-mz.152-35.SE1.bin**

If you do not have a security version of IOS, you must purchase a license for it.

A public/private key pair is used for SSH encryption. Traffic sent to a device is encrypted using a public key and the encrypted traffic is decoded using the private key when it gets to the device. Users are authenticated through a username/password combination. Before generating the keys on a device, you need to set the hostname and password of the switch because the keys are identified using Fully Qualified Domain Name (hostname.domainname).

The steps to enable SSH are as follows:

- Set the hostname and domain name
- Generate crypto keys for encryption; SSH is enabled at this point
- Set other SSH parameters such as idle timeout and authentication-retries (optional)

Mini-lab – Enabling SSH Access

You already know how to add an IP address and default gateway on a switch. Take a router and add the IP address below to the Fast Ethernet interface and connect it to a switch via a straight-through cable. Add an IP address to the management VLAN and ping from the router to the switch.

Router(config)#interface f0/0

Router(config-if)#ip address 192.168.1.1 255.255.255.0

```
Switch(config)#ip default-gateway 192.168.1.1  
Switch(config)#interface vlan 1  
Switch(config-if)#ip address 192.168.1.2 255.255.255.0  
Switch(config-if)#no shut
```



FIG 8.3 – Mini-lab: Enabling SSH Access

An example is shown below:

```
Switch(config)#hostname SwitchOne  
SwitchOne(config)#ip domain-name howtonetwork.com  
SwitchOne(config)#crypto key generate rsa
```

The name for the keys will be: SwitchOne.howtonetwork.com

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

```
SwitchOne(config)#ip ssh time-out 60  
SwitchOne(config)#ip ssh authentication-retries 2  
SwitchOne(config)#line vty 0 15  
SwitchOne(config-line)#transport input ssh  
SwitchOne(config-line)#password cisco
```

You can specify the SSH version using the ip ssh version [1|2] command (version 2 is the default in modern IOS versions).

To verify that SSH is enabled, as well as the version of SSH that is enabled on a switch,

use the show ip ssh command:

```
SwitchOne#show ip ssh  
SSH Enabled - version 2.0  
Authentication timeout: 60 secs; Authentication retries: 2
```

Now you can attempt to connect from the router to the switch. Cisco documentation on how to do this is somewhat light.

```
Router#ssh -l paul 192.168.1.2
```

Open

Password:

```
SwitchOne>
```

If you try to telnet from the router to the switch the connection will be rejected.

```
Router#telnet 192.168.1.2
```

Trying 192.168.1.2 ...

```
% Connection refused by remote host Router#
```

[END OF MINI-LAB]

The output below displays how to permit both Telnet and SSH.

```
Router(config-line)#line vty 0 15  
Router(config-line)#transport input ssh telnet
```

Disable HTTP

You can disable HTTP access using the no ip http server command. Routers can be accessed, managed, and configured via a web page using HTTP, so unless you need to run it you should disable it.

```
Switch(config)#no ip http server
```

To view the status of the HTTP server on the switch:

```
Switch#show ip http server status  
HTTP server status: Disabled  
HTTP server port: 80  
[output truncated]
```

Disable CDP

Cisco Discovery Protocol is a Cisco proprietary data link layer protocol used to

discover the Cisco devices that are attached to a particular device. Although CDP can be a really useful troubleshooting protocol, it can pose security flaws since the device information might be available to anyone who connects to it. Because it runs at layer 2 of the OSI model, it doesn't require an IP address to be configured to exchange information with connected devices.

You can disable CDP in your network or, at least, on devices that are at the edge of your network that connect to other devices that you do not trust.

An example of a CDP output on a switch is shown below:

```
Router#show cdp neighbor detail
```

Device ID: Switch

Entry address(es):

Platform: Cisco 2960, Capabilities: Switch

Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/2

Holdtime: 176

Version :

Cisco Internetwork Operating System Software

IOS (tm) C2960 Software (C2960-I6Q4L2-M), Version 15.1(22)EA4, RELEASE

SOFTWARE(fc1)

Copyright (c) 1986-2005 by Cisco Systems, Inc.

Compiled Wed 18-May-05 22:31 by jharirba

advertisement version: 2

Duplex full

The output above demonstrates why CDP is a very useful troubleshooting tool. To turn off CDP on an entire device, use the no cdp run command:

```
Switch(config)#no cdp run
```

To turn off CDP on an interface, use the no cdp enable interface configuration command. You must know the difference between these two commands for the CCNA exam.

```
Switch(config)#int fast0/2
```

```
Switch(config-if)#no cdp enable
```

Add a Banner Message

Banner messages are displayed when a user connects to a device. Although these messages do not provide any actual security, they can be used to display warning messages and company policies, which can be very useful legally.

When configuring a banner, a delimiting character is selected to tell the router when the banner message is complete. In the example below, the delimiting character is Y:

```
Switch(config)#banner motd Y
```

Enter TEXT message. End with the character “Y”.

```
KEEP OUT OR YOU WILL REGRET IT Y
```

```
Switch(config)#
```

Referring to the output below, when telnetting to the router and the MOTD banner appears, notice that the banner message is truncated. This is because Y was chosen as the delimiting character. To avoid this, always select a delimiting character that is not used in your banner message.

```
Router#telnet 192.168.1.3
```

Trying 192.168.1.3 ...Open

```
KEEP OUT OR Y
```

The banner in the example above is a message of the day (MOTD) banner, which is shown before the user sees the login prompt. Other types of banner messages include:

- Login – shown before the user sees the login prompt
- Exec – shown to user after login prompt; used when you want to hide the banner message from unauthorized users

There are banner inputs as part of the labs at the end of the chapter. I suggest that you learn to configure all three types and test them by logging in to the router. You will have different choices depending on your platform and IOS:

```
Router(config)#banner ?
```

LINE c banner-text c, where “c” is a delimiting character

exec Set EXEC process creation banner

incoming Set incoming terminal line banner

login Set login banner

motd Set Message of the Day banner

prompt-timeout Set Message for login authentication timeout

slip-ppp Set Message for SLIP/PPP

External Authentication Methods

We have explored how to configure local usernames on the router. Although the method is easy, it is not scalable since it can become very difficult to manage many usernames on a router or a switch. A better alternative is to use external authentication methods. External authentication can be provided using either the TACACS+ or RADIUS

protocol.

- TACACS+ (Terminal Access Controller Access Control System Plus) is a Cisco proprietary protocol that operates on TCP port 49. It is used to provide access control to network devices.
- RADIUS (Remote Authentication Dial-In User Service) is an open standard protocol used to provide secure remote access to the network. It operates on UDP ports 1812 and 1813.

To set up a router or a switch to use RADIUS/TACACS+, you need to set up AAA on the device. This is, however, beyond the scope of the CCNA exam. If you are interested in finding out how to go about this, it's covered in CompTIA Network+ and CCNA Security study guides.

Shut Down Unused Ports

Any switch ports that are not being used should be both shut down and placed into an unused VLAN.

```
Switch(config)#interface range f0/10-20
Switch(config-if-range)#switchport access vlan 500
% Access VLAN does not exist. Creating vlan 500
Switch(config-if-range)#shutdown
```

Network Device Clock and NTP

For incident management and logging, it is important for a network device to have accurate timestamps with its logging messages. You can view the time on a device using the show clock command:

```
Switch#show clock
*23:09:45.773 UTC Tue Mar 2 1993
```

To set the time, use the clock command in privileged mode. The commands shown below show how to set the time zone and recurring summertime on a switch:

```
clock timezone CST -6
clock summer-time CDT recurring
clock summer-time CST recurring 2 Sun Mar 2:00 1 Sun Nov 2:00
```

The output below shows how you can set the clock on a Cisco device:

```
Switch#clock set 14:55:05 March 29 2016
1d23h: %SYS-6-CLOCKUPDATE: System clock updated from 17:26:01 CST
```

```
Switch#show clock
```

```
*14:55:7.953 UTC Tue Mar 29 2016
```

Another option for updating the time on a network device is using the Network Time Protocol (NTP), which was discussed earlier. You will need to be able to configure your router as an NTP client for the CCNA exam.

Routers sync with an NTP server over TCP port 123. NTP is now included in the CCNA syllabus and it is quite useful to ensure that all the network devices in the same environment have the same time. A router can be configured to sync with an NTP server using the `ntp server` command:

```
Switch(config)#ntp server 134.84.84.84 prefer  
Switch(config)#ntp server 209.184.112.199
```

Update the IOS

One of the easiest ways to ensure that your switch/router is secure is to maintain the software on your Cisco switch and router. IOS updates not only fix bugs, they also provide feature enhancements. Most of Cisco's stackable switches offer lifetime warranties (which includes software updates) so there are no excuses. You can request that Cisco TAC do a bug sweep if you are concerned about any possible issues before an upgrade.

Disable Unused Services

A recommended best practice for increasing security is to disable unused services. An easy way to check the services running on a router is to use the `service` command in global configuration mode. Just use a ? to list the services, as shown below (output truncated):

```
Router(config)#service ?  
compress-config      Compress the configuration file  
config              TFTP load config files  
counters            Control aging of interface counters  
dhcp                Enable DHCP server and relay agent  
tcp-keepalives-in   Generate keepalives on idle incoming network  
tcp-small-servers   Enable small TCP servers (e.g., ECHO)  
telnet-zeroidle     Set TCP window 0 when connection is idle  
timestamps          Timestamp debug/log messages  
udp-small-servers   Enable small UDP servers (e.g., ECHO)
```

Here is a list of common services that should be disabled (or enabled), along with a

brief description of each service:

- no service pad – rarely used; assembles/disassembles packets in asynchronous networking
- no service config – prevents the switch from getting its configuration file from a tftp server in the network
- no service finger – rarely used; disables the finger protocol
- no ip icmp redirect – prevents the router from sending ICMP redirects, which can help attackers learn about the network topology
- no ip finger – another way to disable the finger service
- no ip gratuitous-arps – disables unsolicited arp responses, which can lead to man-in-the-middle attacks
- no ip source-route – disables user-specified hop-by-hop routing to destination
- service sequence-numbers – enables clarity in logs by giving each log entry a number, which increases sequentially
- service tcp-keepalives-in – prevents the router from keeping hung management sessions open
- service tcp-keepalives-out – prevents the router from keeping hung management sessions open
- no service udp-small-servers – disables echo, chargen, discard, and daytime, which are rarely used
- no service tcp-small-servers – disables echo, chargen, and discard, which are rarely used
- service timestamps debug datetime localtime show-timezone – timestamps each log entry packet (in debug mode) with the date and time, using local time, and shows the time zone
- service password-encryption – encrypts all clear text passwords in the configuration file with type 7 encryption
- service timestamps log datetime localtime show-timezone – timestamps each logged packet (not in debug mode) with the date and time, using local time, and shows the time zone

Using ACLs to Limit Telnet and SSH Access

You can use an access list to determine who can access the VTY lines. First, create the access list, and then assign it to the VTY lines. I've added some other security related commands also.

```
Switch(config)#access-list 11 permit 10.10.9.23
```

```
Switch(config)#access-list 11 permit 10.10.10.7
```

```
Switch(config)#line vty 0 15
Switch(config-line)#access-class 11 in
Switch(config-line)#transport input ssh
Switch(config-line)#transport output telnet ssh
Switch(config-line)#exec-timout 5 0
```

Being able to configure the above is a key CCNA exam topic, so make sure that you have it down cold. Let's do a show run and take a look at the configuration; I also set SSH-only access, which you've seen how to do already. I displayed only the relevant part of the output below:

```
line vty 0 15
access-class 11 in
exec-timeout 5 0
transport input ssh
transport output telnet ssh
```

With that configuration, only host 10.10.9.23 and 10.10.10.7 are allowed to connect, and they must use a local-defined username and password and connect using SSH. If any one of those conditions fails, they will not be able to establish a session with the switch. It's very important to note that you will be applying an access class to a port, NOT an access group or list. If there is no activity on the line for five minutes, it will terminate the session.

Restrict VLAN Information

All VLANs are permitted across a trunk link by default. To further lock down security, you can specify the VLANs that are permitted across a trunk link as shown in the output below:

```
Switch(config)#int fast0/4
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan ?
WORD  VLAN IDs of the allowed VLANs when this port is in trunking mode
add   add VLANs to the current list
all   all VLANs
except all VLANs except the following
none  no VLANs
remove remove VLANs from the current list
```

```
Switch(config-if)#switchport trunk allowed vlan 7-12
```

```
Switch#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/4	on	802.1q	trunking	1
Port	Vlans allowed on trunk			
Fa0/4	7-12			

Change the Native VLAN

As discussed earlier, the native VLAN is used to carry untagged traffic on a trunk link. The default native VLAN on a trunk port is VLAN 1. This creates a degree of predictability. You can improve the security of the network by changing the native VLAN (any valid VLAN number from 2 to 4095 can be used).

The native VLAN can be verified using the show interfaces [int] switchport command as shown below:

```
Switch#show interfaces FastEthernet0/1 switchport
```

Name: Fa0/1

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Voice VLAN: none

The native VLAN can be changed using the switchport trunk native vlan interface-level command. Remember that both sides of the connection must have the same native VLAN.

```
Switch(config-if)#switchport trunk native vlan 888
```

```
Switch#show interfaces FastEthernet0/1 switchport
```

Name: Fa0/1

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 888

Voice VLAN: none

This is one of the key objectives in the CCNA syllabus, so bear it in mind and ensure that you can issue the show and configuration commands above. Native VLANs must match on either side of a trunk link and they must also be manually configured if you want them to be a number other than 1. There is no autodetect feature for native VLANs, so you can't rely on the other side of your trunk link setting itself to match the configured side.

Change the Management VLAN

As you know, a Switched Virtual Interface (SVI) can be created on the switch so you can access it via Telnet/SSH for management purposes. By default, this is VLAN 1. A good security practice is to change this VLAN number. This can be done by creating another VLAN and SVI as shown below:

```
Switch(config)#vlan 3
```

```
Switch(config-vlan)#interface vlan 3
```

```
%LINK-5-CHANGED: Interface Vlan3, changed state to up
```

```
Switch(config-if)#ip address 192.168.1.1 255.255.255.0
```

SNMP

Simple Network Management Protocol (SNMP) allows for the management of your network via one or more network management stations. SNMP uses a system of trap messages to notify the SNMP management station about events in the network. The SNMP management station can also poll the network device to determine its state. Finally, SNMP can also be used to remotely manage your network devices. SNMP is disabled by default. However, you can verify that with the following command:

```
Switch#show snmp
```

```
%SNMP agent not enabled
```

Let's configure SNMP for read-only access (use write access only if it is absolutely necessary) and with an access list:

```
Switch(config)#snmp-server community HoWtOnEtWoRk ro 15
```

```
Switch(config)#access-list 15 permit 10.10.10.15
```

```
Switch(config)#access-list 15 permit 10.10.10.16
```

In the first line, SNMP was configured using the community string HoWtOnEtWoRk; it

has read-only access (that is the ro) and only the two addresses in access list 15 are permitted. It is pretty easy to secure SNMP, you just need to map out what you want to do. SNMPv3 offers authentication and encryption for additional security. We will discuss SNMP again later in this guide.

Securing VTP

If you use VTP in your network, you will certainly want to configure a password. This will ensure that only authorized switches are installed into your VTP domain.

Switch#show vtp password

The VTP password is not configured.

Let's configure the password:

Switch(config)#vtp password IwAnTmYcCnA

Setting device VLAN database password to IwAnTmYcCnA

Switch(config)#end

Switch#show vtp password

VTP Password: IwAnTmYcCnA

You will have to configure the password on each switch in your VTP domain. If a switch does not have the password, it will not participate in VTP.

Switch Port Security

Typically, the switch will learn the MAC address of the device directly connected to a particular port and allow traffic through. The CCNA syllabus now asks, "How do you control who and how many can connect to a switch port?" This is where port security can be of assistance. Cisco switches allow you to control which devices can connect to a switch port or how many of them can connect to it (such as when a hub or another switch is connected to the port).

Enabling Port Security

Port security is disabled by default. Before configuring the port security settings on a port, you have to enable it using the switchport port-security command (the port must be set to access or trunk and not set to dynamic for port security commands to work):

Switch#config terminal

Switch(config)#interface fa0/1

Switch(config-if)#switchport port-security

As soon as port security is enabled, it will apply the default values, which is one host

permitted to connect at a time. If this rule is violated the port will shut down. If you are connected to the actual port you are configuring port security on, you should enable it after configuring other features to ensure that you don't get locked out unintentionally.

Using the switch port's port security feature, you can specify:

- Who can connect to the switch port
- How many devices can connect to the switch port
- Violation action

Who Can Connect?

If you know that only a particular host should be connecting to a switch port, then you can restrict access on that port to the MAC address of that host. This will ensure that no one can unplug the authorized host and connect another one. This is a good option for secure locations. This is done using the following command:

```
switchport port-security mac-address [address]
```

Example

If you want only the host with MAC address 0001.14ac.3298 to connect to port fa0/1 on your switch, then the commands required will be:

```
Switch#config terminal  
Switch(config)#interface fa0/1  
Switch(config-if)#switchport port-security  
Switch(config-if)#switchport port-security mac-address 0001.14ac.3298
```

These commands will not add the MAC address to the CAM table. When a host connects to this port and sends the first packet, the source address of the packet is checked against the configured MAC address. If a match is found then the address is added to the CAM table. The address is purged in the configured aging time if no traffic is seen for that host.

So, do you have to provide each host's MAC address manually? That's a huge task considering the thousands of hosts that a network can have! Well, not really. Port security provides a facility called a sticky address. The switch will use the MAC address of the first host connected to the port as a static MAC address and only that host will be able to connect to the port subsequently. The command required is `switchport port-security mac-address sticky`. The learned address is then added to the running configuration (not the startup configuration).

You cannot configure static secure or sticky secure MAC addresses on a voice VLAN.

How Many Can Connect?

Let's say that you have only one switch port left free and you need to connect five hosts to it. What can you do? Connect a hub or switch to the free port!

Connecting a switch or a hub to a port has implications. It means that the network will have more traffic. If a switch or a hub is connected by a user instead of an administrator, then there are chances that loops will be created. So, it is best that the number of hosts allowed to connect is restricted at the switch level. This can be done using the `switchport port-security maximum` command. This command configures the maximum number of MAC addresses that can source traffic through a port. Consider the following examples:

```
Switch(config-if)#switchport port-security maximum 1  
Switch(config-if)#switchport port-security mac-address sticky
```

The commands above allow only one host to connect to the port. The MAC address of the allowed host is learned automatically and the learned address is added to the running configuration.

```
Switch(config-if)#switchport port-security maximum 3  
Switch(config-if)#switchport port-security mac-address 001a.14e9.8a7d
```

The commands above allow three hosts to connect at the same time, of which one MAC address is static and the other two can vary.

```
Switch(config-if)#switchport port-security maximum 5
```

The command above allows a maximum of five hosts to connect simultaneously. Hosts can vary. If you add any port security to a voice VLAN you must enable at least two MAC addresses as a maximum.

Violation Action

What happens if a violation of security occurs on a switch port? What if five hosts are allowed on a port, but six connect to it? The switch can take one of the following three configured actions:

- Shut down the port – The port is shut down and an SNMP message is sent. The port is put in an err-disabled state and can only be recovered by the administrator.
- Protect – Keep the port up, but do not allow the offending host to send/receive data. No SNMP trap is sent.
- Restrict – Keep the port up, but do not allow the offending host to send/receive

data. Notify the administrator through SNMP and/or syslog.

The three modes can be configured using the switchport port-security violation shutdown|protect|restrict command.

Verify the port security configuration using the show port-security interface command:

```
Switch#show port-security interface FastEthernet0/1
```

Port Security: Enabled

Port status: SecureUp

Violation mode: Shutdown

Maximum MAC Addresses: 5

Total MAC Addresses: 5

Configured MAC Addresses: 3

Aging time: 20 mins

Aging type: Inactivity

SecureStatic address aging: Enabled

Security Violation count: 0

The output above shows that Fast Ethernet 0/1 has been configured with three static MAC addresses and will allow a maximum of five hosts to connect to it. If a violation is detected, then the port (by default) will go into err-disabled state (see the Error Disable Recovery section below) and shut down the port (switch interface). You can see this happening on the switch below, where an unauthorized MAC address connects to the Fast Ethernet 0/2 port.

```
Switch#
```

```
00:55:59: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/2, putting  
Fa0/2 in err-disable state
```

```
Switch#
```

```
00:55:59: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation  
occurred, caused by MAC address 1234.5678.489d on port FastEthernet0/2.
```

```
Switch#
```

```
00:56:00: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2,  
changed state to down
```

```
00:56:01: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to down
```

Another important command is show port-security. This command provides an overview of all the ports that have port security configured.

```
Switch#show port-security
```

Port	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Security Action
Fa0/1	8	7	0	Shutdown
Fa0/2	15	5	0	Restrict
Fa0/3	5	4	0	Protect

Fine-tuning Port Security Configuration

The Cisco IOS port security implementation is very flexible and it offers a number of additional ways to customize configurations.

One of the most useful command sets are the aging time and aging type. This is accomplished using the following commands:

```
Router(config-if)#switchport port-security aging time [value]
```

```
Router(config-if)#switchport port-security aging type [type]
```

The aging time value specifies the time after the learned addresses expire. This allows a new workstation to take the place of the one that has been removed. If the value is set to 0, the aging time is disabled for the specific port.

Aging can be configured to take effect at regular intervals or only during periods of inactivity using the absolute or inactivity keyword in the aging type command. For absolute aging, the addresses expire exactly after the time value configured, while for inactive aging, the addresses expire only if the port does not receive any packets from the secure source address for the specified aging time.

You can verify the aging values using the following commands:

```
Switch#show port-security interface FastEthernet0/1
```

Port Security: Enabled

Port status: SecureUp

Violation mode: Shutdown

Maximum MAC Addresses: 5

Total MAC Addresses: 5

Configured MAC Addresses: 3

Aging time: 20 mins

Aging type: Inactivity

SecureStatic address aging: Enabled

Security Violation count: 0

If you want to disable port security aging for all addresses on a port, you can use the following command:

```
Router(config-if)#no switchport port-security aging time
```

Error Disable Recovery

When certain errors occur on Cisco switches, the erring ports are put into an err-disabled state. This means that the port has been disabled because of an error. A common cause of this kind of error is the violation of a port security policy. An err-disabled interface would be marked as err-disabled in its show interface output as shown below:

```
Switch# show interface f0/10
```

FastEthernet0/10 is down, line protocol is down (**err-disabled**)

An error disabled interface can be manually activated by bouncing the interface (issuing a shutdown, and then a no shutdown command). However, waiting for an administrator to manually recover an error disabled interface might take too long. There is a way to automatically reenable an error disabled port using the errdisable recovery cause global configuration command. You need to be able to predict the root cause of the error to use this option. A sample output is shown below:

```
Switch(config)#errdisable recovery cause ?
```

all Enable timer to recover from all causes

bpduguard Enable timer to recover from bpdu-guard error disable state

dtp-flap Enable timer to recover from dtp-flap error disable state

link-flap Enable timer to recover from link-flap error disable state

pagp-flap Enable timer to recover from pagp-flap error disable state

rootguard Enable timer to recover from root-guard error disable state

udld Enable timer to recover from udld error disable state

The err-disable root causes can vary based on the switch model, but the most common causes are:

- all
- arp-inspection
- bpduguard
- dhcp-rate-limit
- link-flap

- psecure-violation (port security)
- security-violation
- storm-control
- udld

The default recovery timeout is 300 seconds on most switches. This can be changed using the errdisable recovery interval global configuration command:

```
Switch(config)#errdisable recovery interval ?
```

```
[30-86400] timer-interval(sec)
```

You can view the status of the automatic recovery of error disabled interfaces using the show errdisable recovery command. A sample output is shown below:

```
Switch#show errdisable recovery
```

ErrDisable Reason	Timer Status
arp-inspection	Disabled
bpduguard	Disabled
channel-misconfig	Disabled
dhcp-rate-limit	Disabled
dtp-flap	Disabled
gbic-invalid	Disabled
inline-power	Disabled
l2ptguard	Disabled
link-flap	Disabled
mac-limit	Disabled
link-monitor-failure	Disabled
loopback	Disabled
oam-remote-failure	Disabled
pagp-flap	Disabled
port-mode-failure	Disabled
psecure-violation	Enabled
security-violation	Disabled

sfp-config-mismatch Disabled

storm-control Disabled

udld Disabled

unicast-flood Disabled

vmmps Disabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

Interface	Errdisable reason	Time left(sec)
-----------	-------------------	----------------

Fa0/0	psecure-violation	193
-------	-------------------	-----

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 8 exam.

Chapter 8 Labs

Lab 1: Basic Router Security

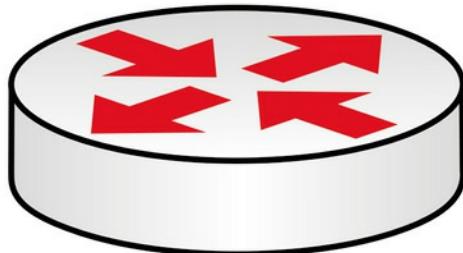


FIG 8.4 – Basic router security

Lab Exercise

Your task is to configure basic access and security features on a router.

Purpose

Learn some basic steps to take to lock down your router.

Lab Objectives

1. Protect enable mode with a password.
2. Enable service password encryption.
3. Protect the Telnet and console lines.
4. Add a banner.
5. Turn off CDP.
6. Configure router logging.

Lab Walk-through

1. Protect enable mode with an enable secret password. Test this by logging out of privileged mode and then logging back in:

```
Router#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)#enable secret cisco
```

```
Router(config)#exit
```

```
Router#
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
Router#exi
```

```
Router con0 is now available
```

Press RETURN to get started.

```
Router>en
```

Password:

Router#

2. Set an enable password and then add service password encryption. This is rarely done on live routers because it is not secure.

```
Router(config)#no enable secret
```

```
Router(config)#enable password cisco
```

```
Router(config)#service password-encryption
```

```
Router(config)#exit
```

```
Router#
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
Router#show run
```

```
Building configuration...
```

```
Current configuration: 480 bytes
```

```
!
```

```
version 15.1
```

```
no service timestamps log datetime msec
```

```
no service timestamps debug datetime msec
```

```
service password-encryption
```

```
!
```

```
hostname Router
```

```
!
```

```
enable password 7 0822455D0A16
```

3. Protect the Telnet lines. Set a local username and password and have users enter this when connecting to the router. You may have more or less VTY lines depending on your platform.

```
Router(config)#line vty 0 ?
```

```
[1-15] Last Line number
```

```
[cr]
```

```
Router(config)#line vty 0 15
```

```
Router(config-line)#login local
```

```
Router(config-line)#exit
```

```
Router(config)#username howtonetwork password cisco
```

You have tested Telnet before, but feel free to add a PC and telnet to the router so that you are prompted for a username and password.

4. Protect the console port with a password. Set one directly on the console port.

```
Router(config)#line console 0  
Router(config-line)#password cisco
```

You can test this by unplugging and plugging your console lead back into the router. You can also protect the auxiliary port on your router if you have one:

```
Router(config)#line aux 0  
Router(config-line)#password cisco
```

5. Protect the Telnet lines by permitting only SSH traffic in. You can also permit only SSH traffic outbound. You will need a security image for this command to work.

```
Router(config)#line vty 0 15  
Router(config-line)#transport input ssh  
Router(config-line)#transport output ssh
```

6. Add a banner message of the day (MOTD). Set the character that tells the router you have finished with your message as X (the delimiting character).

```
Router(config)#banner motd X  
Enter TEXT message. End with the character “X”.  
Do not use this router without authorization. X
```

```
Router(config)#exit  
Router#exit
```

Router con0 is now available
Press RETURN to get started.

Do not use this router without authorization.
Router>

7. Turn off CDP on the entire router. You can disable it on an interface only with the no cdp enable interface command.

```
Router(config)#no cdp run
```

You can test whether this is working by connecting a switch or router to your router before you turn off CDP and issuing the show cdp neighbor (detail) command.

8. Set the router to send logging messages to a host in the network:
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.

```

Router(config)#logging ?
  A.B.C.D IP address of the logging host
  buffered Set buffered logging parameters
  console Set console logging parameters
  host    Set syslog server IP address and parameters
  on      Enable logging to all enabled destinations
  trap    Set syslog server logging level
  userinfo Enable logging of user info on privileged mode enabling
Router(config)#logging 10.1.1.1

```

Lab 2: Switch Security

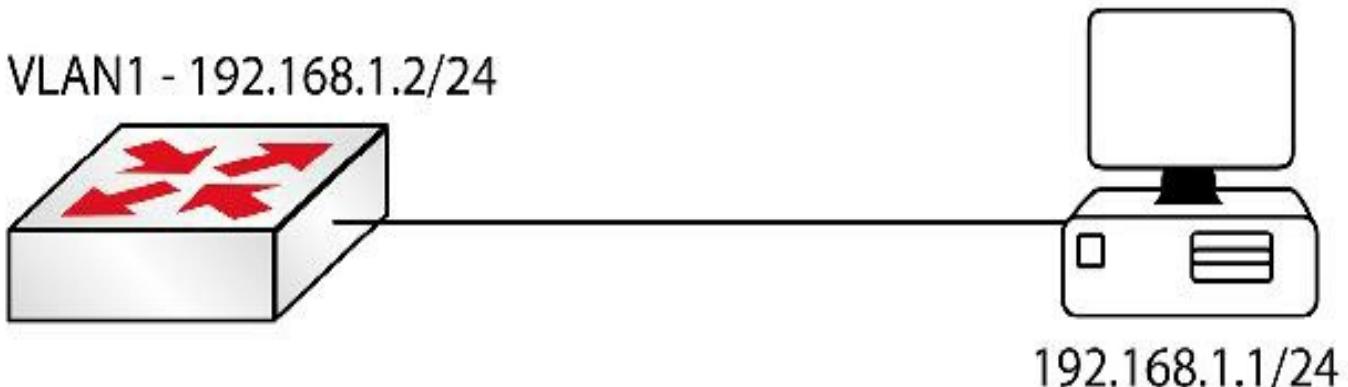


FIG 8.5 – Switch security

Lab Exercise

Your task is to configure basic access and security features on a switch. Please note that your switch will need to have a security image that permits basic security settings.

Purpose

Learn how to apply basic security settings on a Cisco switch.

Lab Objectives

1. Set up Telnet access.
2. Assign a management IP address to the switch.
3. Configure SSH.

Lab Walk-through

1. Connect a PC or laptop to your switch. In addition, set up a console connection for your configuration. The port to which you connect your PC will be the one you configure security settings on in this lab. I have chosen Fast Ethernet 0/1 on my switch.
2. Log in to the vty lines and set up Telnet access referring to a local username and

password.

Switch#conf t

Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#line vty 0 ?

[1-15] Last Line number

[cr]

Switch(config)#line vty 0 15

Switch(config-line)#login local

Switch(config-line)#exit

Switch(config)#username days password cisco

Switch(config)#

3. Add an IP address to VLAN 1 on the switch (all ports are in VLAN 1 automatically). Additionally, add the IP address 192.168.1.1 to your PC's Fast Ethernet interface.

Switch(config)#interface vlan 1

Switch(config-if)#ip address 192.168.1.2 255.255.255.0

Switch(config-if)#no shut

%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan2, changed state to up

Switch(config-if)#^Z

Switch#ping 192.168.1.1 **i Tests connection from switch to PC**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 31/31/32 ms

4. Test Telnet by telnetting from your PC to your switch.

Command Prompt

```
PC>telnet 192.168.1.2  
Trying 192.168.1.2 ...open
```

User Access Verification

```
Username:      days
```

```
Password:
```

```
Switch>
```

5. Your IT manager changes his mind and wants only SSH access, so change this on your vty lines. Only certain models and IOS versions will support the SSH commands.

```
Switch(config)#line vty 0 15
```

```
Switch(config-line)#transport input ssh
```

6. Now telnet from your PC to the switch. Because only SSH is permitted, the connection should fail.

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>telnet 192.168.1.2
Trying 192.168.1.2 ...open

[Connection to 192.168.1.2 closed by foreign host]
PC>
```

Lab 3: Switch Port Security

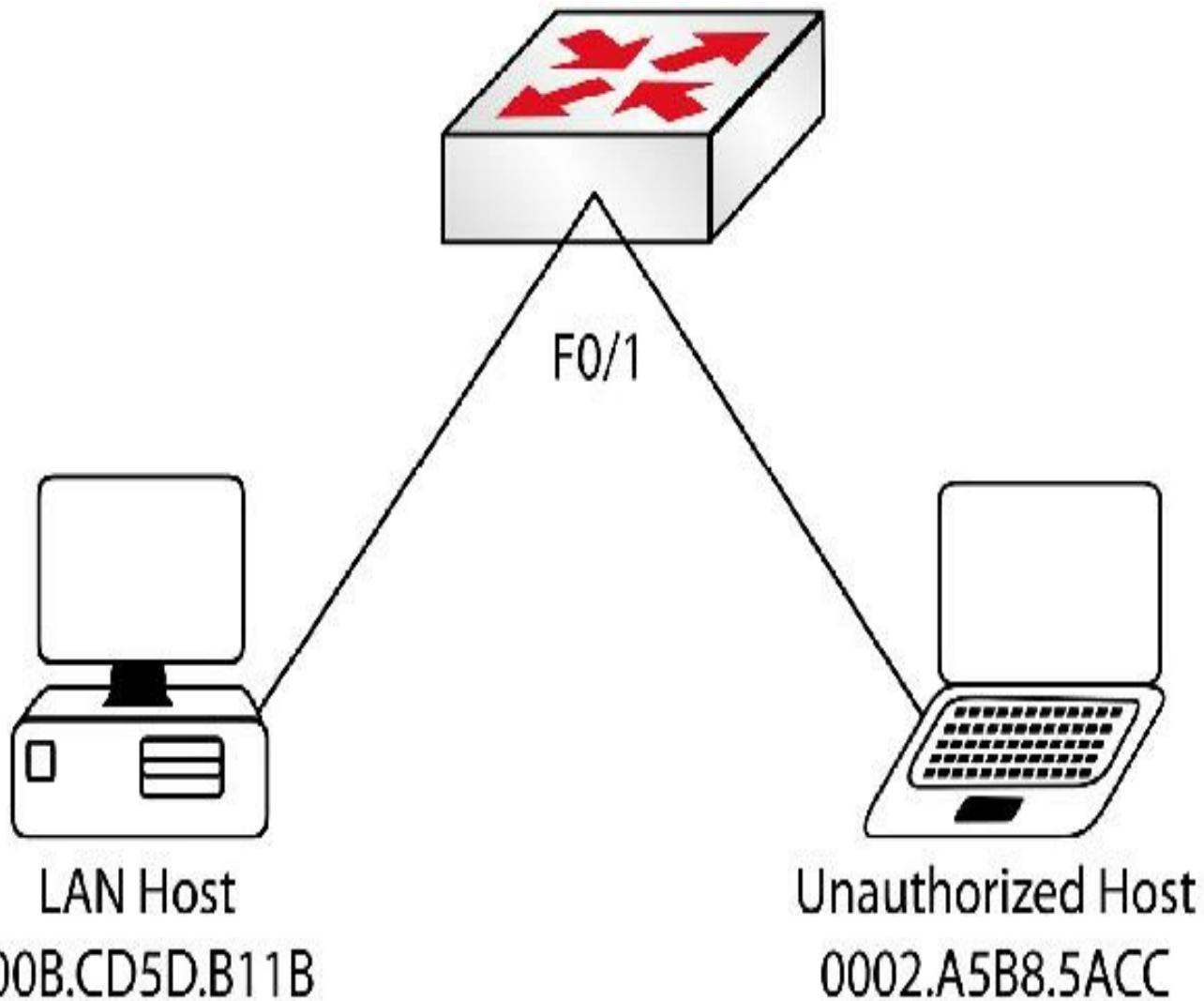


FIG 8.6 – Port security

Lab Exercise

A PC is connected directly to the Fast Ethernet 0/1 interface on your switch. Apply port security permitting only the LAN Host to connect through the port, as well as a static MAC address. If a violation is detected the port should shut down. Plug in the Unauthorized Host to check whether the configuration is working correctly.

Purpose

Security is very crucial for any network. Layer 2 security will ensure that no one can compromise the security from inside the network. You will be expected to know how to configure port security for the CCNA exam.

Lab Objectives

1. Enable port security on fa0/1 on the switch.
2. Configure a static MAC address for port security on fa0/1.

3. Configure violation action on the switch port.
4. Test the port by connecting a different device.

Lab Walk-through

1. Enable port security on fa0/1 on the switch:

```
Switch#configure terminal
Switch(config)#interface fa0/1
Switch(config-if)#switchport port-security
```

(NOTE: The port you apply this command to must be an access port and not left as dynamic. If the interface is connected to another switch, the switch may convert to a trunk. The show interfaces trunk command will display your trunk interfaces:

```
Switch#show interfaces trunk
Port      Mode      Encapsulation  Status        Native vlan
Fa0/1    desirable   802.1q        trunking     1
```

If you attempt to apply the command to a dynamic port you will see the output below:

```
Switch(config-if)#switchport port-security
Command rejected: Not eligible for secure port.
```

To set an interface to access mode, please apply the following command:

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fast0/1
Switch(config-if)#switchport mode access
```

2. Configure a static MAC address for port security on fa0/1. This will be the MAC address you want to permit to pass traffic through the switch port. You will need to alter the MAC address for your own equipment.

```
Switch(config-if)#switchport port-security mac-address 000b.cd5d.b11b
```

(NOTE: You have several other options here. You can set the maximum number of MAC addresses permitted through the switch port, hard set the MAC address, and set the violation action of your settings (see step 3):

```
Switch(config-if)#switchport port-security ?
mac-address Secure mac address
maximum      Max secure addrs
```

violation Security Violation Mode

[cr]

3. Configure violation action on the switch port. You may choose the restrict (inform the network administrator of the violation), shutdown, or protect (only permit frames from the permitted devices) commands:

```
Switch(config-if)#switchport port-security violation ?  
protect Security violation protect mode  
restrict Security violation restrict mode  
shutdown Security violation shutdown mode
```

Set the shutdown command so the port will shut down when it detects a restriction:

```
Switch(config-if)#switchport port-security violation shutdown
```

4. Now plug the authorized host (LAN Host) into your switch port Fast Ethernet 0/1. After about 30 seconds the switch port light will go from amber to green and show as up. You can then check the port security status with the show port-security command:

```
Switch#  
00:40:32: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up  
00:40:33: %LINEPROTO-5-UPDOWN: Line protocol on Interface  
FastEthernet0/1, changed state to up  
Switch#show port-security int fast0/1  
Port Security: Enabled  
Port Status: Secure-up  
Violation Mode: Shutdown  
Aging Time: 0 mins  
Aging Type: Absolute  
SecureStatic Address Aging: Disabled  
Maximum MAC Addresses: 1  
Total MAC Addresses: 1  
Configured MAC Addresses: 1  
Sticky MAC Addresses: 0  
Last Source Address: 000b.cd5d.b11b  
Security Violation Count: 0
```

5. Unplug the LAN Host and plug in the Unauthorized Host. Once the frames reach the switch port it will close down.

```
Switch#
```

```
00:41:04: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1,  
putting Fa0/1 in err-disable state  
00:41:04: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation  
occurred, caused by MAC address 0002.a5b8.5acc on port FastEthernet0/1.  
00:41:05: %LINEPROTO-5-UPDOWN: Line protocol on Interface  
FastEthernet0/1, changed state to down
```

6. The interface has been shut down. You can now check the port security status:

```
Switch#show port-security int fast0/1  
Port Security: Enabled  
Port Status: Secure-shutdown  
Violation Mode: Shutdown  
Aging Time: 0 mins  
Aging Type: Absolute  
SecureStatic Address Aging: Disabled  
Maximum MAC Addresses: 1  
Total MAC Addresses: 1  
Configured MAC Addresses: 1  
Sticky MAC Addresses: 0  
Last Source Address: 0002.a5b8.5acc  
Security Violation Count: 1
```

7. To reenable the interface, plug the authorized LAN Host back in and then add the shut and no shut commands to the Fast Ethernet interface.

One last point: please take some time to use the other port security commands. You can, for example, permit a maximum number of MAC addresses through the port or choose a specific range of allowed addresses.

Running Configuration

```
Switch1#sh run  
Building configuration...
```

```
[output truncated]
```

```
hostname Switch  
!  
interface FastEthernet0/1  
switchport mode access  
switchport port-security
```

switchport port-security violation shutdown
switchport port-security mac-address 000b.cd5d.b11b
!

Chapter 9 — Network Troubleshooting

What You Will Learn in This Chapter

Your Troubleshooting Plan

Layer 1 Troubleshooting

VLAN Troubleshooting

Trunking Troubleshooting

VTP Troubleshooting

IP Addressing Troubleshooting

Access List Troubleshooting

Syllabus Topics Covered

7.0 Troubleshooting

7.1 Troubleshoot and correct common problems associated with IP addressing and host configurations

7.2 Troubleshoot and resolve VLAN problems

 7.2.a Identify that VLANs are configured

 7.2.b Verify that port membership is correct

 7.2.c Verify that correct IP address is configured

7.3 Troubleshoot and resolve trunking problems on Cisco switches

 7.3.a Verify correct trunk states

 7.3.b Verify that correct encapsulation is configured

 7.3.c Verify that correct VLANs are allowed

7.4 Troubleshoot and resolve ACL issues

 7.4.a Verify statistics

 7.4.b Verify permitted networks

 7.4.c Verify direction

 7.4.c (i) Verify interface

7.5 Troubleshoot and resolve layer 1 problems

 7.5.a Framing

 7.5.b CRC (cyclic redundancy check)

 7.5.c Runts

 7.5.d Giants

7.5.e Dropped packets

7.5.f Late collisions

7.5.g Input/Output errors

Troubleshooting the network is not some mysterious skill that can only be carried out by the select few. With a methodical plan and using a step-by-step process, you should quickly be able to troubleshoot, diagnose, and resolve most of the problems you encounter in a computer network.

For the CCNA exam you will be expected to troubleshoot some problems in a simulated network. You must be able to look at network topologies and identify the cause of the problem, and then quickly resolve it. This chapter is primarily focused on showing you the kinds of problems you could come across in the CCNA exam. It is the troubleshooting system that is important here, not the specific problems we will look at.

Before you delve into troubleshooting, it's important that you have a strong grasp of the technology from reading the theory pages several times over and, of course, completing the labs many times over until you no longer need to look up the commands.

When configuring a lab, you will find that the use of IOS show commands will ensure that everything is working as desired. When you are troubleshooting, it will almost always be in a network that has already been configured, or at least partly configured, so you should start off by using the show commands.

Show commands can give you a general overview of the status of a device or drill down to a very low level. You can also use debug commands to track events per service or protocol. Please use debug commands with great care on a live network because they can consume massive amounts of device resources and quickly cause it to crash, which can leave you open to disciplinary or legal issues.

There are many books written about troubleshooting in a Cisco networking environment, and as you progress through your career you should plan to read several of them. Before starting our review of the kinds of problems you will see on the CCNA exam, let's first discuss some general background on troubleshooting.

In the exam you could be tested on your troubleshooting skills in several ways:

- Fix a partly configured or broken network using the simulator
- Log in to devices and answer questions about which parts are broken
- Answer theory-only questions about probable causes of network problems
- Answer questions about the best show or debug command to use in a situation

In fact, you may well have to employ your troubleshooting skills during the configuration lab if it isn't performing as required. Bear in mind that you will be

working with a simulator and not on a live device, so the debug and show commands available will be limited.

Your Troubleshooting Plan

A structured approach is generally the best method to use when troubleshooting networking issues. In a larger organization this includes how problems are reported to the networking team, the troubleshooting procedures followed by the team, and the processes used to include other departments for problems that span organizational units. In smaller organizations the formal structure may not be as well defined, as much of the approach will reside in a single person, namely you.

The initial steps in troubleshooting involve isolating the source of the problem. During this step (if it were a live network) you will want to discuss the symptoms seen by the people reporting the problem. Users in the network may mislead you with wrong information due to their lack of knowledge on the reported symptoms, which means the symptoms may not be the root cause of the problem. Sometimes users think they know what the problem is and present facts that support their conclusion. The best advice during this step is to determine how widespread the problem is.

Attempt to isolate the issue to a single location, building, floor, and, finally, specific network segment or node. One method often used is termed “divide and conquer,” whereby you parse the network into progressively smaller sections until you isolate the location of the problem.

It’s important to gather information from the users but not be misled by them. Some could call in complaining that their e-mail isn’t working, which could be part of a far larger problem or simply due to their PCs running slowly. I’ve also noticed that when working on large networks managed by multiple teams that everyone is quick to blame the network for the problem.

Take all information provided in and then apply your plan. If you know that all services are working on a device but only one user is having an issue, then there is little point in checking to see whether the power supply has failed. Because only one user is affected, it would be unlikely that the problem was caused by a configuration issue or network outage. As such, you would probably check the port the user is plugged into on the network switch.

Once you have isolated the problem in the network topology, then continue to apply structured critical thinking to isolate the root cause of the problem. You may use the OSI model as a reference to help you organize the troubleshooting commands and tools that you have at your disposal. In order to solve many network problems, you will need to

be familiar with Cisco troubleshooting tools as well as those provided by other vendors for use on their products, such as Microsoft for Windows operating systems and the Open Source Foundation for Linux operating systems.

OSI layer 1 (physical) problems include:

- Faulty or broken cables
- Broken or faulty pins/connectors
- No power
- No cable connected or wrong interface
- Failing or damaged interface
- Incorrect cable for the interface

Layer 2 (data link) problems include:

- Incorrect configuration on the interface
- Clock rate missing or incorrect
- Incorrect layer 2 protocol settings
- Faulty network card
- Interface shut down

Layer 3 (network) problems include:

- Misconfigured routing protocol
- Incorrect IP/network addressing
- Incorrect subnet masking

Each of the layers above requires a different set of commands or debugs, although some show commands will display information for multiple layers, such as show interface serial 0/0, for example. The show ip ospf 10 command will not tell you whether an interface has gone down.

A great book to help you with real-life troubleshooting is:

Internetworking Troubleshooting Handbook (Cisco Press).

ISBN: 1587050056

If I had to carry only one book in my briefcase when out in the field, it would be:

Cisco Field Manual: Router Configuration (Cisco Press).

ISBN: 1587050242

These books are somewhat out of date, and the Cisco IOS and devices have come a long way since their publication, but the principles are still sound. With a methodical approach, you should be able to quickly and effectively troubleshoot and resolve any

problem you encounter in the exam and on a live network in the real world.

Network Debugging

We have actually already covered this in the various theory lessons and labs, but it's time to directly address debugging Cisco devices.

Built into the Cisco IOS are a large number of show commands, as you already know, but you also have the ability to view status and error messages in real time, which Cisco refers to as debugging. Some of these messages are printed on the console screen automatically, such as interfaces coming up or going down, duplex errors, or VLAN mismatches. Others require you to issue a specific debug command for that protocol or technology.

If you need to debug a live network (commonly referred to as a production network), then you must exercise extreme caution and probably seek expert advice first. Some debug commands will return a huge amount of information very quickly, meaning that you won't see a command prompt on the screen to turn them off and you will quickly overload your CPU, causing the router to crash. I've seen too many careers end prematurely due to this mistake.

You may not actually have the ability to issue debug commands in the CCNA exam because you are working on router simulators and there is no real traffic passing over the link. I can't guarantee this, of course, but you could easily be asked theoretical debugging questions in the exam.

If you type `debug ?` at a router prompt, you will see several pages of available debugs:

R1#`debug ?`

IUA	ISDN adaptation Layer options
aaa	AAA Authentication, Authorization and Accounting
aal2_xgcpspi	AAL2_XGCP Service Provider Interface
access-expression	Boolean access expression
acircuit	Attachment Circuit information
adjacency	adjacency
arp	IP ARP and HP Probe transactions
asnl	Application Subscribe Notify Layer
aspp	ASPP information

async Async interface information

bgp BGP information

--More—

Usually, I advise using the ?, which helps you see what options are available:

R1#debug ip ?

access-list IP access-list operations

address IP address activity

admission Network admission control debug

auth-proxy Authentication proxy debug

bgp BGP information

cache IP cache operations

dhcp Dynamic Host Configuration Protocol

director Director agent information

dns DNS

eigrp IP-EIGRP information

--More—

And many of these commands have options available or options within options:

R1#debug ip eigrp ?

<1-65535> Autonomous System

neighbor IP-EIGRP neighbor debugging

notifications IP-EIGRP event notifications

summary IP-EIGRP summary route processing

vrf Select a VPN Routing/Forwarding instance

<cr>

For troubleshooting, you will want to use debug commands after you have checked all of your configurations and used your available show commands but still can't see the issue. If you are troubleshooting a WAN connection, you would use the debug outputs to prove to your ISP that the issue isn't on your side of the connection because they will inevitably immediately blame you (as I'm sure you are already aware of if you work on a network team).

We will cover layer 2 debug commands in the relevant WAN sections, but they include:

- debug frame-relay packet
- debug frame-relay events
- debug serial packet
- debug serial interface
- debug ppp packet
- debug ppp authentication
- debug ppp negotiation

You can issue a debug ip packet command but I advise that you never do this on a live network due to the huge amount of output it produces. If you have a small home network you can try it out. It's worth noting that you can debug an access list, but Cisco has made the command somewhat difficult to find:

```
R1#debug ip packet ?
```

<1-199> Access list

<1300-2699> Access list (expanded range)

detail Print more debugging detail

Here is some output from the command above if you really want to see it. Included are s=, which is the source IP address, and d= for the destination address:

```
R1#debug ip packet
```

IP packet debugging is on

```
*Mar 1 03:42:13.359: IP: s=192.168.1.1 (local), d=224.0.0.5 (FastEthernet0/0), len 80, sending broad/multicast
```

You can also drill into more detail if you append the tag detail to the end of the debug:

```
R1#debug ip packet detail
```

IP packet debugging is on (detailed)

```
*Mar 1 03:48:07.059: IP: s=192.168.1.2 (FastEthernet0/0), d=224.0.0.5, len 80, rcvd 0, proto=89
```

Protocol number 89 is used by OSPF. You may recognize 224.0.0.5 as the All OSPF Routers address. Each routing protocol offers its own set of debugs that we will address as they arise.

If you are connected to a remote device via a Telnet connection, then the debug output won't show automatically (because it is printed on the console session by default). You will need to issue the terminal monitor command. You can send the logging messages to

an internal buffer with the logging buffered command.

You may need to add millisecond-level timestamps onto your debugs. Cisco will usually request this for chatty protocols or rapid message exchange such as Integrated Services Digital Network (ISDN) or Point-to-Point Protocol (PPP). You would add the service timestamps debug datetime msec command to do this, and if you issue a show run command you may see that it's on by default depending on your IOS release:

version 15.1

```
service timestamps debug datetime msec
```

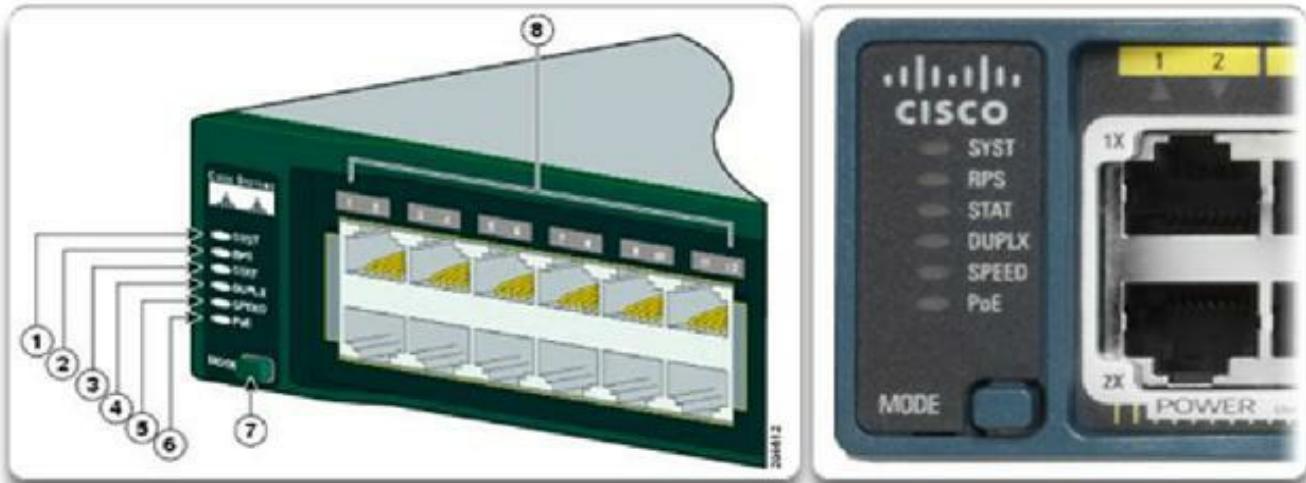
Most importantly, turn off all debugs with the undebug all command, or un all for short. I've managed to type this when the terminal screen was flooded with debug output and I thought the router was about to crash. The command took effect after several agonizing seconds.

Layer 1 Troubleshooting

Troubleshooting network devices at layer 1 might seem like a simple matter but it can become very complicated because there are a lot of parameters to check at this layer.

Although you may laugh, many layer 1 issues can be easily diagnosed, tested, and resolved by swapping out a cable. All Cisco devices have interface LEDs, which indicate various states (read the user's manual for a complete list). No light or a red light usually indicates a faulty cable, so swap it out to test this theory. An amber (orange) light can mean that no keepalives are seen on the line. It can also mean that an incorrect cable has been installed, so ensure that the cable is straight-through (if going from a switch to a host).

Figure 9.1 below shows the common LEDs found on the Cisco 2960 model. The documentation runs into many pages so check the online manual for this model.



Catalyst 2960 Switch LEDs

1	The system LED	5	The port speed LED
2	The RPS LED (if RPS is supported on the switch)	6	The PoE status LED (if PoE is supported on the switch)
3	The port status LED (This is the default mode.)	7	The Mode button
4	The port duplex mode LED	8	The port LEDs

FIG 9.1 – Cisco 2960 LEDs (Image © Cisco Systems)

On Cisco routers and switches you can focus on two commands:

- show controllers – displays hardware component statistics
- show interfaces – displays interface-level statistics

A show controllers output might look similar to the following:

```
R1#show controllers
```

```
Interface FastEthernet0/0
```

```
Hardware is GT96K FE ADDR: 6627DD48, FASTSEND: 6072E6E4, MCI_INDEX: 0
```

```
Bytes_recv 0 Bytes_sent 0 Bytes_sent 0 Frames_sent 0
```

```
total_bytes_RX 0 Total_frames_RX 0 Bcast_frames_recv 0
```

```
Mcast_frames_RX 0 CRC_err 0 Ovr_sized_frames 0
```

```
Fragments 0 Jabber 0 Collision 0
```

```
Late_collision 0 64B frame 0; 65_127B_frames 0
```

```
128_255B_frames 0 256_511B_frames 0 512_1023B_frames 0
```

```
1023_maxB_frames 0 Rx_error 0 Dropped_frames 0
```

```
Mcast_frames_tx 0 Bcast_frames_tx 0 Sml_frame_recv 0
```

[output omitted]

The command above is usually used on a Serial interface, however.

The output above presents hardware-level statistics for all router/switch interfaces and modules. Some of the most interesting parameters are as follows (syllabus items in bold):

- Bytes_recv – total number of bytes received on the interface
- Bytes_sent – total number of bytes sent on the interface
- Frames_sent – total number of frames sent on the interface
- Bcast_frames_recv – total number of broadcast frames received on the interface
- Mcast_frames_RX – total number of multicast frames received on the interface
- **CRC_err** – total number of CRC errors; incremented when the checksum calculated by the sender does not match the checksum calculated by the network device and usually indicates transmission problems that altered the packet (collisions or physical issues)
- Ovr_sized_frames – total number of oversized frames (frames larger than 1514 bytes)
- Fragments – total number of fragments; packets are fragmented when they cross interfaces with a smaller MTU (maximum transmission unit) than the packet's size
- Collision – number of collisions
- **Late_collision** – number of late collisions (collisions detected late in the transmission process); usually indicates a duplex mismatch so you should check to ensure that the duplex and autonegotiation settings are consistent at both ends of the link
- Dropped_frames – number of frames dropped (among the multiple reasons a frame might get dropped, the most common ones include collisions or interface queue full)

Another tool you can use when troubleshooting layer 1 issues is the show interfaces X command:

```
R1#show interfaces FastEthernet0/1
```

FastEthernet0/1 is up, line protocol is up (connected)

Hardware is Gt96k FE, address is c201.0f00.0001 (bia c201.0f00.0001)

MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

Keepalive set (10 sec)

Half-duplex, 10Mb/s, 100BaseTX/FX

ARP type: ARPA, ARP Timeout 04:00:00

Last input never, output never, output hang never

Last clearing of show interface counters never

Input queue: 0/75/0/0 (size/max/drops/flushes);Total output drops: 0

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

0 packets input, 0 bytes

Received 0 broadcasts, 0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

0 watchdog

0 input packets with dribble condition detected

0 packets output, 0 bytes, 0 underruns

0 output errors, 0 collisions, 0 interface resets

0 unknown protocol drops

0 babbles, 0 late collisions, 0 deferred

0 lost carrier, 0 no carrier

0 output buffer failures, 0 output buffers swapped out

Some of the most interesting parameters you can inspect in the output above include the following (syllabus items in bold):

- Interface status – up, line protocol is up means that the interface is physically up and connected to another device at the other end
- MAC address
- MTU (maximum transmission unit) – this is 1500 for Ethernet by default
- Interface bandwidth

- Duplex parameters
- Last input, output – the time since the last packet was successfully received or transmitted by the interface
- Input queue – the number of packets in the input queue (if this number starts to get close to the maximum value, it means that the network device has issues processing the packets compared to the rate it receives them and it might soon start to discard packets)
- **Total output drops** – the number of packets dropped because the output queue is full (a common cause might be traffic from a high bandwidth link being transmitted to a lower bandwidth link)
- Output queue – the status of the output queue (if the number of packets in this queue gets close to the maximum value, packets might get dropped)
- 5 minute input rate and 5 minute output rate – the average input and output rate of the interface in the last five minutes
- **Runts** – the number of frames that are smaller than the minimum IEEE 802.3 frame size (64 bytes for Ethernet), with a bad CRC; usually caused by a duplex mismatch or physical problem (bad cable or port)
- **Giants** – the number of frames that are larger than the maximum IEEE 802.3 frame size (1518 for non-jumbo Ethernet); usually caused by a faulty interface module
- Throttles – the number of times the traffic received on the port is disabled; usually caused by a buffer or processor overload
- Input errors – includes runts, giants, no buffer, CRC, frame, overrun, and ignored counts
- **CRC** – this is incremented when the checksum calculated by the sender does not match the checksum calculated by the network device; usually indicates transmission problems that altered the packet (collisions or physical issues)
- **Frame** – the number of packets received incorrectly; usually caused by collisions or physical cabling problems
- Overrun – the number of times the receiver hardware was unable to forward the received data to a hardware buffer; caused by traffic rates that exceed the ability of the receiver hardware to handle data
- Ignored – the number of received packets ignored because the interface hardware ran low on internal buffers; usually caused by broadcast storms
- Underruns – the number of times the transmitter has ran faster than the switch can handle; this can happen in a high-throughput situation
- Output errors – the sum of all errors that prevented correct data transmission out of the interface

- Collisions – the number of times a collision occurred before the interface transmitted a frame successfully; usually seen on half-duplex interfaces
- **Late collisions** – the number of times a collision is detected late in the transmission process; can be caused by a duplex mismatch or physical cable/port issues
- Deferred – the number of frames that have been transmitted after they were put in hold because the media was busy; usually seen on half-duplex links, where sending and receiving packets cannot happen at once because of the shared media

Most commonly you will be troubleshooting input errors, which will be accompanied by CRC errors. This is a sign that the other side of the connection is set to half-duplex.

Switch1#show int f0/11

FastEthernet0/11 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 0011.9247.db0b (bia 0011.9247.db0b)

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

Keepalive set (10 sec)

Full-duplex, 100Mb/s

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

1117 packets input, 91352 bytes, 0 no buffer

Received 108 broadcasts (0 multicast)

0 runts, 0 giants, 0 throttles

665 input errors, 660 CRC, 0 frame, 0 overrun, 0 ignored

[output truncated]

However, you will also see the error message below printed on your console screen:

00:06:14: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on
FastEthernet0/11 (not full duplex), with Switch1 FastEthernet0/11 (full duplex).

00:06:14: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/11 (not full duplex), with Switch1 FastEthernet0/11 (full duplex).

The same command on the switch connected to f0/11 reveals that the duplex setting is in fact incorrect.

Switch2#show int f0/11

FastEthernet0/11 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 000d.292e.118b (bia 000d.292e.118b)

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

Keepalive set (10 sec)

Half-duplex, 100Mb/s

Switch2(config-if)#int f0/11

Switch2(config-if)#duplex full

You can also use the show ip interface brief command to quickly establish whether any interface is up or down and whether you need to move on to troubleshooting the interface or the encapsulation, duplex settings, etc.

Troubleshooting VLAN Issues

When troubleshooting end-to-end intraVLAN connectivity, you have to make sure that you verify a number of configuration aspects, including:

- IP addressing issues within the VLAN
- VLANs are created
- VLANs are correctly associated to switch ports

Usually, you want each VLAN to be assigned a unique IP subnet to avoid interVLAN communication issues. This means that two hosts that are part of the same VLAN will need to be assigned IP addresses from the same range. Therefore, if two hosts in the same VLAN cannot communicate, the first thing to check is whether they have IP addresses from the same subnet. Don't forget to also check whether they have the same subnet mask.

If there is still no VLAN communication after the IP addressing verification, the next thing to check is whether the specific VLAN is correctly defined on all switches between the two hosts. This is accomplished using the show vlan command:

```
Switch#show vlan
```

VLAN Name	Status	Ports
-----------	--------	-------

1 default	active	Fa1/1
-----------	--------	-------

[output truncated]

If the VLAN is not configured, you can create it using the `vlan x` command on the switch:

```
Switch#configure terminal
```

```
Switch(config)#vlan 5
```

```
Switch(config-vlan)#name CCNA
```

```
Switch(config-vlan)#exit
```

```
Switch#show vlan
```

VLAN Name	Status	Ports
-----------	--------	-------

1 default	active	Fa1/1
-----------	--------	-------

5 CCNA	active	Fa1/1
---------------	---------------	--------------

[output truncated]

After properly creating the VLAN on all switches between the two hosts, you need to check whether the correct ports are associated with the respective VLAN. At the very least, the following ports have to be associated:

- The port connecting to the sender host
- The port connecting to the receiver host
- Any interswitch port connecting multiple switches between the two hosts

To check port-to-VLAN mapping, you can use the `show vlan` command. If you also have trunk connections between switches, you can check whether the specific VLAN is allowed on the trunk using the following command:

```
Switch#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
------	------	---------------	--------	-------------

Fa1/11	on	802.1q	trunking	1
--------	----	--------	----------	---

Fa1/12	on	802.1q	trunking	1
--------	----	--------	----------	---

Port	Vlans allowed on trunk
------	------------------------

Fa1/11	1,3,4,
---------------	---------------

Fa1/12	1,3,4,
---------------	---------------

Port Vlans allowed and active in management domain

Fa1/11 1,3,4,

Fa1/12 1,3,4,

Port Vlans in spanning tree forwarding state and not pruned

Fa1/11 1,3,4,

Fa1/12 1,3,4,

If the specific VLAN is not allowed on the trunk, you can add it using the following command:

```
Switch(config)#int fast1/11
```

```
Switch(config-if)#switchport trunk allowed vlan add 5
```

```
Switch(config-if)#int fast1/12
```

```
Switch(config-if)#switchport trunk allowed vlan add 5
```

Use caution in real-world scenarios: if you forget the add keyword, all previously allowed VLANs on the particular trunk will be removed from the trunk.



```
Switch#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
------	------	---------------	--------	-------------

Fa1/11	on	802.1q	trunking	1
--------	----	--------	----------	---

Fa1/12	on	802.1q	trunking	1
--------	----	--------	----------	---

Port Vlans allowed on trunk

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

Port Vlans allowed and active in management domain

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

Port Vlans in spanning tree forwarding state and not pruned

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

Now that all the trunk ports are carrying the relevant VLAN, you must also add it to the access ports connecting to the hosts:

```
Switch(config)#interface FastEthernet1/2
```

```
Switch(config-if)#switchport access vlan 5
```

```
Switch#show vlan
```

VLAN Name	Status	Ports
<hr/>		
1 default	active	Fa1/1
5 CCNA	active	Fa1/1, Fa1/2

At this point you should have end-to-end VLAN connectivity between two hosts in the same VLAN.

Another useful command is show interface X switchport, which will tell you whether the interface is set to access/trunk/dynamic, which VLAN is native on it, encapsulation type, and other useful information:

```
Switch#show int f0/10 switchport
```

Name: Fa0/10

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: static access

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

Remember that you can apply an IP address to a VLAN. This will also create an SVI for the switch. You can easily see this with the show ip interface brief command. The SVI must be opened with the no shut command in order for it to operate.



You do need to add the no shut command for older versions of IOS.

Switch#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	unassigned	YES	manual	up	up
GigabitEthernet1/2	unassigned	YES	manual	down	down
Vlan1	192.168.1.2	YES	manual	up	up

[output truncated]

Switch#show int vlan 1

Vlan1 is up, line protocol is up

Hardware is CPU Interface, address is 0002.4a4b.27a7 (bia 0002.4a4b.27a7)

Internet address is 192.168.1.2/24

Troubleshooting Trunks

Apart from a hardware or cable fault, most of the trunking problems originate from configuration errors. Common problems are discussed below:

1. **Trunk will not come up** – First, check whether the interface status is up/up using the show ip interface brief or show interface [interface] command. The second thing to check is the mode configured on the switch port. This can be done using the show interface [interface] switchport command. This command will show an output similar to the one given below:

SwitchA#show interface fa1/1 switchport

Name: Fa1/1

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: Disabled

Access Mode VLAN: 0 ((Inactive))

Important points to note are the Administrative Mode, Administrative Trunking Encapsulation, and Negotiation of Trunking lines. These will tell you the mode, the trunking protocol on the port, and DTP (Dynamic Trunking Protocol) status, respectively. Remember that ports set on Auto/Auto mode will not trunk (this was covered previously). A trunking protocol mismatch will not allow the trunk to come up. Also remember that the only trunking protocol is dot1q on the 2960 Switch and it might be ISL on older models of Cisco switches.

2. **Trunk does not carry traffic from relevant VLANs** – Trunks carry the traffic for all VLANs by default. Only two things can cause this problem: allowed list and pruning. The show interface trunk command will show which VLANs are allowed across the trunk and which VLANs are pruned.

```
Switch(config-if)#switchport trunk allowed vlan 10-20
```

```
Switch(config-if)#end
```

```
Switch#show int trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/1	on	802.1q	trunking	1

Port Vlans allowed on trunk

Fa0/1 10-20

3. **Trunk link does not have the same encapsulation at both ends** – This might happen because some Cisco switches allow both ISL and dot1q encapsulation types. If, by mistake, you configure ISL at one end and dot1q at the other, the trunk link will not work. This can be checked using the following command:

```
Switch#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/1	on	802.1q	trunking	1
Fa0/2	on	802.1q	trunking	1

Troubleshooting VTP

VTP problems include the following:

1. **VTP client does not receive or apply information from the server** – The first thing to check is whether the trunk link is configured and active between the

VTP server and the client. This includes trunk links between any switches between the VTP server and client if the client in consideration is not directly connected.

Second, ensure that the VTP domain and password are correct. You can check the domain name with the show vtp status command and the password with the show vtp password command.

Another important factor is the revision number. If the VTP client is an old switch with pre-existing configuration, then it might have a higher revision than the one being advertised by the server. In such situations change the domain of the client to something else and then revert it back to the correct domain. This will reset the revision number on the client. The show vtp status command helps to verify the VTP configuration. You can see plenty of examples of the show commands in the theory and lab sections.

2. **New VTP client caused a change in the VLAN database in the entire network** – This can happen only if the client was brought from a lab or another network (using the same domain name and password, if one was set) and had a higher revision number. This can be verified using the show vtp status command.
3. **VTP pruning is not working correctly** – If there is a VTP transparent switch in between the VTP server and the VTP client, then VTP pruning will not work. Another reason VTP pruning will appear not to be working correctly is the configuration of allowed VLANs on the trunk links. Some VLANs might have been removed manually. This can be verified using the show interface trunk command.

Mini-Lab – Troubleshooting VTP, VLANs, and Trunking

For this lab you will have to play along by breaking it before you fix it. Figure 9.2 below shows a very simple network. A host on either end should be in VLAN 10 and connect across a trunk link to the other host. You have been called into a small business to troubleshoot and resolve the issue. In the CCNA exam you may just have to check configurations and answer questions, or log in to one or more devices and resolve the issue.

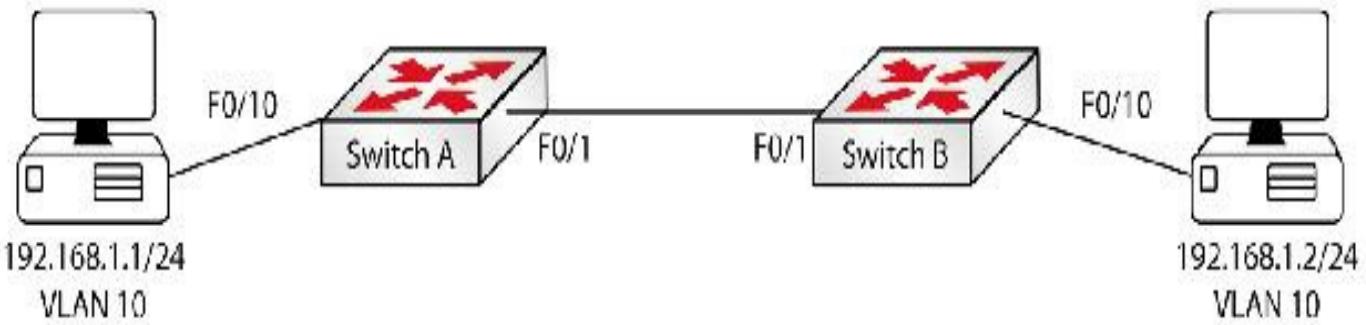


FIG 9.2 – Mini-lab: Troubleshooting VTP, VLANs, and Trunking

You need to add a faulty configuration to both sides before you fix it:

```

SwitchA(config)#int f0/10
SwitchA(config-if)#shut
SwitchA(config-if)#vtp domain howtonetwork
Changing VTP domain name from NULL to howtonetwork
SwitchA(config)#vtp password cisco
Setting device VLAN database password to cisco
SwitchA(config)#int f0/1
SwitchA(config-if)#switchport mode trunk
SwitchA(config-if)#switchport trunk allowed vlan 1-9

```

And you can break a few things on Switch B for good measure:

```

SwitchB#conf t
SwitchB(config)#int f0/1
SwitchB(config-if)#switchport mode access
SwitchB(config-if)#duplex half
SwitchB(config-if)#int f0/10
SwitchB(config-if)#switchport mode access
SwitchB(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SwitchB(config-if)#no shut
SwitchB(config-if)#exit
SwitchB(config)#vtp domain howtonetwork
Domain name already set to howtonetwork.

```

```
SwitchB(config)#vtp password hello
```

Setting device VLAN database password to hello

```
SwitchB(config)#
```

Now pretend that you are troubleshooting the network for the first time. Looking at the diagram above, you would expect certain things to be true from what you already know about VLANs, VTP, and trunking, such as:

- The listed ports should be up
- Hosts should be connected to access ports
- Access ports should be in VLAN 10
- Links should be trunking via being set to trunk or via DTP
- VLAN 10 and the native VLAN should pass across the trunk
- VTP domain and password should match

Being methodical about this, you know that if a port is down nothing interesting will happen. I've removed much of the irrelevant output in the commands below to save space:

1. Are the relevant interfaces up?

```
SwitchA#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
-----------	------------	-----	--------	--------	----------

Fa0/1	unassigned	YES	manual	up	up
-------	------------	-----	--------	----	----

Fa0/10	unassigned	YES	manual	administratively down	down
--------	------------	-----	--------	-----------------------	------

Alarm bells should be ringing because your access port is administratively down. You can easily fix this with the no shut command. You don't need to reissue the show ip interface command again because you will see the interface come up.

2. Is the interface placed into the correct VLAN and set to access?

```
SwitchA#show int f0/10 switchport
```

Name: Fa0/10

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: static access

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

Access Mode VLAN: 1 (default)

Trunking Native Mode VLAN: 1 (default)

The port is set to access but it's in VLAN 1. If you wanted to check this another way you could issue a show vlan brief command. I've removed some output to save space:

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5

This, of course, is easily fixed:

```
SwitchA(config)#int f0/10
SwitchA(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SwitchA(config-if)#end
SwitchA#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/5
10 VLAN0010	active	Fa0/10

3. You can now move on and check the trunking:

```
SwitchA#show interfaces trunk
Port      Mode      Encapsulation  Status      Native
                                         vlan
Fa0/1    on        802.1q        trunking    1
Port      Vlans allowed on trunk
Fa0/1    1-9
```

Good news and bad news. The interface is trunking but only VLANs 1 through 9 are permitted. There is a configuration line blocking your VLAN. In the exam you will probably have only the show run command available as opposed to the show

run interface X, for example. I've pasted in the relevant show run output below to save space:

```
interface FastEthernet0/1
switchport trunk allowed vlan 1-9
switchport mode trunk
```

You will be told in the exam what to allow. You might want to remove the entire configuration line by typing it out again with no in front, or just add VLAN 10.

```
SwitchA(config)#int f0/1
SwitchA(config-if)#switchport trunk allowed vlan 1-10
SwitchA(config-if)#end
SwitchA#show int trunk
Port      Mode      Encapsulation  Status      Native
                                         vlan
Fa0/1    on       802.1q      trunking    1
Port      Vlans allowed on trunk
Fa0/1    1-10
```

4. Check the VTP. This is the final part you need to check. As you know, both sides need to match the domain and password:

```
SwitchA#show vtp status
VTP Version: 2
Configuration Revision: 1
Maximum VLANs supported locally: 255
Number of existing VLANs: 6
VTP Operating Mode: Server
VTP Domain Name: howtonetwork
VTP Pruning Mode: Disabled
VTP V2 Mode: Disabled
VTP Traps Generation: Disabled
MD5 digest: 0xAA 0x0E 0xB1 0xB4 0xEE 0x2F 0xAC 0xC5
Configuration last modified by 0.0.0.0 at 3-1-93 04:26:01
```

Local updater ID is 0.0.0.0 (no valid interface found)

You can't see the password in the configuration but there is a command you can use to check it:

SwitchA#show vtp password

VTP Password: cisco

So far, you can see that the interfaces are correctly trunking or set to access, and the VLAN is applied to the interface and allowed across the trunk. The VTP configuration is also correct. You can move over to Switch B. I'll keep this part shorter because you know the drill by now. Interfaces up, correctly set to access/trunk, VLANs exist and permitted, and VTP configuration agrees:

SwitchB#show ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	unassigned	YES	manual	up	up
FastEthernet0/10	unassigned	YES	manual	up	up

SwitchB#show int f0/10 switchport

Name: Fa0/10

Switchport: Enabled

Administrative Mode: static access

Operational Mode: static access

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: native

Negotiation of Trunking: Off

Access Mode VLAN: 10 (VLAN0010)

Trunking Native Mode VLAN: 1 (default)

SwitchB#show interface trunk

SwitchB#

SwitchB(config)#int f0/1

SwitchB(config-if)#switchport mode trunk

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

SwitchB#show int trunk

Port	Mode	Encapsulation	Status	Native vlan
------	------	---------------	--------	----------------

Fa0/1	on	802.1q	trunking	1
-------	----	--------	----------	---

Port **Vlans allowed on trunk**

Fa0/1 1-1005

SwitchB#show vtp status

VTP Version: 2

Configuration Revision: 1

Maximum VLANs supported locally: 255

Number of existing VLANs: 6

VTP Operating Mode: Server

VTP Domain Name: howtonetwork

VTP Pruning Mode: Disabled

VTP V2 Mode: Disabled

VTP Traps Generation: Disabled

MD5 digest: 0x31 0x48 0xCC 0x20 0x45 0xFB 0x46 0x1F

Configuration last modified by 0.0.0.0 at 3-1-93 03:58:08

Local updater ID is 0.0.0.0 (no valid interface found)

SwitchB#show vtp password

VTP Password: hello

SwitchB(config)#vtp password cisco

Setting device VLAN database password to cisco

You know that the interface was set to half-duplex on Switch B; however, because autoconfiguration is running, Switch A has set itself to the same. If you had hard set it to full-duplex and 100 Mbps, you would have seen errors. This is a small gotcha because the network will work as is but you want the link to work at full capacity, so set full-duplex on both ends (or at least change it back on Switch B):

SwitchB(config)#int f0/1

SwitchB(config-if)#duplex full

Now, all things being equal, you should be able to ping across the link. The output below shows 192.168.1.1 pinging its neighbor on the other side of VLAN 10:

```
PC>ping 192.168.1.2
```

Pinging 192.168.1.2 with 32 bytes of data:

```
Reply from 192.168.1.2: bytes=32 time=1ms TTL=128
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
```

Ping statistics for 192.168.1.2:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 1ms, Average = 0ms

[END OF MINI-LAB]

Troubleshooting Host IP Addressing Issues

There are a few things to consider when troubleshooting host connectivity at layer 3. Let's work on the following example:

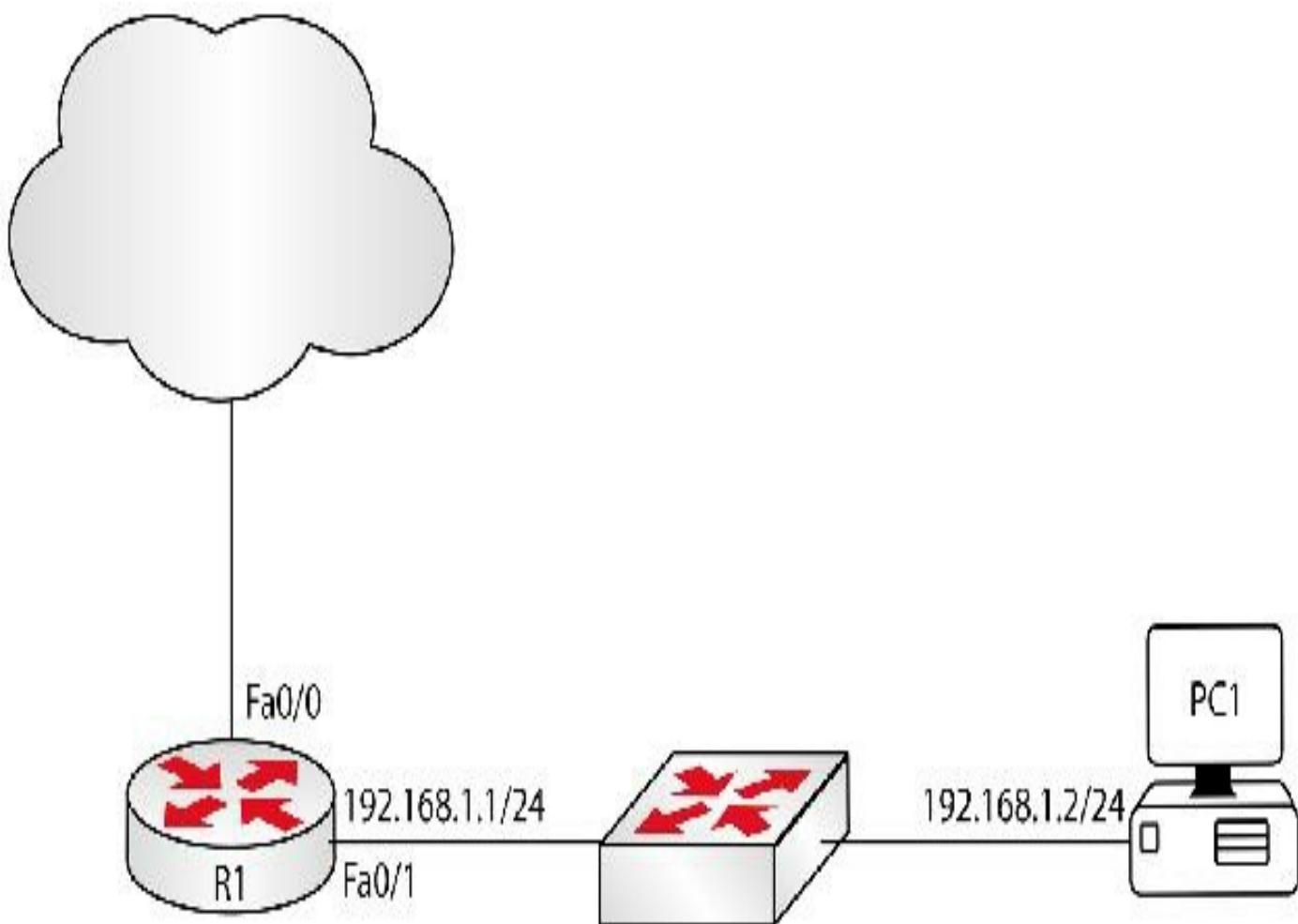


Figure 9.3 – Troubleshooting host IP addressing issues

Usually, when troubleshooting scenarios that involve verifications only at the IP layer, the recommendation is to start the investigation from one end of the network. Let's assume that you will start the troubleshooting process from the Internet cloud and work your way to the user's workstation. The first thing you would do is verify interface Fast Ethernet 0/0 on R1 and ensure that:

1. The interface is in up/up status; and
2. The interface is properly addressed.

You can check the interface up/up status (as well as the IP address and subnet mask) using the following command:

```
R1#show interfaces fa0/1
```

FastEthernet0/1 is up, line protocol is up

Hardware is Gt96k FE, address is c201.0f00.0000 (bia c201.0f00.0000)

Internet address is 192.168.1.1/24

MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

[output truncated]

To verify correct IP addressing (but without seeing the subnet mask) on the Internet-facing interface, use the following command to display a summary:

R1#sho ip int br

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	192.168.1.1	YES	manual	up	up

[output omitted]

You can also use the show ip interface fa0/1 command to see both pieces of information.

After you make sure that everything is okay at the interface level, you will want to make sure that the router has connectivity to the Internet:

R1#ping 8.8.8.8

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:

!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 20/29/52 ms

Based on the output above, the ping is successful so you can shift your attention to the LAN-facing part of the network. Start by inspecting the LAN-facing router interface:

R1#sho ip interface fa0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 192.168.1.1/24

Broadcast address is 255.255.255.255

Address determined by setup command

MTU is 1500 bytes

[output omitted]

The interface is up and the correct IP address is assigned. One additional thing you might want to check on the router is the DHCP server configuration if the host is configured as a DHCP client. In this case you need to pay attention to a few parameters:

- Correct network and subnet statement in the DHCP pool
- Correct default gateway statement in the DHCP pool
- Correct DNS server statement in the DHCP pool

- Correctly configured excluded addresses range

Next, check the switch for correct layer 2 configuration (VLAN assignment). The process for this was demonstrated in a previous section so it won't be repeated here.

The last element in the troubleshooting process is the user workstation. From an IP connectivity perspective, check the following:

- Correct IP address assignment – from the same range as R1's Fast Ethernet 0/0 interface
- Correct subnet mask assignment
- Correct default gateway assignment – this should match the address on R1's interface (192.168.1.1)
- Correct DNS server assignment – this can be manually configured or automatically assigned from the DHCP server
- Make sure that you don't have other workstations in the network with the same IP address because communication errors might occur if duplicate IP addresses are present in the same VLAN

Troubleshooting Access Lists

You can check which access lists are configured on your router with the show ip access-lists command:

```
Router#show ip access-lists
```

```
Standard IP access list 10
```

```
    permit 10.1.1.2
```

```
Extended IP access list 100
```

```
    permit tcp any any eq domain
```

```
    permit udp any any eq 25
```

```
    permit tcp any any eq www
```

You can see an access list on an interface by issuing the show ip interface [interface type/number] command, including in which direction it has been placed:

```
Router#show ip interface Serial0/0
```

```
Serial0/0 is up, line protocol is up
```

```
Internet address is 192.168.1.1/24
```

```
Broadcast address is 255.255.255.255
```

```
Address determined by setup command
```

```
MTU is 1500 bytes
```

```
Helper address is not set
```

```
Directed broadcast forwarding is disabled
```

Outgoing access list is not set

Inbound access list is 10

You can also type show run interface X. Note that this command may not be available on your router depending on the IOS release you are using. It probably won't work on Packet Tracer or in the exam.

```
Router#show run interface Serial0/0
interface Serial0/0
ip address 192.168.1.1 255.255.255.0
ip access-group 10 in
end
```

It is possible to run a debug on access lists. This is outside the scope of the CCNA syllabus (please check the latest syllabus). You would specify the traffic you want to monitor with an access list and then enter the command below in global configuration mode:

```
Router#debug ip packet 101 [detail] i Where 101 is your access list
```

- detail – gives more detailed information in the debug

The debug ip packet command in general will show only process-switched traffic, so it is not much use if CEF (Cisco Express Forwarding) is turned on.



You can see actual statistics per ACL, including how many matches there have been, with the show access-lists X command:

```
Router#show access-lists 100
Extended IP access list 100
    permit tcp host 192.168.1.1 any established (308 matches)
    permit udp host 192.168.1.2 any eq domain (12 matches)
    permit icmp host 192.168.1.3 any
    permit tcp host 192.168.4.5 host 10.0.0.1 gt 1023
    permit tcp host 192.168.4.8 host 10.0.0.2 eq smtp (4 matches)
```

You can clear the counters on an access list with the clear access-list counters command:

```
Router#clear access-list counters ?
```

```
[0-199] Access list number
```

```
WORD Access list name
```

```
[cr]
```

```
Router#clear access-list counters 100
```

```
Router#show access-lists 100
```

Extended IP access list 100

```
 permit tcp host 192.168.1.1 any established
```

```
 permit udp host 192.168.1.2 any eq domain
```

```
 permit icmp host 192.168.1.3 any
```

```
 permit tcp host 192.168.4.5 host 10.0.0.1 gt 1023
```

```
 permit tcp host 192.168.4.8 host 10.0.0.2 eq smtp
```

The majority of ACL issues crop up either from a configuration error, such as putting the wrong network address or wildcard mask in, or forgetting to apply the ACL to an interface. Remember that if you add the ACL to an internal interface it will only inspect traffic when it hits that interface. If, for example, you are trying to block Telnet access coming in from the Internet, your internal interface ACL won't block it.

Also, remember that access lists are processed top-down; if there is a match higher up, the more specific entry further down won't be reached. The router will NOT filter self-generated traffic.

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 9 exam.

PART 2 — ICND2

Chapter 10 — LAN Switching Technologies

What You Will Learn in This Chapter

Spanning Tree Protocol

Rapid Spanning Tree Protocol

Syllabus Topics Covered

1.0 LAN Switching Technologies

1.1 Identify enhanced switching technologies

1.1.a RSTP

1.1.b PVSTP

1.2 Configure and verify PVSTP operation

1.2.a Describe root bridge election

1.2.b Spanning Tree mode

Spanning Tree Protocol (STP) is enabled by default on Cisco switches. Because it works so well it is disregarded by many network engineers who regret doing so when faced with an STP issue, which will quickly bring down even enterprise-level networks. Cisco expects you to have a good grasp of STP and RSTP (Rapid STP) for the exam, including how they operate, how to tweak the configurations, and how to troubleshoot common issues.

Spanning Tree Protocol

Imagine for a moment that you drive a delivery truck in a large city. There is a large network of roads for you to choose from and you know the name of the building you need to reach. Your boss doesn't permit you to use a map or satellite navigation so you have to drive around random streets until you find your destination.

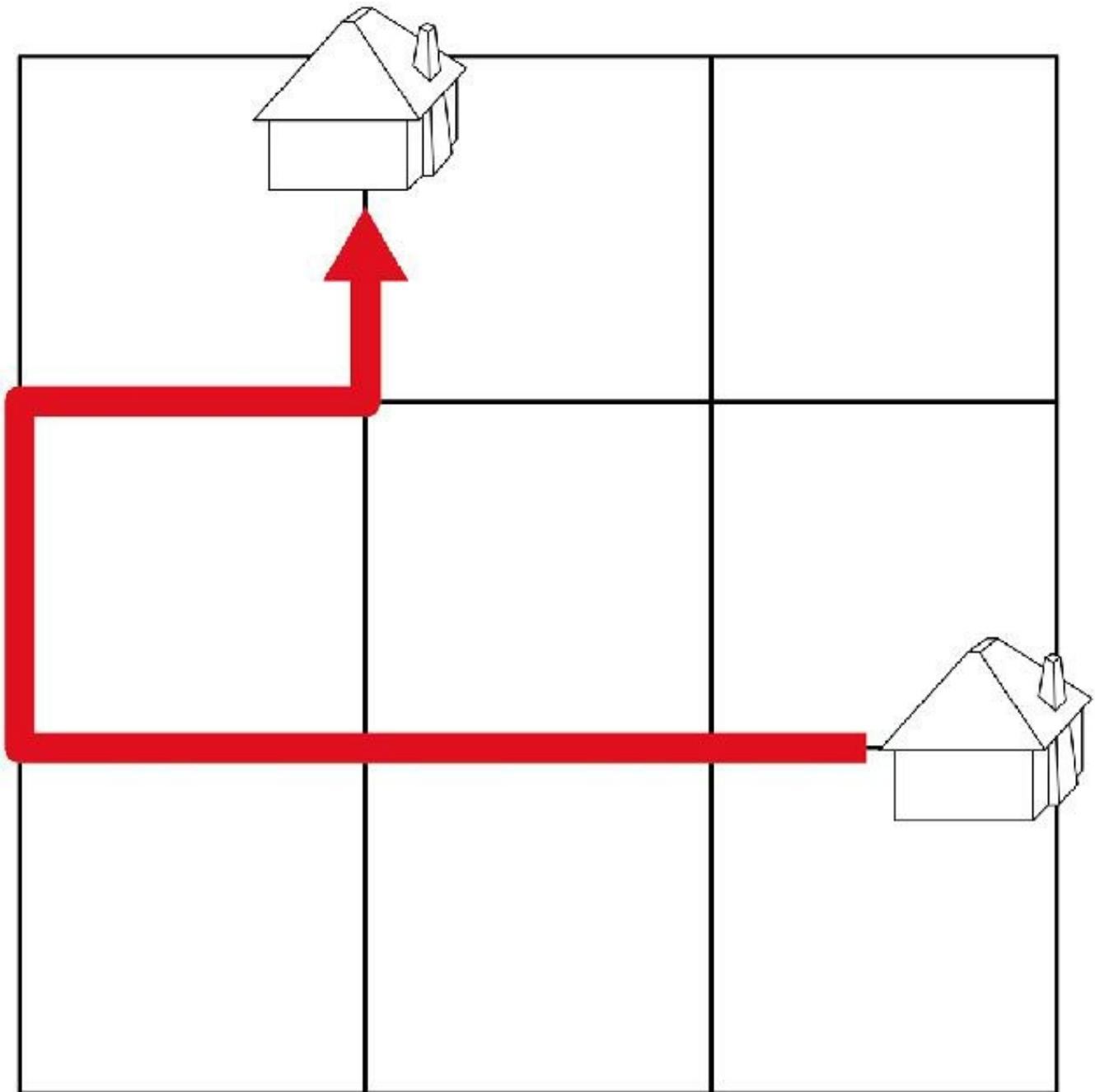


FIG 10.1 – Taking the long route

You eventually reach your destination but the journey was a nightmare, and you still have to find your way back. Having a grid of roads you can take is useful, but if they are all open you run the risk of going around in circles. Now imagine that the next time you need to make the journey, the fastest route has been made available to you because the longer routes have been closed off.

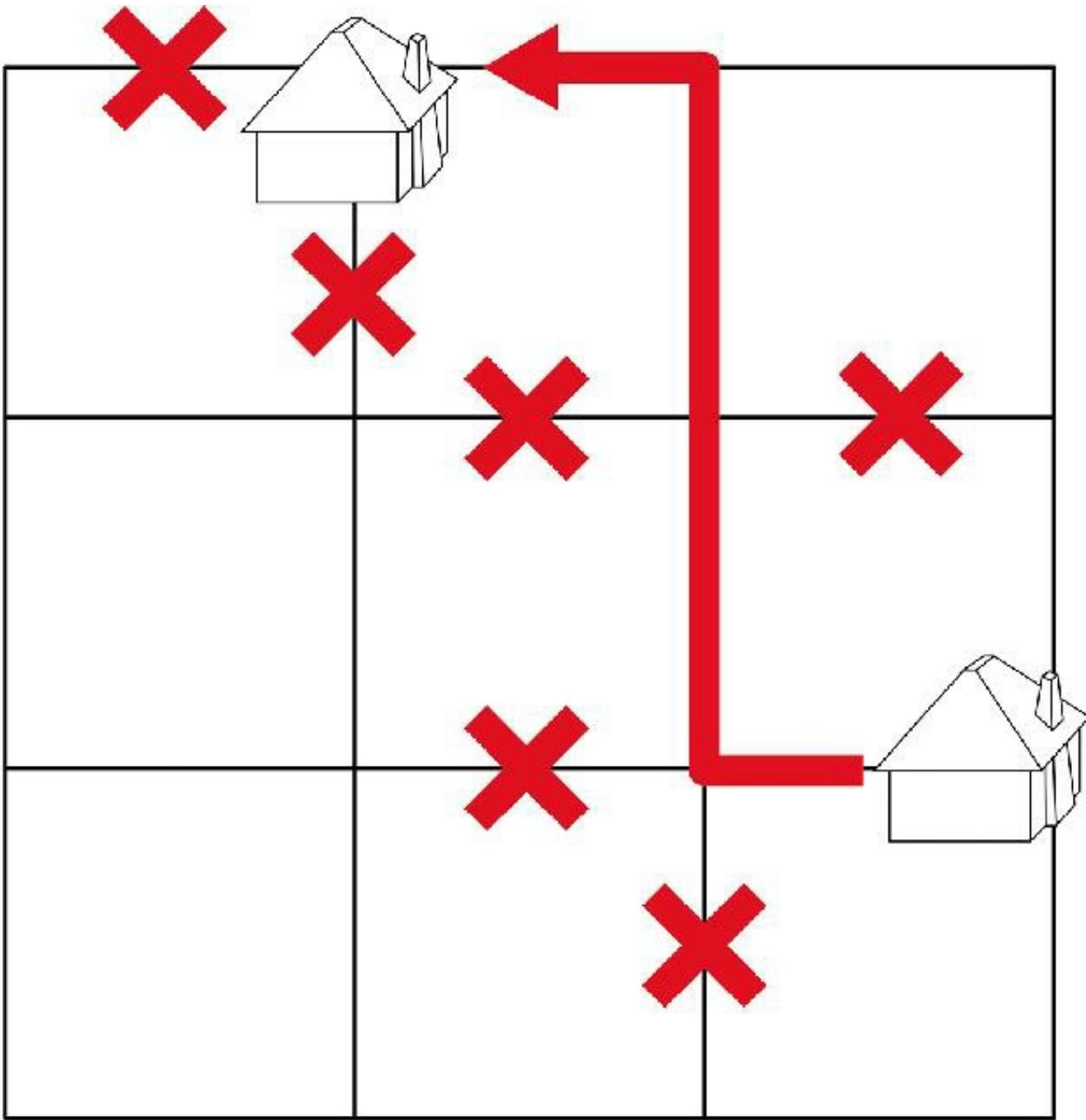


FIG 10.2 – Longer paths shut down

This time it's impossible to become lost. If one of the main routes was no longer available, an alternative route would be opened for you. This is a simplified explanation of how Spanning Tree Protocol works.

STP was originally created by Radia Perlman and was later standardized by IEEE 802.1D as a messaging service between switches designed to provide a loop-free topology in a layer 2 network that has multiple redundant paths. In order to achieve this, STP prevents some interfaces (referred to as ports in this context) from forwarding traffic.

In theory, the remaining ports will all forward traffic in a loop-free network. STP works so well in fact that its importance is often forgotten and many network engineers find it very difficult to troubleshoot (because they don't understand it). When I was working at Cisco TAC, we would regularly have to assist even CCIEs who were struggling to configure or troubleshoot STP issues in their network.

Figure 10.3 below shows a full-mesh network, a good redundant setup where, if one link fails, there are two more links for traffic to go through. However, could this lead to any problems? Let's say a host is connected to port Fa0/1 on Switch A (MAC address AAA) and this switch sends a broadcast out to the network advertising this MAC. Of course, this is desired behavior because we need the switched network to build a map of which MAC addresses are connected to which interfaces.

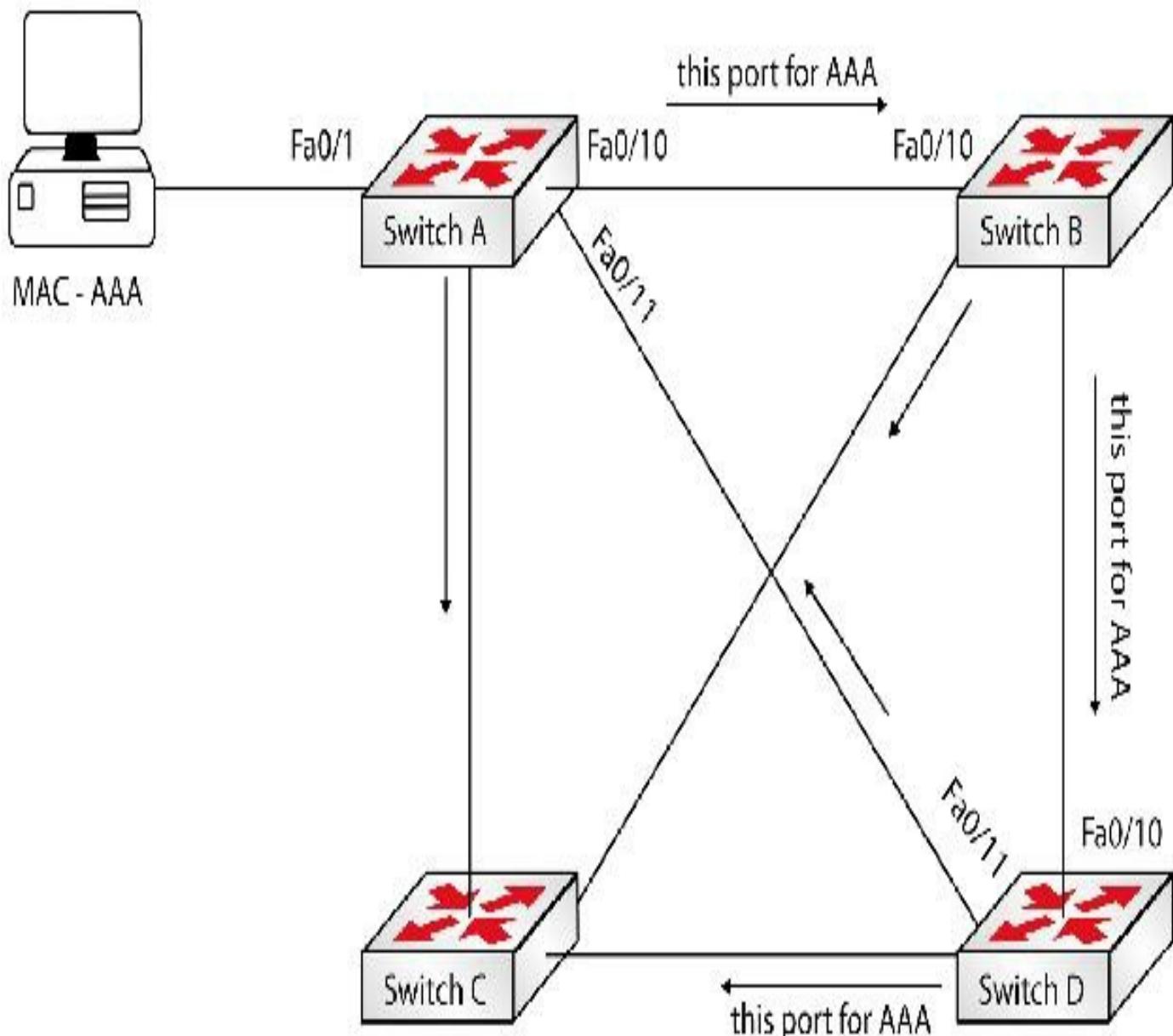


FIG 10.3 – Beginning of a switching loop

Switch A has to forward this frame out of every port except fa0/1. Part of what happens next is shown below:

1. Switch B receives the packet on fa0/10 and sends it out on every port except that one;
2. Switch D receives the packet on fa0/10 and sends it out on every port except that one, but including fa0/11; and
3. Switch A receives the packet on fa0/11 and sends it out on every port except fa0/11, but including fa0/1 and fa0/10!

As a result, not only has the original source received the frame back, but now Switch A has to send the packet back out of fa0/10 also. Soon enough, all ports are advertising the fact that they can reach host AAA. The network will shortly become unusable as demonstrated in Figure 10.4 below.

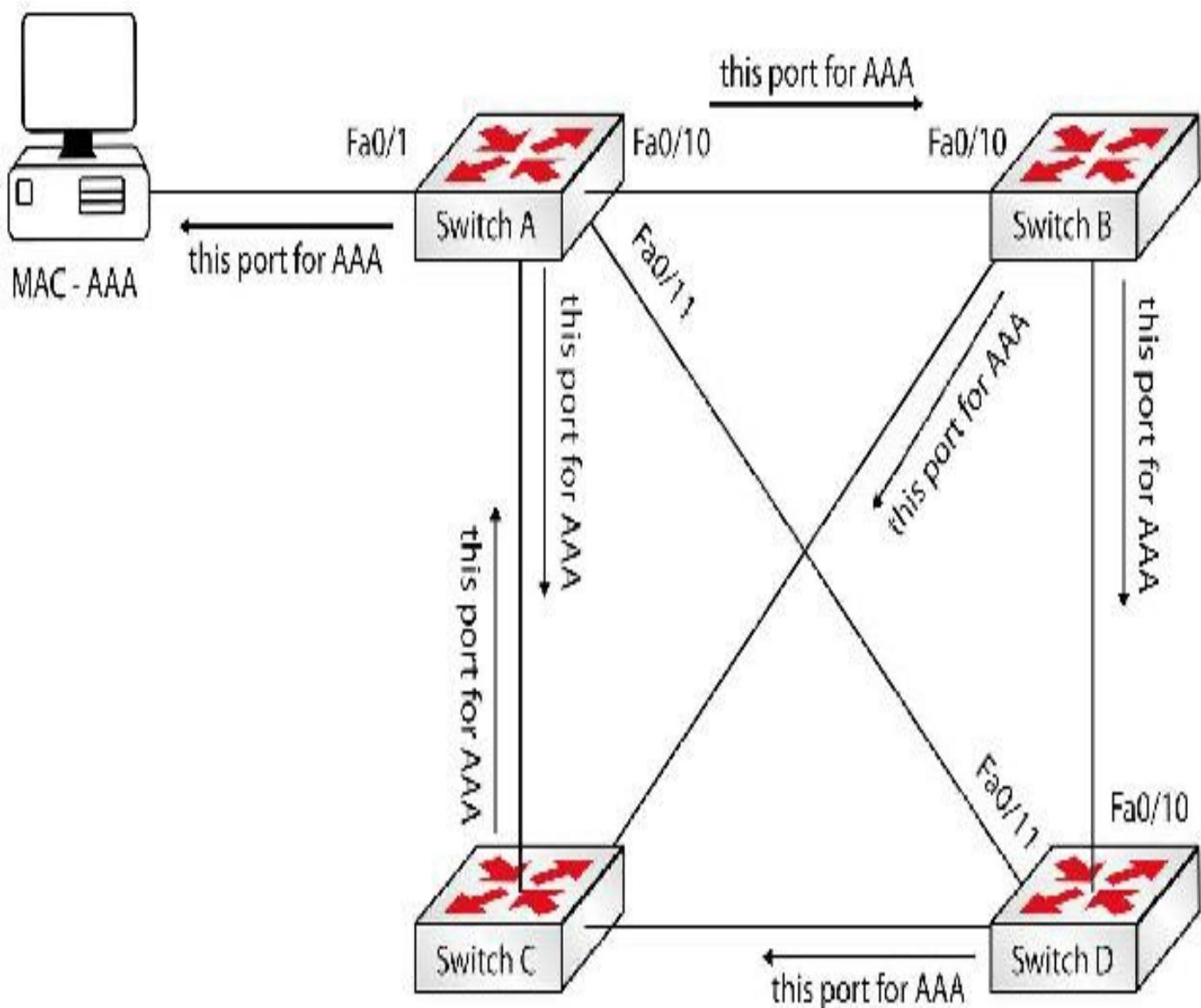


FIG 10.4 – Full switching loop

As you can see in Figure 10.4, a loop has been created. Such loops can bring a network to a grinding halt. Layer 2 LAN protocols have no method to stop traffic from endlessly traveling around, possibly carrying inaccurate information. At layer 3, you can make packets expire after a certain amount of time or after they have traveled a certain distance (using the TTL value or route poisoning, for example).

As layer 2 networks grew, it quickly became evident that a system to prevent loops was needed if LANs were to continue to function.

STP allows switches to communicate with each other so they can create a loop-free topology. It does this by electing a root bridge, which becomes the logical center of the switched network. It then builds a loop-free path leading toward the root bridge. Note that the logical root switch does not have to be at the center of the network physically.

STP is enabled by default on all bridges (switches). This means that you can install several switches, configure VLANs, and STP will work to prevent loops. For this reason, it isn't compulsory to add any configuration, but as the network administrator, you may want to add some of the configuration commands discussed in this section to determine where your layer 2 traffic goes, for example, to a more powerful switch or load balanced per VLAN.

In the network in Figure 10.5 below, the network hosts are missing but are all connected to the access layer switches using VLANs 20 and 30. The load for all the VLAN traffic has been distributed using configuration commands so that each distribution layer switch is the root for just one VLAN. It can switch traffic for both VLANs should the primary switch for that VLAN fail.

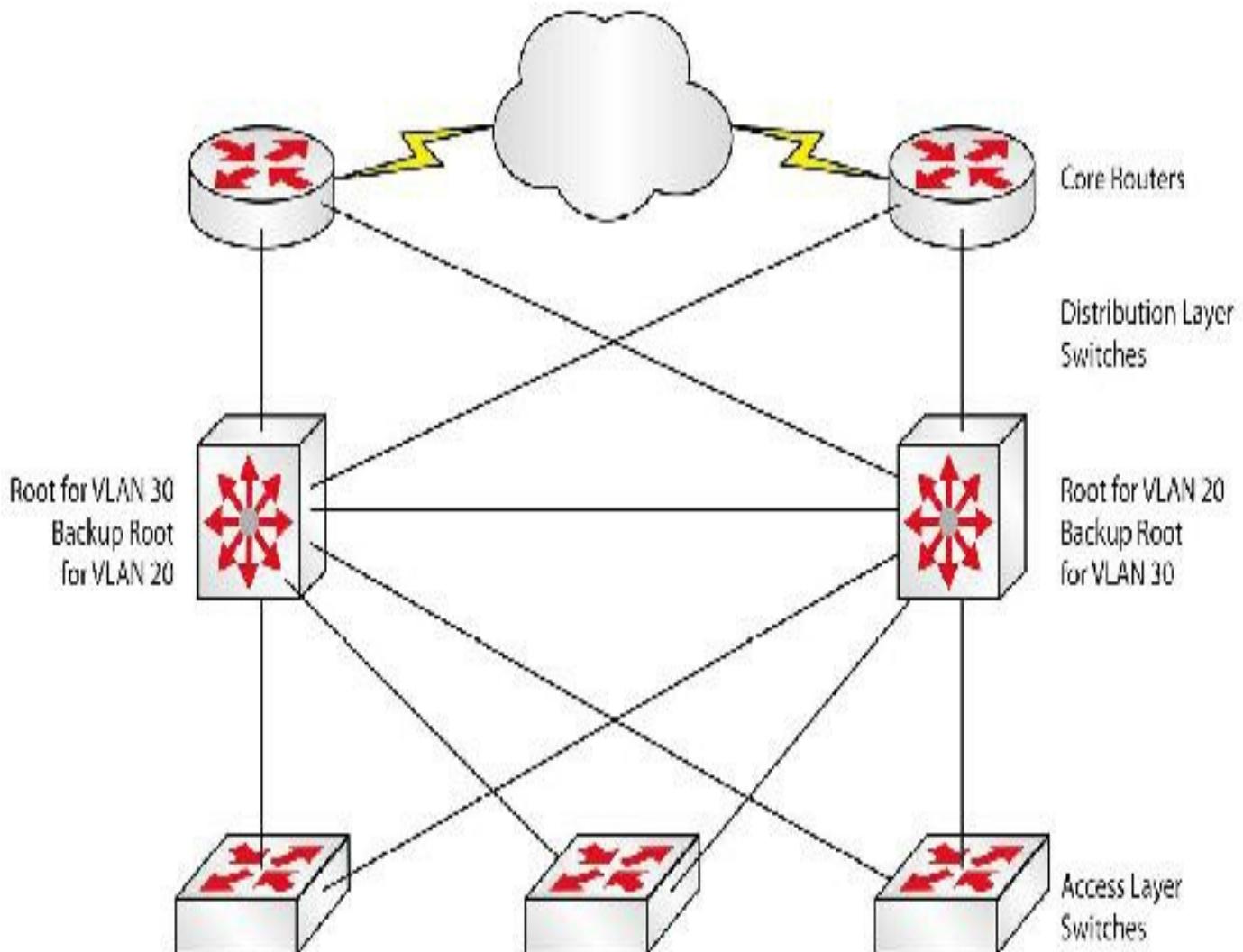


FIG 10.5 – Correct root bridge placement

If you connected all the devices without any configuration, you might find that one of the lower-powered access layer switches becomes the root for all your layer 2 traffic, which is certainly not a desired outcome. This is illustrated in Figure 10.6 below. As a CCNA-level engineer, you would quickly be able to troubleshoot and resolve this type of issue. Although it may seem unlikely, I've had to deal with it on many occasions when taking over from an IT engineer who just expected to plug-and-play multiple network switches.

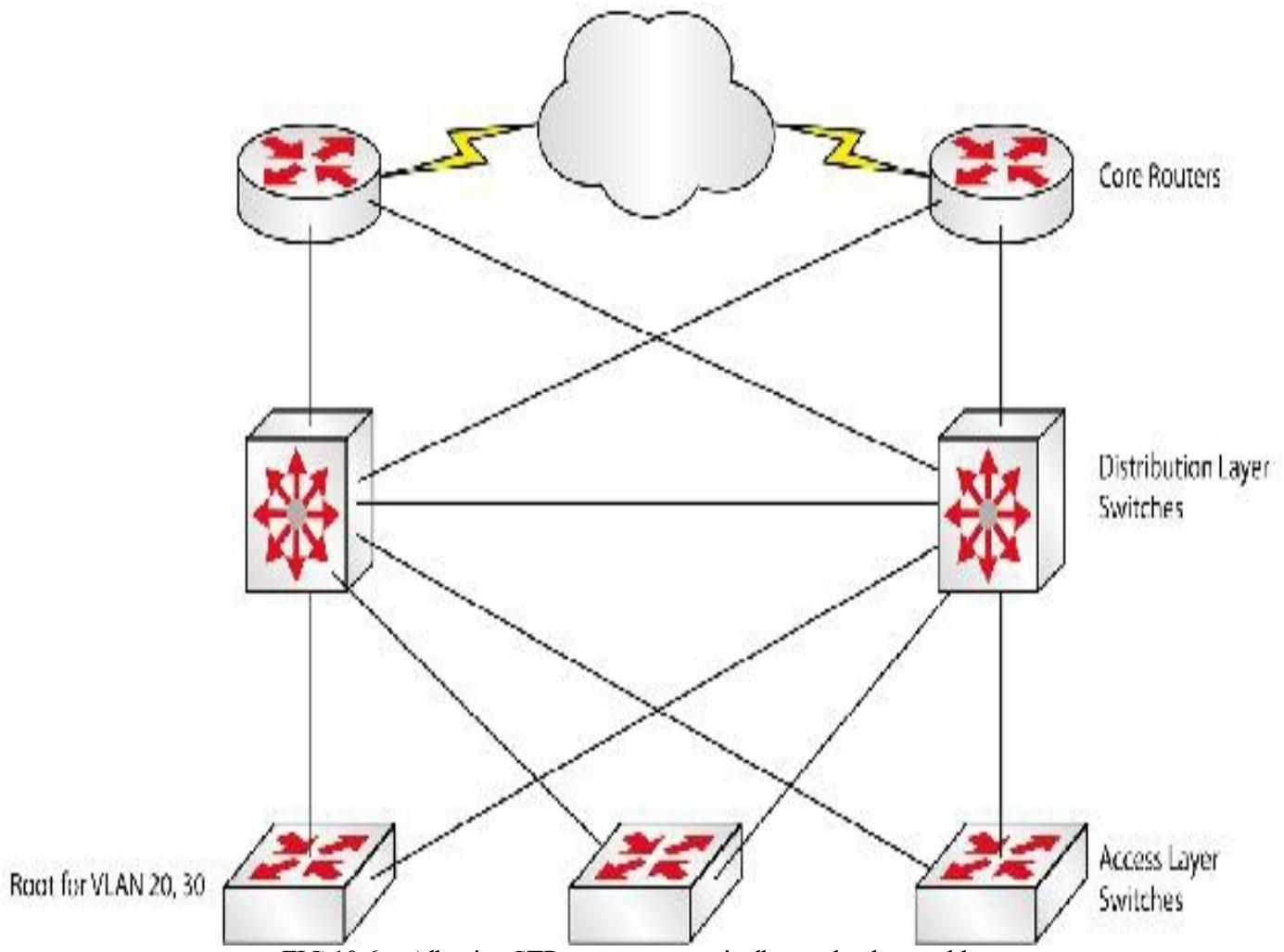


FIG 10.6 – Allowing STP to run automatically can lead to problems

Each bridge runs the Spanning Tree algorithm, which calculates how the loop (as seen in Figure 10.4 above) can be prevented. When STP is applied to a looped LAN topology, all VLANs will be reachable but any open ports that would create a traffic loop are blocked. When it sees a loop in the network it blocks one or more redundant paths, preventing a loop from forming. As you can guess, STP calculates the best-cost path to reach the root bridge and then the best-cost interfaces are put into forwarding state while the others are put into blocking state.

All this is achieved by swapping Bridge Protocol Data Units (BPDUs). STP also uses BPDUs to continually monitor the network to look for failures on switch ports or changes in the network topology. If a change in the LAN is detected, STP can make redundant ports available and close other ports to ensure that the network continues to function loop-free. This entire process can take quite some time but improvements have been developed to speed this up, which we'll cover shortly.

Figure 10.7 below shows a packet capture of a BPDU. There are actually two types—a Topology Change Notification (TCN) BPDU, which is used for topology changes (such

as an interface going down), and a configuration BPDU, which is used for initial STP configuration. Can you tell which one is in the packet capture below?

□ Spanning Tree Protocol

Protocol Identifier: Spanning Tree Protocol (0x0000)

Protocol Version Identifier: Spanning Tree (0)

BPDU Type: Configuration (0x00)

□ BPDU flags: 0x00

0.... = Topology Change Acknowledgment: No

.... ...0 = Topology Change: No

□ Root Identifier: 32768 / 10 / 00:1c:b1:91:9a:00

Root Bridge Priority: 32768

Root Bridge System ID Extension: 10

Root Bridge System ID: Cisco_91:9a:00 (00:1c:b1:91:9a:00)

Root Path Cost: 16

□ Bridge Identifier: 32768 / 10 / 00:21:d8:3d:23:00

Bridge Priority: 32768

Bridge System ID Extension: 10

Bridge System ID: Cisco_3d:23:00 (00:21:d8:3d:23:00)

Port identifier: 0x8004

Message Age: 4

Max Age: 20

Hello Time: 2

Forward Delay: 15

FIG 10.7 – BPDU packet capture

Before you learn more about STP, you need to understand some of the common terms associated with it. To see some of the values on a switch, you would issue the show spanning-tree vlan [vlan#] command. As always, it would be a great idea to get access to a live switch and try these out for yourself.

Switch#show spanning-tree vlan 1

VLAN0001

Spanning tree enabled protocol ieee

Root ID Priority 32769
 Address 0001.4272.A095
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
 Address 0001.C934.3988
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface Role Sts Cost Prio.Nbr Type

Fa0/1 Root FWD 19 128.1 P2p

- **Root ID** – This is the ID of the bridge (switch) assumed to be the root. The switch usually assumes it is the root until the BPDUs have all been exchanged between the switches. You can see the root details, including the root cost (covered later), with the show spanning-tree root command, which I recommend you try during any STP labs:

Switch#show spanning-tree root

Root Hello Max Fwd

Vlan	Root ID	Cost	Time	Age	Dly
------	---------	------	------	-----	-----

Root Port

VLAN0001	32769	0011.9247.db00	0	2	20	15
----------	-------	----------------	---	---	----	----

- **Bridge ID** – The BID is the unique identification number of each switch in the network. It consists of bridge priority, the VLAN ID, and the base MAC address of the switch. In the previous show spanning-tree vlan 1 output, the BID is 32768 plus the VLAN ID, which is 1 and 0001.C934.3988.

The default bridge priority of a Cisco switch is 32768. This is a configurable value between 0 and 61440, but the value has to be in increments of 4096 (i.e., 4096, 8192, 12288, and so on). Priority plays a very big role in STP and how well the network will function, which we will examine in detail shortly.

- **Root bridge** – All switches in the network take part in an election to decide the

root of the spanning tree; this then leads to them making further decisions, such as which redundant path to block and which to open. The election is won by the switch with the lowest BID. Switches that do not become a root bridge are called non-root bridges.

- **BPDU** – A Bridge Protocol Data Unit contains information exchanged between switches to select a root bridge, as well as to configure the network after that. A decision on which port to block is made after examining BPDUs from neighbors. Cisco switches send BPDUs every two seconds by default. This value can be configured from one second to 10 seconds.
- **Root port** – Each switch has to have a path to the root bridge, if not directly connected. The root port is the directly connected link or the fastest path to the root bridge from a non-root bridge. The root bridge will never have a root port, which is always closest to the root bridge.
- **Port cost** – Each port has a cost that is determined by the bandwidth of the link. Port cost determines which of the redundant links will not be blocked. The lower the cost, the better it is. Port cost also determines which port will become the root port if multiple paths to the root bridge exist. Default port costs are shown in Table 10-1 below:

Table 10-1: STP port costs

Link Speed	STP Cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
10 Gbps	2

It's important that you learn the costs above for the exam so that you can look at a diagram and determine the best path. As a BPDU traverses the switched network, the cost is incremented. This is how the switch determines the least cost.

- **Designated port** – The bridges on a network segment collectively determine which bridge has the least-cost path from the network segment to the root. The port connecting this bridge to the network segment is then the designated port for the segment. Ports that are not selected as a designated port are called non-designated ports. Designated ports point away from the root bridge.

Port States in STP

Switch ports (the interfaces on the switch) running STP can be in one of five states. Please ensure that you understand all of these for the CCNA exam:

1. Blocking
2. Listening
3. Learning
4. Forwarding
5. Disabled

User data is only passed by a port in forwarding state.

First State – Blocking

None of the ports will transmit or receive any data, but they will listen to BPDUs. The BPDU carries various pieces of information that are used by STP to determine which state the ports should be in and what the STP topology should be.

Second State – Listening

The switch listens for frames but does not learn or act on them. The switch does receive the frames but discards them before any action is taken. MAC addresses are not placed into the CAM table while the port is listening.

Third State – Learning

The switch will start to learn MAC addresses it can see and will populate its CAM table with the addresses and the ports on which they were found. In this state, the switch will start to transmit its own BPDUs.

Fourth State – Forwarding

The switch has learned MAC addresses and corresponding ports and populates its CAM table with this information. The switch can now forward traffic.

Fifth State – Disabled

In disabled state, the port will receive BPDUs but will not forward them to the switch processor. Instead, it discards all incoming frames from both the port and other forwarding ports on the switch.

The port states are transitional and allow other BPDUs to arrive in good time from other switches. Port transition times are shown below and the process from start to finish can typically take 50 seconds (15 seconds each for learning and listening and 20 seconds for the MAX age timer, which will be covered shortly):

- Initialization to blocking
- Blocking to listening
- Listening to learning (15 seconds)
- Learning to forwarding (15 seconds)
- Forwarding to disabled (if there is a failure)

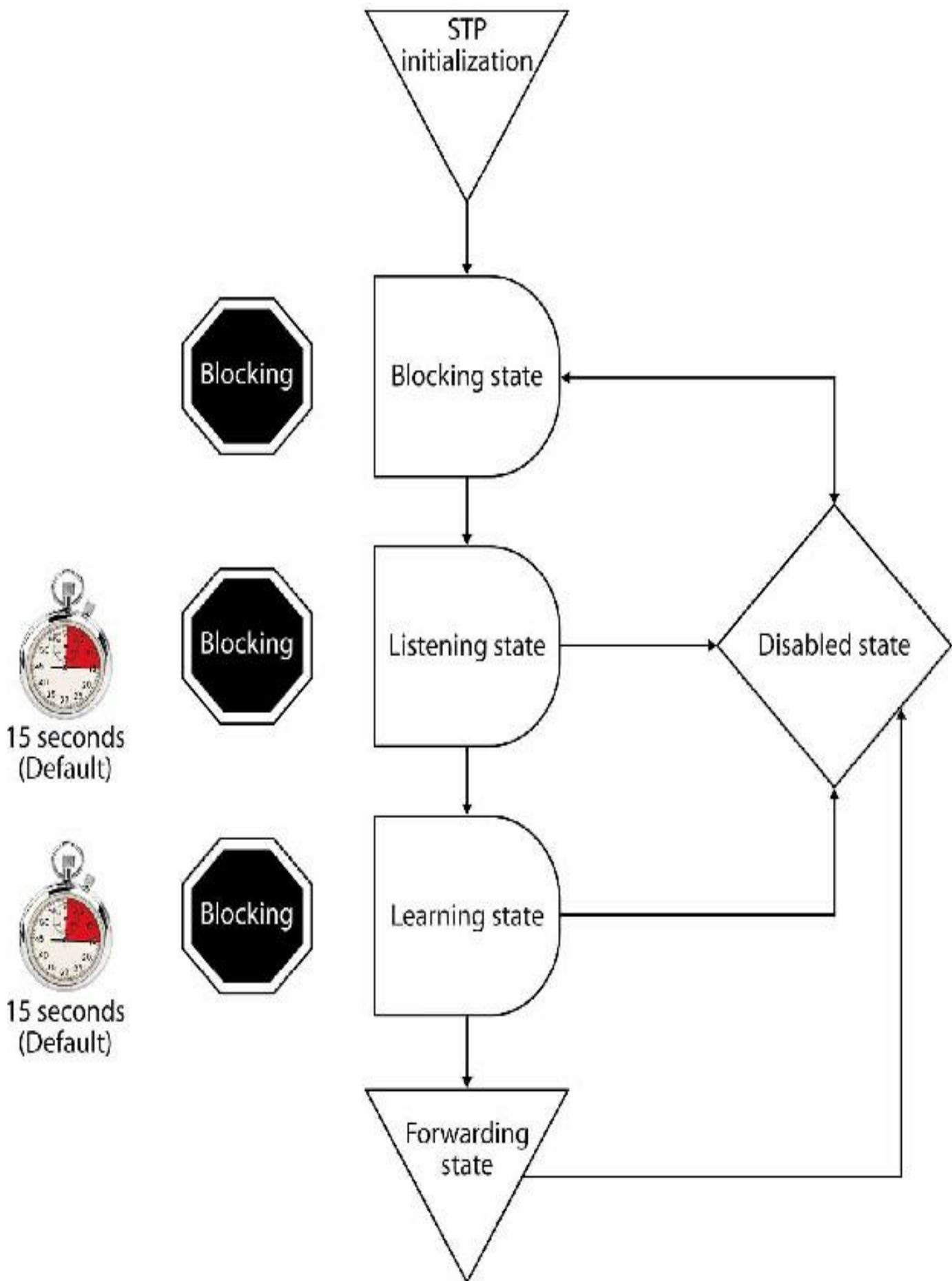


FIG 10.8 – STP port states

All ports start at the blocking state (except for a few exceptions, which will be discussed later). After STP convergence, some ports will transition to the listening, learning, and, finally, forwarding states and the rest will remain in the blocking state. Keeping this and the time needed to transition from one state to another in mind, a layer 2 network running STP takes 50 seconds to start switching data!

STP Convergence

Remember that STP works by selecting a root bridge in the LAN. It is selected by comparing the Bridge ID of each switch, and the switch with the lowest BID wins. As the network administrator, you can manually configure which switch you prefer to be the root (and secondary root). We'll look at how to do this shortly.

STP can be considered converged after three steps have taken place (all ports will either be blocking or forwarding):

1. Elect one root bridge (switch)
2. Elect root ports
3. Elect designated ports

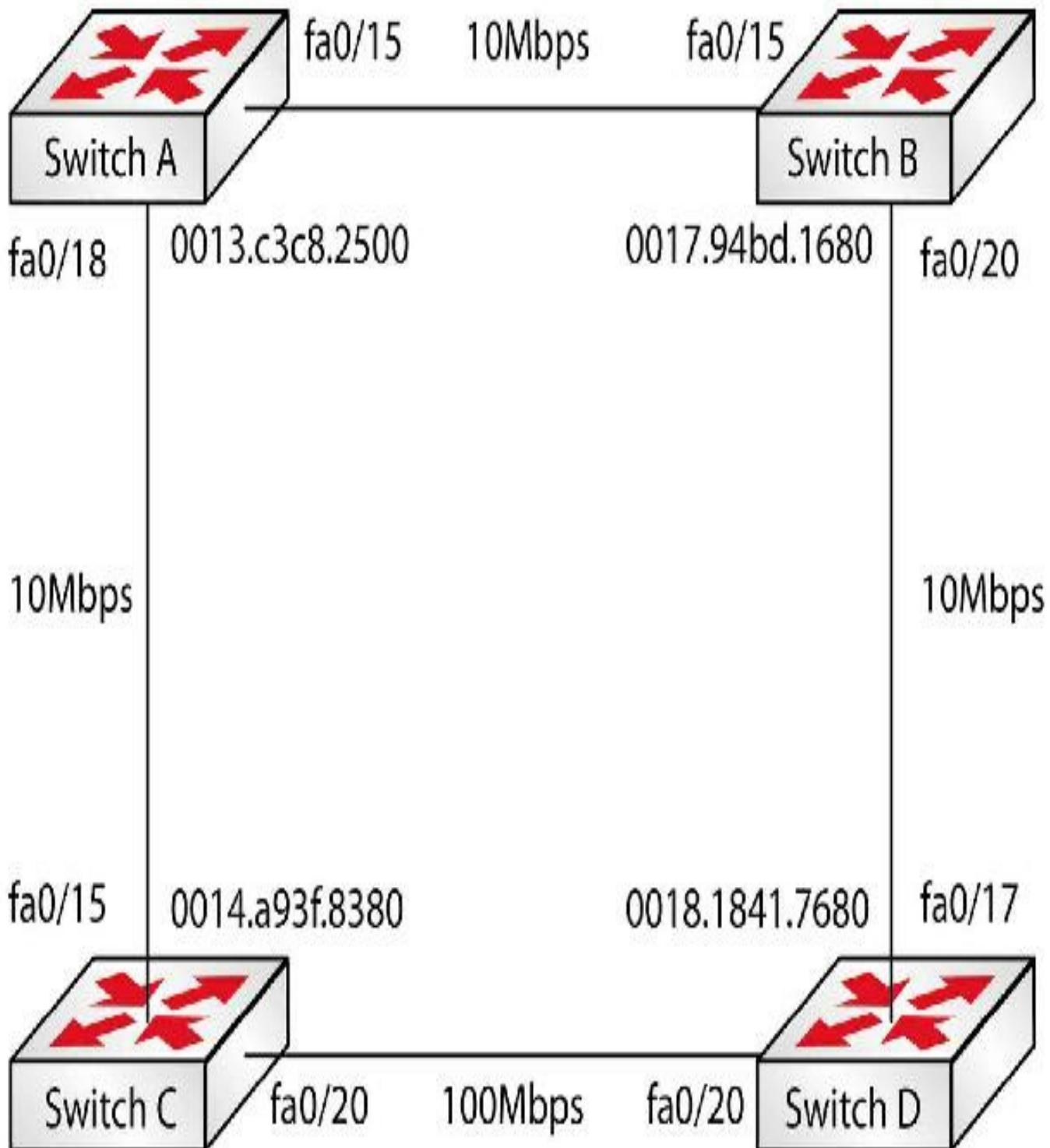


FIG 10.9 – STP convergence

We will use the network shown in Figure 10.9 above to go through the STP convergence process. VLAN 5 has been configured and all the interfaces shown have been placed into VLAN 5.

Elect One Root Bridge

The bridge with the lowest BID becomes the root bridge. The BID used to consist of

just two values in an 8-byte field—the bridge priority (32768 by default), which is two bytes, and the base MAC address of the backplane or supervisor module (depending on the switch model), which is six bytes.

Here is an extract from a show version command on a 2960 Switch. You can see the base MAC address:

32K bytes of flash-simulated non-volatile configuration memory.

Base ethernet MAC Address: 00:0C:BE:D4:3C:40

Motherboard assembly number: 34-7410-05

The 802.1T standard introduced an extended system ID in order to conserve MAC addresses while still allowing for a unique BID. With extended system ID enabled (VLAN 10 in this example), the bridge priority is set to either 4096 as a minimum or a multiple of 4096, depending on which bridge priority bits are set. The default priority of 32768 is a multiple of 4096. Figure 10.10 below shows the old BID format and the new one below that:

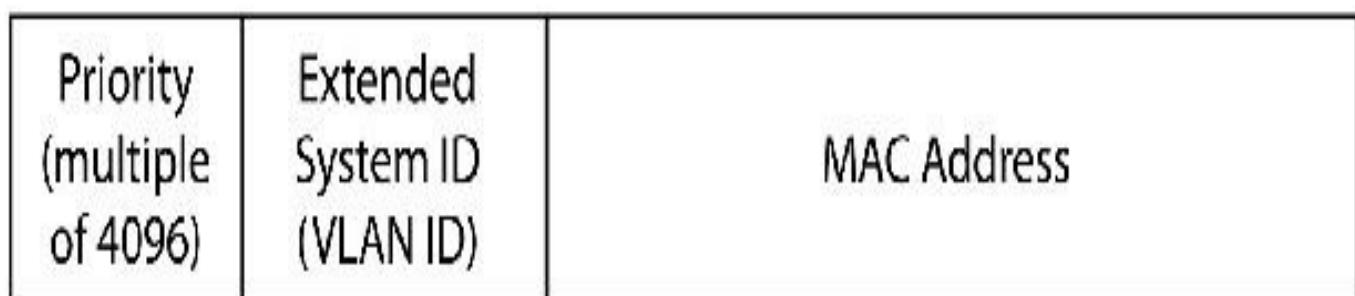


FIG 10.10 – Bridge ID format

The output below is from a 3550 Switch. The command below won't work on Packet Tracer:

VTP-Switch-1#show spanning-tree vlan 10 bridge

Vlan	Bridge ID	Hello	Max	Fwd	Time	Age	Dly	Protocol
VLAN0010	4106 (4096,10) 000d.bd06.4100	2	20	15	ieee			

Here you can see that the priority value is 4096, to which is added the VLAN ID of 10 and the MAC address.

The root bridge on a VLAN is selected by an election. Each switch running STP passes its BID information using BPDUs. BPDUs are multicast frames that are sent out every two seconds from every port (you can see the Hello time in the output above). This is necessary to maintain a loop-free topology. The bridge with the lowest ID is selected as the root bridge. This means that the switch with the lowest priority is elected as the root bridge. If all the switches have the same priority, then the switch with the lowest MAC address is selected as the root bridge, which in this instance is Switch A with MAC 0013.c3e8.2500.

All ports on the root bridge are set as designated ports and are always set to the forwarding state. We will discuss port roles shortly.

STP elections take place using the following order:

- Lowest root bridge ID
- Lowest root path cost to the root bridge
- Lowest sender bridge ID
- Lowest sender port ID

In the network in Figure 10.9, the priority of all the switches has been left at default, so the switch with the lowest MAC address will be selected as the root bridge. In this case it will be Switch A. To verify this, issue the show spanning tree vlan [vlan#] command on Switch A :

SwitchA#show spanning-tree vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 32773 i **32768 plus 5 (the VLAN ID)**

Address 0013.c3e8.2500

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)

Address 0013.c3e8.2500

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/15	Desg	FWD	100	128.15	P2p
Fa0/18	Desg	FWD	100	128.18	P2p

Notice that there is no Cost field displayed under the Address field because this switch is the root for this VLAN. If you wanted Switch C to be the root bridge, then you would need to give it a lower priority than 32773 using either of the following commands:

Switch(config)#spanning-tree vlan 5 root primary

Switch(config)#spanning-tree vlan 5 priority 8192

The first command lets the switch set its own priority for the specified VLAN to 4096 less than the lowest spanning tree switch priority value. You can also configure the secondary root bridge on any switch you want to take over as the root bridge in case the current root bridge fails.

The second command lets you manually choose the priority. Which command you choose will depend on your network policy; however, you need to know both for the CCNA exam and beyond so do try them both. If you added the priority 8192 command to a switch and the root primary to the second switch, then the second switch would set its priority to 4096, thus becoming the root bridge.

Let's check the show spanning-tree output on Switch C (once you have made it the root with one of the commands above) and on Switch A. The VLAN information is added to the priority value in the output:

SwitchC#show spanning-tree vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 8197

Address 0014.a93f.8380

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 8197 (priority 8192 sys-id-ext 5)
Address 0014.a93f.8380
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300

SwitchA#show spanning-tree vlan 5

VLAN0005

Spanning tree enabled protocol ieee
Root ID Priority 8197
Address **0014.a93f.8380**
Cost 100
Port 18 (FastEthernet0/18)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)

Address 0013.c3e8.2500
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300

Note that Switch A now shows Switch C's MAC address as the root bridge's MAC address and Switch C says that it is the root bridge. Now please set the priority value on Switch C back to 32768 and make Switch A the root bridge for the next section. Note that even though you set the priority value back to 32768 the VLAN # will be added to that value, so for VLAN 5 the priority will be 32673.

Elect Root Ports

For non-root bridges, there will be only one root port. The root port will be the port with the lowest path cost to the root bridge (i.e., the best/fastest path). The root port will be set to forwarding state.

Path cost is the cost of transmitting a frame to the root bridge. The value is set according to the bandwidth of the link on the LAN, so the slower the link, the higher the cost.

In order to determine which port should be a root port, STP runs through the decision-making process below:

- Lowest root bridge ID
- Lowest root path cost to the root bridge
- Lowest sender bridge ID
- Lowest sender port ID

The primary factor in deciding the root bridge is the root bridge ID and in deciding the

root ports on the non-root bridges is the path cost, which is cumulative (i.e., each path cost is added to the frame as it traverses the network). If there is a tie in the path cost, then STP moves down the list in order to make a decision, finally coming to the sender bridge ID and the port ID.

In the network below, Switch B's and Switch C's fa0/15 ports will be the root ports. Switch D has two options—fa0/17 toward Switch B and fa0/20 toward Switch C. The total cost of the link on fa0/17 is 200 (10 Mbps = 100). The total cost of the link on fa0/20 is 119 (10 Mbps = 100 and 100 Mbps = 19), so fa0/20 will be the root port for Switch D and fa0/17 will be blocked. For default costs, see Table 10-1. Costs are cumulative, adding up in the BPDU cost field as the frame traverses the network.

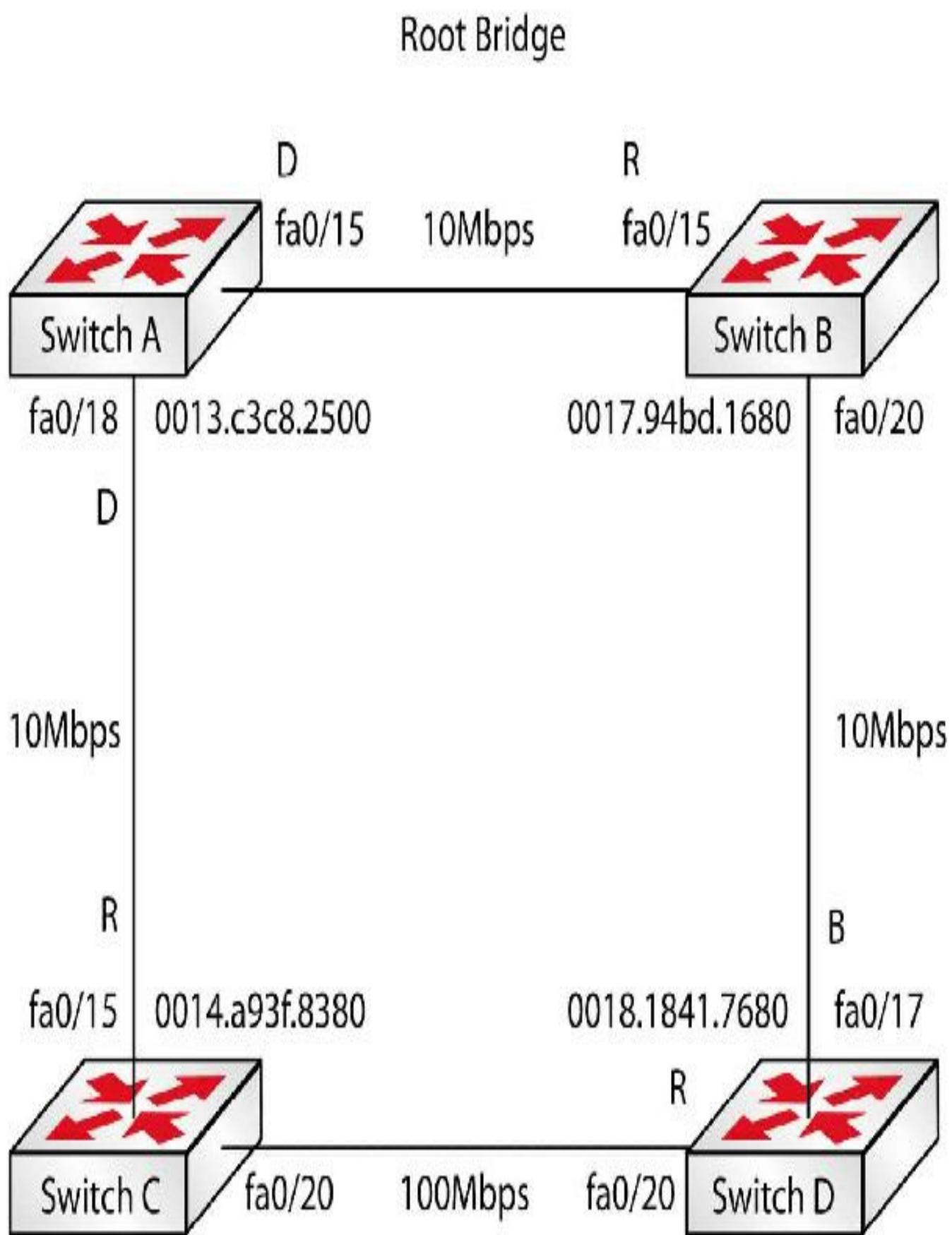


FIG 10.11 – STP port statuses

Let's verify Switch D's root port using the show spanning-tree [vlan#] command:

```
SwitchD#show spanning-tree vlan 5
VLAN0005
  Spanning tree enabled protocol ieee
  Root ID Priority  32773
    Address  0013.c3e8.2500
    Cost      119
    Port      20 (FastEthernet0/20)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID Priority  32773 (priority 32768 sys-id-ext 5)
    Address  0018.1841.7680
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
<hr/>					
fa0/17	Altn	BLK	100	128.17	P2p
fa0/20	Root FWD	19		128.20	P2p

You can see the cumulative cost is 119, which is the 100 Mbps link (19) plus the 10 Mbps link (100) (i.e., the interface port costs). The port priority always defaults to 128 plus the interface number. P2P indicates a point-to-point connection.

If you want to make fa0/17 on Switch D a root port instead of fa0/20, then you will need to change the cost on fa0/17 so that the total cost would be something better (less) than the current cost of 119, which you can see in the output above. To do this, you need the cost of interface fa0/17 to be less than 19. This can be adjusted using the spanning-tree cost # interface-level command (or spanning-tree vlan # cost # command if you just want to affect one VLAN, which is preferred) and can be used on fa0/17, as shown in the output below:

```
SwitchD(config)#int fa0/17
SwitchD(config-if)#spanning-tree vlan 5 cost 1
SwitchD(config-if)#do show spanning-tree vlan 5
The “do” command lets you issue a show command while in config mode
[output truncated]
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
<hr/>					
fa0/17	Root	LIS	1	128.17	P2p

fa0/20 Altn BLK 19 128.20 P2p

You can see that fa0/17 becomes the root port with a cost of 1 (as specified in the command) and fa0/20 goes into a blocking (BLK) state. Notice that fa0/17 is in the listening (LIS) state because it needs to transition through the listening and learning states, and the port can spend up to 15 seconds in each of these states.

Elect Designated Ports

If a switch has redundant ports connecting it to a LAN segment (another downstream switch or hub, for example), then the port with the lowest cost will be elected the designated port. Designated ports forward BPDUs onto the LAN segment and traffic to and from the LAN segment. In simple terms, the designated port becomes the only link for the LAN segment toward the rest of the network and the root bridge. By default, all ports on a root switch are designated ports. The criteria used for electing the designated port on a segment are listed in order below:

- Lowest root bridge ID
- Lowest root path cost to the root bridge
- Lowest sender bridge ID
- Lowest sender port ID

In Figure 10.11 above, the fa0/20 port on Switch C will be the designated port for the link to Switch D. This is because Switch C has the lowest cost to the root bridge. The root bridge is the same and Switch C has a better path cost to the root bridge (100) versus Switch D's path cost (200). (Switch D's Fa0/17 interface has been reverted to its normal cost.) If there were multiple links, then an election would have taken place. Let's verify this on Switch C:

```
SwitchC#show spanning-tree vlan 5
VLAN0005
  Spanning tree enabled protocol ieee
  Root ID Priority  32773
    Address 0013.c3e8.2500
    Cost 100
    Port 15 (Fasthernet0/15)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)
    Address 0014.a93f.8380
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
  Interface      Role Sts Cost    Prio.Nbr Type

```

fa0/15	Root FWD 100	128.15 P2p
fa0/20	Desg FWD 19	128.20 P2p

In summary, all root ports forward information to the root bridge and designated ports send traffic away from the root bridge. Any remaining ports will be non-designated and set to the blocking state. Blocking ports listen to configuration BPDUs but do not send or forward them.

It's worth noting that on larger networks you could have multiple switch segments separated by routers, and for each of these segments you would have one root bridge with all ports in designated (forwarding) state and neighbor switches with one root port.

Mini-lab – STP Operations

You can put much of what you have learned so far into context with this mini-lab. First, you'll use two switches connected with two Fast Ethernet links. You've already learned how to make ports access ports and put them into a VLAN so I'll skip these steps. All ports are in VLAN 5.

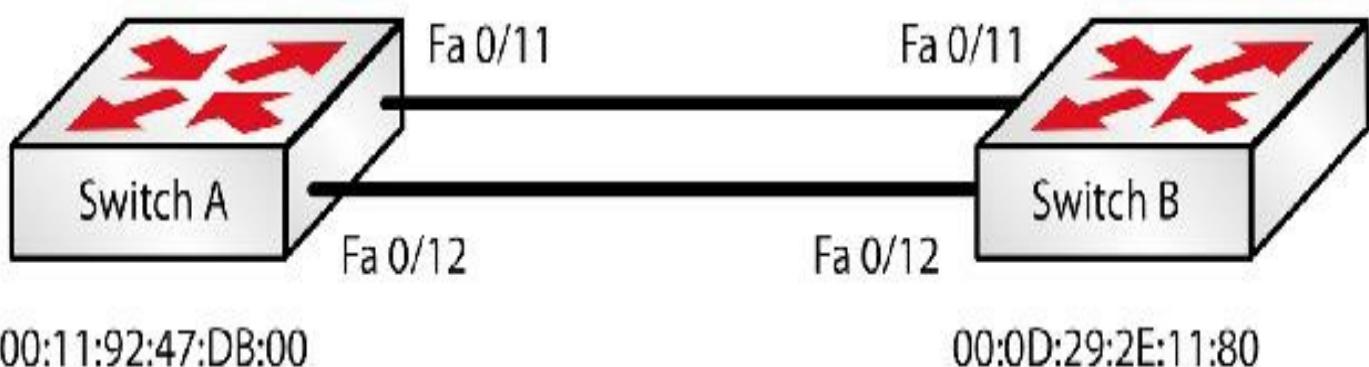


FIG 10.12 – Mini-lab: STP Operations

In Figure 10.12 above, you can see that Switch B has a lower base MAC address (you must choose the switch with the lowest MAC address as your Switch B) so it would be chosen as the root bridge. The root bridge should have all ports as designated/forwarding (which is another way to tell if a switch is the root bridge, by the way).

SwitchB#show spanning-tree vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 32773

Address 000d.292e.1180

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)

Address 000d.292e.1180

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

Fa0/11	Desg	FWD	19	128.11	P2p
--------	------	------------	----	--------	-----

Fa0/12	Desg	FWD	19	128.12	P2p
--------	------	------------	----	--------	-----

So far, you can see that Switch B is in fact the root bridge. Now check Switch A.

SwitchA#show span vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 32773

Address 000d.292e.1180

Cost 19

Port 11 (FastEthernet0/11)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)

Address 0011.9247.db00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

Fa0/11	Root	FWD	19	128.11	P2p
--------	-------------	------------	----	--------	-----

Fa0/12	Altn	BLK	19	128.12	P2p
--------	------	-----	----	--------	-----

One port is set to root and forwarding. You can see F0/12 is blocking. It's also set to alternate (Altn), which means that it's an alternative path between two switches. There

must have been a tie as STP went down the list:

- Lowest root bridge ID
- Lowest root path cost to the root bridge
- Lowest sender bridge ID
- Lowest sender port ID

The root bridge ID would have been identical, as with the path cost and sender bridge ID. This leaves the port ID. STP would have checked the MAC address of both ports and set the lowest to forward. You can confirm this by checking the interface MAC addresses.

SwitchA#show int f0/11

FastEthernet0/11 is up, line protocol is up (connected)

 Hardware is Fast Ethernet, address is 0011.9247.db0b (bia 0011.9247.db0b)

SwitchA#show int f0/12

FastEthernet0/12 is up, line protocol is up (connected)

 Hardware is Fast Ethernet, address is 0011.9247.db0c (bia 0011.9247.db0c)

The MAC addresses have been allocated sequentially, F0/11 is lower as it ends in b as opposed to c for F0/12.

I'd like you to do this lab on your own equipment. Obviously, your Switch B may not be the root because it might have a higher MAC address, so check first which one is the root and then name it Switch B or just force it to be the root with one of the two command options you already know.

First, enable debugs for STP events and then add timestamps on the debug messages (these commands probably won't work on Packet Tracer):

SwitchA#debug spanning-tree events

SwitchA(config)#service timestamps debug datetime msec

Next, shut the root port on Switch A to force the timers to start:

SwitchA#conf t

Enter configuration commands, one per line. End with CNTL/Z.

SwitchA(config)#int f0/11

SwitchA(config-if)#shut

*Mar 1 00:29:34.627: STP: VLAN0005 new root port Fa0/12, cost 19

*Mar 1 00:29:34.627: STP: VLAN0005 Fa0/12 -] listening

00:29:36: %LINK-5-CHANGED: Interface FastEthernet0/11, changed state to administratively down

*Mar 1 00:29:36.627: STP: VLAN0005 sent Topology Change Notice on Fa0/12

00:29:37: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to down

SwitchA(config-if)#end

SwitchA#show span vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 32773

Address 000d.292e.1180

Cost 19

Port 12 (FastEthernet0/12)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)

Address 0011.9247.db00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 15

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/12	Root	LRN	19	128.12	P2p

You can see the debug messages appear and that F0/12 has gone into the learning (LRN) state. The debug messages below show you the learning state move into forwarding.

SwitchA#

*Mar 1 00:29:34.627: STP: VLAN0005 Fa0/12 -] learning

*Mar 1 00:30:04.627: STP: VLAN0005 Fa0/12 -] forwarding

SwitchA#show span vlan 5

VLAN0005

Spanning tree enabled protocol ieee

Root ID Priority 32773

```

Address 000d.292e.1180
Cost    19
Port    12 (FastEthernet0/12)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32773 (priority 32768 sys-id-ext 5)
Address 0011.9247.db00
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Role Sts Cost Prio.Nbr Type
-----
```

Fa0/12 Root FWD 19 128.12 P2p

The total time from blocking at 00:29:34.627 to learning was 00:30:04.627. The counter shows 29 minutes and 34 seconds past midnight to 30 minutes and 4 seconds, so the entire process took 30 seconds. It was faster than the usual 50 seconds because it was a very simple topology with no switches in between.

[END OF MINI-LAB]

STP Timers

We have briefly mentioned these but it's worth looking at STP timers separately.

Three timers monitor and age BPDUs:

- Hello
- Forward delay
- Max age

Although you shouldn't change these without advice from a Cisco TAC, I'll demonstrate how to below.

- Hello – sent by the root bridge every two seconds by default
 Switch(config)#spanning-tree vlan 1 Hello-time ?
 <1-10> number of seconds between generation of config BPDUs
- Forward delay – the default 15 seconds switches wait while they build their bridging table (the listening and learning states each use this 15 second timer)
 Switch(config)#spanning-tree vlan 1 forward-time ?
 <4-30> number of seconds for the forward delay timer

- Max age – how long a BPDU is stored before it is flushed from the table (a new BPDU should be received every two seconds; this timer is set to 20 seconds (Hello interval multiplied by 10) and if it is reached it usually indicates a link failure); the interface then moves to the listening state

Switch(config)#spanning-tree vlan 1 max-age ?

<6-40> maximum number of seconds the information in a BPDU is valid

The total time for STP to recover from a link failure is 20 seconds max age plus 15 seconds listening plus 15 seconds learning, which is 50 seconds to recovery.

Switch#show spanning-tree
detail

VLAN0001 is executing the ieee compatible Spanning Tree Protocol

Bridge Identifier has priority 32768, sysid 1, address 0011.9247.db00

Configured Hello time 2, max age 20, forward delay 15

We are the root of the spanning tree

[output truncated]

Cisco's Enhancements to STP

STP, as we know, keeps the network loop-free but reconvergence can take up to 50 seconds. That is a very long time in networking terms. For almost a minute, data cannot flow across the network. In most cases this is a critical issue, especially for important network services.

To deal with this issue (before the industry standard for Rapid STP was ratified), Cisco added the following features to STP implementation on its switches:

- PortFast
- UplinkFast
- BackboneFast

PortFast

PortFast is typically enabled on an interface connected directly to a host. If you have a laptop or a server connected to a switch port, then you know that:

- It will not need to listen to BPDUs because it is not a layer 2 device
- It will not create loops because it has a single link to the layer 2 network

Therefore, you can safely disable Spanning Tree on such ports. It is very important to ensure that such ports never have an STP-enabled layer 2 device connected on them

(think port security!), or else a loop or a breakdown of the network is quite possible. You will even get a warning message on certain switches stating this when you enable PortFast on a switch port!

When you configure a switch port as PortFast, STP will skip the listening and learning states and the port will transition to forwarding state when it comes up so it will never be blocked. Other manuals state that STP is disabled on a port using PortFast; however, this is not the case because the port can still send and forward BPDUs. This is not a problem when the port is connected to a network device that does not send or respond to BPDUs, such as the NIC in a workstation, for example. However, this may result in a switching loop if the port is connected to a device that does send BPDUs, such as another switch.

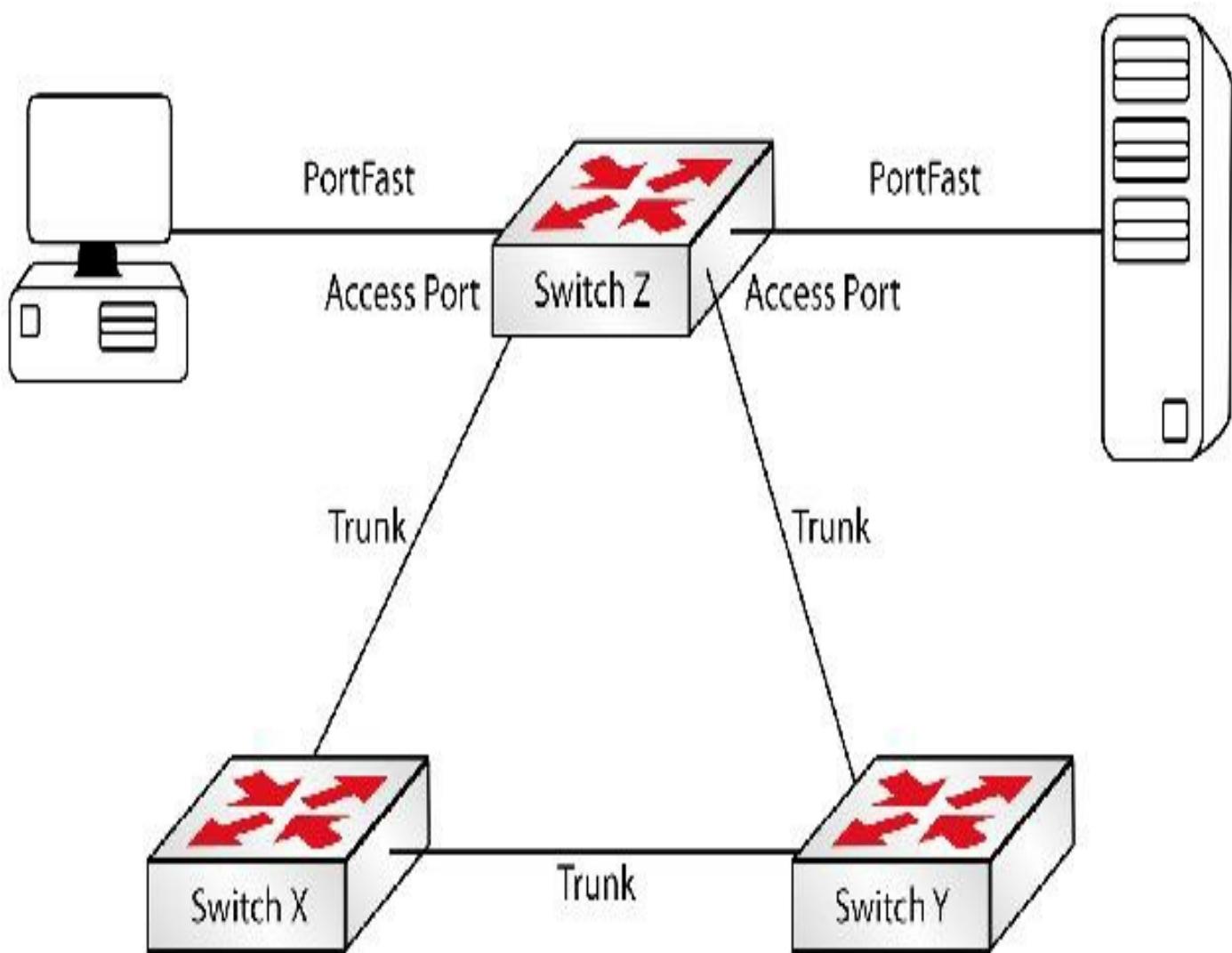


FIG 10.13 – PortFast

The command to configure PortFast is `spanning-tree portfast`. Note the system-generated warning message:

```
SwitchA(config)#int FastEthernet0/44
```

```
SwitchA(config-if)#spanning-tree portfast
```

%Warning: PortFast should only be enabled on ports connected to a single host.

Connecting hubs, concentrators, switches, bridges, etc... to this interface when PortFast is enabled can cause temporary bridging loops. Use with CAUTION

%PortFast has been configured on FastEthernet0/44 but will only take effect when the interface is in a nontrunking mode.

UplinkFast

The purpose of UplinkFast is to optimize convergence when an uplink on an access layer switch fails. Let's consider the network shown in Figure 10.14 below:

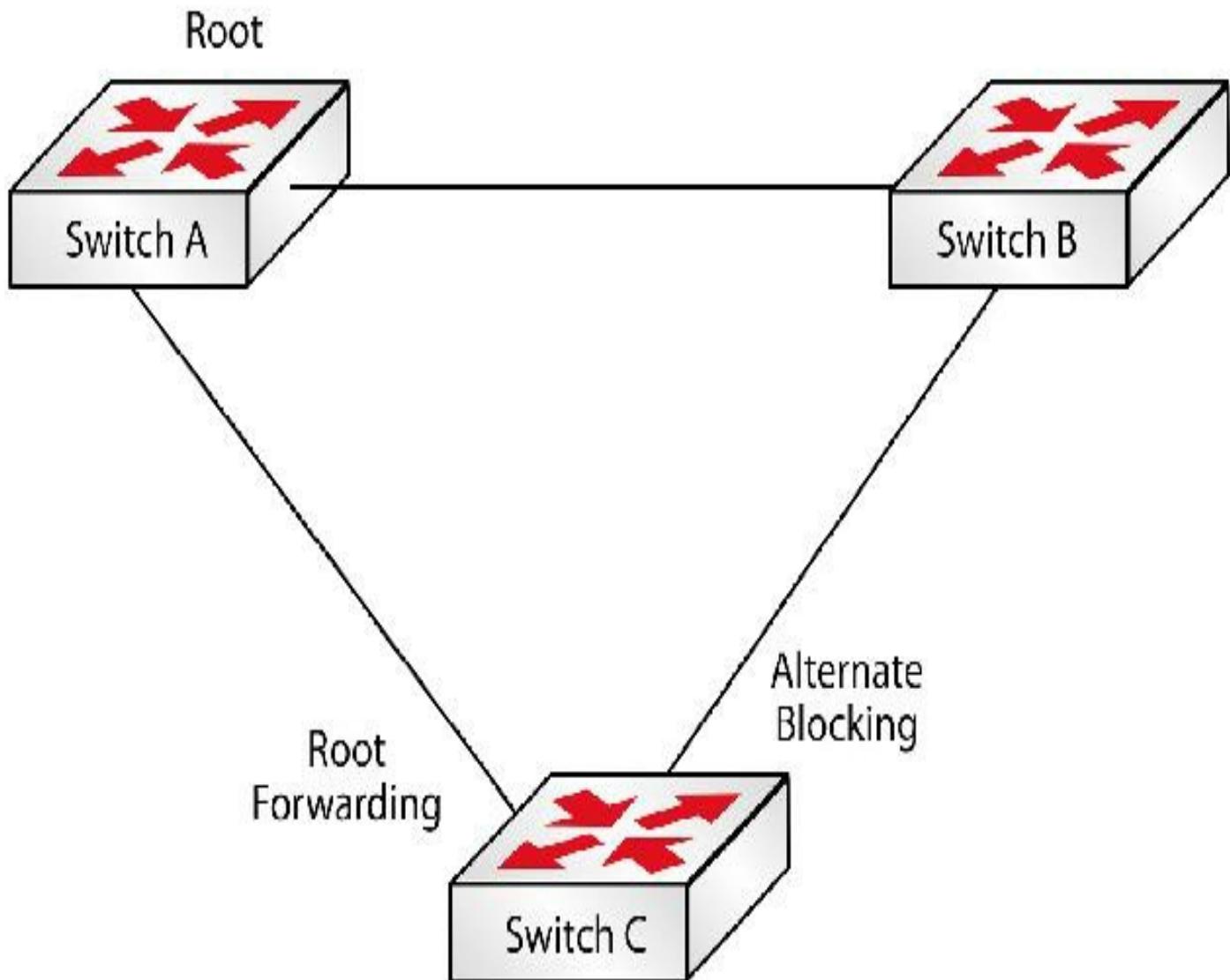


FIG 10.14 – Redundant links to the root bridge

If the link between Switch C and Switch A fails for some reason, UplinkFast will almost immediately transition the alternate port to the forwarding state as per Figure

10.15 below:

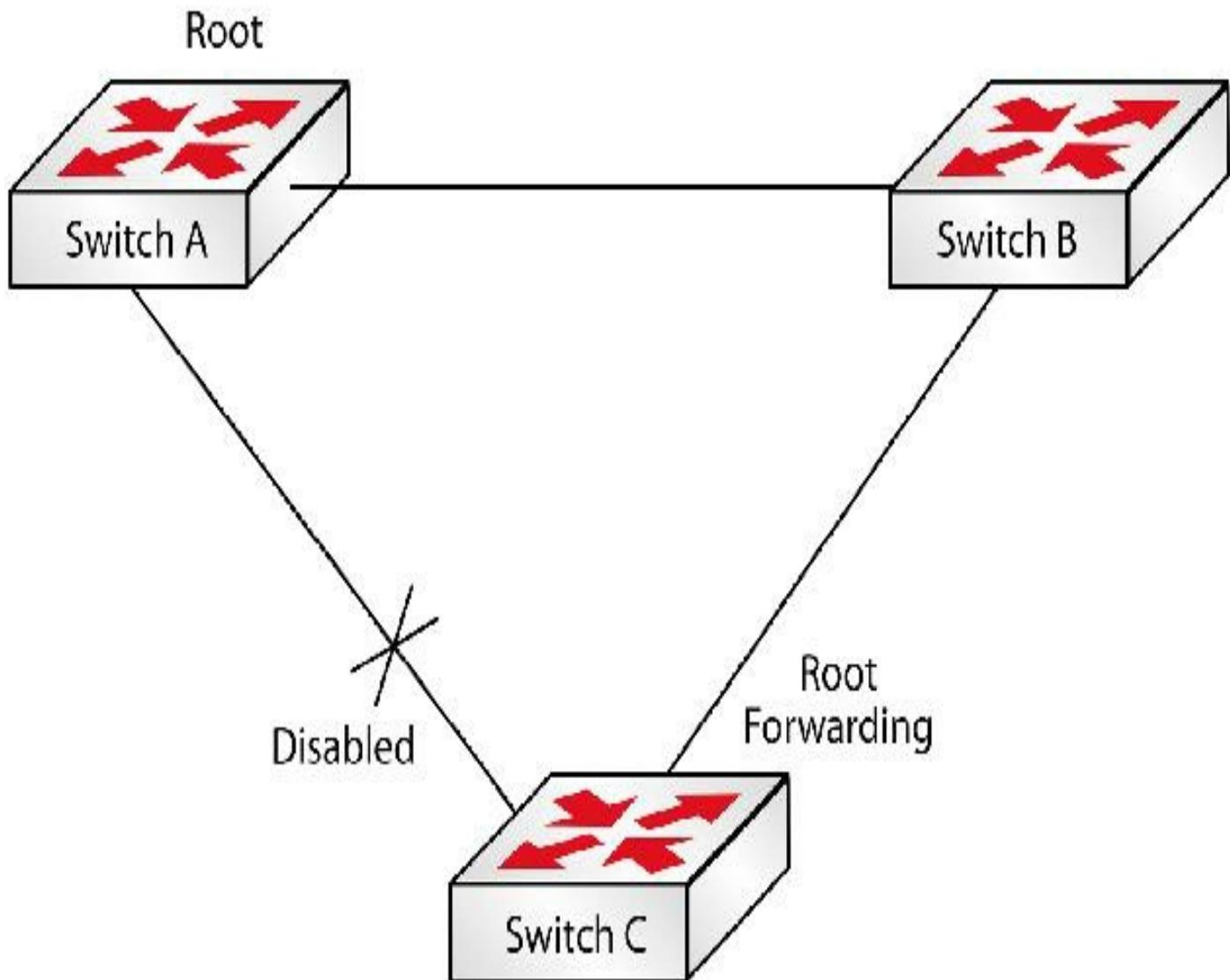


FIG 10.15 – UplinkFast detects disabled port

Mini-lab – Configuring UplinkFast

You will need three switches for this lab. Set all ports as access ports in VLAN 5. Of course, you might have a different root from mine due to MAC addressing. You know how to set the switch to be the root so feel free to do this. The network in Figure 10.16 below does not have UplinkFast enabled yet.

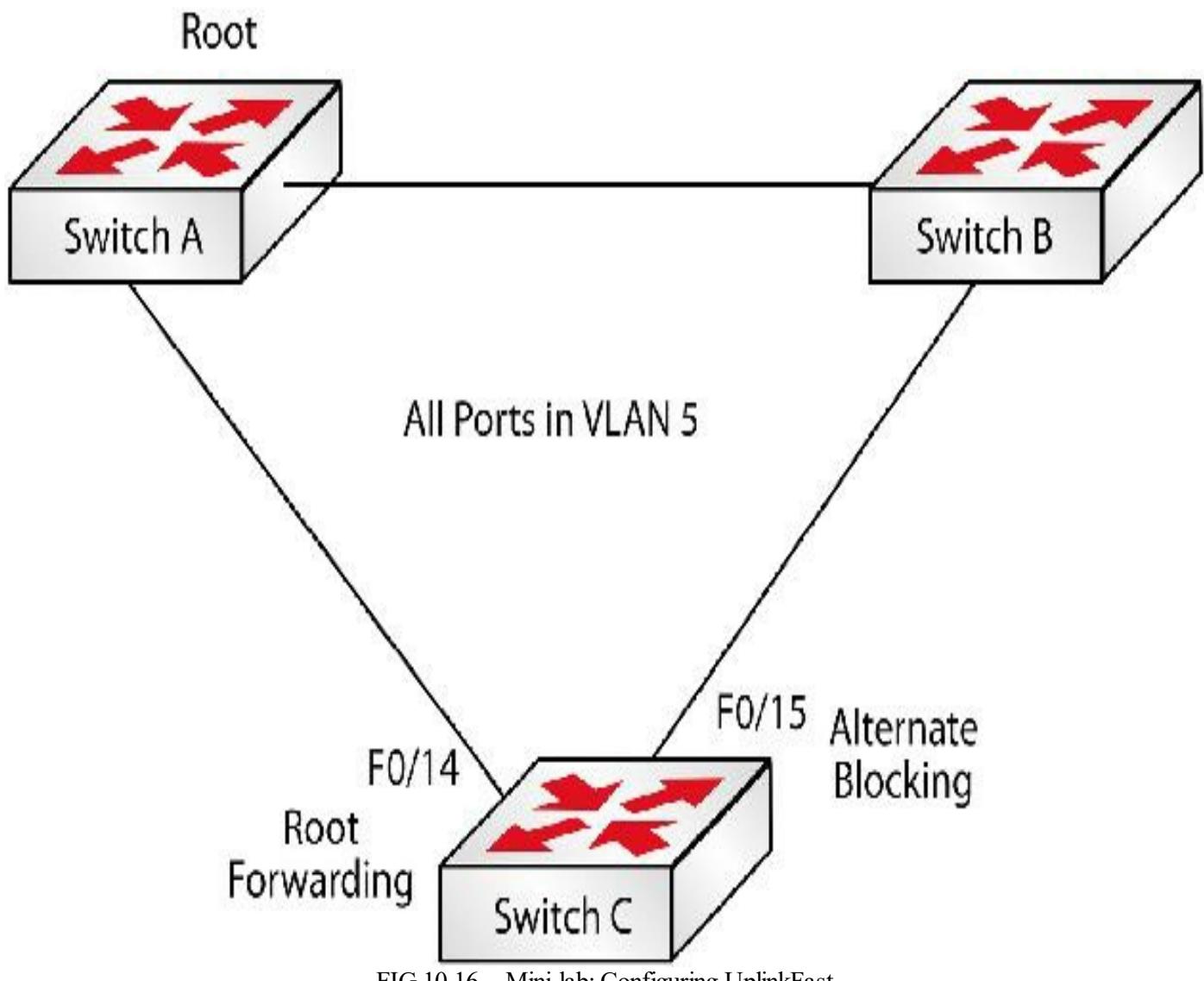


FIG 10.16 – Mini-lab: Configuring UplinkFast

In Figure 10.16 above, Switch A is the root bridge. Now consider the following output from Switch C:

```

SwitchC#show spanning-tree vlan 5
VLAN0005
  Spanning tree enabled protocol ieee
  Root ID Priority  32773
    Address  0013.c3e8.2500
    Cost      19
    Port     14 (FastEthernet0/14)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID Priority  32773 (priority 32768 sys-id-ext 5)
    Address  0017.94bd.1680
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

```

Aging Time 300

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/14	Root	FWD	19	128.14	P2p	
Fa0/15	Altn	BLK	19	128.15	P2p	

SwitchC#show spanning-tree uplinkfast
UplinkFast is disabled

Use the following debug commands on the switch:

```
SwitchC#debug spanning-tree event  
Spanning Tree event debugging is on  
SwitchC#debug spanning-tree uplinkfast  
Spanning Tree uplinkfast debugging is on
```

These debugs will show you STP events and UplinkFast messages. They probably won't work on Packet Tracer. Now shut down port fa0/14 on Switch C, which is currently the root port as per the output above.

Because UplinkFast brings up the alternate port so quickly, enable milliseconds on the debugs with the service timestamps debug datetime msec command in global configuration mode:

```
SwitchC(config-if)#shutdown  
*Mar 2 22:14:30.504: STP: VLAN0005 new root port Fa0/15, cost 19  
*Mar 2 22:14:30.504: STP: VLAN0005 Fa0/15 -] listening  
*Mar 2 22:14:30.504: STP: UFAST: removing prev root port Fa0/14 VLAN0005 port-id 800E  
*Mar 2 22:14:32.420: %LINK-5-CHANGED: Interface FastEthernet0/14, changed state to administratively down  
*Mar 2 22:14:32.504: STP: VLAN0005 sent Topology Change Notice on Fa0/15  
*Mar 2 22:14:33.420: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed state to down  
*Mar 2 22:14:45.504: STP: VLAN0005 Fa0/15 -] learning  
*Mar 2 22:15:00.504: STP: VLAN0005 Fa0/15 -] forwarding
```

Note that the time taken for F0/15 to transition to the forwarding state is 30 seconds. This is faster than the expected 50 seconds because the listening and learning times were short in this P2P link between switches and no other hosts/switches are connected here.

Next, enable no shutdown on the F0/15 port and then enable uplinkfast on Switch C and repeat the process:

```
SwitchC(config)#spanning-tree uplinkfast
```

```
SwitchC(config)#exit
```

```
SwitchC#show spanning-tree vlan 5
```

[output truncated]

Uplinkfast enabled

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/14	Root	FWD	3019	128.14		P2p
Fa0/15	Altn	BLK	3019	128.15		P2p

```
SwitchC(config)#int fa0/14
```

```
SwitchC(config-if)#shutdown
```

*Mar 2 22:28:23.300: STP: VLAN0005 new root port Fa0/15, cost 3019

*Mar 2 22:28:23.300: STP FAST: UPLINKFAST: make_forwarding on VLAN0005

FastEthernet0/15 root port id new: 128.15 prev: 128.14

*Mar 2 22:28:23.300: %SPANTREE_FAST-7-PORT_FWD_UPLINK: VLAN0005

FastEthernet0/15 moved to Forwarding (UplinkFast).

*Mar 2 22:28:23.300: STP: UFAST: removing prev root port Fa0/14 VLAN0005 port-id 800E

*Mar 2 22:28:25.216: %LINK-5-CHANGED: Interface FastEthernet0/14, changed state to administratively down

*Mar 2 22:28:25.300: STP: VLAN0005 sent Topology Change Notice on Fa0/15

*Mar 2 22:28:26.216: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed state to down

```
SwitchC(config-if)#do show spanning-tree vlan 5
```

[output truncated]

Uplinkfast enabled

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/15	Root	FWD	3019	128.15		P2p

Note that the time taken for fa0/15 to transition to the forwarding state has changed from 30 seconds downtime to less than a second with UplinkFast enabled. Now that you have

seen the difference it makes, let's define what exactly it does.

[END OF MINI-LAB]

If a switch has multiple links toward the root bridge, then UplinkFast marks the redundant link as an alternate port and brings it up quickly in case the root port fails. This is possible because blocked ports keep listening for BPDUs.

When you enable UplinkFast on a switch globally (rather than per port), the switch does three things:

1. Increases the root priority to 49152
2. Sets the port costs to 3000
3. Tracks alternate root ports (ports on which root Hellos are received)

You can see this in the outputs below (which are truncated to save space). Bear in mind that you have to add the VLAN # as well as the original port cost, which for Fast Ethernet is 19.

Cisco recommends caution when using UplinkFast. You should enable it on switches that have blocked ports so the access layer switch does not become a root or transit switch (one that forwards frames between other switches). Note that your switch was the root for VLAN 1, but after enabling UplinkFast this is no longer the case. The large root priority value coupled with the large costs per link make this switch unlikely to become the root.

```
Switch#show spanning-tree vlan 1
```

```
VLAN0001
```

```
  Spanning tree enabled protocol ieee
```

```
  Root ID Priority 32769
```

```
    Address 0017.0e31.d180
```

This bridge is the root

```
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
  Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
```

```
    Address 0017.0e31.d180
```

```
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
    Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

```

Fa0/1      Desg FWD 19    128.1  P2p
Fa0/3      Desg FWD 19    128.3  P2p
Switch(config)#spanning-tree uplinkfast
Switch(config)#end
Switch#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID Priority  32769
    Address  000c.3018.3700
    Cost      3019
    Port      7 (FastEthernet0/7)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

```

```

Bridge ID Priority  49153 (priority 49152 sys-id-ext 1)
  Address  0017.0e31.d180
  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Aging Time 300

```

Uplinkfast enabled

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

Fa0/1	Desg	FWD	3019	128.1	P2p
Fa0/3	Desg	FWD	3019	128.3	P2p

BackboneFast

UplinkFast works by finding alternate ports for directly connected links. Similarly, BackboneFast works by finding an alternate path when an indirect link to the root port goes down. The difference between these two processes is that the indirect link doesn't have the option of bypassing the max age timer. This time, the switch learns about a failure due to the lack of Hellos from other switches. If a failure is learned this way, the switch has to wait for the max age timer to expire before trying to change the topology using STP.

BackboneFast allows any switch learning about an indirect failure to send a Root Link Query (RLQ) BPDU out of the port the Hello was expected on, asking the neighbor

switch if Hellos are still being received from the root. If the RLQ response states that there has been a direct link failure, it can converge and bypass the max age timer.

Let's consider the network in Figure 10.17 below:

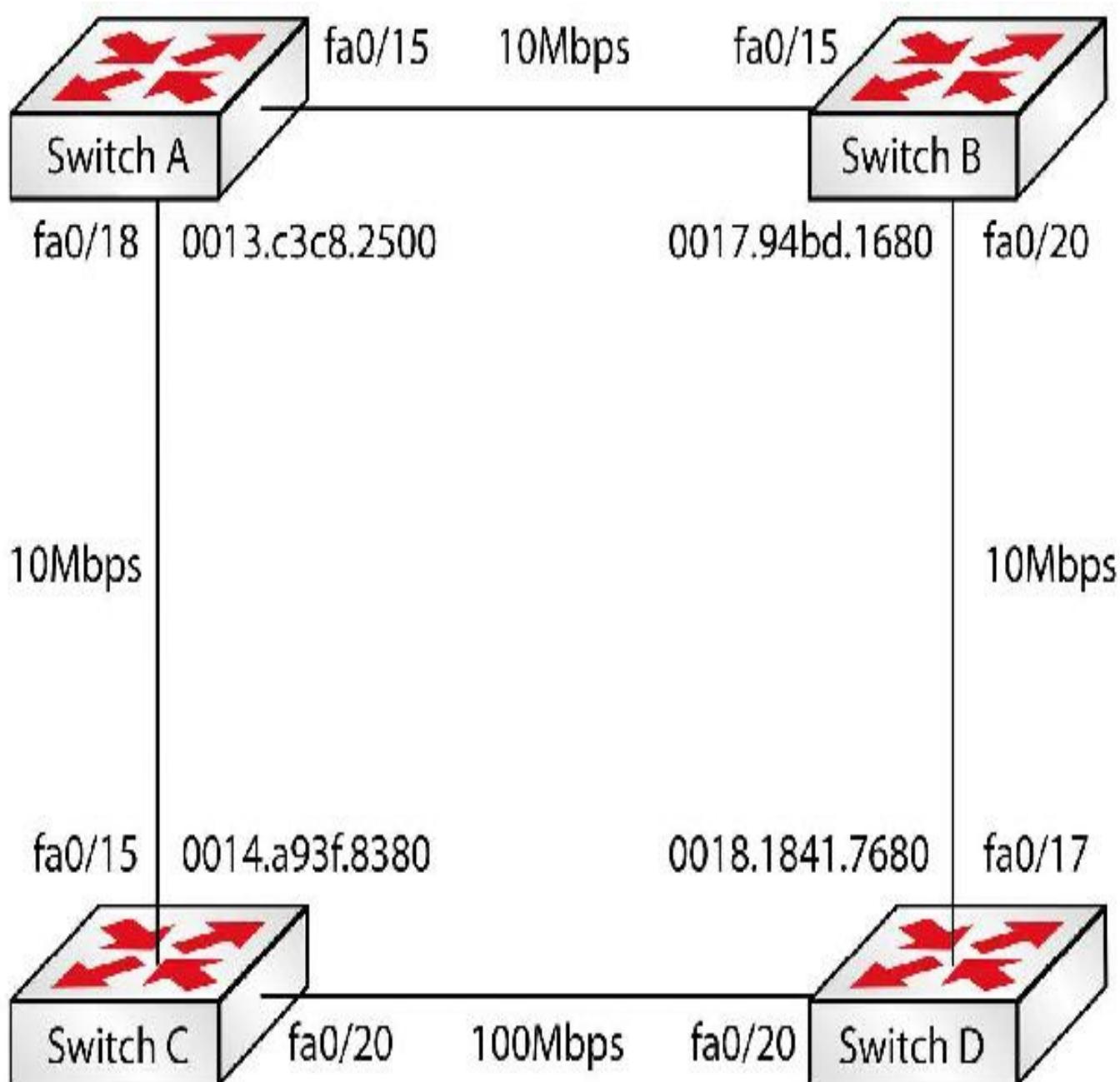


FIG 10.17 – Redundant path to the root bridge

Switch A is the root bridge in Figure 10.17. F0/20 on Switch D is the root port.

Let's assume that the link between Switch A and Switch C goes down. Switch C will advertise itself as the root bridge to Switch D. This BPDU is known as an inferior BPDU. Switch B discards this new information since it knows that Switch A is the root bridge and Switch C is a non-root bridge. Eventually, Switch C will receive a BPDU from Switch D and mark F0/20 as its root port toward Switch A. BackboneFast ensures

a quick failover as soon as the inferior BPDU is received. It saves roughly 20 seconds out of the 50 seconds of convergence time.

Configuring BackboneFast can be accomplished easily with the command below:

```
Switch(config)#spanning-tree backbonefast
```

BackboneFast must be enabled on all switches in order for this feature to work.

STP Security

As you have learned, PortFast disables STP on a switch port, but an important fact is that a PortFast switch port will keep listening for BDPUs. If someone adds a switch to a port that has been configured as PortFast, the consequences will be unpredictable and in some cases disastrous. To guard against this situation, Cisco provides the BPDU Guard, BPDU Filter, and Root Guard features.

BPDU Guard

If a switch is plugged into a switch port configured as PortFast, it could change the STP topology without the administrator knowing about it and could even bring down the network. To prevent this, BPDU Guard can be configured on the switch port. With this configured, if a BPDU is received on a switch port, it will be put into an err-disabled mode and an administrator will have to bring up the port. This can be configured on the port using the `spanning-tree bpduguard enable` command.

The administrator must recover this port via the command line by issuing a `shutdown` command and then a `no shutdown` command on the interface (i.e., bounce the interface). Until this is done, the status light on the port will show as amber and frames cannot pass.

BPDU Guard enabled
on interface. Port is
err-disable when
BPDU is received

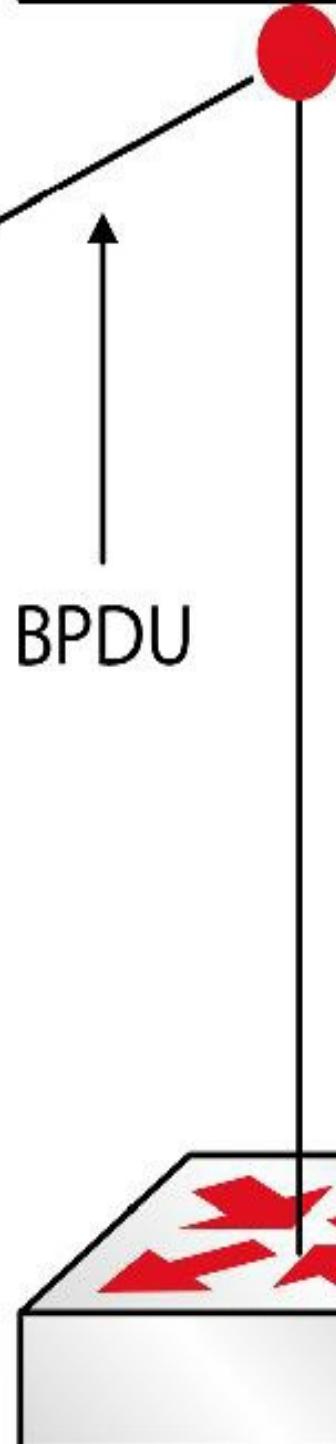
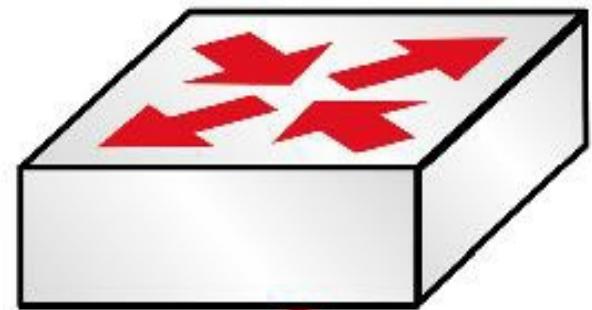


FIG 10.18 – BPDU Guard

Mini-lab – Configuring BPDU Guard

In Figure 10.19 below, I've connected a PC to F0/1 on Switch 0. You can configure any IP address on your PC connected to your switch.

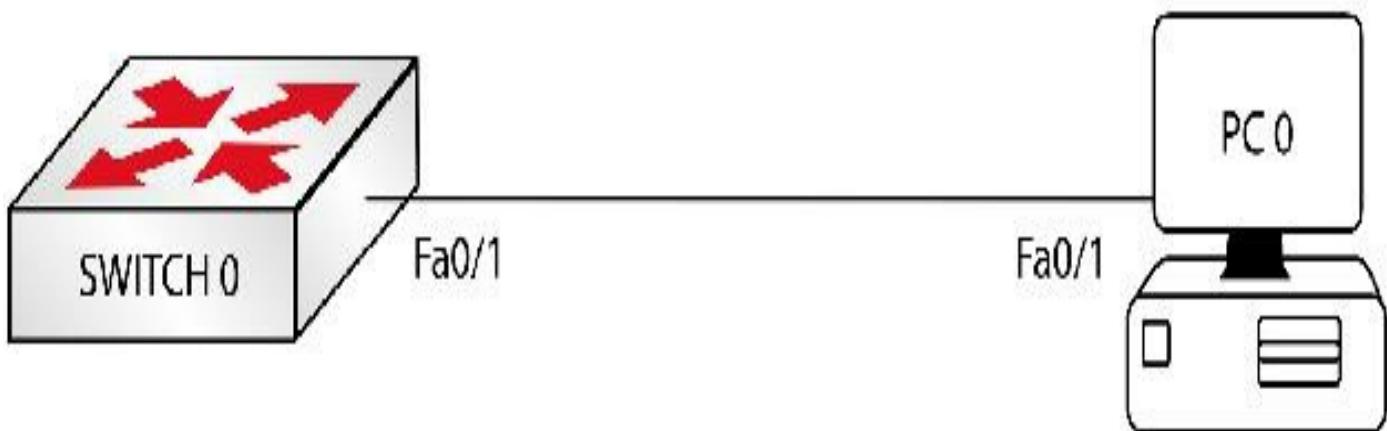


FIG 10.19 – Mini-lab: Configuring BPDU Guard

```
Switch0#config t
Switch0(config)#int f0/1
Switch0(config-if)#switchport mode access
Switch0(config-if)#spanning-tree bpduguard enable
Switch0(config-if)#end
Switch0#show int f0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Switch0#show run int f0/1
Building configuration...
Current configuration : 89 bytes
interface FastEthernet0/1
switchport mode access
spanning-tree bpduguard enable
```

end

The port will operate normally until I swap the PC for another switch, which sends a BPDU causing Switch 0 to shut interface F0/1:

%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port FastEthernet0/1 with BPDU Guard enabled. Disabling port.

%PM-4-ERR_DISABLE: bpduguard error detected on 0/1, putting 0/1 in err-disable state

Switch0#show int f0/1

FastEthernet0/1 is down, line protocol is down (**err-disabled**)

Hardware is Lance, address is 00e0.a3b4.7601 (bia 00e0.

[END OF MINI-LAB]

BPDU Filter

When BPDU Filter is configured on a switch port that has been configured as PortFast, it will prevent the port from sending and receiving BPDUs on that port. This effectively disables STP on the port. This is unlike the behavior seen with BPDU Guard, where the port is put into an err-disabled mode. BPDU Filter can be enabled on the switch port using the spanning-tree bpdufilter enable command:

Switch(config-if)#spanning-tree bpdufilter enable

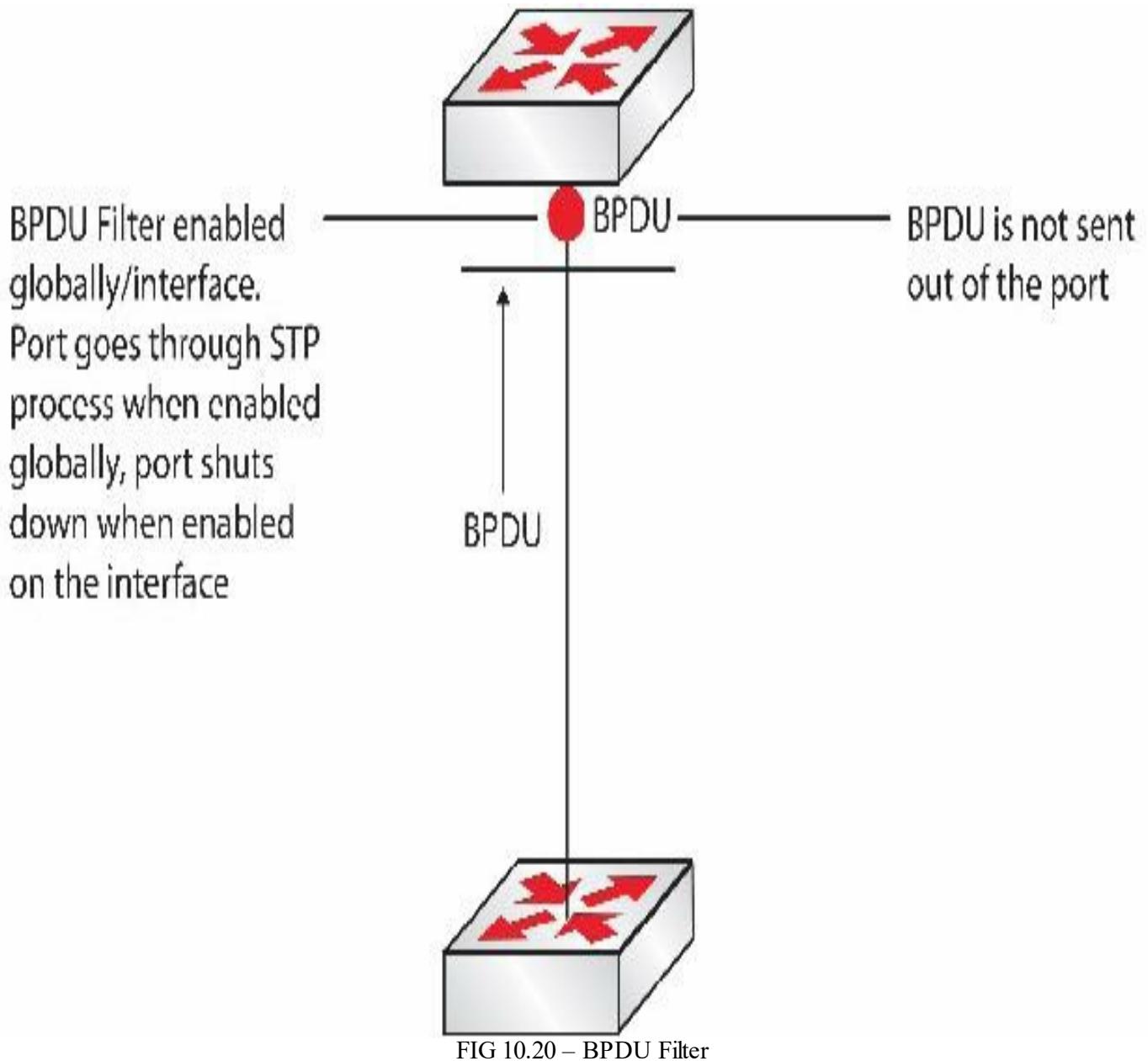


FIG 10.20 – BPDU Filter

Root Guard

Root Guard is configured per port and, as with BPDU Guard, it monitors for incoming BPDUs. Root Guard is designed to prevent the port from becoming a root port. If a superior BPDUs are received on the port, the port is placed into a root-inconsistent state, preventing it from forwarding or receiving frames until the superior BPDUs cease.

As mentioned, Root Guard is enabled on an interface. The command to configure it is shown below:

```
Switch(config-if)#spanning-tree guard root
```

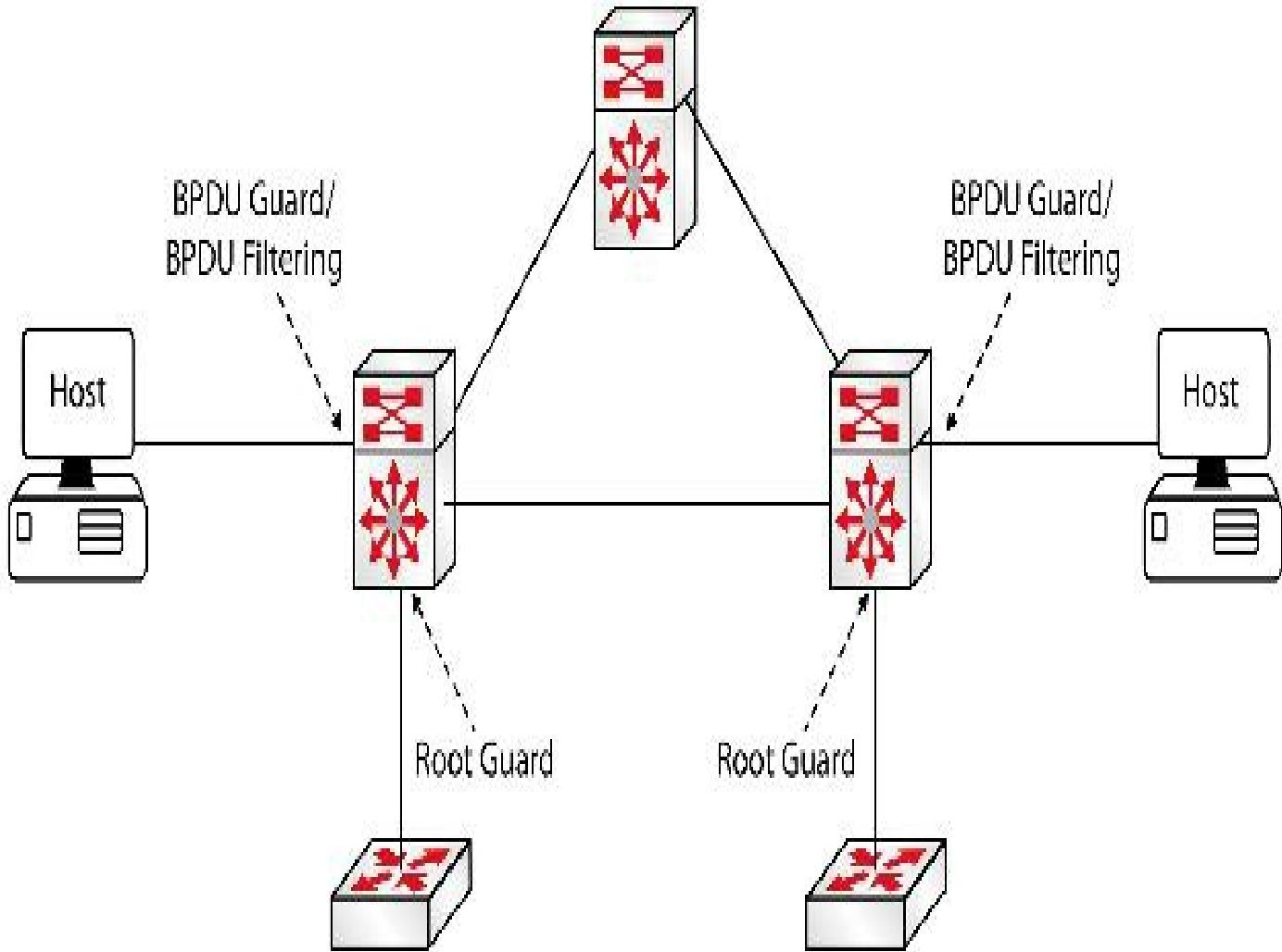


FIG 10.21 – Root Guard

Rapid Spanning Tree Protocol

The features discussed in the previous section—PortFast, UplinkFast, and BackboneFast—were added by Cisco, and because of this they worked only on Cisco switches. IEEE added these features to a new STP protocol called Rapid Spanning Tree Protocol (RSTP) under the (layer 2) 802.1W standard. The goal of RSTP is to improve STP convergence.

NOTE: People using a home lab and wanting to configure RSTP will need a 2960T Catalyst Switch as a minimum hardware requirement.

Similar to traditional Spanning Tree, RSTP will also elect a root bridge using the same parameters as STP. All RSTP ports will be in a forwarding state (designated ports), while other ports could be an alternate port, root port, backup port, or disabled. RSTP has defined variations of BPDUs, new port roles and states, and backward compatibility with 802.1D switches.

RSTP dramatically improves STP convergence times by using a few key concepts:

- Transitions from the discarding (rather than blocking) state to the learning state, thereby bypassing the listening state
- Integration and standardization of Cisco's PortFast, UplinkFast, and BackboneFast
- Waits for three missed Hellos on a root port instead of 10

RSTP has simplified the STP logic where possible, as well as defined link types and port roles, to speed up convergence times.

RSTP Link Types

802.1D was devised when shared hubs were in common usage on live networks. On modern networks, most links are point-to-point (switch-to-switch). The remaining link types must be connected to a host of some sort (edge), so PortFast logic would need to be applied here (ports set to forward immediately).

RSTP allows a switch to query its neighbor on point-to-point links to establish its status. It would do this, for example, if no periodic Hello was received. As with BackboneFast, the neighbor switch would respond stating whether it had lost its neighbor.

There are three RSTP link types:

- Point-to-point – switch-to-switch
- Shared – switch connects to a hub (other switches connect to the hub)
- Edge – a host device is connected (end-user)

RSTP Port Roles

As mentioned earlier, RSTP has defined new port roles, adding alternate and backup ports. Table 10-2 below lists these roles.

Table 10-2: RSTP port roles

Root Port	This elected port forwards data in the active topology.
Designated Port	This is an elected port that forwards data for every switched LAN segment.
Alternate Port	This is an alternative path to the root bridge but is different from the root port path.
Backup Port	This port provides a redundant path (less desirable) to a segment to which another switch port already connects. (They can only exist when there are two ports connected between the switches.)
Disabled	This type of port does not participate in the active topology.

There is a port type known as an edge port, which is considered to be the same as a port

configured with the spanning-tree portfast command. The root port and the designated port are the same as the 802.1D root port and designated port. The alternate port is similar to the UplinkFast concept of tracking alternate paths to the root, thus preventing the loss of a switch's root port.

The backup port is a new concept and its role is to prevent the loss of the designated port attached to a shared link when there is another physical port attached to the same shared LAN, as shown in Figure 10.22 below.

RSTP Port States

A comparison of the STP port states and the RSTP port states is shown in Table 10-3 below. You can see that the listening and disabled states have been removed from RSTP, and it has a new role of discarding.

Table 10-3: Comparison of STP port states and RSTP port states

Operational Status	STP Port State	RSTP Port State	Port in Active Topology
Enabled	Blocking/Disabled	Discarding	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

A discarding port does not forward or receive frames or learn source MAC addresses. As you can see, once a port is set to forward, it participates in the active topology and behaves in the same manner as an 802.1D port. The listening state is no longer required because RSTP actively queries its neighbors, thereby preventing any loop creation during convergence.

Figure 10.22 below illustrates the various port roles and states. You can see which switch is the root by the fact that all ports are set to designated (marked as DP).

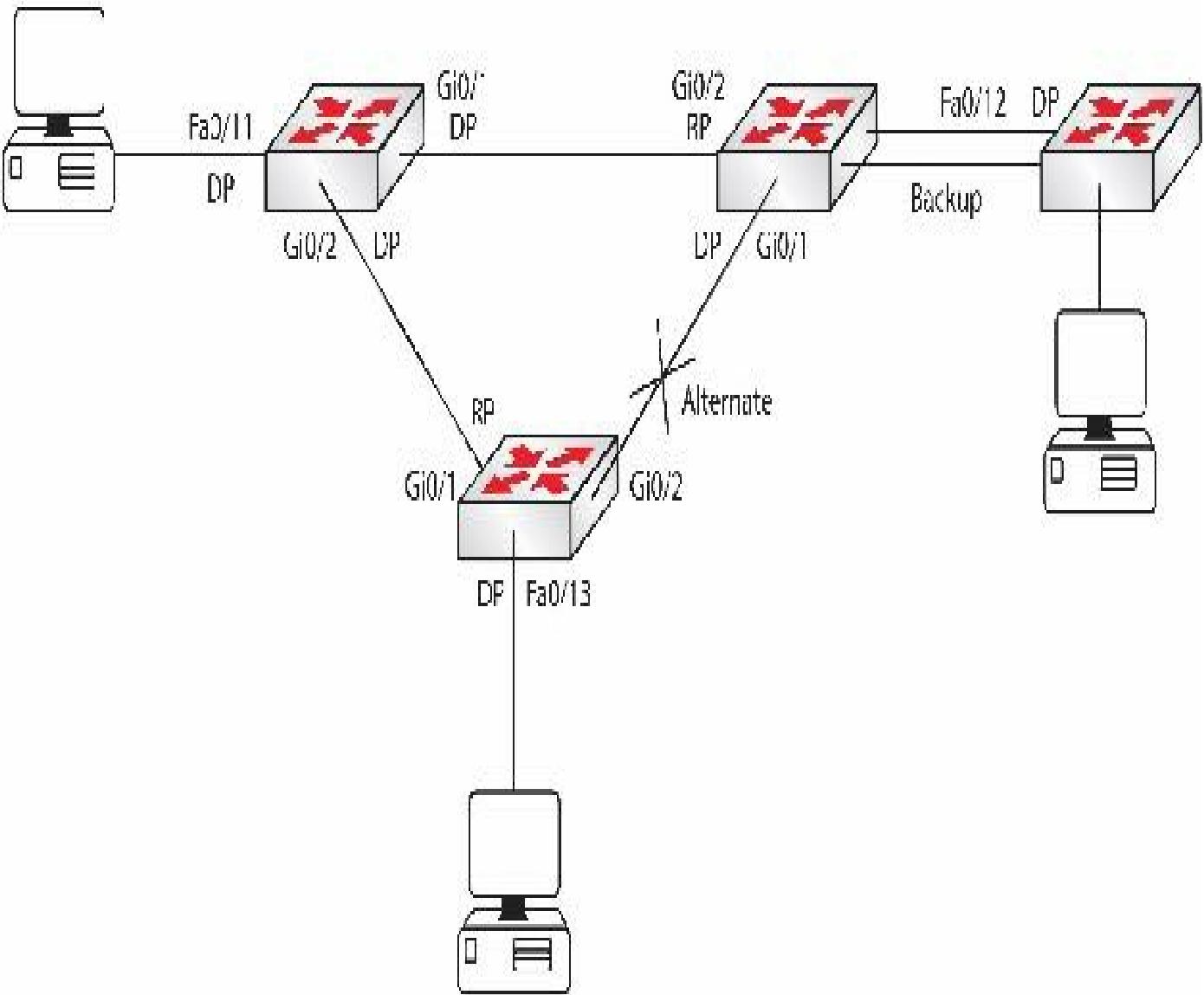


FIG 10.22 – RSTP port roles and states

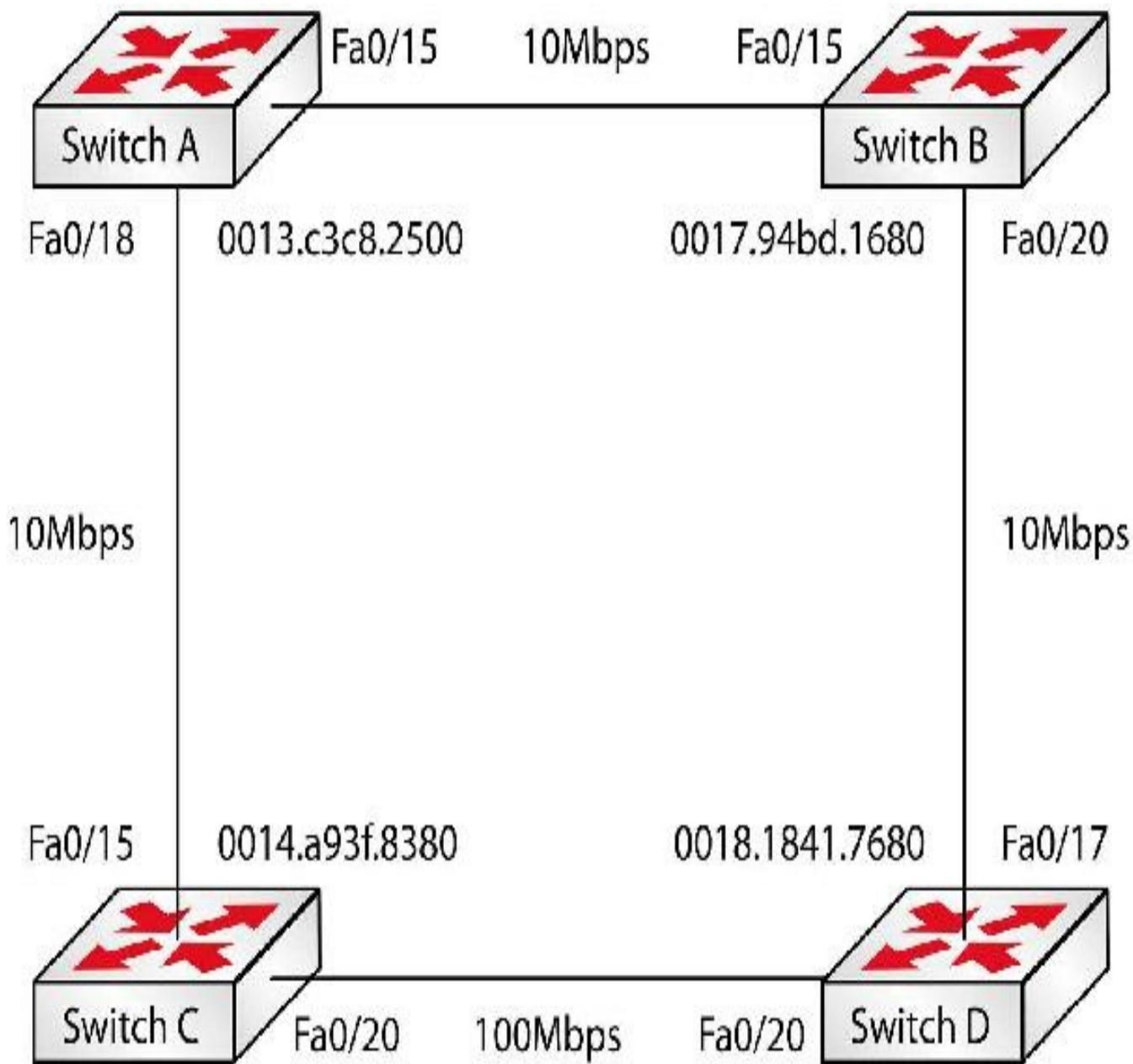
You can see in Figure 10.22 above that an alternate port has received a more useful BPDU from another switch on the same segment. This port will be put into the discarding state. The backup port has received a more useful BPDU from the same switch they are on. This is considered to be a backup port for the designated port on the same switch.

Per-VLAN STP and per-VLAN Rapid STP

This is a good time to introduce you to another very significant change that Cisco made to STP.

When the original bridging standard (802.1D) was drafted, VLANs did not exist. Hence, one Spanning Tree instance worked across the entire switch. Eventually, VLANs were introduced and they created different logical networks on the same switch. This gave rise to the need to have different topologies for load balancing and flexible Spanning

Trees. A strong reason for implementing Per-VLAN STP on a switch is for efficient utilization of the ports on the switch. This is illustrated with the following network:



Let's assume that all the switches have two VLANs configured. Switch D has two ways to reach Switch A. If one STP instance was running across the network, then fa0/17 would be in the blocked state. With two STP instances running, you can have fa0/20 blocked for one VLAN and fa0/17 blocked for another and utilize both links by load balancing traffic across them.

To achieve this, Cisco added the Per-VLAN Spanning Tree Plus (PVST+) feature on its switches. When 802.1W (RSTP) was introduced by the IEEE, it still did not accommodate multiple Spanning Tree instances on a switch. Cisco introduced Per-

VLAN Rapid Spanning Tree Plus (PVRST+) to support Rapid Spanning Tree instances on each VLAN on the switch. PVST+ and PVRST+ both provide the same functionality across both 802.1D and 802.1W standards. PVST+ has only three port states (discarding, learning, and forwarding), while STP has five port states (blocking, listening, learning, forwarding, and disabled).

Figure 10.24 below shows a simplified version of how this works. You can see the physical topology on the left and then the logical topologies on the right for two other VLANs. Each has a different root bridge and blocked and forwarding ports.

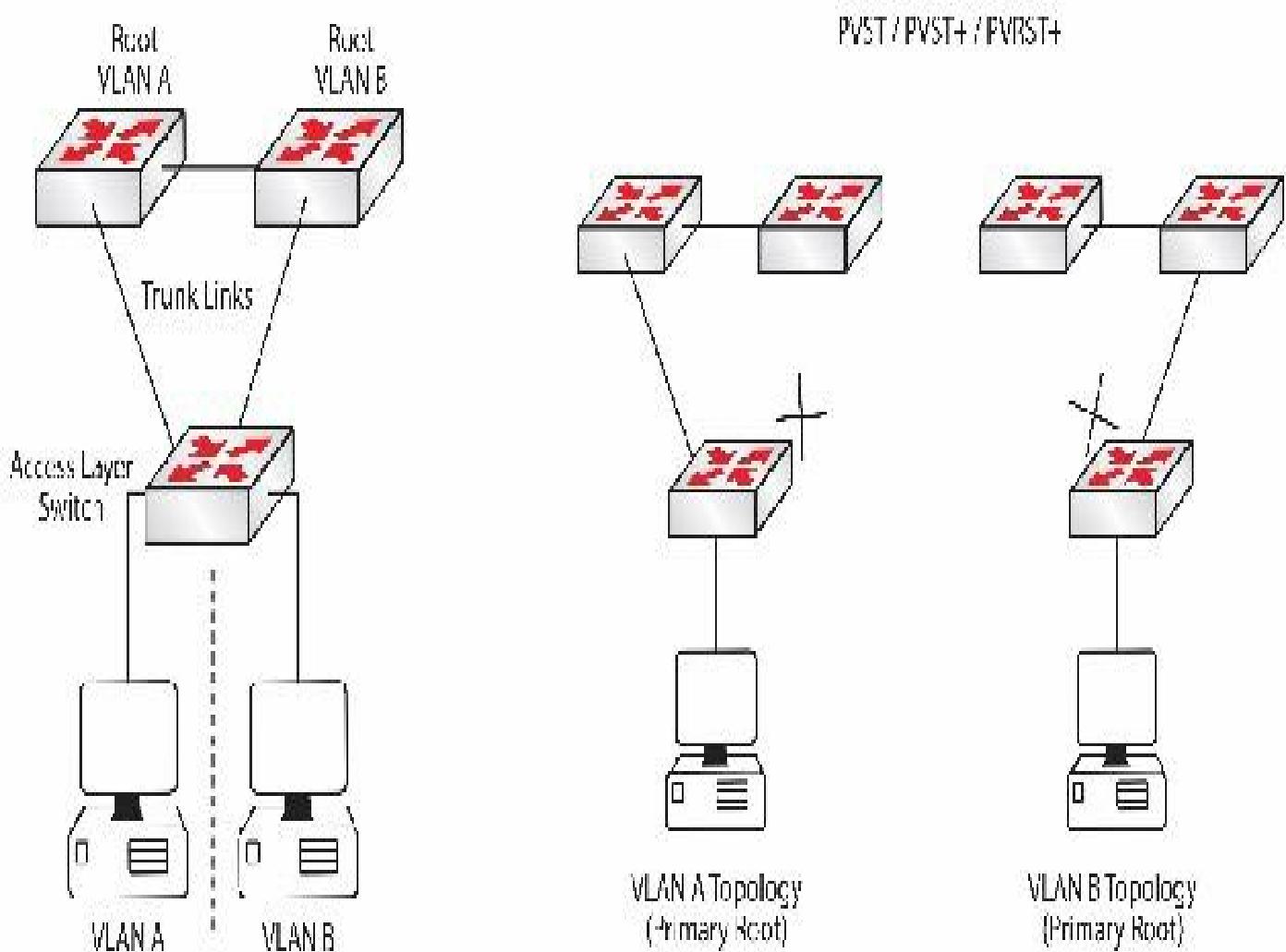


FIG 10.24 – Physical versus logical PVST+ topology

PVST+ and PVRST+ both change the BID in the BPDU by adding the VLAN number to the configured priority. PVRST+ is a combination of PVST+ and RSTP and it provides rapid (under one second) convergence, with the added benefit of PVST+.

Mini-lab – Configuring PVRST+

You can use any switch to create VLAN 10.

```
Switch#show spanning-tree vlan 10
VLAN0010
Spanning tree enabled protocol ieee
Root ID Priority 24586
Address 0015.63f6.b700
Cost 3019
Port 107 (FastEthernet3/0/1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
Bridge ID Priority 49162 (priority 49152 sys-id-ext 10)
Address 000f.f794.3d00
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
```

[output truncated]

To enable RSTP for each VLAN in the switched network, use the following command:

```
Switch(config)#spanning-tree mode rapid-pvst
```

This is all that is needed if you need only one instance of STP. Later on in this section will show what is needed to enable load-sharing capabilities.

Using the show spanning-tree vlan [vlan#] command, you can verify which type of Spanning Tree is running:

```
Switch#show spanning-tree vlan 10
VLAN0010
Spanning tree enabled protocol rstp
Root ID Priority 24586
Address 0015.63f6.b700
Cost 3019
Port 107 (FastEthernet3/0/1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
Bridge ID Priority 49162 (priority 49152 sys-id-ext 10)
Address 000f.f794.3d00
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
UplinkFast enabled but inactive in rapid-pvst mode
```

[output truncated]

Two items are of interest in the output above. The first is the Spanning tree enabled protocol rstp and the second is the sys-id-ext 10. This shows that the bridge priority was configured as 49152 and VLAN id 10 was added to it.

[END OF MINI-LAB]

Load Balancing Using RSTP/STP

How can load balancing be achieved in the network shown in Figure 10.23 above if VLAN 1 and VLAN 5 are being used in the LAN? You can achieve this by configuring Switch A with a better priority for VLAN 1 and configuring Switch B with a better priority for VLAN 5. This can be done using the following commands:

```
SwitchA(config)#spanning-tree vlan 1 priority 4096  
SwitchB(config)#spanning-tree vlan 5 priority 4096
```

The show spanning-tree output for both VLANs on Switch D to verify load balancing is shown below:

```
SwitchD#show spanning-tree  
VLAN0001  
Spanning tree enabled protocol ieee  
Root ID Priority 4097  
Address 0013.c3e8.2500
```

[output truncated]

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/17	Desg	FWD	119	128.17	P2p
Fa0/20	Root	FWD	19	128.20	P2p

```
VLAN0005  
Spanning tree enabled protocol ieee  
Root ID Priority 4101  
Address 0017.94bd.1680
```

[output truncated]

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/17	Root	FWD	19	128.17	P2p
Fa0/20	Desg	FWD	119	128.20	P2p

You can see that the root bridge for VLAN 1 is Switch A, whereas the root bridge for VLAN 5 is Switch B. Fa0/20 is the root port for VLAN 1 and Fa0/17 is the root port for VLAN 5.

End of Chapter Questions

Please also visit www.howtonetwork.com/ccnasimplified to take the free Chapter 10 exam.

Mark on Figure 10.25 below the root bridge and the interface states on each switch for designated, blocking, and root:

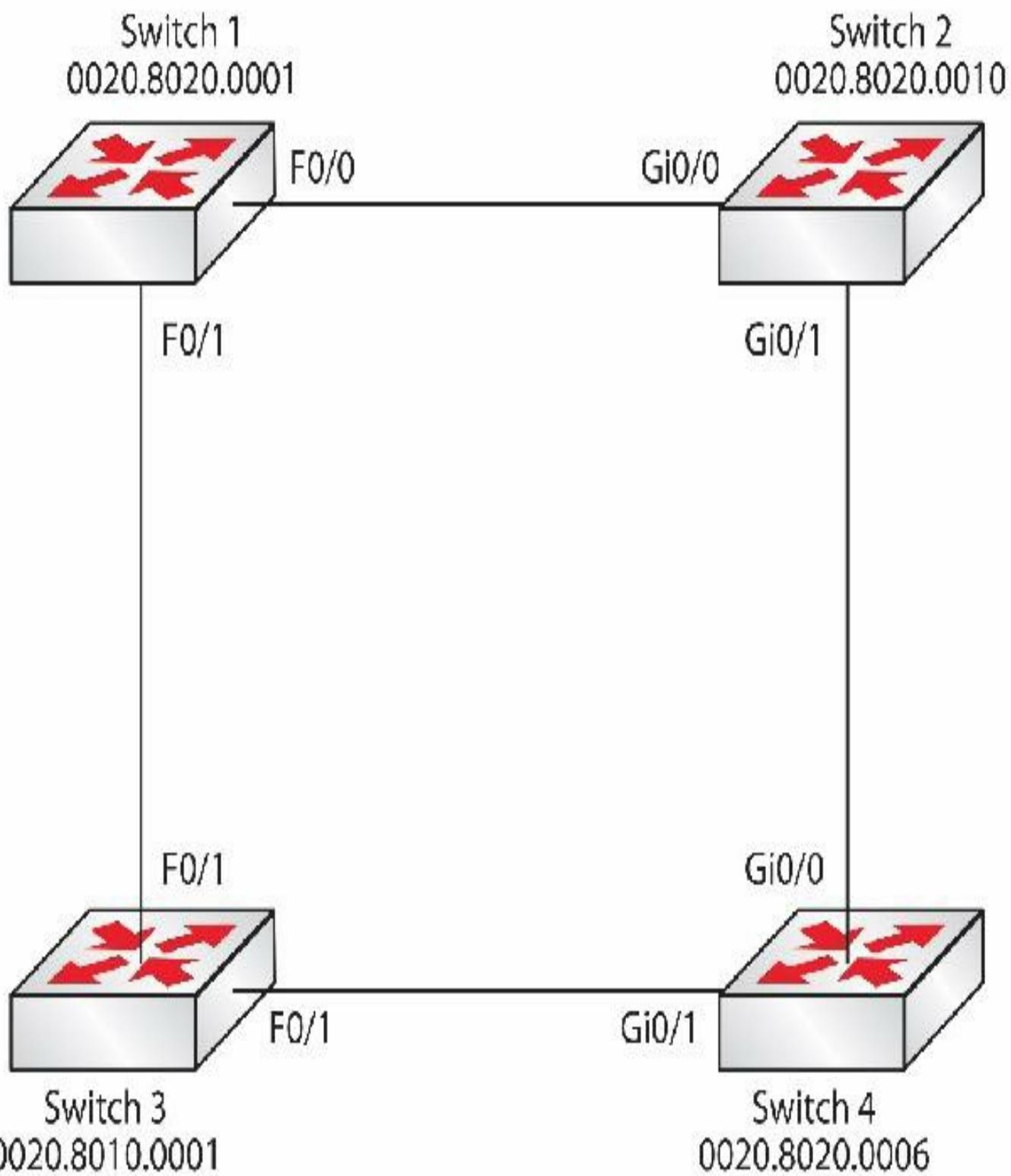


FIG 10.25 – Mark the port roles

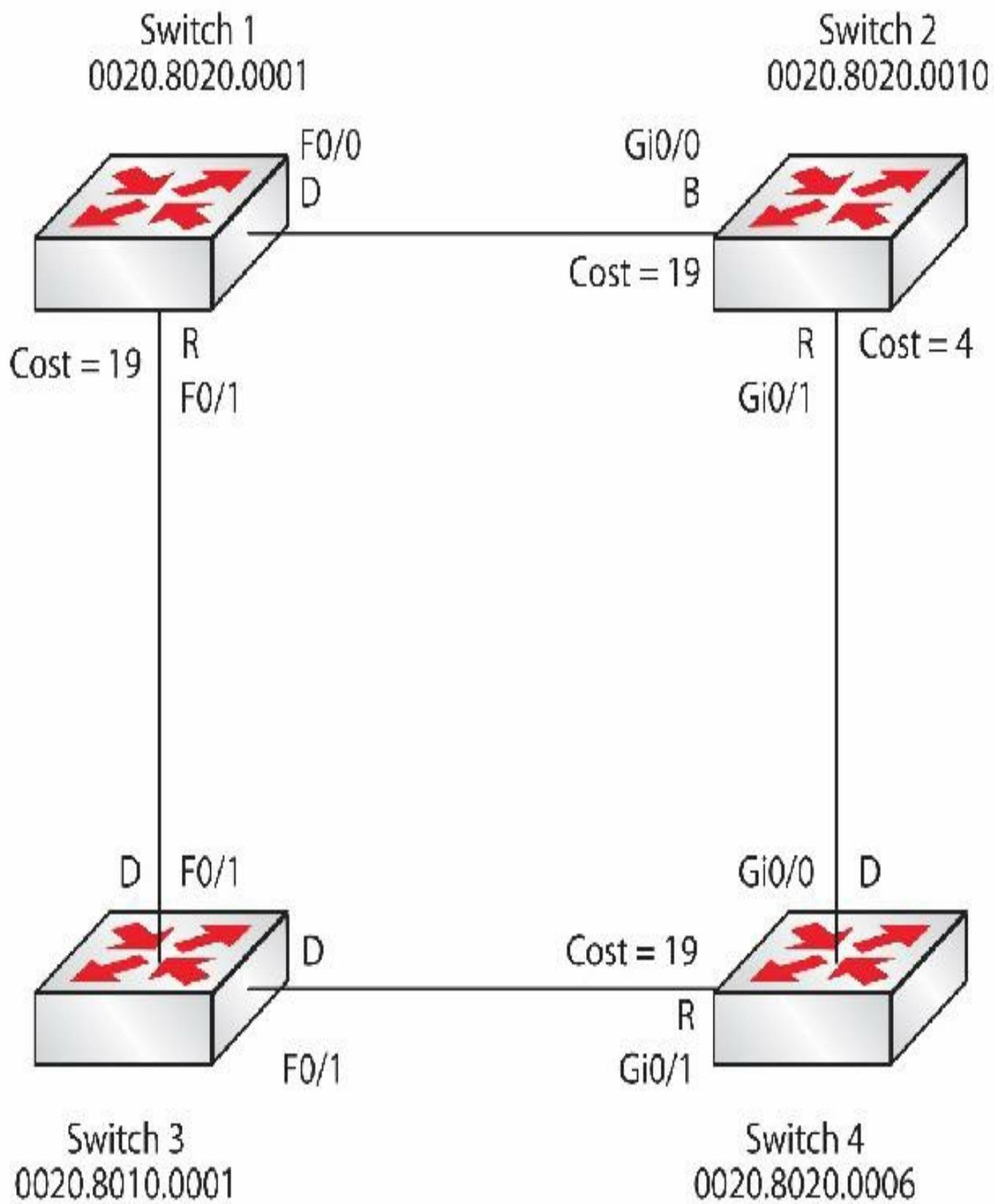


FIG 10.26 – Solution

Chapter 10 Labs

Lab 1: Spanning Tree Protocol

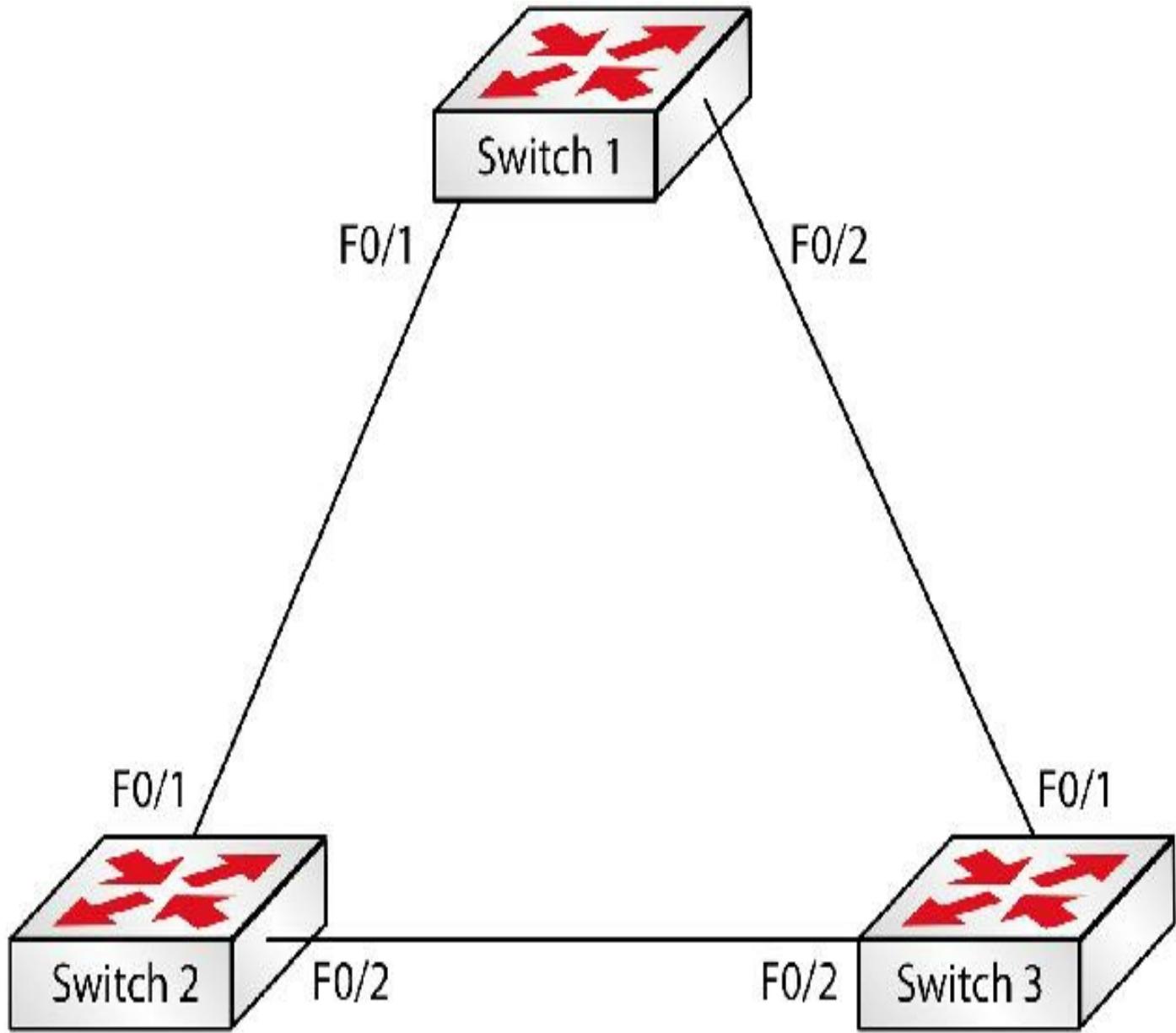


FIG 10.27 – STP election

VLAN 2 and VLAN 3 exist on all switches.

Lab Exercise

Your task is to configure the network above so that Switch 2 is always the root bridge for VLAN 2 and Switch 3 is the root bridge for VLAN 3. You will need to connect the three switches together with crossover cables.

Purpose

STP is a very important topic for the CCNA exam and you can expect to be tested on

both your theoretical knowledge and hands-on ability. For this lab you will configure the switches to ensure that the correct switch is the root bridge. In the real world, if the incorrect switch becomes the root bridge, the network will experience delays.

Lab Objectives

1. Configure Ports F0/1 and F0/2 on all switches as 802.1Q trunks (default on 2960 Switches).
2. Create VLAN 2 and VLAN 3 on all the switches.
3. Configure bridge priority on Switch 2 and Switch 3 for VLAN 2 and VLAN 3, respectively.

Lab Walk-through

1. First, check the status of the switches. The VTP domain name should match. You will also want to check whether there are any interfaces already trunking. You will check the outputs on Switch 1. You can do the same on Switch 2 and Switch 3.

Switch1#sh vtp status

VTP Version: 2

Configuration Revision: 1

Maximum VLANs supported locally: 250

Number of existing VLANs: 10

VTP Operating Mode: Server

VTP Domain Name: howtonetwork

VTP Pruning Mode: Disabled

VTP V2 Mode: Enabled

VTP Traps Generation: Disabled

MD5 digest: 0xB7 0xE3 0x3A 0x57 0x1D 0x41 0x42 0x40

Configuration last modified by 0.0.0.0 at 3-1-93 01:11:38

Local updater ID is 0.0.0.0 (no valid interface found)

You can see that the VTP domain name is howtonetwork. Please set all the switches to the same VTP domain name. Please also make sure that all switches are set to the VTP server (this should be the default but it varies depending on IOS version and whether another user changed this).

Switch2(config)#vtp domain howtonetwork

And server commands:

Switch2(config)#vtp mode ?

client Set the device to client mode.

server Set the device to server mode.
transparent Set the device to transparent mode.

To configure the switches for trunking on relevant ports, follow the commands below:

```
Switch#configure terminal  
Switch1(config)#hostname Switch1  
Switch1(config)#interface range fa0/1 - 2  
Switch1(config-if-range)#switchport mode trunk  
Switch#configure terminal  
Switch(config)#hostname Switch2  
Switch2(config)#interface range fa0/1 - 2  
Switch2(config-if-range)#switchport mode trunk  
Switch#configure terminal  
Switch(config)#hostname Switch3  
Switch3(config)#interface range fa0/1 - 2  
Switch3(config-if-range)#switchport mode trunk
```

Please note—your switch ports may be numbered 1/1, 1/2, and so on depending on your model. The interface range command may not work on your switch if it has an older IOS release, so you will have to set the configurations per interface. Please also note that this command seems to have changed as IOS levels changed, meaning that you don't need gaps for one IOS release but you do need gaps for another, as demonstrated below:

```
SwitchC(config)#int range f0/1-24  
SwitchC(config-if-range)#shut  
SwitchA(config)#int range f0/1-24  
^  
% Invalid input detected at “^”marker.
```

```
SwitchA(config)#int range f0/1 - 24  
You can now check which interfaces are set to trunking:
```

```
Switch1#sh int trunk  
Port      Mode      Encapsulation  Status      Native  
                  vlan  
Fa0/1    on       802.1q        trunking   1  
Fa0/2    on       802.1q        trunking   1  
Port      Vlans allowed and active in management domain
```

Fa0/1 1

Fa0/2 1

[output truncated]

2. To create VLANs on all switches, enter the following commands (your model may require you to input them individually):

Switch1(config)#vlan 2,3

Switch2(config)#vlan 2,3

Switch3(config)#vlan 2,3

You should then be able to see that VLANs 2 and 3 are part of the Spanning Tree:

Switch1#show int trunk

Port	Mode	Encapsulation	Status	Native
------	------	---------------	--------	--------

vlan

Fa0/1	on	802.1q	trunking	1
-------	----	--------	----------	---

Fa0/2	on	802.1q	trunking	1
-------	----	--------	----------	---

Port	Vlans allowed and active in management domain
------	---

Fa0/1	1,2,3
-------	-------

Fa0/2	1,2,3
-------	-------

Switch1#show vlan brief

1	default	active
---	---------	--------

Fa0/3, Fa0/4, Fa0/5, Fa0/6

Fa0/7, Fa0/8, Fa0/9, Fa0/10

Fa0/11, Fa0/12, Fa0/13, Fa0/14

Fa0/15, Fa0/16, Fa0/17, Fa0/18

Fa0/19, Fa0/20, Fa0/21, Fa0/22

Fa0/23, Fa0/24, Gig1/1, Gig1/2

2	VLAN0002	active
---	-----------------	--------

3	VLAN0003	active
---	-----------------	--------

3. Check the switches to see which is the root bridge for VLANs 2 and 3. Some of the output is omitted and, of course, your output will be different due to MAC addresses; you may also have a different root bridge due to the bridge priority/MAC addressing. Note that you will see different fields depending on your switch model or if you are using Packet Tracer.

```
Switch2#show spanning-tree vlan 2
```

VLAN0002

Spanning tree enabled protocol ieee

Root ID Priority 32770

Address 0009.7c87.9081

Cost 19

Port 1 (FastEthernet0/1)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bri. ID Priority 32770 (priority 32768 sys-id-ext 2)

Address 0008.21a9.4f80

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Port ID	Design.	Port
-----------	---------	---------	------

Name	Prio.Nbr	Cost	Sts	Cost	Bridge ID	ID	Prio.Nbr
------	----------	------	-----	------	-----------	----	----------

Fa0/1	128.1	19	FWD	19	32770	0009.7c87.9081	128.1
-------	-------	----	-----	----	-------	----------------	-------

Fa0/2	128.2	19	FWD	19	32770	0008.21a9.4f80	128.2
-------	-------	----	-----	----	-------	----------------	-------

And now issue the same command on Switch 3. The output below is slightly different due to different versions of code and switch models:

```
Switch3#show spanning-tree vlan 2
```

VLAN0002

Spanning tree enabled protocol ieee

Root ID Priority 32770

Address 0009.7c87.9081

Cost 19

Port 1 (FastEthernet0/1)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bri. ID Priority 32770 (priority 32768 sys-id-ext 2)

Address 000f.23a6.8940

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

Fa0/1	Root FWD	19	128.1	P2p
Fa0/2	Altn BLK	19	128.2	P2p

The Spanning Tree cost for a 100 Mbps interface is 19, and you can see that output in the Cost field.

You can issue a show interface fast 0/1 command on Switch 1 to verify that the MAC address owns the MAC address allocated as the root:

```
Swtch1#show int fast0/1
FastEthernet1/1 is up, line protocol is up
  Hardware is Fast Ethernet, address is 0009.7c87.9081
```

4. Do the same for VLAN 3 to see where the root is.

```
Switch2#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID Priority  32771
    Address 0009.7c87.9081
    Cost     19
    Port     1 (FastEthernet0/1)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

The MAC address above belongs to Switch 1 again.

5. Configure bridge priority on Switch 2 and Switch 3 for VLAN 2 and VLAN 3, respectively. You want Switch 2 to be the root for VLAN 2 and Switch 3 to be the root for VLAN 3. You can use the trusty ? to give you more information:

```
Switch2(config)#spanning-tree vlan 2 ?
  forward-time Set the Forward Delay for the spanning tree
  hello-time   Set the Hello interval for the spanning tree
  max-age     Set the Max Age interval for the spanning tree
  priority    Set the bridge priority for the spanning tree
  root        Configure switch as root
[cr]
```

```
Switch2(config)#spanning-tree vlan 2 priority 4096
Switch3(config)#spanning-tree vlan 3 priority 4096
```

Next, issue the show spanning-tree vlan # command to check that the respective switches are the roots for the desired VLANs:

```

Switch2#show spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID Priority  4098
    Address  0008.21a9.4f80
This bridge is the root
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bri. ID Priority  4098 (priority 4096 sys-id-ext 2)
    Address  0008.21a9.4f80
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
Interface Port ID          Desig. Port
                           ID Prio.Nbr
Name   Prio.Nbr Cost  Sts Cost Bridge ID
-----
Fa0/1   128.1   19   FWD 0   4098 0008.21a9.4f80 128.1
Fa0/2   128.2   19   FWD 0   4098 0008.21a9.4f80 128.2

```

If you do the same for VLAN 3 on Switch 3, you will see that it is the root for that VLAN:

```

Switch3#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID Priority  4099
    Address  000f.23a6.8940
This bridge is the root
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Bri. ID Priority  4099 (priority 4096 sys-id-ext 3)
    Address  000f.23a6.8940
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
Interface  Role Sts Cost  Prio.Nbr Type
-----
Fa0/1      Desg FWD 19     128.1  P2p
Fa0/2      Desg FWD 19     128.2  P2p

```

Running Configuration

[VLAN information won't show on a show run]

```
Switch1#sh run
```

```
Building configuration...
```

```
[output truncated]
```

```
hostname Switch1
```

```
!
```

```
interface FastEthernet0/0
```

```
!
```

```
interface FastEthernet0/1
```

```
switchport mode trunk
```

```
!
```

```
interface FastEthernet0/2
```

```
switchport mode trunk
```

```
!
```

```
Switch2#sh run
```

```
Building configuration...
```

```
[output truncated]
```

```
hostname Switch2
```

```
!
```

```
spanning-tree vlan 2 priority 4096
```

```
!
```

```
interface FastEthernet0/0
```

```
!
```

```
interface FastEthernet0/1
```

```
switchport mode trunk
```

```
!
```

```
interface FastEthernet0/2
```

```
switchport mode trunk
```

```
!
```

```
Switch3#sh run
```

```
Building configuration...
```

```
[output truncated]
```

```
hostname Switch3
```

```
!
```

```
spanning-tree vlan 3 priority 4096
```

```
!
interface FastEthernet0/0
!
interface FastEthernet0/1
switchport mode trunk
!
interface FastEthernet0/2
switchport mode trunk
!
```

Chapter 11 — Understanding EtherChannels

What You Will Learn in This Chapter

EtherChannels

Configuring EtherChannels

Syllabus Topics Covered

1.0 LAN Switching Technologies

1.1 Identify enhanced switching technologies

1.1.c EtherChannels

EtherChannel is a Cisco-specific term that is used to describe a logical bundle of physical Ethernet interfaces. EtherChannels solve the issue of connected switches with multiple parallel connections only being able to connect to one of these links due to STP blocking the other links. STP can now treat these multiple links as if they were one physical link.

EtherChannels

You may hear other vendors refer to this technology as:

- Link aggregation
- Link teaming
- NIC teaming
- Port channeling

The idea behind this technique is to take multiple interfaces and bundle (bond) them to increase the amount of throughput between two devices. Link aggregation can use any type of interfaces (e.g., 100 Mbps, 1 Gbps, 10 Gbps) but usually all the interfaces within a bundle must have the same capacity, so you can't connect Fast Ethernet interfaces to Gigabit Ethernet, for example.

Such connections are used in scenarios in which we need a high amount of traffic traveling between two network devices, usually in data center environments at the core or distribution layer. The most common use for link aggregation is connecting two switches.

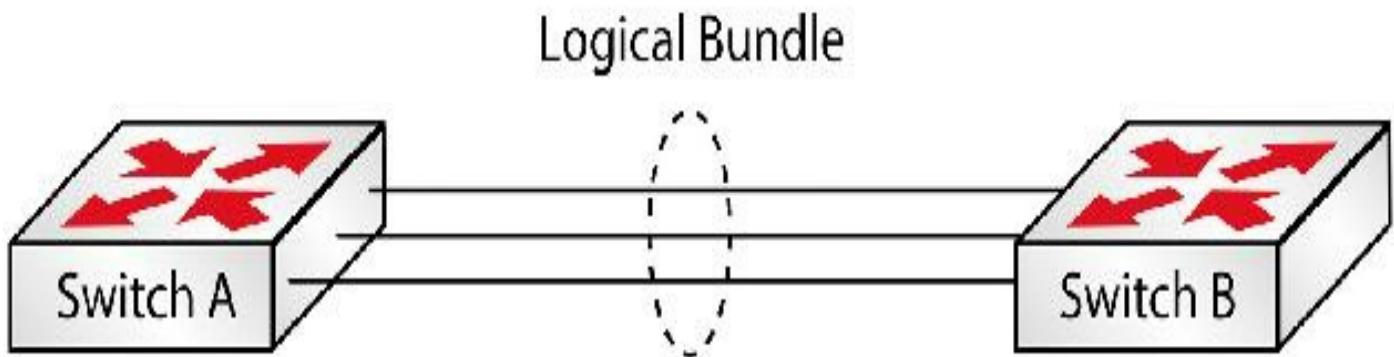


FIG 11.1 – Link aggregation

Besides the increase in throughput, interface bonding also offers another important benefit, which is redundancy. If one link in the bundle fails, the rest keep forwarding traffic and the logical link (the bundle) remains active. In order for the logical link to go down, all the component physical links must go down.

Another benefit of link aggregation is optimizing ports in STP. With multiple physical interfaces, only one of those interfaces would be in forwarding mode (to prevent loops). When the interfaces are bundled together, they are treated as one logical Spanning Tree interface and set in forwarding mode, if there are no redundant EtherChannels. This means that if one of the links in the bundle should fail for any reason, there will be no need for STP convergence to take place. The entire bundle would need to fail for this to happen.

In order for an EtherChannel to form, all ports or interfaces should have matching settings, such as interfaces with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. The DTP settings won't affect EtherChannel formation.

EtherChannel (link aggregation) is a very popular technology and is typically deployed between the distribution and the core layers or between core devices, where increased availability and scalability is needed. However, link aggregation is usually disabled on interfaces that are facing end-users. Without the ability to use EtherChannels, the network in Figure 11.2 below would only be able to utilize one link between the top two switches:

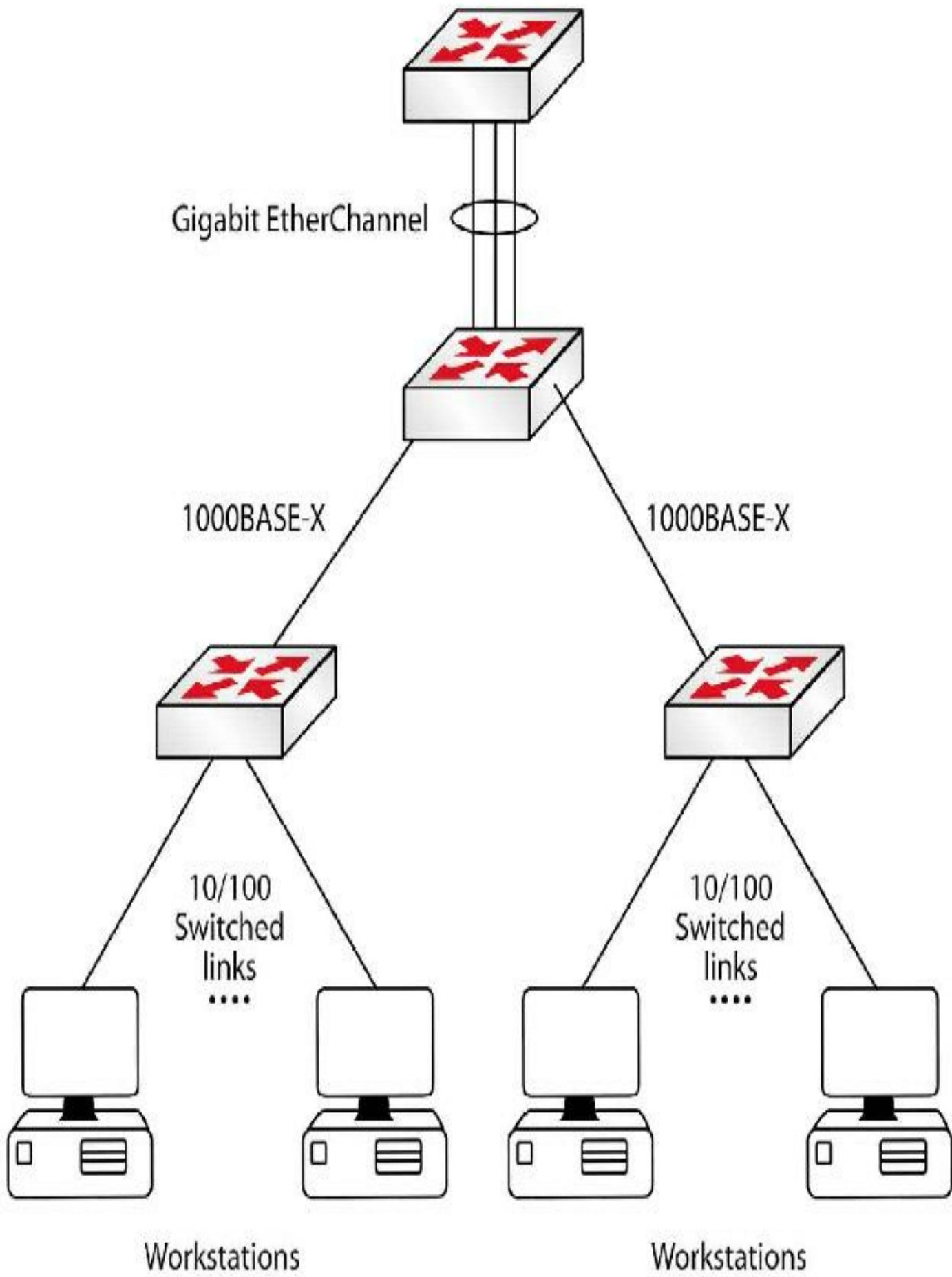


FIG 11.2 – Multiple links utilized by EtherChannel

Two commonly used link aggregation protocols are:

- LACP (Link Aggregation Control Protocol) – an open standard protocol
- PAgP (Port Aggregation Protocol) – a Cisco proprietary protocol

Link Aggregation Control Protocol

LACP is described in the IEEE 802.3AD specification as a standard for aggregating multiple physical Ethernet links into a single logical link. LACP is incompatible with PAgP, so both ends of a link need to run LACP before a logical bundle (EtherChannel) can be formed.

The physical ports in a link bundle should have the same physical properties (such as speed and duplex) and trunk encapsulation type and must be operating on the same layer of the OSI model (either as switch ports or as routed ports). Traffic is balanced across the active members of a port channel. If an interface within a port channel fails, then traffic is rebalanced across the remaining active interfaces.

LACP supports autonegotiation of port channels by exchanging LACP packets. LACP requires ports to operate in full-duplex mode (unlike PAgP, which supports half-duplex). LACP frames are sent to a special multicast group address for IEEE 802.3 slow protocols: 01-80-C2-00-00-02. LACP frames are encoded with the EtherType value 0x8809. Figure 11.3 below shows a captured LACP frame:

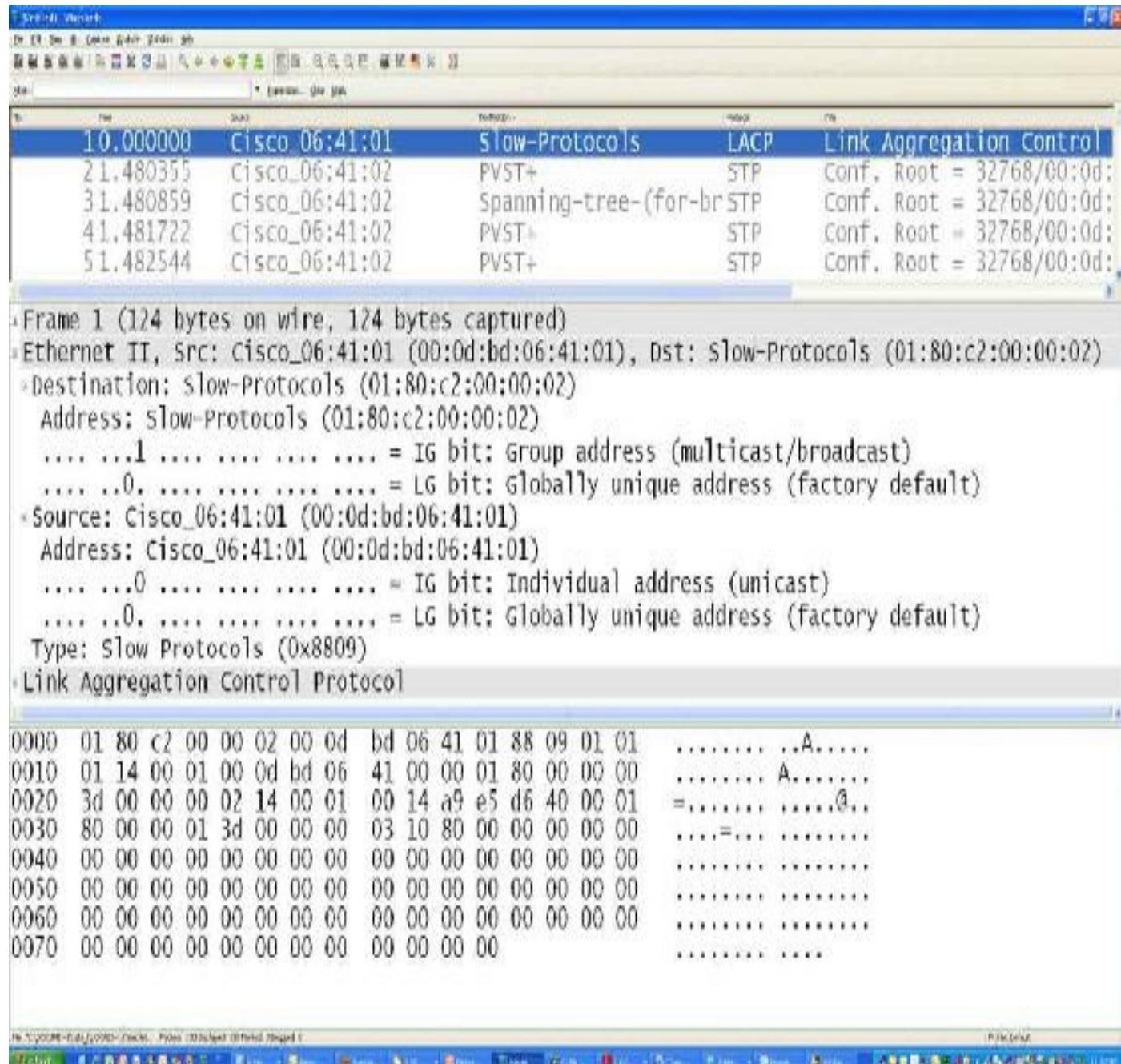


FIG 11.3 – LACP frame capture

LACP Port Modes

LACP supports two modes of operations, which are active and passive.

In LACP active mode, a port actively negotiates LACP. This means that the port initiates negotiations with the remote end. If one of the ends of a link is in active mode, then LACP negotiation will be initiated and a port channel will be formed if all the physical attributes match.

In LACP passive mode, ports will not initiate LACP negotiation. However, they will respond if the other end initiates. This means that if both ends of a link are in passive

mode, then an EtherChannel will not be formed.

Table 11-1 below shows the different LACP combinations and the result when trying to establish an EtherChannel between two switches:

Table 11-1: LACP combinations

Switch 1 LACP Mode	Switch 2 LACP Mode	EtherChannel Result
Passive	Passive	No EtherChannel Formed
Passive	Active	EtherChannel Formed
Active	Active	EtherChannel Formed
Active	Passive	EtherChannel Formed

Port Aggregation Protocol

PAgP is Cisco's proprietary protocol for automatic negotiation of EtherChannels. Just like LACP, PAgP sends packets between EtherChannel-capable ports to negotiate an EtherChannel. These frames are sent to the multicast address 01-00-0C-CC-CC-CC. An output from a packet sniffer is shown below:

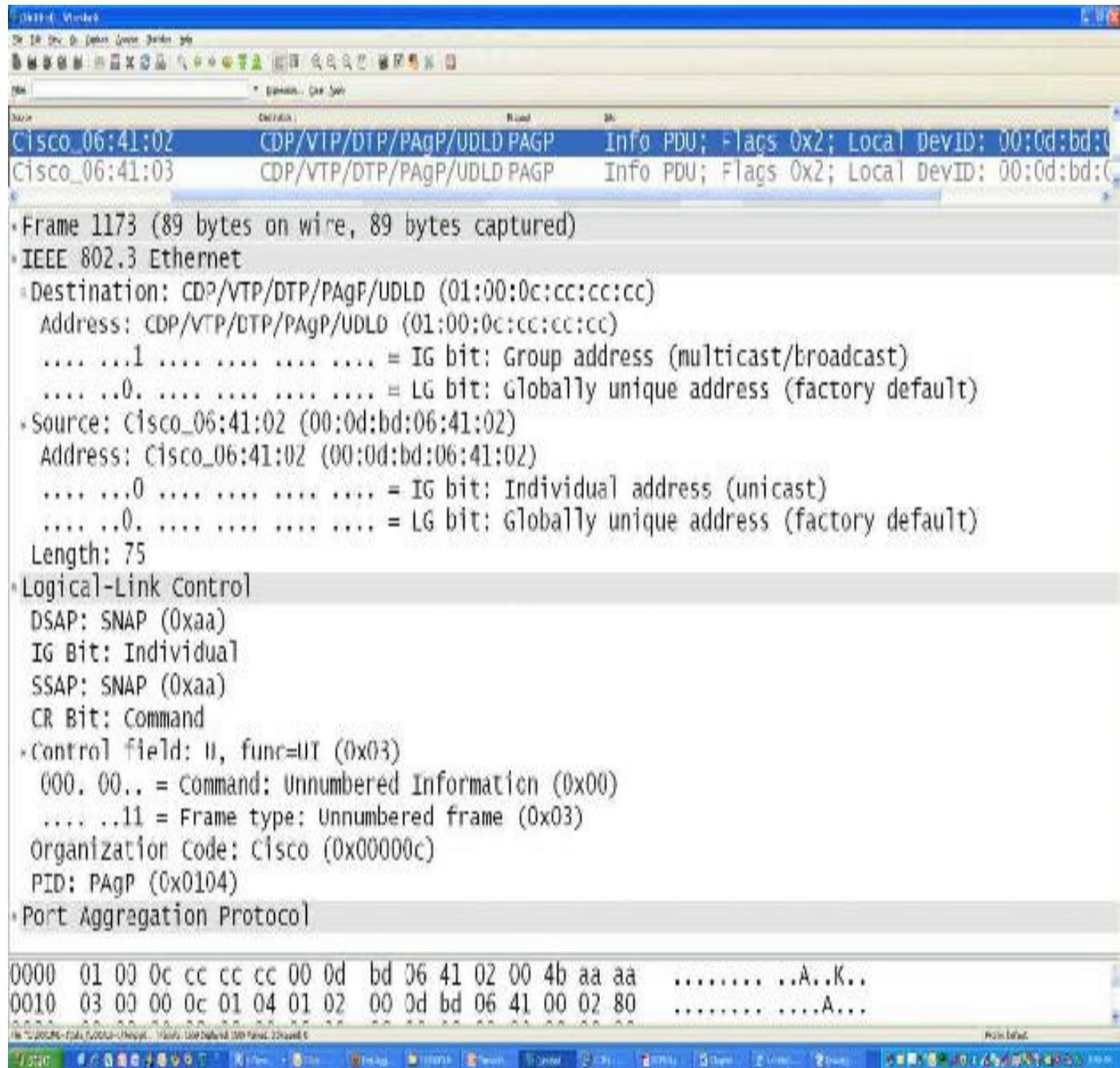


FIG 11.4 – PAgP frame capture

NOTE: You will be expected to identify PAgP or LACP from the multicast MAC address, so make sure that you remember these.

PAgP Port Modes

PAgP also operates in two modes, like LACP, as described below.

Ports in auto mode will not initiate PAgP negotiation but will respond when they receive a PAgP packet. This is similar to the passive mode in LACP. Two ports operating in PAgP auto mode will NOT negotiate an EtherChannel.

Ports in desirable mode will actively initiate PAgP negotiations. This means that an

EtherChannel will be negotiated as long as the other end is running PAgP, whether in auto mode or desirable mode.

Table 11-2 below shows the different PAgP combinations and the result when trying to establish an EtherChannel between two switches:

Table 11-2: PAgP combinations

Switch 1 PAgP Mode	Switch 2 PAgP Mode	EtherChannel Result
Auto	Auto	No EtherChannel Formed
Auto	Desirable	EtherChannel Formed
Desirable	Auto	EtherChannel Formed
Desirable	Desirable	EtherChannel Formed

Configuring EtherChannels on Cisco IOS

PAgP and LACP configurations on Cisco devices are similar but each implementation presents some particularities. We will cover each of them in the sections below.

As a general rule, both PAgP and LACP configurations require the component port to share a number of settings, including:

- Same speed and duplex
- Allowed VLAN list
- STP cost for each VLAN
- STP priority for each VLAN
- STP PortFast settings

An easy way to ensure that the configurations are exactly the same is to use the interface range command. This allows you to apply the same configuration on multiple interfaces at the same time.

Mini-lab – PAgP Configuration

Let's assume that you want to configure two ports in a PAgP EtherChannel carrying VLAN 100. The command to group the ports in an EtherChannel is channel-group [group_id] mode [mode] in interface configuration mode. Use desirable mode for both ports. The number of channel groups you can configure depends on your platform. As of the writing of this chapter, the limit for the 2960 Switch model is six.

```
2960(config)#int f0/3
```

```
2960(config-if)#channel-group ?
```

<1-6> Channel group number

On my Cisco 3550 switch you have 64 available:

3550(config-if)#channel-group ?

<1-64> Channel group number

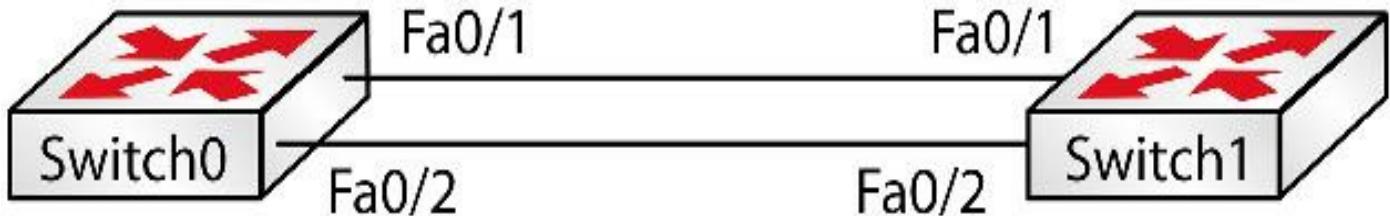


FIG 11.5 – Mini-lab: PAgP Configuration

Switch0#conf t

Enter configuration commands, one per line. End with CNTL/Z.

Switch0(config)#interface range f0/1-2

Switch0(config-if-range)#switchport mode access

Switch0(config-if-range)#switchport access vlan 100

% Access VLAN does not exist. Creating vlan 100

Switch0(config-if-range)#channel-group 6 mode ?

active Enable LACP unconditionally

auto Enable PAgP only if a PAgP device is detected

desirable Enable PAgP unconditionally

on Enable Etherchannel only

passive Enable LACP only if a LACP device is detected

Switch0(config-if-range)#channel-group 6 mode desirable

Creating a port-channel interface Port-channel 6

Now repeat this configuration on Switch 1.

You can verify the EtherChannel configuration using a number of commands:

- show etherchannel [id] port-channel
- show etherchannel [id] brief
- show etherchannel [id] detail
- show etherchannel summary
- show etherchannel port-channel

- show etherchannel protocol

Let's test a couple of these commands on your newly created EtherChannel. You can see that a port channel interface is created and it matches the number allocated to the channel group. Check that the port channel is present first.

Switch0#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/2	unassigned	YES	unset	down	down
Vlan1	unassigned	YES	unset	administratively	down down
Port-channel 6	unassigned	YES	unset	down	down
[output truncated]					

Note that the command below will show you which interfaces are in which group, as well as whether PAgP or LACP is in use. You will need to remember this for the troubleshooting part of the exam.

Switch1#show etherchc summary

Flags: D - down P - 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

u - unsuitable for bundling

w - waiting to be aggregated

d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

Group Port-channel Protocol Ports

-----+-----+-----+

6 Po6(SU) PAgP Fa0/1(P) Fa0/2(P)

Switch0#show etherchannel port-channel

Channel-group listing:

Group: 6

Port-channels in the group:

Port-channel: Po6 (Primary Aggregator)

Age of the Port-channel = 00d:00h:18m:06s

Logical slot/port = 2/1 Number of ports = 3

GC = 0x00000000 HotStandBy port = null

Port state = Port-channel

Protocol = LACP

Port Security = Disabled

Ports in the Port-channel:

Index	Load	Port	EC state	No of bits
-------	------	------	----------	------------

Index	Load	Port	EC state	No of bits
0	00	Fa0/2	Active	0
0	00	Fa0/3	Active	0
0	00	Fa0/1	Active	0

Time since last port bundled: 00d:00h:15m:51s Fa0/1

Switch1#show etherchannel port-channel

Channel-group listing:

Group: 6

Port-channels in the group:

Port-channel: Po6

Age of the Port-channel = 00d:00h:03m:55s

Logical slot/port = 2/6 Number of ports = 2

GC = 0x00000000 HotStandBy port = null

Port state = Port-channel

Protocol = PAGP

Port Security = Disabled

Ports in the Port-channel:

Index	Load	Port	EC state	No of bits
-------	------	------	----------	------------

-----+-----+-----+-----

0	00	Fa0/1	Desirable-S1	0
---	----	-------	--------------	---

0	00	Fa0/2	Desirable-S1	0
---	----	-------	--------------	---

Time since last port bundled: 00d:00h:03m:55s Fa0/2

[END OF MINI-LAB]

You might want to note that the port channel load balances traffic across its member interfaces on a per-flow method based on any of the sources and destinations below:

- Destination IP
- Destination MAC
- Source and destination IP
- Source and destination MAC
- Source IP
- Source MAC

The default behavior differs from platform to platform but, in general, if you want a more granular distribution of traffic on the EtherChannel, it is recommended that you configure load balancing based on the source and destination IP hash. This is done in global configuration mode, as follows:

```
Switch(config)#port-channel load-balance src-dst-ip
```

To verify the EtherChannel load balancing configuration, use the following commands:

```
Switch#show etherchannel load-balance
```

```
Source XOR Destination ip address
```

Mini-lab – LACP Configuration

LACP configuration is similar to PAgP configuration. However, you should choose an LACP-specific port mode (active or passive) in the channel-group command:

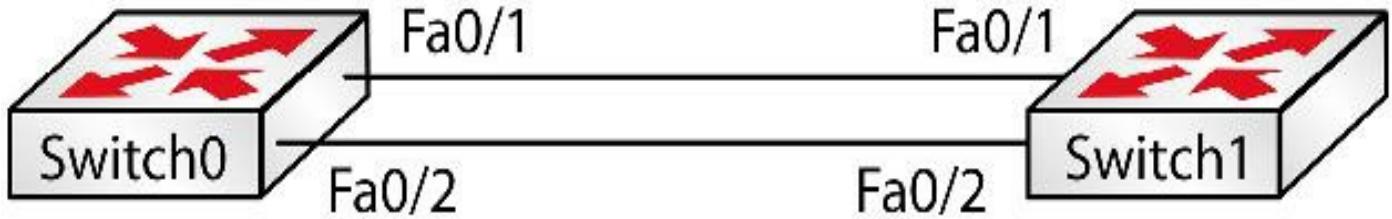


FIG 11.6 – Mini-Lab: LACP Configuration

Here is the configuration for Switch 0; configure Switch 1 yourself:

```
Switch0(config)#interface range FastEthernet1/1-2
Switch0(config-if-range)#switchport mode access
Switch0(config-if-range)#switchport access vlan 100
Switch0(config-if-range)#channel-group 6 mode active
```

Try out some of the previous show commands.

The load balancing customization is achieved in the same way as with PAgP:

```
Switch0(config)#port-channel load-balance src-dst-ip
[END OF MINI-LAB]
```

Layer 3 EtherChannel Configuration

EtherChannels can also be configured as layer 3 logical interfaces. This is done by converting the port to a routed port using the no switchport command and assigning an IP address on the port channel interface. This is accomplished as follows (you won't be able to do this on a 2960 Switch because it doesn't generally support layer 3 services and protocols):

```
Switch#configure terminal
Switch(config)#interface port-channel 10
Switch(config-if)#no switchport
Switch(config-if)#ip address 10.0.0.1 255.255.255.0
Switch(config-if)#no shut
Switch(config-if)#do sho ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	up	
FastEthernet0/1	unassigned	YES	unset	up	

```
Port-channel10 10.0.0.1      YES manual up          up
```

Port Channel Mode On

There is another method to configure EtherChannels using the interface command channel-group # mode on. This is neither PAgP nor LACP (i.e., no protocol is involved in setting up the channel or monitoring it). If you use the on command for one side you cannot use any LACP or PAgP commands on the other side (i.e., active, passive, auto, or desirable).

If both ends of an EtherChannel are hardcoded as on, the EtherChannel will be formed, without a negotiating protocol.

The configuration on a switch on one of my live Cisco racks (access layer switch 1 [ALS1]) is shown below:

```
ALS1(config-if)#channel-group 1 mode on
```

```
Creating a port-channel interface Port-channel 1
```

```
1d08h: %LINK-3-UPDOWN: Interface Port-channel11, changed state to up
```

```
1d08h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel11, changed state to up
```

```
ALS1(config-if)#end
```

```
ALS1#show etherchannel summary
```

Flags: D - down P - in port-channel

I - stand-alone s - suspended

H - Hot-standby (LACP only)

R - Layer3 S - Layer2

u - unsuitable for bundling

U - in use f - failed to allocate aggregator

d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

Group	Port-channel	Protocol	Ports
-------	--------------	----------	-------

-----+-----+-----+

1	Po1(SU)	-	Fa0/12(P)
---	---------	---	-----------

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 11 exam.

Chapter 11 Labs

Lab 1: LACP EtherChannels

Lab Exercise

Your task is to configure the network in Figure 11.7 below. You will place three ports into a VLAN and then create an LACP EtherChannel between them. When you have completed the lab, reload the switches and configure one side as passive. Then repeat the process using PAgP.

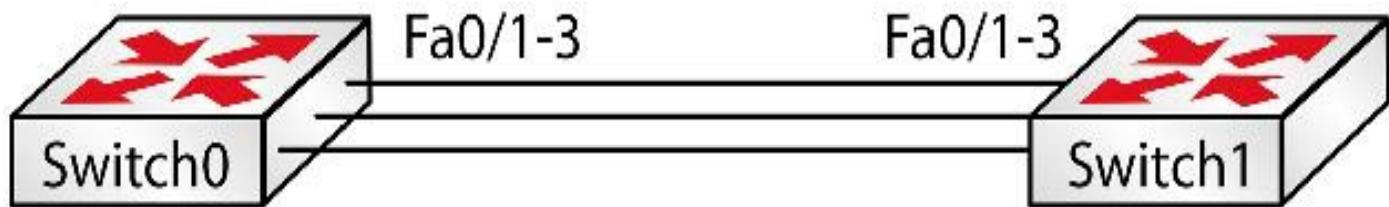


FIG 11.7 – LACP EtherChannels Lab

Text in Courier New font indicates commands that can be entered on the router.

Purpose

This two-switch lab will teach you the basics of EtherChannel.

Lab Objectives

1. Use the ports listed in Figure 11.7.
2. Use the interface range command to group ports Fast 0/1-3.
3. Make all ports access and place into VLAN 100.
4. Configure EtherChannel LACP on both switches.
5. Use show commands to verify your configurations.

Lab Walk-through

1. Configure the VLAN on the switches.

```
Switch(config)#host Switch0
```

```
Switch0(config)#interface range fast0/1-3
```

```
Switch0(config-if-range)#switchport mode access
```

```
Switch0(config-if-range)#switchport access vlan 100
```

2. Create the EtherChannel.

```
Switch0(config-if-range)#channel-group ?
```

```
<1-6> Channel group number
```

```
Switch0(config-if-range)#channel-group 1 mode active
```

```
Creating a port-channel interface Port-channel 1
```

3. Now repeat the commands on Switch 1.

```
Switch(config)#host Switch1
```

```
Switch1(config)#interface range fast0/1-3
```

```
Switch1(config-if-range)#switchport mode access
```

```
Switch1(config-if-range)#sw access vlan 100
```

```
% Access VLAN does not exist. Creating vlan 100
```

```
Switch1(config-if-range)#channel-group 1 mode active
```

```
Creating a port-channel interface Port-channel 1
```

4. Issue show commands to display the EtherChannel information.

```
Switch0#show etherchannel summary
```

Flags: D - down P - 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

u - unsuitable for bundling

w - waiting to be aggregated

d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

Group Port-channel Protocol Ports

Group	Port-channel	Protocol	Ports
1	Po1(SU)	LACP	Fa0/1(P) Fa0/2(P) Fa0/3(P)

Show Runs

```
hostname Switch0
```

```
!
```

```
spanning-tree mode pvst
```

```
!
```

```
interface FastEthernet0/1
```

```
switchport access vlan 100
```

```
channel-group 1 mode active
```

```
switchport mode access
!
interface FastEthernet0/2
switchport access vlan 100
channel-group 1 mode active
switchport mode access
!
interface FastEthernet0/3
switchport access vlan 100
channel-group 1 mode active
switchport mode access
!
interface FastEthernet0/4
!
hostname Switch1
!
spanning-tree mode pvst
!
interface FastEthernet0/1
switchport access vlan 100
channel-group 1 mode active
switchport mode access
!
interface FastEthernet0/2
switchport access vlan 100
channel-group 1 mode active
switchport mode access
!
interface FastEthernet0/3
```

```
switchport access vlan 100  
channel-group 1 mode active  
switchport mode access  
!
```

Chapter 12 — Router Architecture

What You Will Learn in This Chapter

Router Architecture

Router Boot-up Sequence

Managing the IOS

IOS Licensing

Syllabus Topics Covered

2.0 IP Routing Technologies

- 2.1 Describe the boot-up process of Cisco IOS routers
- 2.2 Configure and verify the operation status of a Serial interface
- 2.3 Manage Cisco IOS files
 - 2.3.a Boot preferences
 - 2.3.b Cisco IOS images (15)
 - 2.3.c Licensing
 - 2.3.c (i) Show license
 - 2.3.c (ii) Change license

We covered two important syllabus subjects in Chapter 5 in the ICND1 section—routing metrics and administrative distances—because they were a better fit for basic IP routing technologies.

After STP issues, the highest number of support calls we dealt with at Cisco TAC from Cisco experts was managing IOS files and recovering lost passwords. Perhaps because it's such an apparently simple subject it is ignored, but if you think about it, losing the IOS file or forgetting the configuration password ranks as a network emergency. For this reason, you would do well to learn the subjects in this chapter by heart.

Router Architecture

For the CCNA exam, you must be very familiar with the types of memory used by Cisco routers and how it is utilized. You should also understand the router boot sequence and what can go wrong. Cisco could also test you on the new licensing format for their IOS.

Router Memory

Cisco routers ship with several different types of memory. Each memory module performs a specific function, and you will be expected to know which does what for the

exam.

ROM

Read-only memory is used to store a tiny operating system called the bootstrap. This helps the router boot up and then pulls the main operating system, or IOS, into memory from flash. ROM chips cannot usually be upgraded because they are soldered onto the motherboard.

Boot ROM is a special kind of EEPROM that houses files that are used to load the IOS and to present the ROMmon mode as a fallback, in case there is no IOS file present or it is corrupted. In the ROMmon mode, limited commands are allowed to recover the router and load another IOS. You can see the router is in ROMmon mode via one of the prompts below, depending on your router model:

```
>  
Rommon>
```

ROM can sometimes contain an RXBOOT image. This contains a Mini IOS, which can then be used to upload a full IOS. It is also referred to as the boot loader and it is primarily used to perform some router maintenance activities. You will see RXBOOT on old models of Cisco routers, such as the 2500 Series.

DRAM

Dynamic RAM (random-access memory) is used by a router to store the running configuration, which describes the active configuration state of the router. This comprises the commands and instructions the router is currently using. Any changes you make to the router's configuration are automatically stored in DRAM. They will all be lost when the router is powered down, which makes this type of memory volatile. In order to save the instructions, you have to tell the router to save the running configuration in NVRAM (non-volatile RAM) and rename it startup-config:

```
R1#copy running-config startup-config
```

```
Destination filename [startup-config]?
```

Of course, you would shorten this to copy run start.

DRAM also is used as a buffer to temporarily store packets and routing tables. The main system image (also called IOS) is also loaded into DRAM while the router is running. You can see how the split is made when you issue the show version command, which shows the DRAM divided into two numbers. For example, my router (see below) has 16 MB DRAM split into 14 MB and 2 MB (14 MB is for the IOS, running configuration, and routing tables, and 2 MB is for buffering packets that cannot be processed yet).

A memory split is normal in most models of Cisco routers and it can be changed with the memory size command. This change is not recommended unless under the direction of a Cisco TAC.

When you issue a show version command on a router, you can see the DRAM memory split:

14336 KB / 2048 KB of memory

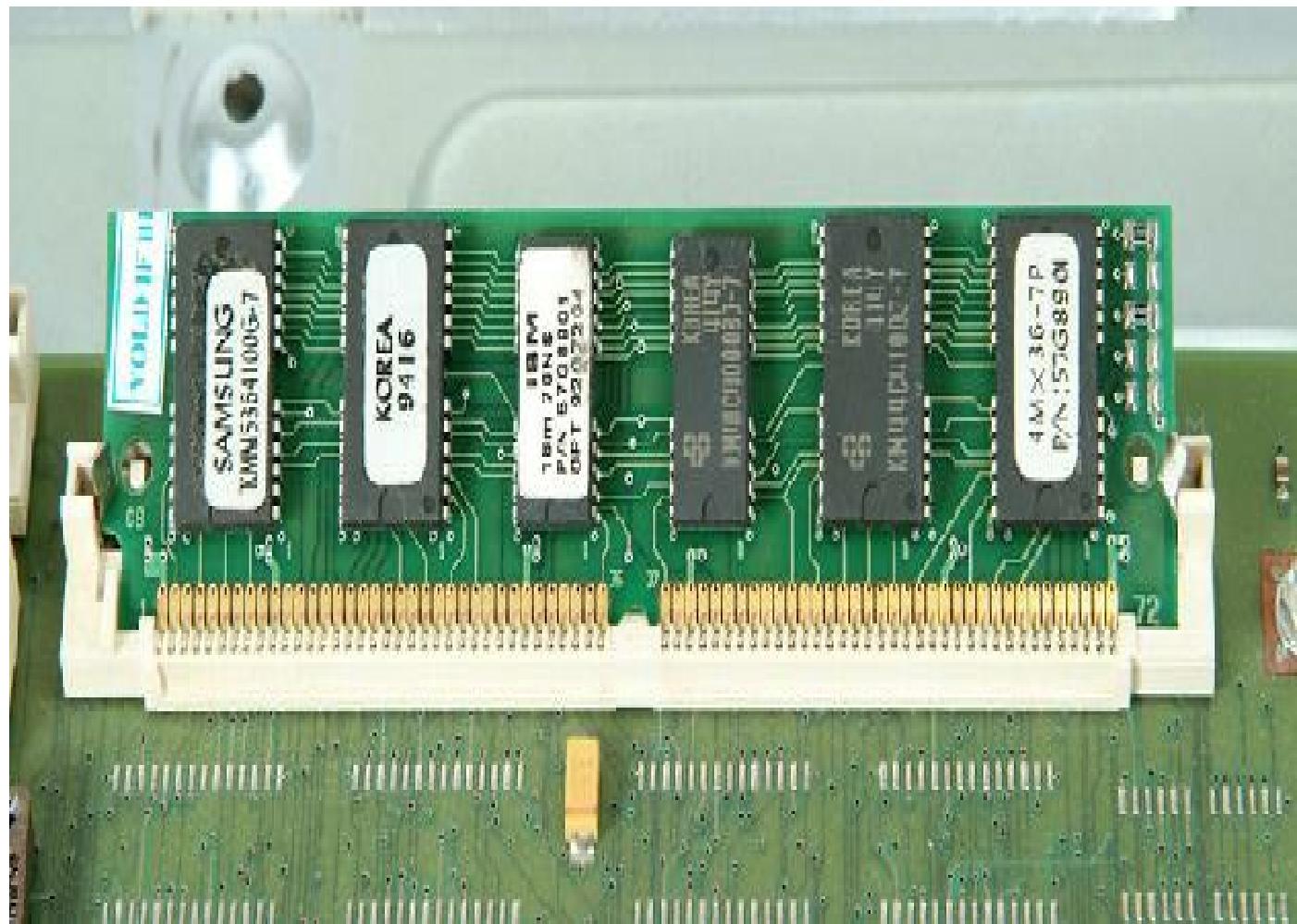


FIG 12.1 – 16 MB DRAM stick

Flash Memory

Flash is used by the router to store the main operating system, or IOS. Flash memory is normally in the form of EEPROM SIMM chips on the motherboard. Flash memory can also be added to the router in the form of PCMCIA cards on some router models. Modern routers also have USB ports available for storing multiple IOS files, as well as router configurations.



FIG 12.2 – Two 8 MB sticks of flash memory

When the router boots, the IOS is decompressed out of flash memory and loaded into DRAM. An example of a flash image stored on the router can be seen below. Your output will differ if you use the `dir flash:` command.

```
Router#show flash
```

```
System flash directory:
```

```
Length Name/status
```

```
c2900-universalk9-mz.SPA.151-1.M4.bin.
```

```
[output truncated]
```

c2900 refers to the model of the router, in this case the 29xx Series, which is running a universal IOS image. The K9 image consists of strong cryptographic features such as AES/3DES encryption. Next, mz indicates that it's stored in RAM (m) in Zipped (z) format. SPA stands for digitally Signed Production version and A is the key version used to sign the software. Finally, 151-1.M4 is the version and release within that version.

You can view the internetwork operating system (IOS) on a router with the show flash or the show version command, although this may differ depending on your router model. You can delete the files in flash memory with the delete flash:filesystem command:

```
RouterA#delete flash:c2900-universalk9-mz.SPA.151-4.M4.bin
```

If you accidentally delete a flash file system, you can recover it with the undelete command. The specific steps depend on the type of file system you have on your router.

NVRAM

Non-volatile RAM is used by the router to store the router's startup configuration. Imagine having to reconfigure the router every time you wanted to reload it. Because NVRAM is non-volatile, it will not lose information when power is removed. The startup configuration is transferred to DRAM every time you reload the router and is renamed running config.

You can compress it with the service compress-config command if you have a very large configuration file in NVRAM. It will, however, take longer for the image to decompress into DRAM.

```
Router(config)#service compress-config
```

The command to copy files from a TFTP server to the flash memory of a router is copy tftp flash. You will then be prompted to enter the IP address of the other host where the new flash file is, as shown below (not that we aren't in configuration mode):

```
RouterA#copy tftp flash  
Address or name of remote host []? 10.10.10.1
```

You will then have to enter the name of the flash image on the other router:

```
Source filename []? / c2900-universalk9-mz.SPA.151-4.M4.bin  
Destination filename [c2900-universalk9-mz.SPA.151-4.M4.bin]? 
```

You may be prompted to erase the flash memory on your router before copying and transferring the file if you have an older model router. You can actually store multiple IOS images in flash memory. You should issue a show version or dir flash: command before you initiate an IOS upgrade to ensure that you have sufficient space (flash and DRAM) available. If you don't do this first, you may find that the router will erase the current IOS and then the upgrade will fail, leaving you with no IOS image on your router.

The output below shows the results of a dir flash: command. You can see that there is an

IOS file present but there is also 221896413 bytes free. You need to be able to work out what this is in MB. I tend to count six across from the right and the number before that usually is the number of MB, so from the output below you still have 221 MB free.

Router#dir flash:

Directory of flash0:/

3 -rw- 33591768 <no date> c2900-universalk9-mz.SPA.151-4.M4.bin

2 -rw- 28282 <no date> sigdef-category.xml

1 -rw- 227537 <no date> sigdef-default.xml

255744000 bytes total (221896413 bytes free)

When the router reloads, your new flash image should be present.

Other options are to issue the copy flash tftp command if you want to store a backup copy or the copy running config tftp command if you want to back up your running configuration file.

You can run a debug on TFTP traffic with the debug tftp command. The output below shows more options for TFTP:

Router#copy tftp ?

flash: Copy to flash: file system

running-config Copy configuration from system

startup-config Copy startup configuration from system

Table 12-1: Router memory and configuration file location

Memory	Usage
ROM	Bootstrap IOS
DRAM	Running config/routing tables/buffers
EEPROM/Flash	IOS storage
NVRAM	Startup config

CPU

The CPU is where all of the processing takes place on the router. Cisco CPUs are generally not upgradeable. You can easily see the CPU type and how much memory you have on your router by typing the show version command at the router prompt. Some of the output below has been truncated.

Router#show version

Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 15.1(4)M7, RELEASE SOFTWARE (fc3)

Copyright (c) 1986-2008 by Cisco Systems, Inc.

ROM: ROM: 3700 Software (C3725-ADVENTERPRISEK9-M), Version 15.1(4)M7, RELEASE SOFTWARE (fc3) i **ROM code**

BOOTLDR: 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c)

System image file is flash: c3725-adventerprisek9-mz.151-4.M7 i **Flash Image**

Cisco 3725 (R7000) processor (revision 0.1) with 124928K/6144K bytes of memory. i **DRAM**

Processor board ID 18086064, with hardware revision 00000003 i **CPU**

32K bytes of non-volatile configuration memory. i **NVRAM**

16384K bytes of ATA System CompactFlash (Read/Write) i **EEPROM/FLASH**

You can actually check the contents of your router or switch memory with the dir command. Note from the output below that you need to add : to the file system that you want to check.

Router#dir ?

/all: List all files

/recursive: List files recursively

all-filesystems: List files on all filesystems

archive: Directory or file name

cns: Directory or file name

flash: Directory or file name

null: Directory or file name

nvram: Directory or file name

system: Directory or file name

tar: Directory or file name

tmpsys: Directory or file name

usbflash0: Directory or file name

xmodem: Directory or file name

ymodem: Directory or file name

[: Output modifiers

[cr]

Router#dir usbflash0:

Directory of usbflash0:/

```
1 -rw- 47438932 Jun 19 2014 13:03:46 +00:00 c1841-adventurese9-mz.151-  
4.M7.bin
```

You can also check the USB device attached with the commands below:

Router#show usb?

controllers Controllers

device USB Device information

driver USB Driver information

port USB Port information

tree USB Device Tree

Router#show usb device

Host Controller: 1

Address: 0x1

Device Configured: YES

Device Supported: YES

Description: Disk

Manufacturer: USB

Router Boot-up Sequence

When you power on a Cisco IOS router, it performs a series of verifications and processes before the IOS (operating system) is loaded and started. The standard sequence of steps is as follows:

1. On power on, the router first performs the POST (power-on self-test). The purpose of this verification test is to check that all the components of the router are present and operating correctly (processor, memory, fans, interfaces, modules, etc). The POST procedure is stored in ROM (read-only memory). The system then checks the configuration register setting. 0x2102 means check the startup configuration file for any commands directing where to load the IOS from. 0x2142 means check flash and TFTP in that order, and if nothing is found

boot from ROM.

2. The bootstrap loads the Cisco IOS software. The bootstrap is a program in ROM used to activate other software components. It is responsible for finding the IOS versions available on the device and loading the proper one. The IOS is loaded from flash memory.
3. The IOS looks for a valid configuration file in NVRAM (the startup configuration) if the configuration register is set to 0x2102.
4. If the IOS successfully finds a valid startup configuration file in NVRAM, it will load the file and run it, making the router operational. If the IOS does not find a valid startup configuration file in NVRAM, it will start in setup mode, allowing the user to define basic configuration settings before the IOS is fully operational. Any additional modifications to the configuration will be stored in RAM. The updated configuration can be saved to the startup configuration in NVRAM using the following command: `copy running-config startup-config`.

Figure 12.3 below shows you the boot-up process in full. Ensure that you can draw this from memory for the exam.

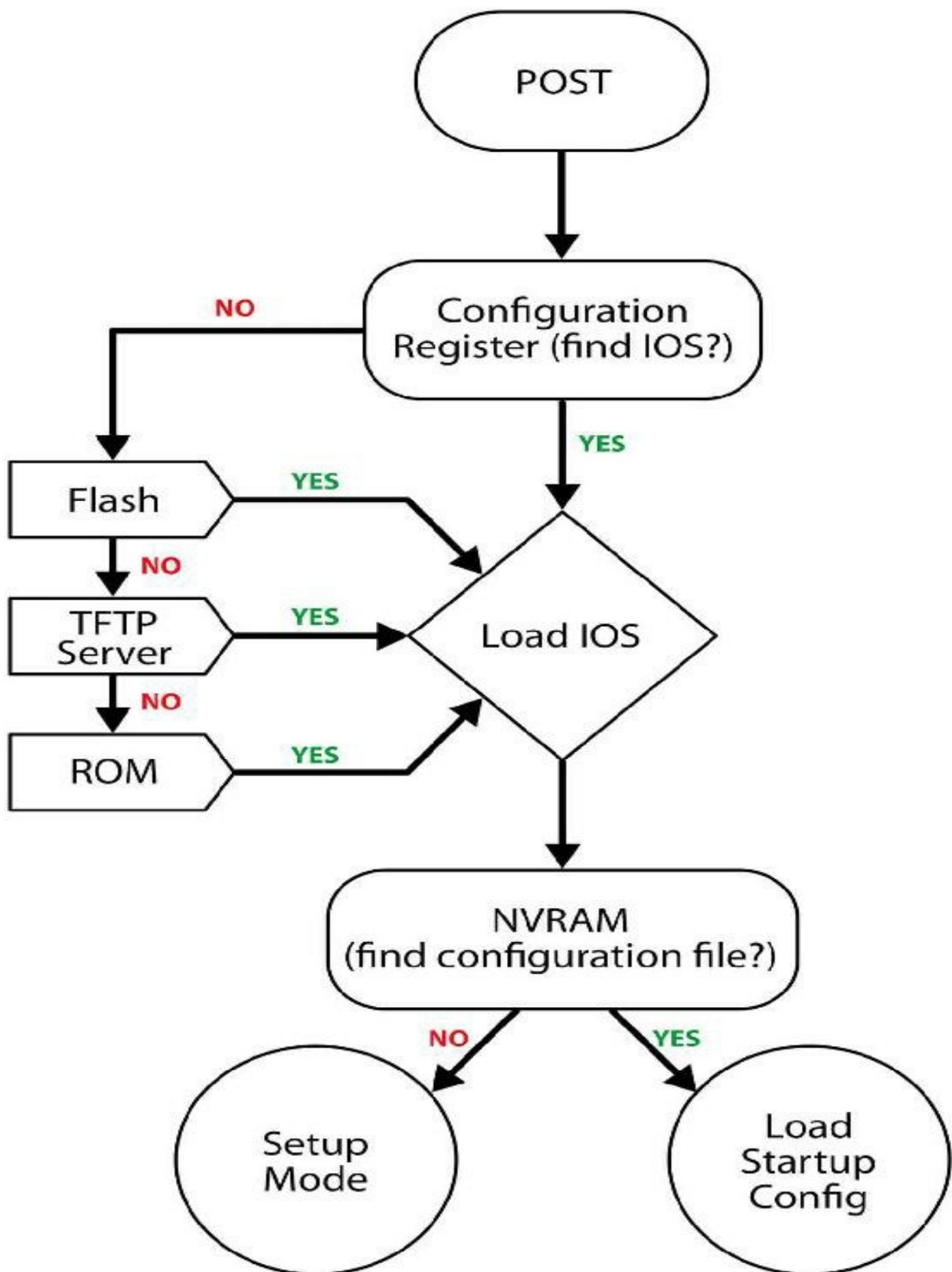


FIG 12.3 – Router boot-up sequence

Setup mode was described earlier in this guide. You will recognize it from the output below:

Would you like to enter the initial configuration dialog? [yes/no]:

% Please answer “yes” or “no”.

Remember also that you should never type yes because you will enter a Q and A mode where the router will attempt to configure itself based on your replies to certain questions. Figure 12.4 below shows which memory types are accessed during the boot stages.

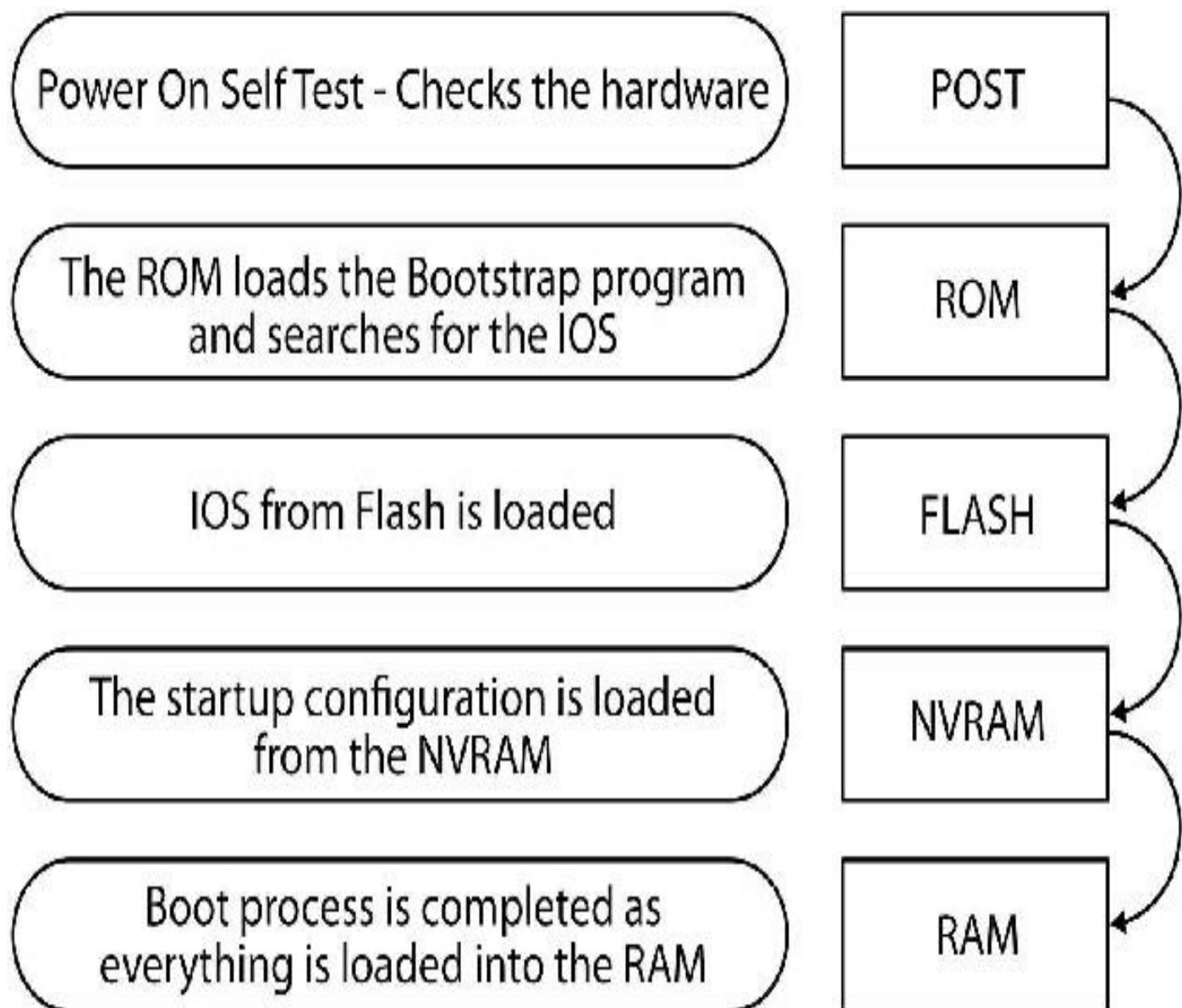


FIG 12.4 – Router booting sequence

Managing the IOS

We all know how important it is to back up our network servers. Most companies have robust backup procedures in place. However, in my time working for the Cisco core team, I discovered that a great many companies do not back up their router and switch configurations. Even expert-level Cisco engineers were unfamiliar with backup and recovery procedures, leaving their network extremely vulnerable in the case of a memory failure. Many network disasters can be avoided or easily resolved by performing and storing regular backups of configuration and IOS files.

Whenever a change is made to the running configuration, the change is stored in volatile DRAM. Once you are comfortable with the changes made, you should copy the configuration to NVRAM using the copy run start command. If you issued the copy start run command, the contents of NVRAM would be transferred into DRAM. If you fail to do this the router will reboot using the configuration originally left in NVRAM.

An easy way to save the configuration to a location in the network is using TFTP. You need to have reachability to the device that has the TFTP software installed. The command to do this is the copy running-config tftp:

Router#copy running-config tftp: i **You need to include the colon**

R1 con0 is now available

Press RETURN to get started.

```
R1>en
R1#copy running-config tftp
Address or name of remote host []? 192.168.1.103
Destination filename [r1-config]?
!!
1335 bytes copied in 4.436 secs (301 bytes/sec)
R1#
```

FIG 12.5 – Screenshot of the copy running-config tftp command

You also could have copied the startup-config file if you wanted to. Here are some options for the running configuration file:

Router#copy running-config ?

flash: Copy to flash file

ftp: Copy to current system configuration

startup-config Copy to startup configuration

tftp: Copy to current system configuration

Similarly, you can copy an IOS image onto a TFTP server. It is useful to back up IOS images before upgrading to a newer version. Since the IOS image is located in flash memory, the command to do this is shown below:

Router#copy flash tftp:

The router will prompt for the TFTP server, which should be a reachable server with TFTP software installed. An example of TFTP software is 3cDaemon. The router will also prompt you for the source and destination filenames.

The general syntax for copy commands is copy [source] [destination], so if you want to reverse the order, the command would be:

Router#copy tftp flash:

You should practice these commands until you know them by heart. They are very useful in emergency scenarios when you need to upgrade a device.



Backing up your files and configurations is absolutely vital. Many companies have gone out of business because they did not bother to back up their startup configurations. By the time they get back on line, they may have lost a huge sum of money. Do not ever let this happen to you or your client.

Booting Options

Although the router will usually boot using one image contained in flash, you can actually instruct it to boot from an image held in a network server or from one of the multiple images held in flash. Sometimes an IOS image may be too large to store in flash, so you would host it on a server and boot from that.

The commands differ slightly depending on which boot options you want to configure. Try all of the options on your own Cisco router. It's important for the exam and in the real world that you know which options you have to boot your router from. Flash is the most common but you can also use a TFTP server if you are directly connected.

Router(config)#boot system ?

WORD TFTP filename or URL

flash Boot from flash memory

ftp Boot from a server via ftp

mop Boot from a Decnet MOP server
rcp Boot from a server via rcp
tftp Boot from a tftp server

For flash:

```
Router(config)#boot system flash ?  
WORD System image filename  
[cr]
```

For TFTP:

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#boot system tftp c1841-adventerprisek9-mz.151-4.M7.bin?
Hostname or A.B.C.D Address from which to download the file
[cr]

```
Router(config)#boot system tftp c1841-adventerprisek9-mz.151-4.M7.bin
```

For USB:

```
Router(config)#boot system flash usbflash0: c1841-adventerprisek9-mz.151-4.M7.bin
```

The router will usually boot the system IOS from flash memory or a TFTP server if need be.

Cisco IOS Licensing

The IOS software that runs on Cisco routers has different levels of functionality. The features enabled on this software (the router operating system) are priced differently depending on the users' needs. Some companies will need advanced IP routing features, others will require advanced voice and video, and others will require cryptographics and enhanced security.

Prior to IOS version 15, these different functionalities were grouped in a number of feature sets that could be purchased by buying the specific license from Cisco:

- IP Base (default)
- IP Voice
- Advanced Security
- SP Services
- Enterprise Base
- Advanced IP Services
- Enterprise Services

- Advanced Enterprise Services (full feature set)

You might argue that you need all these feature sets, considering you can have all the features enabled with the Advanced Enterprise Services (AES) license. However, the AES feature set is the most expensive and most organizations do not need all the features included in that feature set. A company might only need a few features covered in the Advanced IP Services license, for example, so it can purchase a limited functionality feature set, paying less for the license.

With standard IOS-based routers (ISR G2), starting with IOS version 15, there are only four feature sets available (referred to as technology packages) that come in a single Universal IOS image:

- IP Base – the default entry-level package
- DATA – includes advanced data features like MPLS
- Unified Communications – includes advanced IP telephony features
- Security – includes advanced security features like IPS and VPN

Each of these feature sets can be activated based on a license. Licenses can be of two types:

- Evaluation license – functions for a trial period of 60 days
- Permanent license

To activate a specific technology package license on top of the standard Universal IOS, you have to purchase the software package you want to install in order to receive a Product Activation Key (PAK). The PAK key will be converted to a license number (.lic file) on the Cisco website and this license number will be installed on the router to activate a specific feature set.

If you want to see the licenses available on the router, you can use the following command:

Router#show license all

License Store: Primary License Storage

StoreIndex: 0 Feature: ipbasek9 Version: 1.0

License Type: Permanent

License State: Active, In Use

License Count: Non-Counted

License Priority: Medium

License Store: Primary License Storage

License Type: Permanent

License State: Active, In Use

License Count: Non-Counted

License Priority: Medium

To install a new license on an IOS router, you have to first upload the .lic file to the router's flash memory using a TFTP server. After this step, you need to issue the following command:

```
Router#license install flash0:[license_file_name]
```

To uninstall a specific license, you need to disable it first using the following command:

```
Router(config)#license boot module c3900 technology-package [package_name]  
disable
```

And then clear the license:

```
Router#license clear [package_name]
```

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 12 exam.

Chapter 12 Labs

Lab 1: Copy Startup Config Using TFTP

The physical topology is shown in Figure 12.6 below:

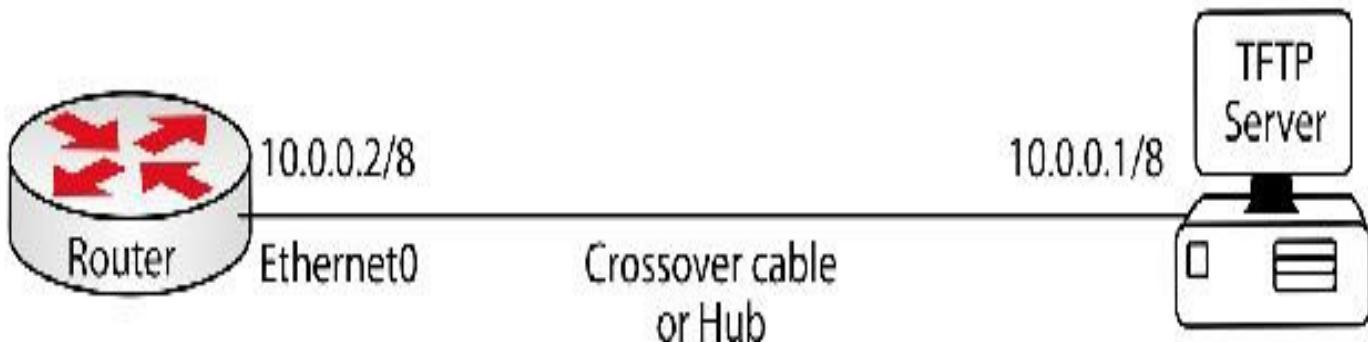


FIG 12.6 – TFTP lab

Lab Exercise

Your task is to configure the IP addressing as specified in Figure 12.6.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Backing up the router's configuration is a crucial part of your backup and disaster avoidance procedures. You will also need to use a TFTP server if you want to upgrade your router's IOS. Familiarity with using a TFTP server is a fundamental skill for a Cisco engineer.

Lab Objectives

1. Configure the router's Ethernet interface.
2. Put TFTP software onto your PC.
3. Connect the PC and router with a crossover cable or using a hub or switch.
4. Ping across the Ethernet link.
5. Copy the startup configuration from the router to the TFTP server.

Lab Walk-through

1. Configure the network shown in Figure 12.6. If you need help, look at some of the other labs you have already configured.

```
Router#config t
```

```
RouterA(config)#interface FastEthernet0
```

```
RouterA(config-if)#ip address 10.0.0.2 255.0.0.0
```

```
RouterA(config-if)#no shut
```

2. Install TFTP software onto your PC, making it a TFTP server. You can find this software at websites such as www.solarwindsuk.net. Install the software on the root of your C drive. Alternatively, use a server inside Packet Tracer and turn on TFTP.

Make sure the PC and the router are both on the same subnet. Change the IP address of the PC to 10.0.0.1 255.0.0.0.

Ping the PC from the router to confirm IP connectivity.

```
Router#ping 10.0.0.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms

Copy the startup configuration to the TFTP server:

```
Router#ping 10.0.0.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

```
Router#copy start tftp:
```

Address or name of remote host []? 10.0.0.1

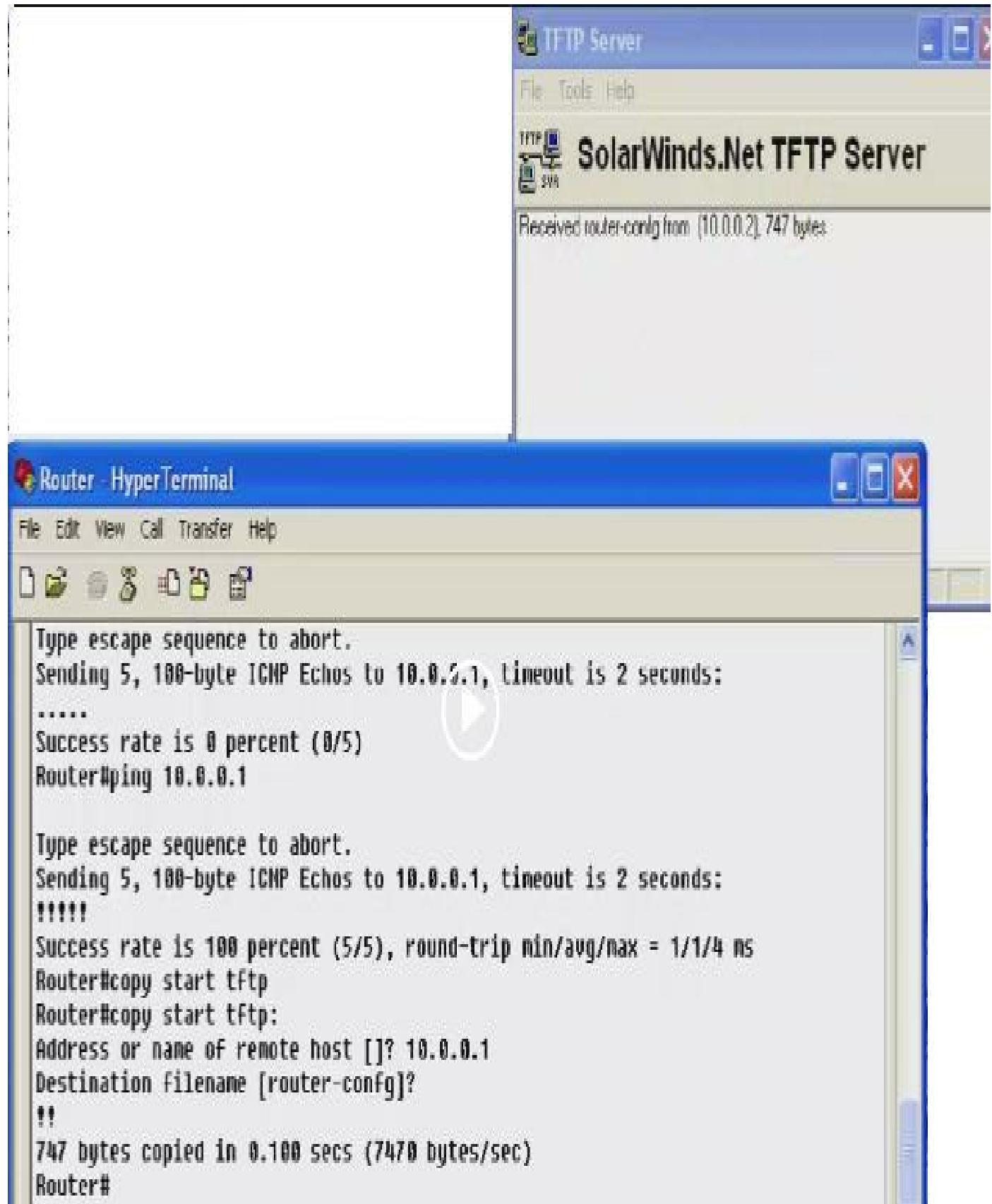
Destination filename [router-config]?

!!

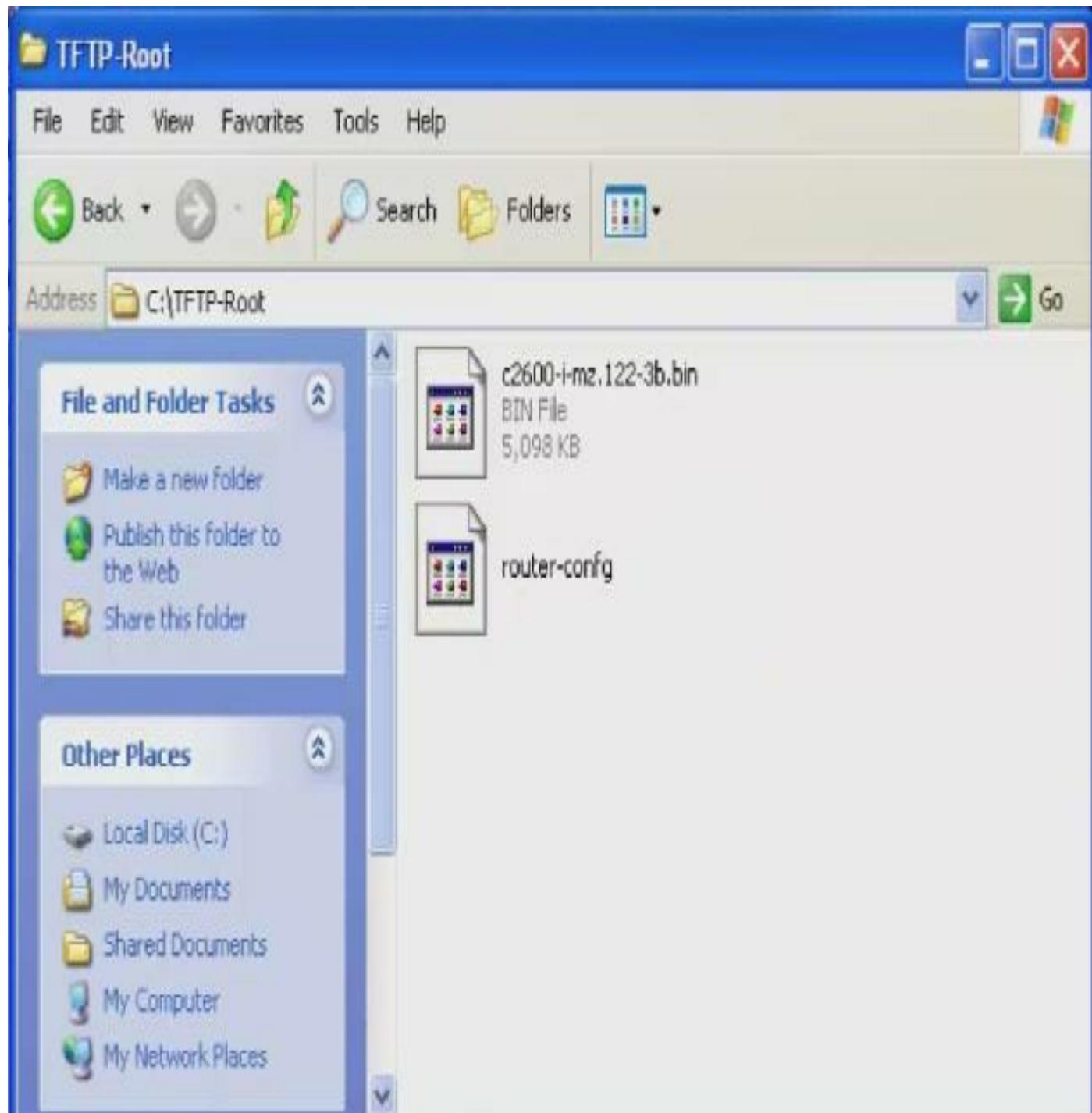
747 bytes copied in 0.256 secs

```
Router#
```

3. Check the TFTP log to make sure that the file has been received.



You can look for the configuration file in Windows Explorer.



4. Reload the router. You can use the copy tftp: start command:

```
Router#ping 10.0.0.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

```
Router#copy tftp: start
```

```
Address or name of remote host []? 10.0.0.1
```

```
Source filename []? router-config i Note the spelling
Destination filename [startup-config]? i Just press Enter here
Accessing tftp://10.0.0.1/router-config...Accessing tftp://10.0.0.1/router-config...
Loading router-config .from 10.0.0.1 (via Ethernet0): !
[OK - 421/4096 bytes]
[OK]
747 bytes copied in 37.980 secs (11 bytes/sec)
Router#
00:18:04: %SYS-5-CONFIG_NV_I: Nonvolatile storage configured from
tftp://10.0.0.1/router-config by console
```

ALWAYS, ALWAYS NAME THE ROUTER'S STARTUP CONFIGURATION AS startup-config. DOING OTHERWISE WILL PREVENT THE ROUTER FROM BOOTING CORRECTLY.

Chapter 13 — Advanced OSPF Features

What You Will Learn in This Chapter

Advanced OSPF Concepts

Syllabus Topics Covered

2.5 Configure and Verify Multi-area OSPF

- 2.5.a Neighbor adjacencies
- 2.5.b OSPF states
- 2.5.c Configure OSPFv2
- 2.5.d Configure OSPFv3
- 2.5.e Router ID (covered in ICND1)
- 2.5.f Understand LSA types and purposes

In previous versions of the CCNA exam, you were expected to have only a cursory understanding of OSPF. The latest version now requires you to have a very good grounding in this subject. You must now understand how OSPF establishes adjacencies with neighbors as well as LSA types. This is for both single- and multi-area OSPF.

Much of this chapter is actually included in the CCNP ROUTE exam syllabus, so if you find that the material is difficult it's understandable. But, as with anything, stick with it, type out any IOS commands you see, go through the labs, and it will all start to make sense.

Advanced OSPF Concepts

OSPF is designed to quickly locate changes in the network topology and calculate the best path available for network traffic to then take. The entire OSPF network is divided into areas. Within these areas, specific link state advertisements (LSAs) are flooded to neighbor routers. This ensures that routers in other OSPF areas will not be affected by changes or problems such as links flapping.

Unlike RIP, OSPF operates within a hierarchy of areas (which is explained below). At the highest level is the autonomous system (AS), which is a logical construct for a collection of networks under a common administration. OSPF is an intra-AS (interior gateway) routing protocol. An exterior routing protocol exchanges routing information between autonomous systems. An example of this is BGP, which is outside the CCNA RS syllabus but is covered in detail in the CCNP ROUTE exam.

An AS can be divided into a number of areas with continuous network addressing,

depending on the customer's requirements. Routers with multiple interfaces can participate in multiple areas if required per design. These routers are called area border routers (ABRs) and they maintain separate topological databases for each area. We will cover OSPF router types shortly.

Because OSPF uses flooding to advertise LSAs, areas are introduced to limit how far the LSAs are flooded across the network (some LSAs are only flooded within an area). Every router within the same area will hold the same routing database. A router with all of its interfaces in the same area is known as an internal router.

OSPF will only function on a router with at least one active interface (that is, an interface showing as up/up). OSPF will send out Hello packets from the interface using multicast address 224.0.0.5, which is also called the AllSPFRouter address. If the OSPF link is on an NBMA link (such as Frame Relay) the OSPF packet will be unicast rather than multicast.

When the OSPF packet is verified by the other router, a neighbor relationship is formed between the two routers. Each router floods its link state database to every other OSPF router, and in this way a loop-free path to every route is built.

A topological database is essentially an overall picture of the network and it contains the collection of LSAs received from all routers in the same area ONLY. Because routers within the same area share the same information, they have identical topological databases.

When multiple areas are created, two different types of OSPF routes can be seen, depending on whether the source and the destination are in the same or different areas. Intra-area routing occurs when the source and destination networks (or subnets) are in the same area and inter-area routing occurs when they are in different areas.

An OSPF backbone (also known as area 0) is responsible for distributing routing information between other non-zero areas. All routers with at least one interface in area 0 are known as backbone routers.

Any router with all of its interfaces within the same area is known as an internal router (IR). Any router acting as a connection between routers running other routing protocols or instances of OSPF is known as an autonomous system boundary router (ASBR). Any router with interfaces in more than one area (area 0 and another area) is known as an area border router (ABR). More details on OSPF router types are provided in the following sections.

Designated Router and Backup Designated Router

On broadcast networks such as Ethernet (known as multi-access), it would not be

efficient for OSPF to flood the links with advertisements to every one of its neighbors. Neither would it be efficient for every router to become adjacent. This is because each router in the network would need to send LSAs to all the other routers and this consumes resources like memory, CPU, and bandwidth.

Without a designated router (DR) (and an optional backup designated router, or BDR) in a broadcast network, you would have the situation you see in Figure 13.1 below:

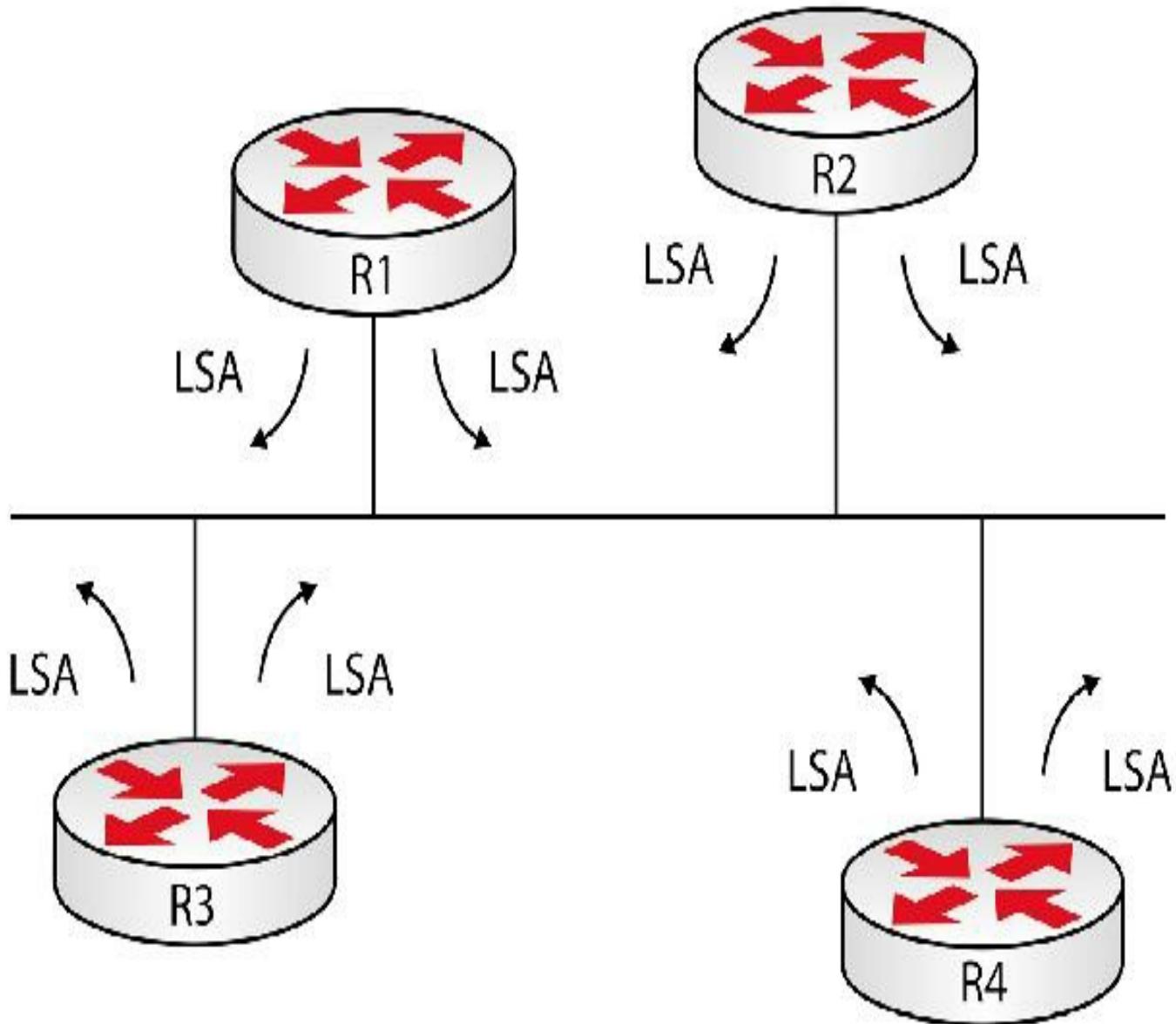


FIG 13.1 – Broadcast network without a DR

There would be $n(n-1)/2$ adjacencies formed, so in the network in Figure 13.1 above you would require six. If you had a small business using only eight routers you would require 28. Each of these routers would be exchanging $n-1$ LSAs, leading to a large amount of your bandwidth being used purely for OSPF traffic. Once you add a DR, you have the situation you see in Figure 13.2 below:

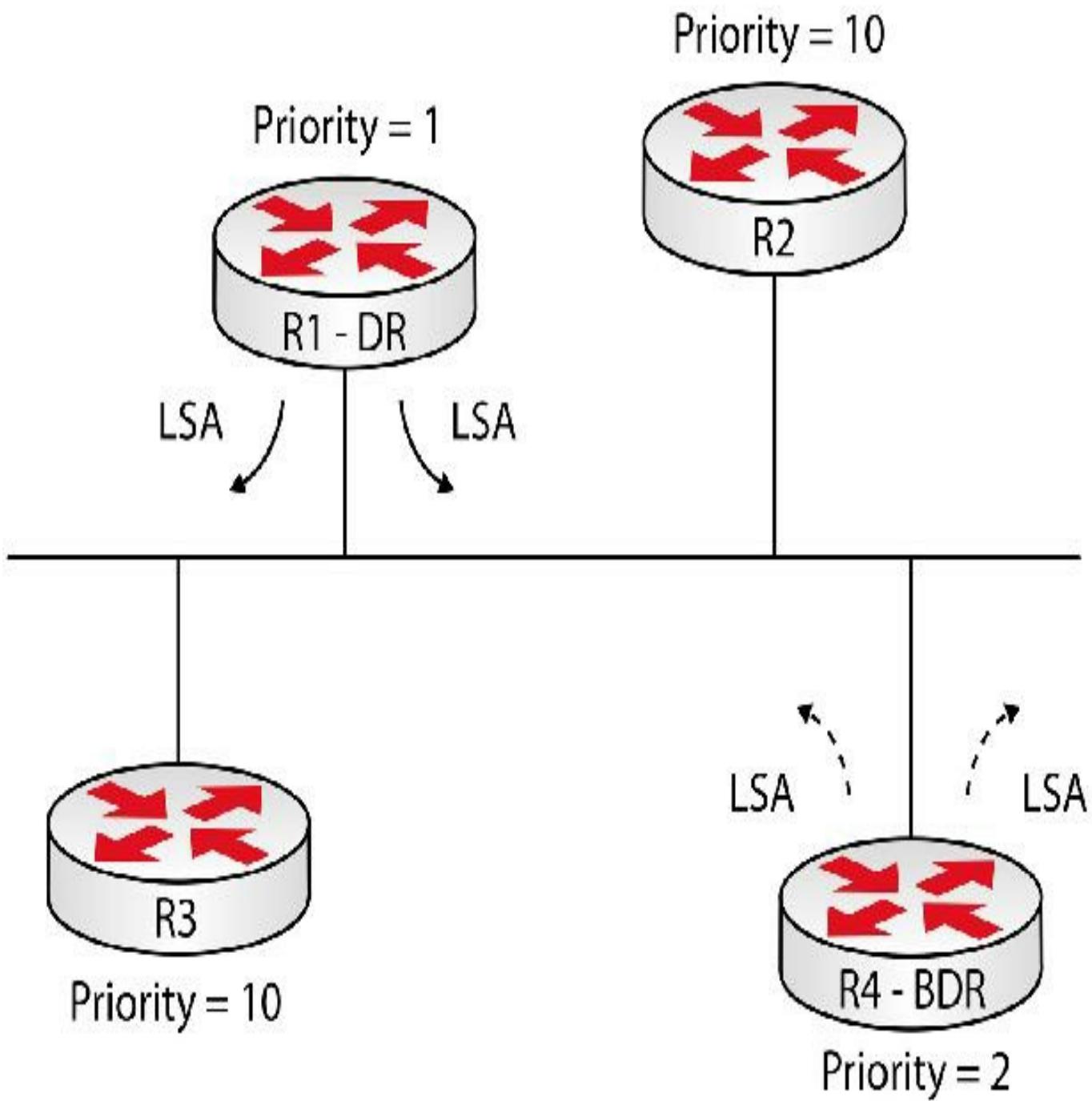


FIG 13.2 – Broadcast network with a DR

In this instance, OSPF will elect one router as the DR, which will listen to LSAs on multicast address 224.0.0.6 and flood them on 224.0.0.5. In addition to the DR, a BDR is usually elected, which will take over the role of the DR should the DR fail for any reason. Any new OSPF routers will form an adjacency only with the DR and the BDR, which creates redundancy.

The DR and BDR are elected when the OSPF process starts. The router with the highest OSPF priority is selected. If the OSPF priority is the same, then the router with the highest Router ID (RID) is elected. Once they have been elected, even if a router with a

higher priority joins the network, a replacement will not be selected. They will only be replaced in the event that the DR and the BDR routers fail.

Once elected, OSPF uses the DR and the BDR routers as follows:

- To reduce the number of adjacencies required on the segment
- To advertise the routers on the multi-access segment
- To ensure that updates are sent to all routers on the segment

Any router not elected DR or BDR is listed as DROther, and these routers will establish an adjacency only with the DR and BDR.

This all takes place per multi-access network, so in the network below you have two DRs and two BDRs (one for each broadcast domain).

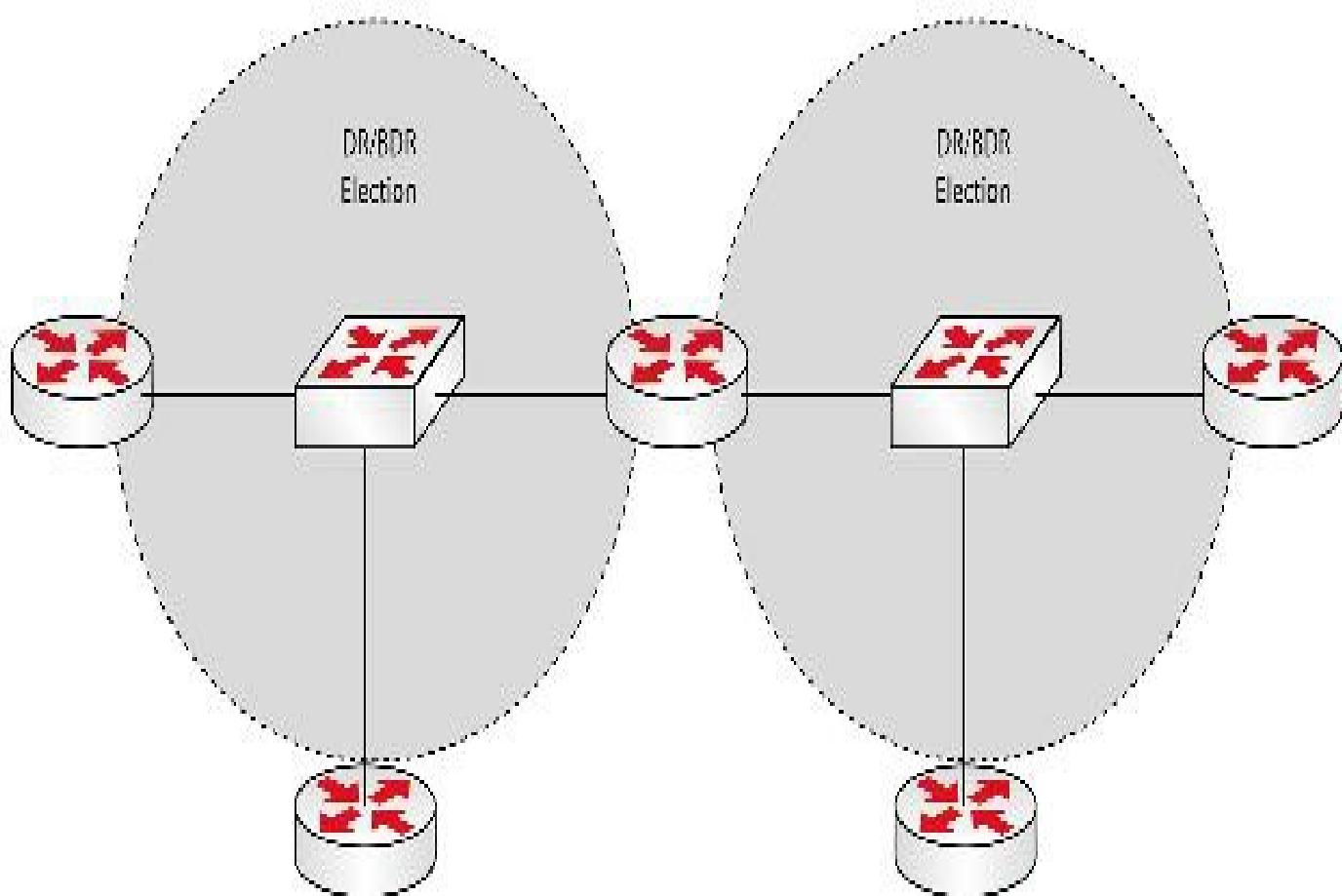


FIG 13.3 – A DR/BDR for each multi-access segment

Establishing Adjacencies

When routers are configured for OSPF, they transition through various states before an adjacency is established. The routers exchange different information at each state and different criteria have to be met before they advance to the next state. The different states include Down, Attempt, Init, 2-Way, Exstart, Exchange, Loading, and Full, as

described below:

1. The Down state is the first OSPF state; Hello packets haven't been received from a neighbor on that interface.
2. The Attempt state is valid only in non-broadcast networks. In this state, Hello packets have been sent but no information has been received from the configured neighbor.
3. The Init state is reached when the OSPF Hello packets are received from a neighbor. To proceed beyond this state, some parameters such as the OSPF area, timer values, and authentication must match.
4. The 2-Way state indicates that bidirectional communication has been established between the OSPF neighbor(s). A router transitions to this state when it has received a Hello packet with its own RID in the Neighbor field and the Hello packet parameters match on the two routers. In multi-access networks, the DR and BDR routers are elected during this phase. OSPF neighbors between non-DR/BDR routers (two DRothers) do not proceed beyond this state.
5. The initialization of database synchronization happens in the Exstart state. The neighbors elect a Master and a Slave. OSPF routers exchange database descriptor (DBD) packets. Database descriptors contain only the headers of the LSA. They describe the contents of the entire link state database.
6. The Exchange state is where routers describe their link state databases using DBD packets. Each DBD packet must be explicitly acknowledged, and the sending router allows only one outstanding DBD at a time. Routers also use LSR (link state request) packets to request an instance of the LSA. When requesting missing information, the M (More) bit is set to indicate that there is some missing information. When the database exchange is completed, the M bit is set to 0.
7. In the Loading state, OSPF routers send LSR packets to request more recent instances of LSAs that have not been received during the Exchange state. The updates sent are placed in a link state retransmission link until acknowledgments are received. When OSPF routers receive LSRs, they respond with an LSU (link state update) containing the required information.
8. The Full state shows that the databases are fully exchanged and both neighbors have the same view of the network. At this point, the relationships are added to the local database and advertised in an LSU packet. Also, best routes are calculated using Djisktra's algorithm and added to the routing table.

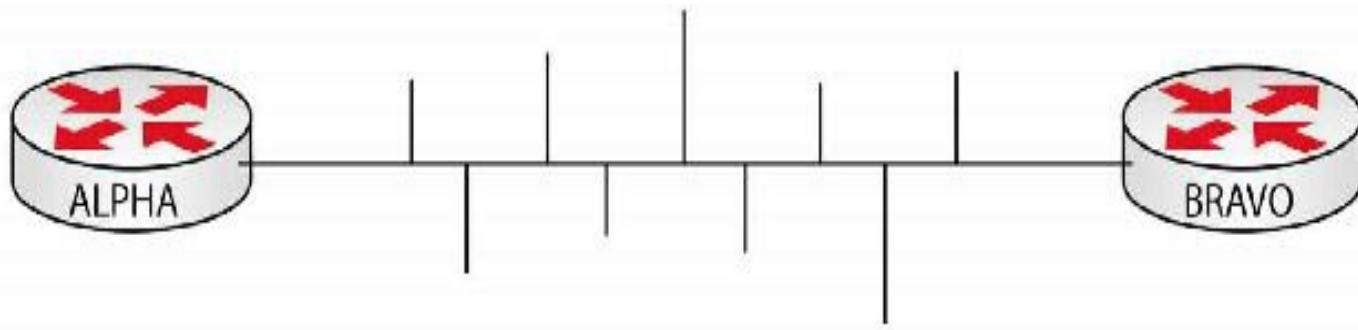


FIG 13.4 – The adjacency process

You can debug the OSPF adjacency process with the debug ip ospf adj command, but be cautious because it can generate a huge amount of traffic. Note how I turn off all debugs at the bottom of the following output:

```
R1#debug ip ospf adj
```

OSPF adjacency events debugging is on

```
R1#clear ip ospf process
```

Reset ALL OSPF processes? [no]: y

```
*Mar 1 01:24:22.687: OSPF: Interface FastEthernet0/0 going Down
```

```
*Mar 1 01:24:22.687: OSPF: 1.1.1.1 address 192.168.1.1 on FastEthernet0/0 is dead,  
state DOWN
```

[output truncated]

```
*Mar 1 01:24:24.243: OSPF: Neighbor change Event on interface FastEthernet0/0
```

```
*Mar 1 01:24:24.243: OSPF: DR/BDR election on FastEthernet0/0
```

```
*Mar 1 01:24:24.243: OSPF: Elect BDR 1.1.1.1
```

```
*Mar 1 01:24:24.247: OSPF: Elect DR 192.168.1.2
```

```
*Mar 1 01:24:24.247: DR: 192.168.1.2 (Id) BDR: 1.1.1.1 (Id)
```

```
*Mar 1 01:24:27.755: OSPF: Rcv LS UPD from 192.168.1.2 on FastEthernet0/0 length  
64 LSA count 1
```

```
*Mar 1 01:24:27.795: OSPF: Rcv LS UPD from 192.168.1.2 on FastEthernet0/0 length  
64 LSA count 1
```

```
R1#
```

```
*Mar 1 01:24:29.747: OSPF: Build router LSA for area 0, router ID 1.1.1.1, seq  
0x80000004, process 1
```

```
R1#un all
```

All possible debugging has been turned off

```
R1#
```

OSPF Priority

Although you can force the router to become a DR/BDR election with the router-id command or by setting a high Loopback address, the preferred method is to use the

interface command ip ospf priority. Please refer back to the ICND1 section on OSPF if you need to refresh your understanding of the Router ID.

```
R1(config)#int f0/0
```

```
R1(config-if)#ip ospf priority ?
```

[0-255] Priority

The command above will have no effect if the DR/BDR election has already taken place. The default priority is 1, and if you don't want the interface to take part in the DR/BDR election, you can set it to 0. If you wanted it to be the DR, you would set it to a high priority on the segment, such as 200, and leave the others as default.

```
R1#show ip ospf int f0/0
```

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.1/24, Area 0

Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 10

Transmit Delay is 1 sec, State DR, **Priority 1**

You can actually force a DR/BDR election on a segment using the clear ip ospf process command. Of course, you would never use this command on a live network unless you have the necessary permission and a network outage is scheduled.

```
R1#clear ip ospf process
```

Reset ALL OSPF processes? [no]: yes

*Mar 1 01:19:26.711: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on FastEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached

*Mar 1 01:19:26.831: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on FastEthernet0/0 from LOADING to FULL, Loading Done

OSPF Router Types

OSPF routers are described by their location and function in the hierarchical OSPF network. Some of the common OSPF router types include:

- Area border routers
- Autonomous system boundary routers
- Internal routers
- Backbone routers

Figure 13.5 below shows a basic OSPF network with two areas, the OSPF backbone area 0 and another normal OSPF area (area 2). R2 has an external BGP (Border

Gateway Protocol) neighbor relationship with R1. This diagram illustrates the different OSPF router types in a network. BGP isn't actually covered in the CCNA RS exam but it's useful to refer to it to illustrate an example of router types.

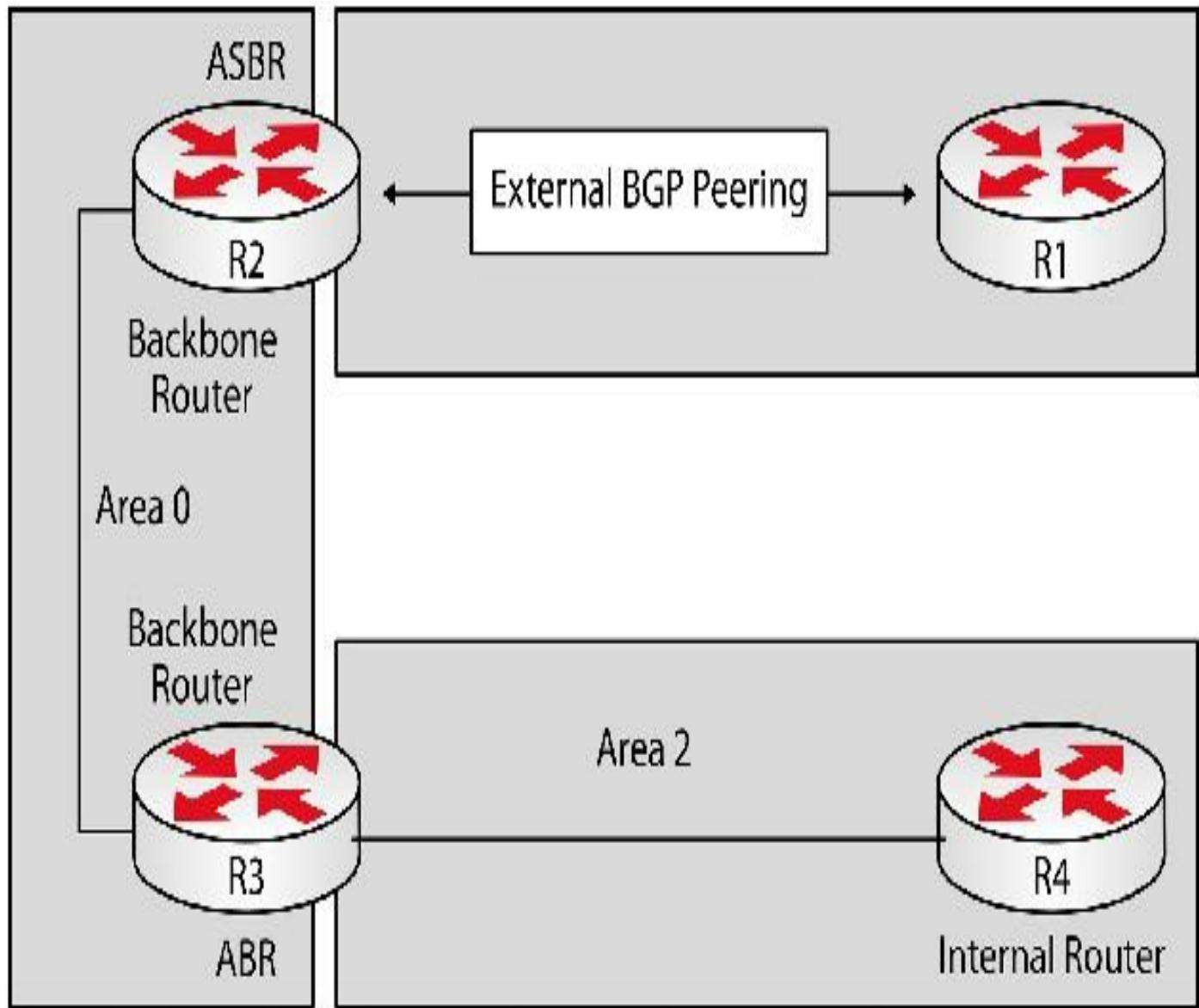


FIG 13.5 – Additional OSPF router types

The ABR connects one or more area(s) to the OSPF backbone area (i.e., area 0). An ABR must have at least one interface in area 0 and another interface in a different area. Since ABRs are members of different areas, they keep a separate SPF database for each of the areas they connect to and summarize the link state information between the areas. An ABR should be a high-end model for this reason. In Figure 13.5 above, R3 is an ABR that connects area 0 and area 2.

The ASBR defines the boundary of an OSPF autonomous system. In the context of OSPF this would be any router that is injecting routes from a different route source into OSPF. A different route source can be another routing protocol or even a different OSPF

process. In Figure 13.5, R2 is an ASBR since it has interfaces in both OSPF and BGP. Internal routers have all their interfaces in a single OSPF area, but it doesn't have to be in area 0. R4 in Figure 13.5 is an internal router since its only interface resides in area 2.

Backbone routers have at least one interface in area 0. In Figure 13.5, R2 and R3 are backbone routers.

As shown in Figure 13.5, OSPF routers can have multiple roles. For instance, R2 is both an ASBR and a backbone router, while R3 is both an ABR and a backbone router.

OSPF Link State Advertisements

OSPF uses link state advertisements to notify its neighbors about changes in the network. LSAs are used to build the OSPF database. There are many kinds of OSPF LSAs but we will only focus on the ones required for the CCNA exam. For reference, Figure 13.6 below shows OSPF in three areas, which are already populated with network addresses.

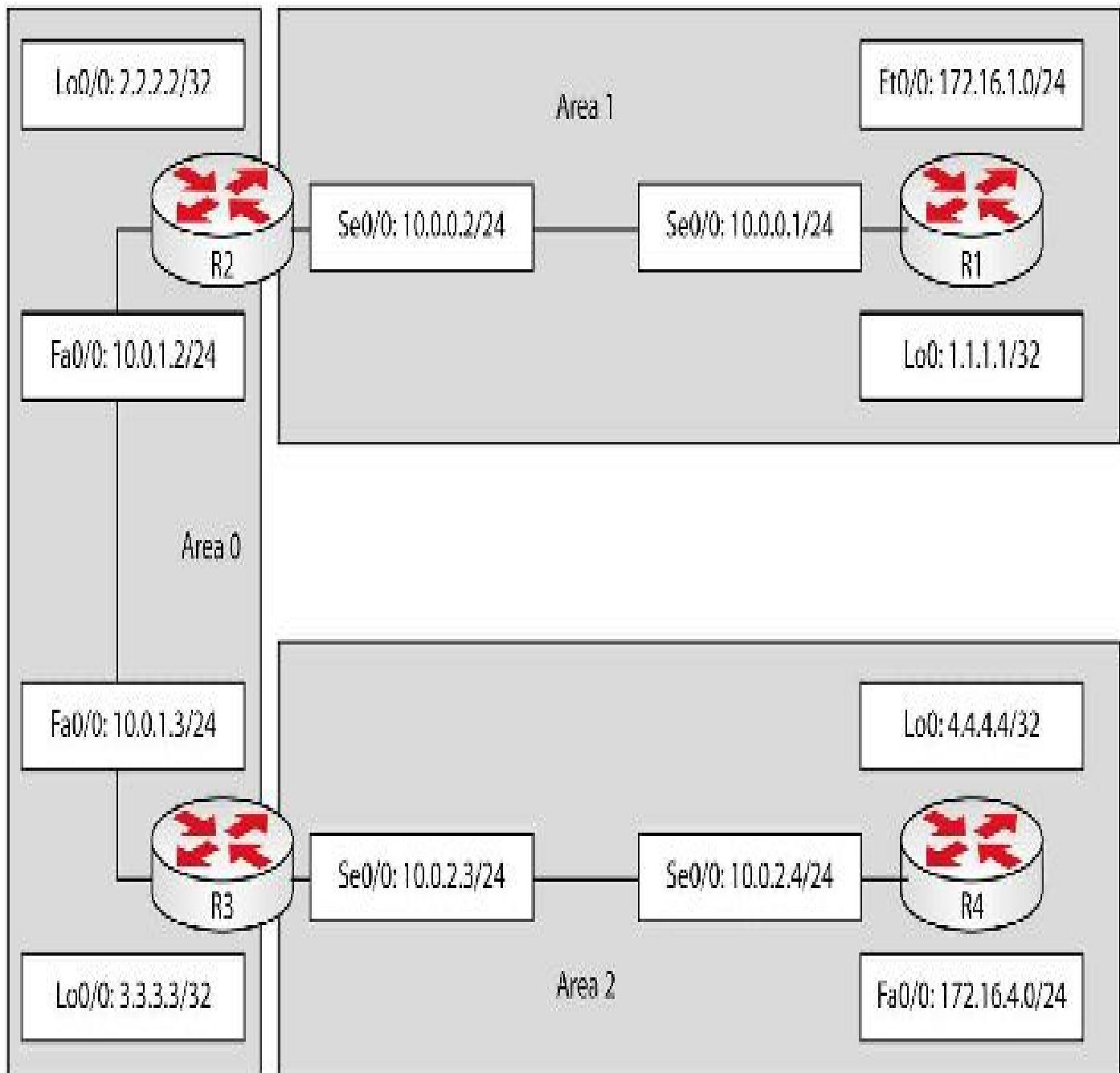


FIG 13.6 – Understanding OSPF LSAs

You can view the contents of the link state database using the `show ip ospf database` command. This command prints out a summary of the LSAs. You can further specify the level of detail by adding specific keywords, which can be viewed using the `?`

R1#`show ip ospf database ?`

adv-router	Advertising Router link states
asbr-summary	ASBR Summary link states
database-summary	Summary of database
external	External link states

network	Network link states
nssa-external	NSSA External link states
opaque-area	Opaque Area link states
opaque-as	Opaque AS link states
opaque-link	Opaque Link-Local link states
router	Router link states
self-originated	Self-originated link states
summary	Network Summary link states
	Output modifiers

[cr]

Some of the common OSPF LSA types include:

- LSA Type 1 (Router LSA)
- LSA Type 2 (Network LSA)
- LSA Type 3 (Summary LSA)
- LSA Type 4 (ASBR Summary LSA)
- LSA Type 5 (External Summary LSA)

When you issue the show ip ospf database command, you won't see the LSA number but you will see the type, such as "Router link states" for LSA Type 1 or "Summary net link states" for LSA Type 3.

Router Link State Advertisements (Type 1)

Type 1 LSAs are generated by each router type, from backbone and stub to NSSA and non-stub. The LSAs list the originating router's RID. This LSA contains the directly connected links in that area. For each link type, the LSA contains a Link ID and an ADV router, which is the advertising RID. Each router in an area floods Type 1 LSAs throughout the area. If a router is in multiple areas, it will generate a Type 1 LSA for each area that it is connected to.

For example, R2 and R3 would generate Type 1 LSAs for area 0, as shown below:

R3#show ip ospf database

OSPF Router with ID (3.3.3.3) (Process ID 3)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link Count
2.2.2.2	2.2.2.2	704	0x80000005	0x0048A2	2

3.3.3.3 3.3.3.3 424 0x80000004 0x003AA4 2

[output truncated]

In the output above, Age is the maximum age of the link, and Seq# and Checksum verifies the link state integrity. The flooding scope of a Type 1 LSA is within the area, so Types 1 LSAs generated in an area are not flooded to the next area. To view Type 1 LSAs you can use the show ip ospf database router command. There are more in-depth details on Type 1 LSAs, but those are beyond the scope of the CCNA exam.

Network Link State Advertisements (Type 2)

Type 2 LSAs are used to advertise routers on a multi-access segment, such as an Ethernet network. The LSA is originated by the DR and flooded within the area (the flooding scope is within the area, just like for Type 1). The network allows the routers in an area to know about all the routers on a multi-access segment.

The Type 2 LSA is generated by the DR and includes the Link ID, which is the DR address, the mask, and the Router IDs of the routers in that network (attached routers). Type 2 information can be viewed using the show ip ospf database network command. The output of the command on R3 is shown below

R3#show ip ospf database network

OSPF Router with ID (3.3.3.3) (Process ID 3)

Net Link States (Area 0)

Routing Bit Set on this LSA

LS age: 248

Options: (No TOS-capability, DC)

LS Type: Network Links

Link State ID: 10.0.1.2 (address of Designated Router)

Advertising Router: 2.2.2.2

LS Seq Number: 80000008

Checksum: 0x8E7B

Length: 32

Network Mask: /24

Attached Router: 2.2.2.2

Attached Router: 3.3.3.3

Summary Link State Advertisements (Type 3)

Type 3 Summary LSAs provide information about destinations outside the local area (i.e., inter-area routing information). Unlike Type 1 LSAs, Type 3 LSAs do not provide topological information; instead, they summarize the topological information received in Type 1 and 2 LSAs into a prefix and its associated cost. Type 3 LSAs are generated by the ABR and the flooding is described as follows:

- Type 3 LSAs are advertised from a non-backbone area to area 0 for each intra-area route (LSA Type 1 and Type 2)
- Type 3 LSAs are flooded from OSPF area 0 to other non-backbone areas for both intra-area routes within area 0 and inter-area routes (Type 2 LSAs that are received from other areas)

We will explore Type 3 LSAs with the same topology used in previous examples:

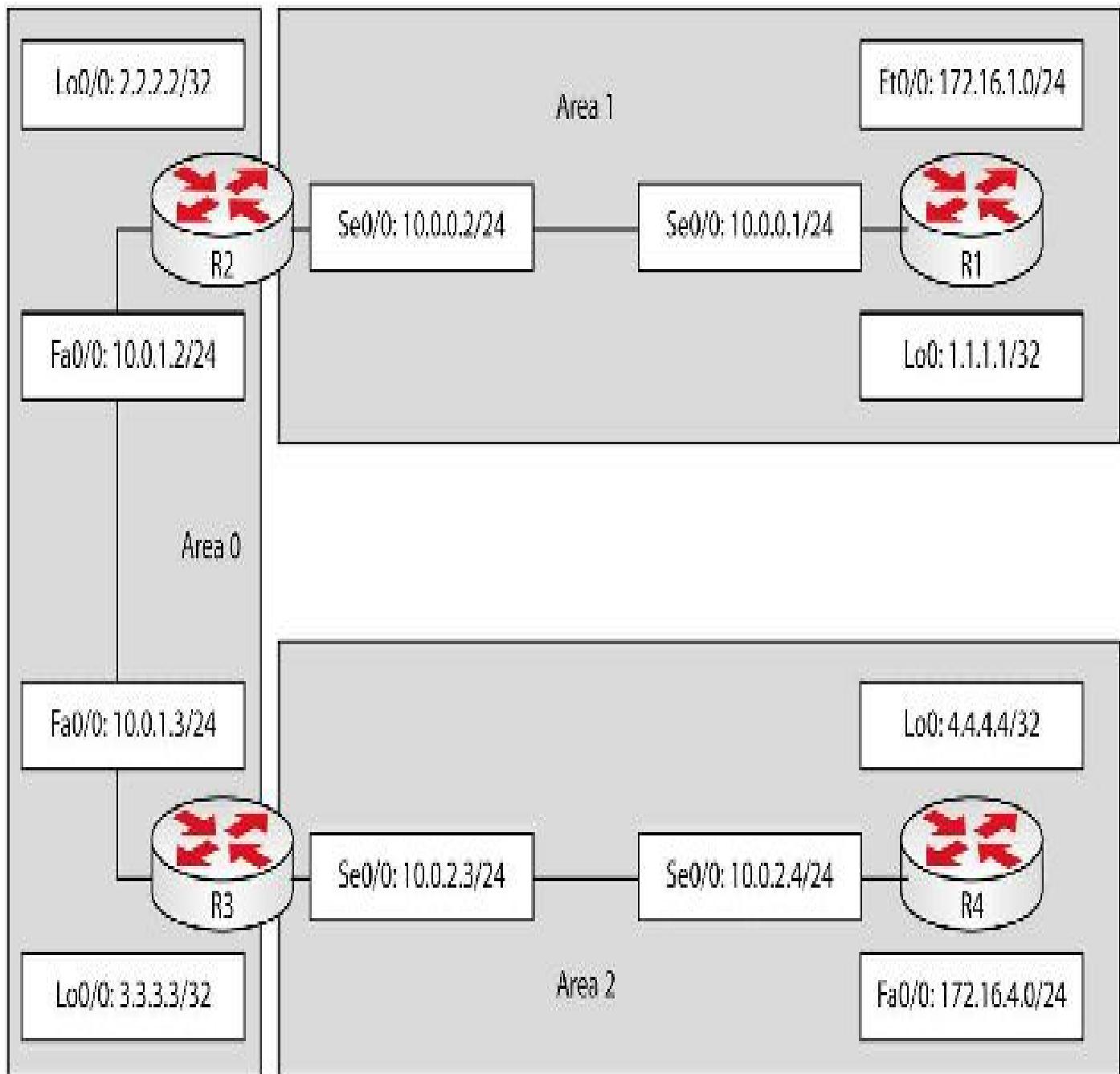


FIG 13.7 – Understanding OSPF LSAs

The `show ip ospf database summary [options]` command can be used to display the Summary LSAs in the link state database (LSDB). The `show ip ospf database summary` command provides detailed information on each Type 3 LSA. The outputs include the Link ID, the network mask, and the cost. A sample output is shown below:

R4#`show ip ospf database summary`

OSPF Router with ID (4.4.4.4) (Process ID 4)

Summary Net Link States (Area 2)

Routing Bit Set on this LSA

LS age: 1612

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 1.1.1.1 (summary Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000001

Checksum: 0x9753

Length: 28

Network Mask: /32

TOS: 0 Metric: 66

Routing Bit Set on this LSA

LS age: 1612

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 2.2.2.2 (summary Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000001

Checksum: 0xE640

Length: 28

Network Mask: /32

TOS: 0 Metric: 2

Routing Bit Set on this LSA

LS age: 1677

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 3.3.3.3 (summary Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000001

Checksum: 0xAE75

Length: 28

Network Mask: /32

TOS: 0 Metric: 1

Routing Bit Set on this LSA

LS age: 1487

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 10.0.0.0 (summary Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000003

Checksum: 0x35AE

Length: 28

Network Mask: /24

TOS: 0 Metric: 65

[output truncated]

NOTE: The metric of the route shown in the output of the command is the distance from the ABR. The receiving router would then add its own interface metric to determine the overall metric of the route. You should keep this in mind to prevent confusion when viewing the LSDB.

You can zoom in on a particular link state ID using the show ip ospf database summary [Link State ID] command, as shown below:

R4#**show ip ospf database summary 172.16.1.0**

OSPF Router with ID (4.4.4.4) (Process ID 4)

Summary Net Link States (Area 2)

Routing Bit Set on this LSA

LS age: 76

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 172.16.1.0 (summary Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000002

Checksum: 0x8D99

Length: 28

Network Mask: /24

TOS: 0 Metric: 75

ASBR Summary Link State Advertisements (Type 4)

Type 4 LSAs are generated by the ABR and are used to provide information on how to reach an ASBR. This is different from Type 3, which provides information on how to reach prefixes in an area to another area.

To generate a Type 4 LSA, there must be an ASBR in the area. An ASBR usually injects routes from another domain into OSPF via redistribution. Route redistribution is beyond the scope of the CCNA exam.

External Summary Link State Advertisements (Type 5)

Type 5 LSAs provide information about destinations outside the OSPF domain. This means they provide information about routes learned from other routing sources outside OSPF. Type 5 LSAs are generated by the ASBR (the router that connects the OSPF domain to another routing source) and they are flooded across the entire domain by default.

As mentioned earlier, routes are injected from other routing sources into OSPF through a process known as redistribution, which is beyond the scope of the CCNA exam.

OSPF Areas

OSPF can burden router CPUs due to flooding and database maintenance requirements. Flooding also has a big impact on bandwidth. For this reason, OSPF has been designed to allow router interfaces to be placed into logical groups called areas. Routers within an area will not maintain detailed information about routers outside their specific area.

The advantages of grouping OSPF interfaces into logical areas include:

- Reduced size of the link state database, resulting in less impact on router memory
- Less impact on a router's CPU because of the smaller database
- LSA flooding is limited within an area, reducing bandwidth usage

OSPF specification defines special OSPF areas based on the types of LSAs that are allowed in the areas:

- Backbone (area 0)
- Stub areas
- Totally stubby areas
- Not-so-stubby areas
- Totally not-so-stubby areas
- Non-backbone, non-stub area

Backbone

All traffic must pass through the backbone and all areas must be connected to area 0. Area 0 cannot be partitioned (i.e., it must be continuous). You can extend area 0 with the use of virtual links.

Stub Areas

Stub areas do not allow external LSAs. In stub areas, Type 5 LSAs are not allowed into the area. This means that Type 5 LSAs originated from other areas are filtered from getting into a stub area by the ABR. This also means that there is no ASBR in that area. Since Type 4 LSAs are used to describe an ASBR, Type 4 LSAs are also not allowed in a stub area. To ensure that a stub area can reach external destinations from other areas, the ABR injects a summary default route into the stub area.

All routers in a stub area MUST be configured with the area [area-id] stub command. Consider the example below:

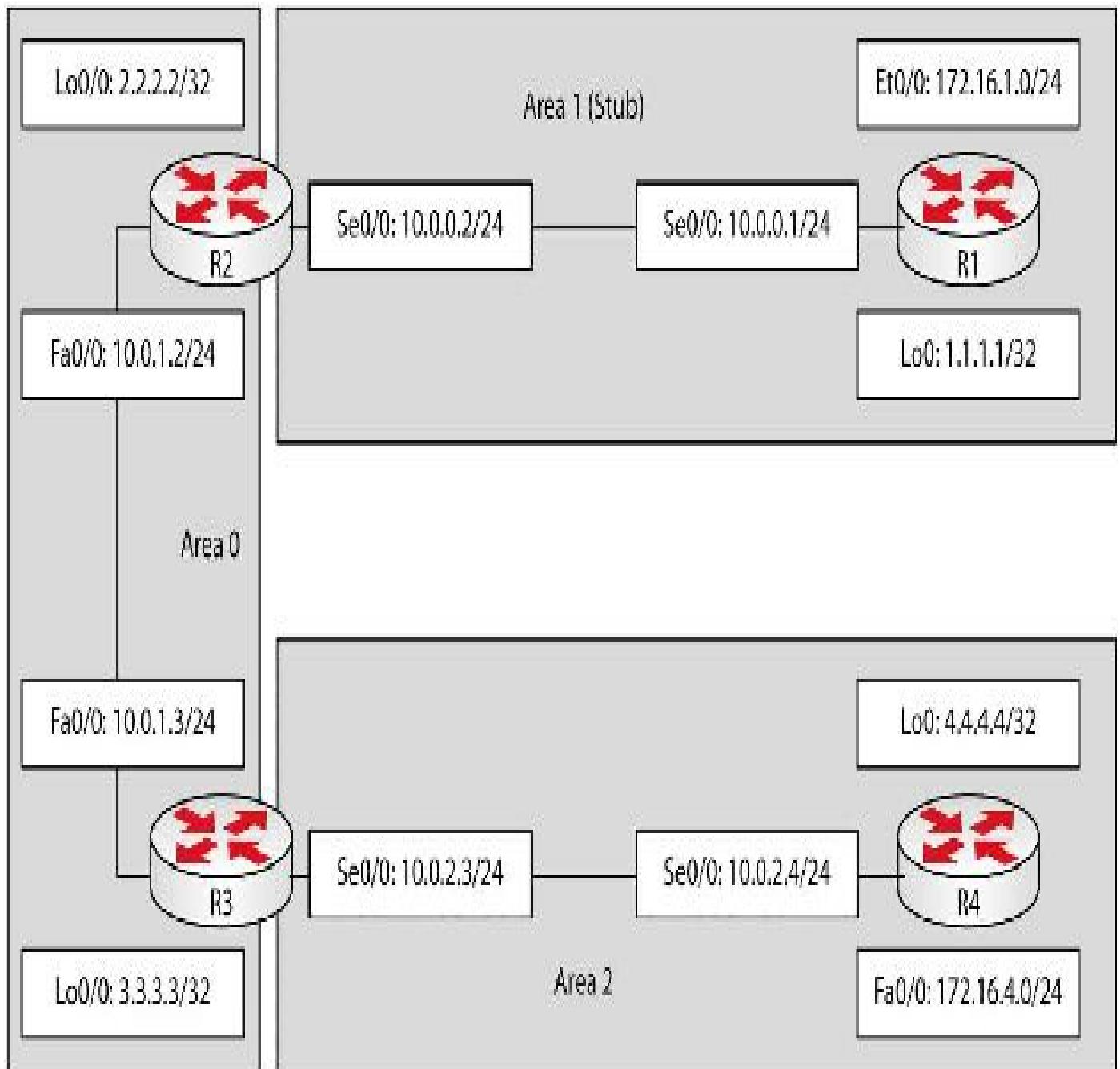


FIG 13.8 – OSPF stub area

To configure area 1 above as a stub area, the area 1 stub command is issued on R1 and R2:

```
R1(config)#router ospf 1
R1(config-router)#area 1 stub
R1(config-router)#exit
R2(config)#router ospf 2
R2(config-router)#area 1 stub
```

```
R2(config-router)#exit
```

Since R2 is the ABR, it originates a default summary route (Type 3 LSA) into area 1. You can see the route using the show ip ospf database summary 0.0.0.0 command:

```
R2#show ip ospf database summary 0.0.0.0
```

OSPF Router with ID (2.2.2.2) (Process ID 2)

Summary Net Link States (Area 1)

LS age: 199

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 0.0.0.0 (summary Network Number)

Advertising Router: 2.2.2.2

LS Seq Number: 80000004

Checksum: 0x6FC3

Length: 28

Network Mask: /0

TOS: 0 Metric: 1

Note that the inter-area (Type 3) Summary LSAs are not filtered in stub areas. You can confirm this by viewing the LSDB on R1.

```
R1#show ip ospf database
```

OSPF Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link Count
1.1.1.1	1.1.1.1	1829	0x80000011	0x008AD1	3
2.2.2.2	2.2.2.2	85	0x80000012	0x001852	2

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
0.0.0.0	2.2.2.2	85	0x80000005	0x006DC4
2.2.2.2	2.2.2.2	85	0x80000003	0x001517
3.3.3.3	2.2.2.2	85	0x80000003	0x00F036

4.4.4.4	2.2.2.2	85	0x80000003 0x00459D
10.0.1.0	2.2.2.2	85	0x80000003 0x00E345
10.0.2.0	2.2.2.2	85	0x80000003 0x005B8C
172.16.4.0	2.2.2.2	85	0x80000003 0x004CE5

Totally Stubby Areas

To further prune Type 3 LSAs from getting into an area, you can configure the ABR in a stub area to stop injecting Type 3 LSAs using the no-summary keyword in the area [area-id] stub command. This makes the OSPF area a totally stubby area. You can configure area 1 as a totally stubby area by including the no-summary keyword on R2:

```
R2(config)#router ospf 2
R2(config-router)#area 1 stub no-summary
R2(config-router)#exit
```

The result of this is that all the Type 3 LSAs will be filtered except for the default route, which will be the only way for the internal routers in a stub area to egress the area. You can confirm this by viewing the LSDB on R1:

```
R1#show ip ospf database
```

OSPF Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link Count
1.1.1.1	1.1.1.1	1288	0x80000012	0x0088D2	3
2.2.2.2	2.2.2.2	1579	0x80000012	0x001852	2

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
0.0.0.0	2.2.2.2	581	0x80000006	0x006BC5

The output above shows that the only summary LSA in area 1 is the default route originated by the ABR (2.2.2.2); all other LSAs have been pruned.

Not-so-stubby Areas

Not-so-stubby areas (NSSAs) are those that inject routes from another routing domain while suppressing external routes from other areas. Just like in stub areas, Type 4 and Type 5 LSAs are suppressed from getting into an NSSA. However, a new Type 7 LSA is used to inject external routing information into an NSSA. Type 7 LSAs are flooded

within the area and are converted to a Type 5 LSA by the ABR and then forwarded on to other areas.

All routers in an NSSA must be configured using the area [area-id] nssa command as shown below:

```
R2(config)#router ospf 1  
R2(config-router)#area 1 nssa  
R2(config-router)#exit
```

There are other details on the Type 7 LSA but they are beyond the scope of the CCNA exam. For further information about NSSAs, you can read RFC 1587 or a study guide for the CCNP Route exam.

Totally Not-so-stubby Areas

Like totally stubby areas, totally not-so-stubby areas (TNSSAs) are an extension of NSSAs, which further prune Type 3 LSAs. This means that LSA Types 3, 4, and 5 are pruned from the totally not-so-stubby areas. The only exception is the default Summary LSA (Type 3), which allows the TNSSA to reach the rest of the OSPF domain. In summary, TNSSAs have the following features:

- Type 7 LSAs are allowed into the area and are converted into Type 5 LSAs at the NSSA ABR
- Types 3, 4, and 5 LSAs are not allowed into the area
- The default route is injected as a Type 3 Summary LSA

Just as for totally stubby areas, the no-summary keyword is used to configure a TNSSA. This command is only required on the ABR. Once this command is applied, the Summary LSAs (inter-area routes) received in the NSSA would be filtered, and this makes the area a totally not-so-stubby area. The output below shows how this is configured; assume that Router 2 is the ABR:

```
R2(config)#router ospf 1  
R2(config-router)#area 1 nssa no-summary  
R2(config-router)#exit
```

Non-backbone, Non-stub Area

This is a normal OSPF area but not area 0. All LSAs except Type 7 are flooded into this area. Table 13-1 below shows a summary of the OSPF areas and LSA types:

Table 13-1: Summary of OSPF areas and LSA types

OSPF Areas and LSA Types

Areas	Type 1	Type 2	Type 3	Type 4	Type 5	Type 7
Backbone (Area 0)	Yes	Yes	Yes	Yes	Yes	No
Stub Area	Yes	Yes	Yes	No	No	No
Totally Stubby Area	Yes	Yes	No	No	No	No
Not-so-Stubby Area	Yes	Yes	Yes	Yes	No	Yes
Totally Not-so-stubby Area	Yes	Yes	No	No	No	Yes
Non-backbone Area	Yes	Yes	Yes	Yes	Yes	No

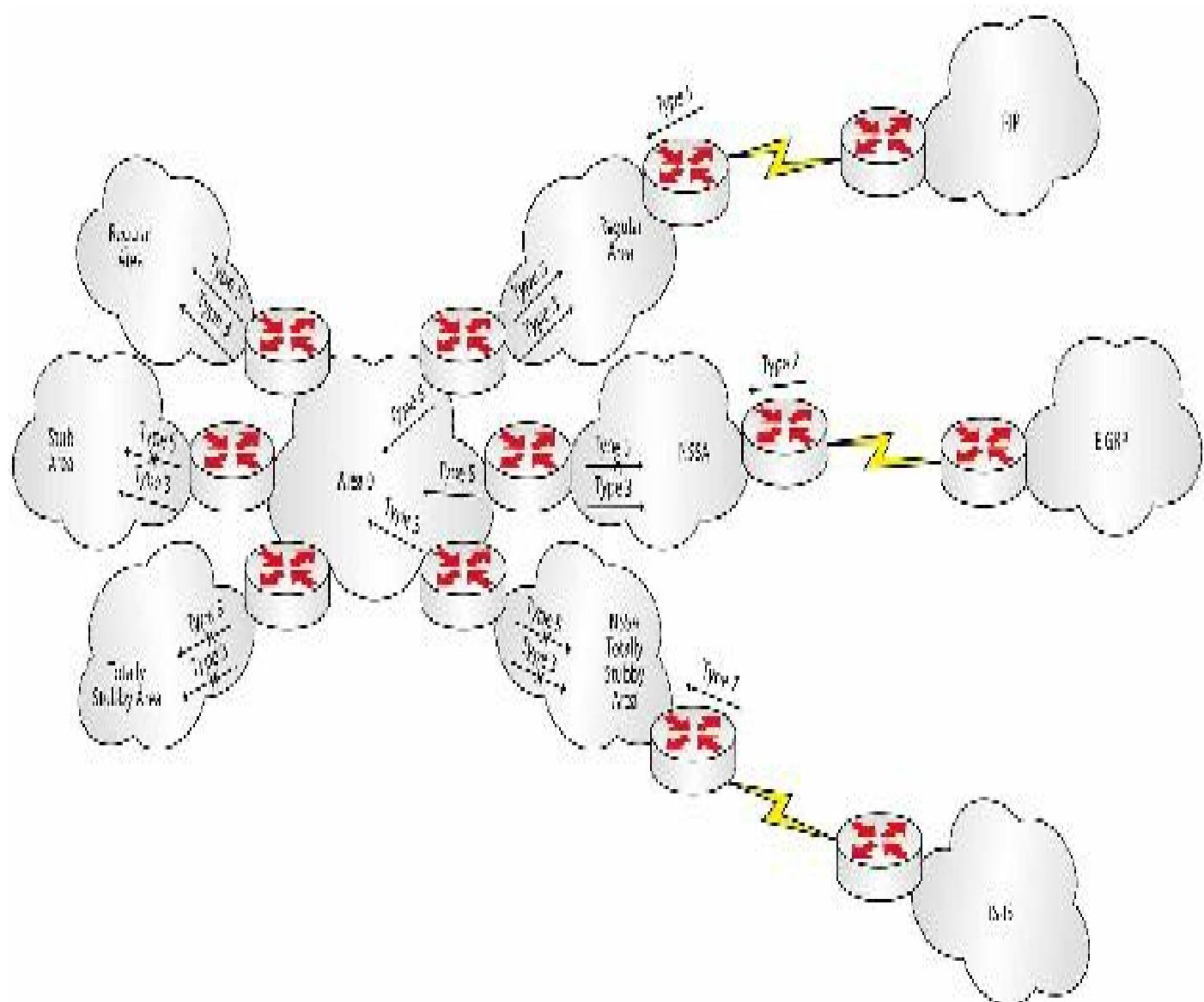


FIG 13.9 – OSPF areas and LSA types

Mini-lab – Configuring Multi-area OSPF

We already examined how to configure single-area OSPF in Chapter 6. Configuring

multi-area OSPF leverages on the same concept. As mentioned earlier, when you are configuring multi-area OSPF, one of the areas must be area 0 and non-zero areas should connect to area 0.

The steps to configure multi-area OSPF are as follow:

1. Plan the network using Cisco OSPF design principles and establish which routers will be the DR/BDR, if required.
2. Assign Router IDs using Loopback interfaces or the router-id [ip address] command. This step is optional but is strongly encouraged.
3. Enable OSPF with the router ospf [process ID] command.
4. Configure the interfaces you want to participate in OSPF with the network x.x.x.x [wildcard] [area #] command.
5. Configure neighbor support commands, if required (for NBMA, for example), with the neighbor [ip address] command.
6. Optionally, add any special area types, such as NSSA, stub, etc., with the following command:

```
R1(config-router)#area 1 ?
```

nssa Specify a NSSA area

stub Specify a stub area

virtual-link Define a virtual link and its parameters

[output truncated]

7. Optionally, add any authentication, summarization, or fine-tuning of Hellos, etc.

Figure 13.10 below shows a simple OSPF network with three non-zero areas attached to Loopback interfaces on the routers:

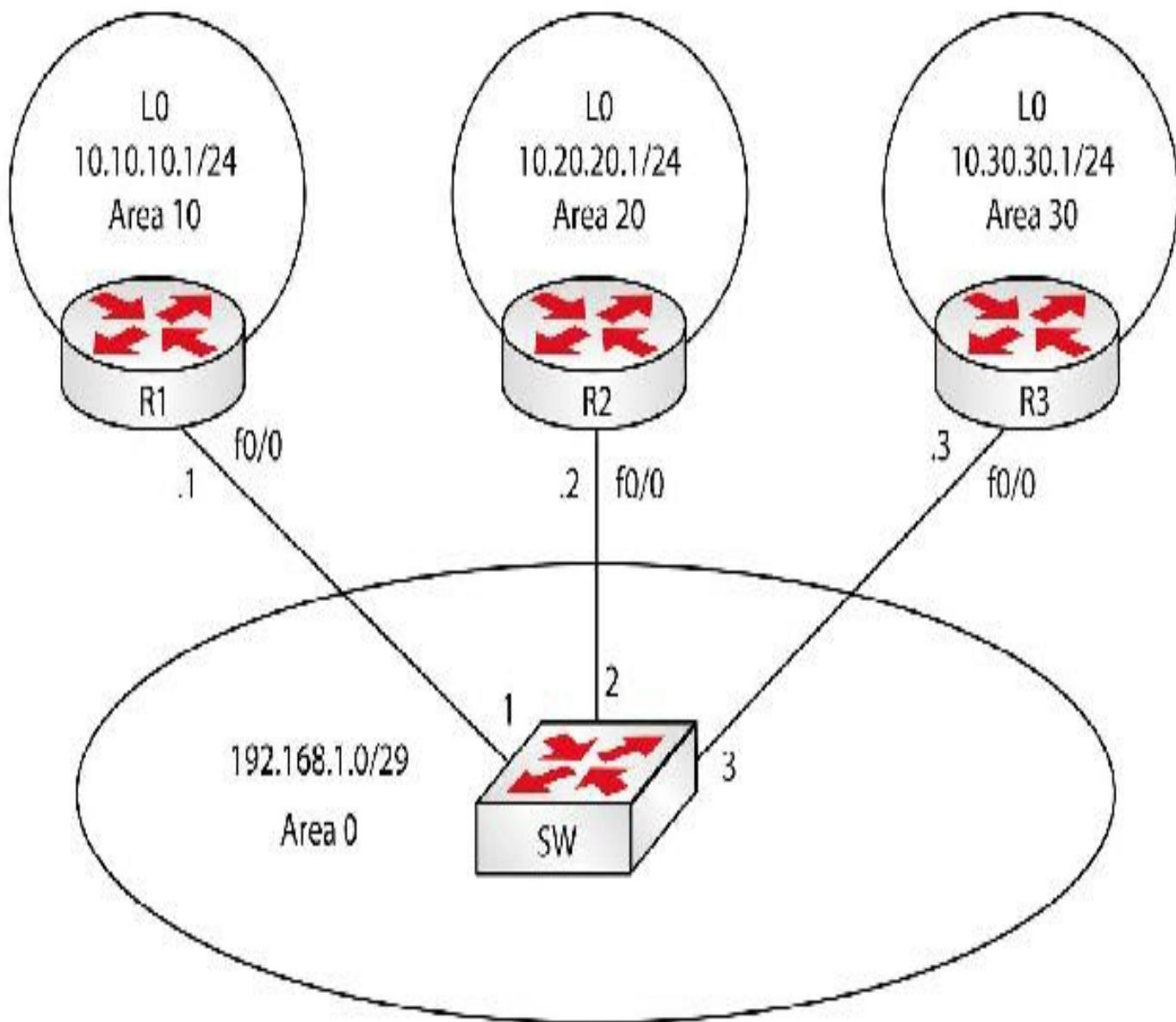


FIG 13.10 – Simple OSPF four-area network

As a small test, I'll add the configuration for Router 1 only. You can copy my commands but replace the correct networks and areas as per Figure 13.10 above. Please note that I ping R2 and R3 once I've added the IP addresses to their F0/0 interfaces.

Router 1 configuration:

```
R1(config-if)#ip address 192.168.1.1 255.255.255.248
R1(config-if)#no shut
R1(config-if)#end
R1#ping 192.168.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
..!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 12/22/32 ms

R1#ping 192.168.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 12/27/56 ms

R1#

Next, I'll add the Loopback 0 IP address and the OSPF configuration. If I wanted to force this router to be the DR, I would have had to add a high Loopback address or a high Router ID or the ip ospf priority command.

R1(config)#int lo0

R1(config-if)#ip add 10.10.10.1 255.255.255.0

R1(config-if)#exit

R1(config)#router ospf 1

R1(config-router)#router-id 200.200.200.200

R1(config-router)#network 192.168.1.0 0.0.0.7 area 0

R1(config-router)#network 10.10.10.0 0.0.0.255 area 10

R1(config-router)#end

R1#

As you configure R2 and R3, you should see the messages below appear on the console of R1. Go ahead and configure the other two routers.

*Mar 1 00:08:56.439: %OSPF-5-ADJCHG: Process 1, Nbr 10.20.20.1 on FastEthernet0/0 from LOADING to FULL, Loading Done

R1#

*Mar 1 00:09:57.299: %OSPF-5-ADJCHG: Process 1, Nbr 10.30.30.1 on FastEthernet0/0 from LOADING to FULL, Loading Done

You can now explore what has taken place with a few OSPF commands. You should usually start with the show ip protocols command:

R1#show ip protocols

Routing Protocol is "ospf 1" Outgoing update filter list for all interfaces is not set

 Incoming update filter list for all interfaces is not set

 Router ID 200.200.200.200

 It is an area border router

 Number of areas in this router is 2. 2 normal 0 stub 0 nssa

 Maximum path: 4

Routing for Networks:

10.10.10.0 0.0.0.255 area 10

192.168.1.0 0.0.0.7 area 0

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway	Distance	Last Update
200.200.200.200	110	00:07:25
10.30.30.1	110	00:04:07
10.20.20.1	110	00:04:57

Distance: (default is 110)

In the output above, you can see the Router ID, which networks you are advertising for which areas, and the routing information sources. The show ip ospf neighbor command should reveal the two neighbors:

R1#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.20.20.1	1	FULL/BDR	00:00:30	192.168.1.2	FE0/0
10.30.30.1	1	FULL/DROTHER	00:00:36	192.168.1.3	FE0/0

R1#

A show ip route will reveal the networks learned:

R1#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B – BGP,

D - EIGRP, EX - EIGRP external, O - OSPF,

IA - OSPF inter area, N1 - OSPF NSSA external type 1,

N2 - OSPF NSSA external type 2, E1 - OSPF external type 1,

E2 - OSPF external type 2

[output truncated]

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

C 10.10.10.0/24 is directly connected, Loopback0

O IA 10.30.30.1/32 [110/11] via 192.168.1.3, 00:08:18, FastEthernet0/0

O IA 10.20.20.1/32 [110/11] via 192.168.1.2, 00:09:08, FastEthernet0/0

192.168.1.0/29 is subnetted, 1 subnets

C 192.168.1.0 is directly connected, FastEthernet0/0

R1#

The show ip ospf interface command will tell you if you are advertising the correct IP address and wildcard mask. You can also confirm timers, neighbors, process ID, network type, and cost. You can also establish whether the router is the DR for the segment.

R1#show ip ospf interface f0/0

FastEthernet0/0 is up, line protocol is up

Internet Address 192.168.1.1/29, Area 0

Process ID 1, Router ID 200.200.200.200, Network Type BROADCAST, Cost: 10

Transmit Delay is 1 sec, State DR, Priority 1

Designated Router (ID) 200.200.200.200, Interface address 192.168.1.1

Backup Designated router (ID) 10.20.20.1, Interface address 192.168.1.2

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:02

Neighbor Count is 2, Adjacent neighbor count is 2

Adjacent with neighbor 10.20.20.1 (Backup Designated Router)

Adjacent with neighbor 10.30.30.1

[output truncated]

The show ip ospf database command will produce a lot of information, even though you are working with a very small network in this lab. Rather than referring to the link state as Type1, Type 2, and so on, the database uses Router IDs to identify the advertising routers.

R1#show ip ospf database

OSPF Router with ID (200.200.200.200) (Process ID 1)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.20.20.1	10.20.20.1	1308	0x80000002	0x009E4B	1
10.30.30.1	10.30.30.1	1253	0x80000002	0x002F91	1
200.200.200.200	200.200.200.200	1317	0x80000003	0x00DA2F	1

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.1.1	200.200.200.200	1256	0x80000002	0x00D43B

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
10.10.10.1	200.200.200.200	1450	0x80000001	0x009861
10.20.20.1	10.20.20.1	1304	0x80000001	0x00AB2A
10.30.30.1	10.30.30.1	1263	0x80000001	0x002E7F

Router Link States (Area 10)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
200.200.200.200	200.200.200.200	1454		0x80000001	0x005D6E 1

Summary Net Link States (Area 10)

Link ID	ADV Router	Age	Seq#	Checksum
10.20.20.1	200.200.200.200	1304		0x80000001 0x0016C5
10.30.30.1	200.200.200.200	1254		0x80000001 0x002F98
192.168.1.0	200.200.200.200	1457		0x80000001 0x007F2D

R1#

You can also run debugs in the network if you want but these are generally run as part of the troubleshooting process. Common debugs for OSPF include debug ip ospf adj and debug ip ospf events:

```
R1#debug ip ospf ?
adj          OSPF adjacency events
database-timer OSPF database timer
events        OSPF events
flood         OSPF flooding
hello         OSPF Hello events
lsa-generation OSPF lsa generation
mpls          OSPF MPLS
nsf           OSPF non-stop forwarding events
packet        OSPF packets
retransmission OSPF retransmission events
rib           OSPF RIB
spf           OSPF spf
tree          OSPF database tree
```

As you can imagine, there is far more to OSPF theory, configuration, and troubleshooting than we have covered here, but our first priority must be to cover the CCNA-level subjects. You could add authentication, tune the Hello/Dead/Retransmit intervals, and add route filtering, redistribution summarization, virtual links, and

network types, as well as configure the same network using NBMA.

I recommend that, for completeness, you repeat the mini-lab above but configure OSPF under the interface, which is the second configuration method but achieves the same result. You need to know both methods for the exam.

[END OF MINI-LAB]

Mini-lab – Configuring and Verifying OSPFv3 Multi-area in Cisco IOS Software

OSPFv3 configuration is usually accomplished with the following sequence of steps:

1. Globally enable IPv6 routing.
2. Configure an OSPFv3 process and assign a unique Router ID; remember that the RID is a 32-bit address, so OSPFv3 would not automatically select an IPv6 IP. If you don't have any IPv4 addresses on the device, you must explicitly assign a RID.
3. Enable IPv6 on the relevant interfaces (either using manually configured addresses or by enabling IPv6 functionality using the `ipv6 enable` command so that link-local addresses can be autogenerated).
4. Enable OSPFv3 routing on the relevant interfaces.
5. Fine-tune OSPFv3 settings (filtering, area types) in router configuration mode.

This process is illustrated in Figure 13.11 below:

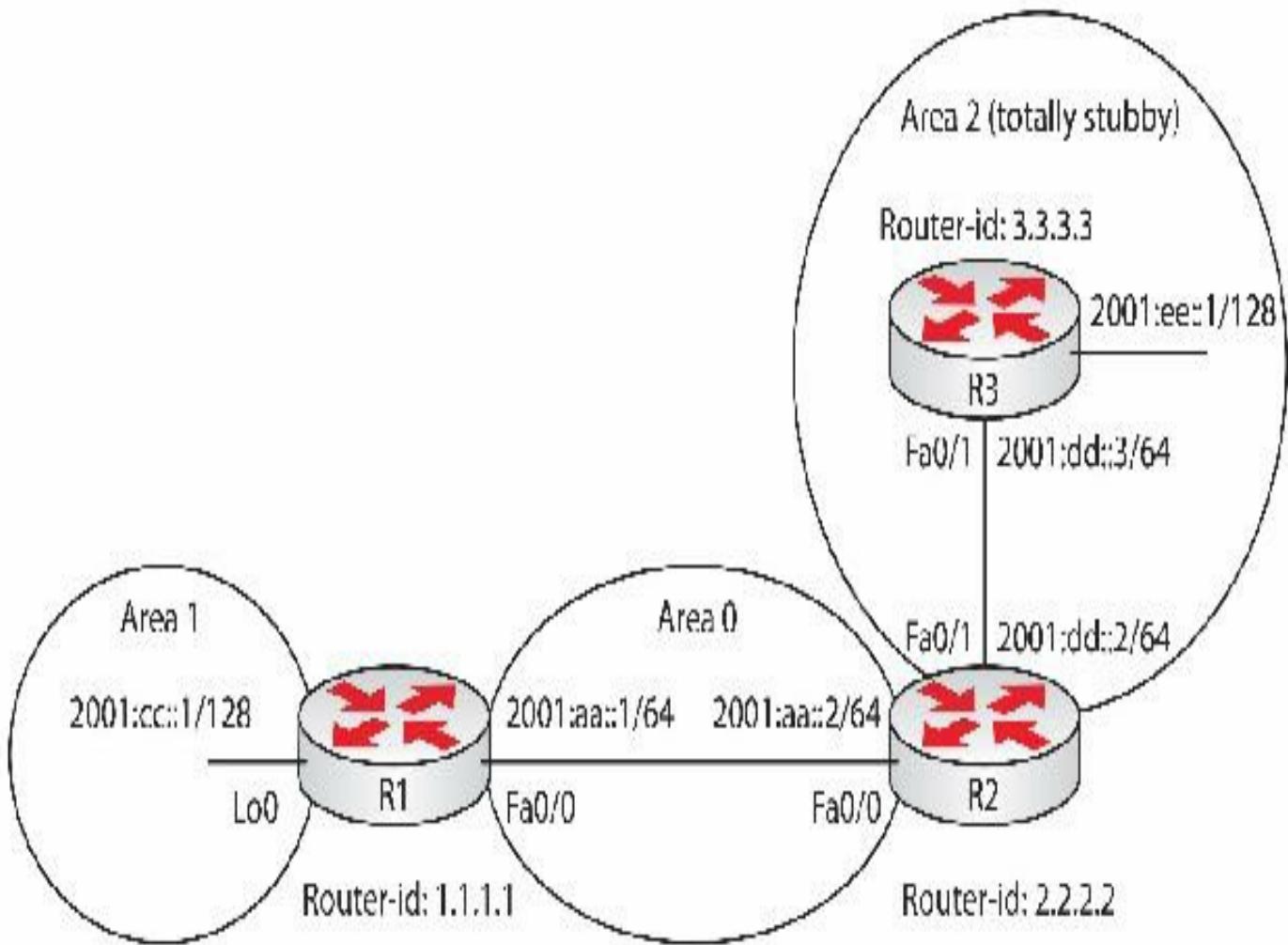


FIG 13.11 – Configuring multi-area OSPFv3 in Cisco IOS Software

Start by enabling IPv6 routing on the three routers. This can be easily overlooked, but without this command you cannot run any kind of IPv6 routing protocol.

```
R1(config)#ipv6 unicast-routing
```

```
R2(config)#ipv6 unicast-routing
```

```
R3(config)#ipv6 unicast-routing
```

Next, define OSPFv3 process 1 on all routers and configure the relevant Router ID values. You MUST configure the RID manually because you do not have any sort of IPv4 address configured on the routers, so the OSPF processes cannot pull an IPv4 value from any interface.

```
R1(config)#ipv6 router ospf 1
```

```
R1(config)#router-id 1.1.1.1
```

```
R2(config)#ipv6 router ospf 1
```

```
R2(config)#router-id 2.2.2.2
```

```
R3(config)#ipv6 router ospf 1
```

```
R3(config)#router-id 3.3.3.3
```

Continue by configuring IPv6 addressing on the relevant interfaces.

```
R1(config)#interface FastEthernet0/0
```

```
R1(config-if)#ipv6 address 2001:AA::1/64
```

```
R1(config)#interface Loopback0
```

```
R1(config-if)#ipv6 address 2001:CC::1/128
```

```
R2(config)#interface FastEthernet0/0
```

```
R2(config-if)#ipv6 address 2001:AA::2/64
```

```
R2(config)#interface FastEthernet0/1
```

```
R2(config-if)#ipv6 address 2001:DD::2/64
```

```
R3(config)#interface FastEthernet0/1
```

```
R3(config-if)#ipv6 address 2001:DD::3/64
```

```
R3(config)#interface Loopback0
```

```
R3(config-if)#ipv6 address 2001:EE::1/128
```

Next, enable OSPFv3 routing on the inter-router interfaces:

```
R1(config)#interface FastEthernet0/0
```

```
R1(config-if)#ipv6 ospf 1 area 0
```

```
R1(config)#interface Loopback0
```

```
R1(config-if)#ipv6 ospf 1 area 1
```

```
R2(config)#interface FastEthernet0/0
```

```
R2(config-if)#ipv6 ospf 1 area 0
```

```
R2(config)#interface FastEthernet0/1
```

```
R2(config-if)#ipv6 ospf 1 area 2
```

```
R3(config)#interface FastEthernet0/1
```

```
R3(config-if)#ipv6 ospf 1 area 2
```

```
R3(config)#interface Loopback0
```

```
R3(config-if)#ipv6 ospf 1 area 2
```

Finalize the configuration by configuring area 2 as a totally stubby area. This command

has to be configured on all the routers in the area.

```
R2(config)#ipv6 router ospf 1
```

```
R2(config)#area 2 stub no-summary
```

The OSPFv3 process goes through the states (DOWN – INIT – 2WAY – EXSTART – EXCHANGE – LOADING – FULL) just like in OSPFv2. You can verify that the OSPFv3 adjacencies have formed:

```
R1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/DR	00:00:31	4	FastEthernet0/0

```
R2#show ipv6 ospf nei
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/BDR	00:00:36	4	FastEthernet0/0
3.3.3.3	1	FULL/BDR	00:00:35	5	FastEthernet0/1

```
R3#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/DR	00:00:33	5	FE0/1

From the output above, you can see that R1 and R3 have a single OSPFv3 adjacency, while R2 has two adjacencies. These are identified by the Router IDs of the neighbor routers.

Let's analyze the link state database entries on Router 1:

```
R1#show ipv6 ospf database
```

OSPFv3 Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	1908	0x80000007	0	1	B
2.2.2.2	1696	0x80000007	0	1	B

Net Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Rtr count
2.2.2.2	148	0x80000004	4	2

Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
1.1.1.1	1898	0x80000001	2001:CC::1/128
2.2.2.2	1686	0x80000001	2001:DD::/64
2.2.2.2	1470	0x80000001	2001:EE::1/128

Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
1.1.1.1	1648	0x80000004	4	Fa0/0
2.2.2.2	150	0x80000005	4	Fa0/0

Intra Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Ref-lsttype	Ref-LSID
2.2.2.2	1673	0x80000002	4096	0x2002	4

Router Link States (Area 1)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	1910	0x80000001	0	0	B

Inter Area Prefix Link States (Area 1)

ADV Router	Age	Seq#	Prefix
1.1.1.1	1910	0x80000001	2001:AA::/64
1.1.1.1	1687	0x80000001	2001:DD::/64
1.1.1.1	1471	0x80000001	2001:EE::1/128

Intra Area Prefix Link States (Area 1)

ADV Router	Age	Seq#	Link ID	Ref-lsttype	Ref-LSID
1.1.1.1	1910	0x80000001	0	0x2001	0

From the LSDB output above, you can see that R1 is advertising an inter-area prefix link state into area 0 for its Loopback address (2001:CC::1/128) and receives LSAs for the networks connected to R2 and R3 from area 0: 2001:DD::/64 and 2001:EE::1/128. R1 has entries for both area 0 and area 1.

Next, let's examine the LSDB on Router 2:

R2#show ipv6 ospf database

OSPFv3 Router with ID (2.2.2.2) (Process ID 1)

Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
------------	-----	------	-------------	------------	------

1.1.1.1	1916	0x80000007	0	1	B
---------	------	------------	---	---	---

2.2.2.2	1693	0x80000007	0	1	B
---------	------	------------	---	---	---

Net Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Rtr count
------------	-----	------	---------	-----------

2.2.2.2	145	0x80000004	4	2
---------	-----	------------	---	---

Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
------------	-----	------	--------

1.1.1.1	1907	0x80000001	2001:CC::1/128
---------	------	------------	----------------

2.2.2.2	1683	0x80000001	2001:DD::/64
---------	------	------------	--------------

2.2.2.2	1466	0x80000001	2001:EE::1/128
---------	------	------------	----------------

Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
------------	-----	------	---------	-----------

1.1.1.1	1650	0x80000004	4	Fa0/0
---------	------	------------	---	-------

2.2.2.2	150	0x80000005	4	Fa0/0
---------	-----	------------	---	-------

Intra Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
------------	-----	------	---------	------------	----------

2.2.2.2	1673	0x80000002	4096	0x2002	4
---------	------	------------	------	--------	---

Router Link States (Area 2)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
------------	-----	------	-------------	------------	------

2.2.2.2	1572	0x80000003	0	1	B
---------	------	------------	---	---	---

3.3.3.3	1478	0x80000003	0	1	None
---------	------	------------	---	---	------

Net Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Rtr count
------------	-----	------	---------	-----------

2.2.2.2	1572	0x80000001	5	2
---------	------	------------	---	---

Inter Area Prefix Link States (Area 2)

ADV Router	Age	Seq#	Prefix
------------	-----	------	--------

2.2.2.2	1677	0x80000001	::/0
---------	------	------------	------

Link (Type-8) Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Interface
2.2.2.2	1954	0x80000002	5	Fa0/1
3.3.3.3	1851	0x80000001	5	Fa0/1

Intra Area Prefix Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Ref-lsttype	Ref-LSID
2.2.2.2	1849	0x80000001	5120	0x2002	5
3.3.3.3	1755	0x80000001	0	0x2001	0

From the output above, you can see that R2 receives an LSA for R1's Loopback on area 0. It also advertises networks connected to R3 into area 0 toward R1. On the other side, in area 2, toward R3, you can see that R2 does not advertise specific networks because area 2 is configured as totally stubby. This is because it will advertise only a default route to reach networks outside of this area. You can see the default route entry as an inter-area prefix link state in area 2, which is ::/0.

Next, let's examine the LSDB on Router 3:

R3#show ipv6 ospf database

OSPFv3 Router with ID (3.3.3.3) (Process ID 1)

Router Link States (Area 2)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
2.2.2.2	1558	0x80000003	0	1	B
3.3.3.3	1462	0x80000003	0	1	None

Net Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Rtr count
2.2.2.2	1558	0x80000001	5	2

Inter Area Prefix Link States (Area 2)

ADV Router Age Seq# Prefix

2.2.2.2 1663 0x80000001 ::/0

Link (Type-8) Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Interface
2.2.2.2	1663	0x80000002	5	Fa0/1
3.3.3.3	1557	0x80000001	5	Fa0/1

Intra Area Prefix Link States (Area 2)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
2.2.2.2	1562	0x80000001	5120	0x2002	5
3.3.3.3	1466	0x80000001	0	0x2001	0

You can see that the LSDB on R3 only includes a default route because it is configured in a totally stubby area.

Next, let's examine the OSPFv3 routing tables on all three routers to make sure that relevant networks are reachable from each end of the network. This is accomplished using the show ipv6 route command:

R1#show ipv6 route ospf

IPv6 Routing Table - 6 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6,

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea,

IS - ISIS Summary, O - OSPF intra, OI - OSPF inter,

OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,

ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

OI 2001:DD::/64 [110/20]

via FE80::C002:2DFF:FE6C:0, FastEthernet0/0

OI 2001:EE::1/128 [110/20]

via FE80::C002:2DFF:FE6C:0, FastEthernet0/0

R2#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6,

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea,

IS - ISIS summary, O - OSPF intra, OI - OSPF inter,

OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,

ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

OI 2001:CC::1/128 [110/10]

via FE80::C001:FFF:FE00:0, FastEthernet0/0

O 2001:EE::1/128 [110/10]

via FE80::C003:2EFF:FED4:1, FastEthernet0/1

R3#show ipv6 route ospf

IPv6 Routing Table - 5 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6,

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea,

IS - ISIS summary, O - OSPF intra, OI - OSPF inter,

OE1 - OSPF ext 1, OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,

ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

OI ::/0 [110/11]

via FE80::C002:2DFF:FE6C:1, FastEthernet0/1

You can see that R1 includes an OSPFv3 routing table entry for both the network between R2 and R3 and for the Loopback interface on R3. R2 contains routing table entries for R1's Loopback and for R3's Loopback because it shares an area with each of these routers. Finally, R3 contains no specific routing table entries; it just contains a default route advertised by R2, pointing to R2 as the exit point out of area 2, because area 2 is configured as totally stubby.

You might have noticed two new LSA types in the outputs above: Link LSA (Type 8) and Intra Area Prefix LSA (Type 9). They are new to OSPFv3 and are used to separate link state information from prefix information. Details on these LSA types are beyond the scope of the CCNA syllabus. If you want to learn more about the differences between OSPFv2 and OSPFv3, please see Section 2 of RFC 5340.

[END OF MINI-LAB]

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 13 exam.

Chapter 13 Labs

You would normally never use OSPF on two routers and would certainly not use three different areas on two routers. However, we've done so here just so you understand how they operate and how to configure areas for the exam.

Lab 1: Multi-area OSPF

The physical topology is shown in Figure 13.12 below:

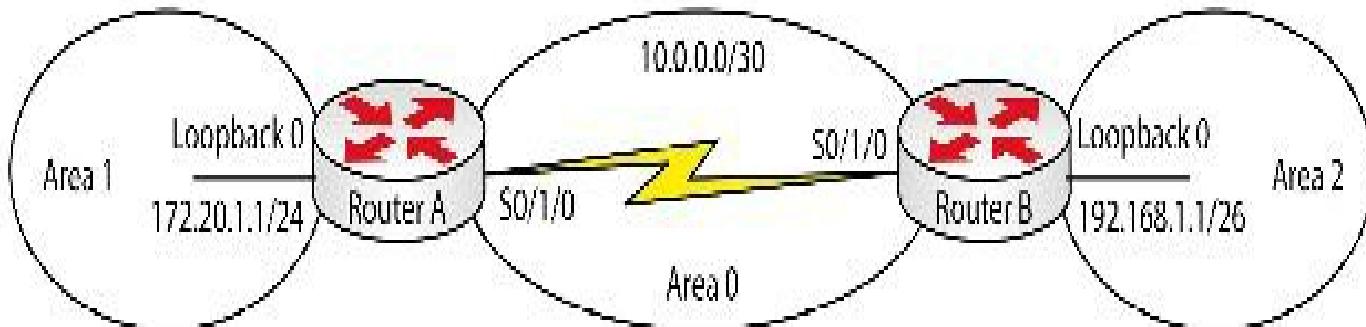


FIG 13.12 – Multi-area OSPF Lab

Lab Exercise

Your task is to configure the network in Figure 13.12 to allow full connectivity using the OSPF routing protocol. Place the point-to-point link into area 0, the Loopback interface of Router A into area 1, and the Loopback interface of Router B into area 2.

Purpose

OSPF is a very popular routing protocol and is in wide use today. You will need to have a good working knowledge of it for the CCNA exam and as a Cisco engineer.

Lab Objectives

1. Configure IP addresses on the router interfaces.
2. Configure OSPF as per the description in the previous section.
3. Verify correct OSPF multi-area functionality.
4. Verify the routing table.

Lab Walk-through

1. Configure all the IP addresses on the topology above. Make sure that you can ping across the Serial link.

```
RouterA(config)# interface s0/1/0
```

```
RouterA(config-if)#ip address 10.0.0.1 255.255.255.252
```

```
RouterA(config)# interface lo0
```

```
RouterA(config-if)#ip address 172.20.1.1 255.255.255.0
```

```
RouterB(config)# interface s0/1/0
```

```
RouterB(config-if)#ip address 10.0.0.2 255.255.255.252
```

```
RouterB(config)# interface lo0
```

```
RouterB(config-if)#ip address 192.168.1.1 255.255.255.192
```

2. Add OSPF to Router A. Put the network on Loopback 0 in area 1 and the 10 network in area 0.

```
RouterA(config)#router ospf 4
```

```
RouterA(config-router)#network 172.20.1.0 0.0.0.255 area 1
```

```
RouterA(config-router)#network 10.0.0.0 0.0.0.3 area 0
```

```
RouterA(config-router)#^Z
```

```
RouterA#
```

```
%SYS-5-CONFIG_I: Configured from console by console
```

```
RouterA#show ip protocols
```

Routing Protocol is ospf 4

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 172.20.1.1

Number of areas in this router is 2. 2 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

172.20.1.0 0.0.0.255 area 1

10.0.0.0 0.0.0.3 area 0

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

172.20.1.1	110	00:00:09
------------	-----	----------

Distance: (default is 110)

3. Add OSPF on Router B. Put the Serial interface network in area 0 and the Loopback network in OSPF area 2.

```
RouterB(config)#router ospf 2
```

```
RouterB(config-router)#net 10.0.0.0 0.0.0.3 area 0
```

```
00:22:35: %OSPF-5-ADJCHG: Process 2, Nbr 172.20.1.1 on Serial0/1/0 from  
LOADING to FULL, Loading Done
```

```
RouterB(config-router)#net 192.168.1.0 0.0.0.63 area 2
```

```
RouterB(config-router)# ^Z  
RouterB#show ip protocols
```

Routing Protocol is ospf 2

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.1.1

Number of areas in this router is 2. 2 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

10.0.0.0 0.0.0.3 area 0

192.168.1.0 0.0.0.63 area 2

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

172.20.1.1	110	00:01:18
------------	-----	----------

192.168.1.1	110	00:00:44
-------------	-----	----------

Distance: (default is 110)

4. Check the routing table on your routers. Look for the OSPF-advertised network. You will see an IA, which means IA – OSPF inter-area. You will also see the AD for OSPF, which is 110.

RouterA#sh ip route

[output truncated]

10.0.0.0/30 is subnetted, 1 subnets

C 10.0.0.0 is directly connected, Serial0/1/0

172.20.0.0/24 is subnetted, 1 subnets

C 172.20.1.0 is directly connected, Loopback0

192.168.1.0/32 is subnetted, 1 subnets

O IA 192.168.1.1 [110/65] via 10.0.0.2, 00:01:36, Serial0/1/0

RouterA#

5. Issue some of the available OSPF commands on either router.

RouterA#sh ip ospf ?

[1-65535] Process ID number

border-routers Border and Boundary Router Information

database Database summary

interface Interface information

neighbor Neighbor list

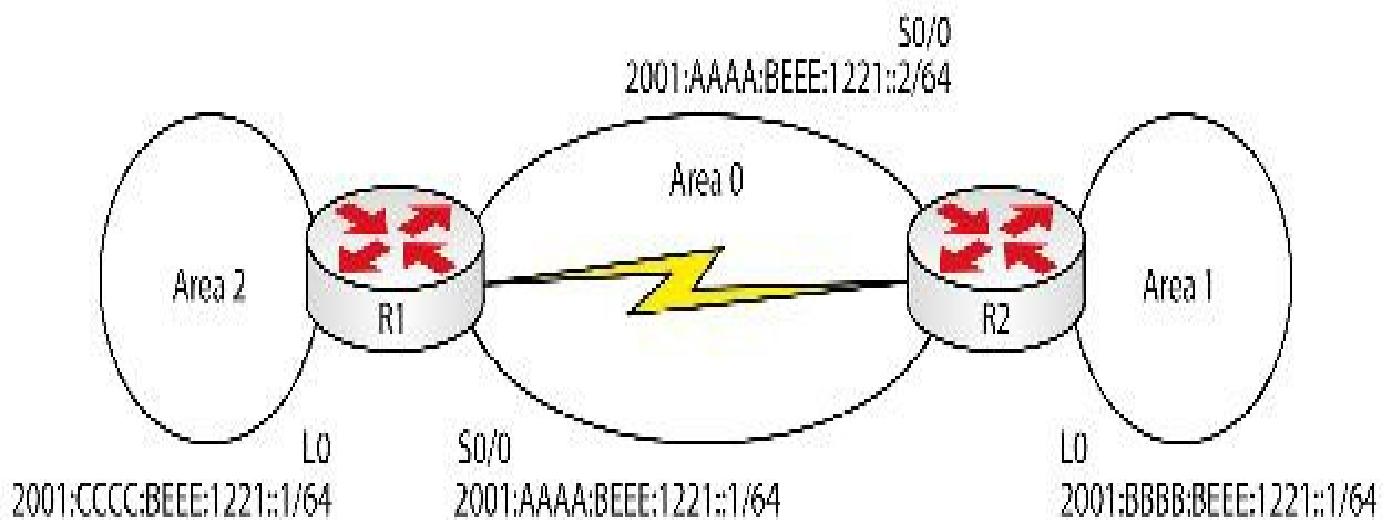
Now redo this lab but put the OSPF configuration under each interface directly. Make sure that you reload all the routers first or at least remove all the current OSPF configurations. Here is a hint for one interface:

```
RouterA(config)#interface s0/0
```

```
RouterA(config-if)#ip ospf 1 area 0
```

Lab 2: Multi-area OSPFv3

The physical topology is shown in Figure 13.13 below:



Lab Exercise

Your task is to configure the network in Figure 13.13 to allow full connectivity using the OSPF routing protocol. Place the point-to-point link into area 0, the Loopback interface of Router 2 into area 1, and the Loopback interface of Router 1 into area 2.

Purpose

Once again, OSPF is a very popular routing protocol and is in wide use today. You will need to have a good working knowledge of it for the CCNA exam and as a Cisco engineer.

Lab Objectives

1. Configure IP addresses on the router interfaces.
2. Configure OSPF as per the description in the previous section.
3. Verify correct OSPF multi-area functionality.
4. Verify the routing table.

Lab Walk-through

1. Configure all the IP addresses on the topology above. Make sure that you can ping across the Serial link. Add clock rates if necessary.

```
Router(config)#hostname R1  
R1(config)#ipv6 unicast-routing  
R1(config)#interface s0/0  
R1(config-if)#ipv6 address 2001:AAAA:BEEE:1221::1/64  
R1(config-if)#no shut  
R1(config)#interface lo0  
R1(config-if)#ipv6 address 2001:CCCC:BEEE:1221::1/64
```

```
Router(config)#hostname R2
```

```
R2(config)#ipv6 unicast-routing  
R2(config)#int s0/0  
R2(config-if)#ipv6 address 2001:AAAA:BEEE:1221::2/64  
R2(config)#no shut  
R2(config)# interface lo0  
R2(config-if)#ipv6 address 2001:BBBB:BEEE:1221::1/64
```

2. Ping from R1 to R2 to test your connection.

```
R1#ping ipv6 2001:AAAA:BEEE:1221::2  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 2001:AAAA:BEEE:1221::2, timeout is 2  
seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/4 ms
```

3. Add your OSPFv3 configurations to the interfaces.

```
R1(config)#ipv6 router ospf 1  
R1(config-router)#router-id 1.1.1.1  
R1(config-router)#int s0/0  
R1(config-if)#ipv6 ospf 1 area 0  
R1(config-if)#int lo0  
R1(config-if)#ipv6 ospf 1 area 2  
R1(config-router)# ^Z
```

```
R2(config)#ipv6 router ospf 1  
R2(config-router)#router-id 2.2.2.2  
R2(config-router)#int s0/0
```

```
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#int lo0
R2(config-if)#ipv6 ospf 1 area 1
R2(config-if)#

```

4. Check the routing table on your routers. Look for the OSPF-advertised network. You will see an IA, which means OI – OSPF inter-area. You will also see the AD for OSPF, which is 110.

```
R1#show ipv6 route
```

IPv6 Routing Table - 6 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B – BGP,

U - Per-user Static route, M - MIPv6, 1 - ISIS L1,

I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1,

OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,

ON2 - OSPF NSSA ext 2, D - EIGRP, EX - EIGRP external

C 2001:AAAA:BEEE:1221::/64 [0/0]

 via ::, Serial0/0

L 2001:AAAA:BEEE:1221::1/128 [0/0]

 via ::, Serial0/0

OI 2001:BBBB:BEEE:1221::1/128 [110/64]

 via FE80::C00A:8FF:FEFC:0, Serial0/0

C 2001:CCCC:BEEE:1221::/64 [0/0]

 via ::, Loopback0

L 2001:CCCC:BEEE:1221::1/128 [0/0]

 via ::, Loopback0

L FF00::/8 [0/0]

 via ::, Null0

R1#

```
R1#ping ipv6 2001:BBBB:BEEE:1221::1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2001:BBBB:BEEE:1221::1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

R1#

5. Issue some of the available OSPF commands on either router.

R1#show ipv6 protocols

IPv6 Routing Protocol is “connected”

IPv6 Routing Protocol is “static”

IPv6 Routing Protocol is “ospf 1”

Interfaces (Area 0):

Serial0/0

Interfaces (Area 2):

Loopback0

Redistribution:

None

R1#

R1#show ipv6 ospf 1

Routing Process “ospfv3 1” with ID 1.1.1.1

It is an area border router

SPF schedule delay 5 secs, Hold time between two SPFs 10 secs

Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs

LSA group pacing timer 240 secs

Interface flood pacing timer 33 msec

Retransmission pacing timer 66 msec

Number of external LSA 0. Checksum Sum 0x000000

Number of areas in this router is 2. 2 normal 0 stub 0 nssa

Reference bandwidth unit is 100 mbps

Area BACKBONE(0)

Number of interfaces in this area is 1

SPF algorithm executed 3 times

Number of LSA 8. Checksum Sum 0x033ABF

Number of DCbitless LSA 0

Number of indication LSA 0

Number of DoNotAge LSA 0

Flood list length 0

Area 2

Number of interfaces in this area is 1

SPF algorithm executed 2 times

Number of LSA 4. Checksum Sum 0x02F03C

Number of DCbitless LSA 0

 Number of indication LSA 0

 Number of DoNotAge LSA 0

 Flood list length 0

R1#show ipv6 ospf database

 OSPFv3 Router with ID (1.1.1.1) (Process ID 1)

 Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	68	0x80000002	0	1	B
2.2.2.2	830	0x80000002	0	1	B

 Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
1.1.1.1	59	0x80000001	2001:CCCC:BEEE:1221::1/128
2.2.2.2	873	0x80000001	2001:BBBB:BEEE:1221::1/128

 Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
1.1.1.1	829	0x80000001	6	Se0/0
2.2.2.2	883	0x80000001	6	Se0/0

 Intra Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
1.1.1.1	829	0x80000001	0	0x2001	0
2.2.2.2	886	0x80000001	0	0x2001	0

 Router Link States (Area 2)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
------------	-----	------	-------------	------------	------

1.1.1.1 71 0x80000001 0 B

Inter Area Prefix Link States (Area 2)

ADV Router Age Seq# Prefix

1.1.1.1 72 0x80000001 2001:AAAA:BEEE:1221::/64

1.1.1.1 72 0x80000001 2001:BBBB:BEEE:1221::1/128

Intra Area Prefix Link States (Area 2)

ADV Router Age Seq# Link ID Ref-lstype Ref-LSID

1.1.1.1 71 0x80000001 0 0x2001 0

R1#

R1#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
-------------	-----	-------	-----------	--------------	-----------

2.2.2.2	1	FULL/ -	00:00:34	6	Serial0/0
---------	---	---------	----------	---	-----------

R1#

R1#show ipv6 ospf interface

Serial0/0 is up, line protocol is up

Link Local Address FE80::C009:8FF:FEFC:0, Interface ID 6

Area 0, Process ID 1, Instance ID 0, Router ID 1.1.1.1

Network Type POINT_TO_POINT, Cost: 64

Transmit Delay is 1 sec, State POINT_TO_POINT,

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:02

Index 1/1/1, flood queue length 0

Next 0x0(0)/0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 3

Last flood scan time is 0 msec, maximum is 0 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 2.2.2.2

Suppress Hello for 0 neighbor(s)

Loopback0 is up, line protocol is up

Link Local Address FE80::C009:8FF:FEFC:0, Interface ID 12
Area 2, Process ID 1, Instance ID 0, Router ID 1.1.1.1
Network Type LOOPBACK, Cost: 1
Loopback interface is treated as a stub Host

R1#

Show Runs

hostname R1

!

ipv6 unicast-routing

!

interface Loopback0

no ip address

ipv6 ospf 1 area 2

ipv6 address 2001:CCCC:BEEE:1221::1/64

!

!

interface Serial0/0

no ip address

ipv6 address 2001:AAAA:BEEE:1221::1/64

ipv6 enable

ipv6 ospf 1 area 0

clock rate 2000000

!

ipv6 router ospf 1

router-id 1.1.1.1

log-adjacency-changes

!

End

hostname R2

```
!  
ipv6 unicast-routing  
!  
interface Loopback0  
no ip address  
ipv6 address 2001:BBBB:BEEE:1221::1/64  
ipv6 ospf 1 area 1  
!  
interface Serial0/0  
no ip address  
ipv6 address 2001:AAAA:BEEE:1221::2/64  
ipv6 ospf 1 area 0  
clock rate 2000000  
!  
ipv6 router ospf 1  
router-id 2.2.2.2
```

Chapter 14 — Enhanced Interior Gateway Routing Protocol (EIGRP)

What You Will Learn in This Chapter

EIGRP

Syllabus Topics Covered

2.6 Configure and Verify EIGRP (Single AS)

- 2.6.a Feasible distance/feasible successors/administrative distance
- 2.6.b Feasibility condition
- 2.6.c Metric composition
- 2.6.d Router ID
- 2.6.e Auto summary
- 2.6.f Path selection
- 2.6.g Load balancing
 - 2.6.g (i) Unequal
 - 2.6.g (ii) Equal

2.7 Passive interface

EIGRP was originally a Cisco proprietary protocol, so you could not use it on non-Cisco equipment, but it was converted into an open standard in 2013 to allow integration with third-party equipment. It was developed by Cisco to address the shortcomings of IGRP, the legacy proprietary routing protocol that was also developed by Cisco. It filled a gap in the market for an easy-to-configure, scalable protocol that would work on routers not capable of running CPU-intensive and complex-to-configure OSPF.

EIGRP

Enhanced Interior Gateway Routing Protocol is a hybrid routing protocol that maps directly to IP protocol 88. It's referred to as hybrid because it uses features from both distance vector and link state protocols. It also uses distance vector measurements along with triggered updates. Designed to overcome some of the limitations of the now-obsolete routing protocol IGRP, the following are enhancements provided by this protocol (among other things):

- The ability to support VLSM
- Reliable Transport Protocol (RTP)

- The DUAL finite-state machine
- Neighbor recovery and discovery
- Protocol-dependent modules

The main advantages of using EIGRP include faster convergence, VLSM support, low CPU utilization, it scales well to large networks, and it has MD5 authentication built in. It is the only protocol that supports multiple routed protocols (through its protocol-dependent modules) because it features separate tables maintained for each routed protocol (IP, IPX, IPv6, etc). Once an EIGRP network is converged, only changes are propagated via incremental updates as opposed to the entire routing table.

EIGRP is a classless routing protocol and it uses Diffusing Update Algorithm (DUAL) to calculate loop-free routes based on information collected. DUAL tracks all the routes advertised by neighbor routers and selects the best path based on composite metrics consisting of bandwidth, delay, reliability, and load. DUAL allows for rapid convergence, making it a good routing protocol choice for many networks.

EIGRP automatically summarizes networks at major network boundaries. Figure 14.1 below is an example of how a badly designed network will be affected if EIGRP is allowed to summarize at the major network boundary:

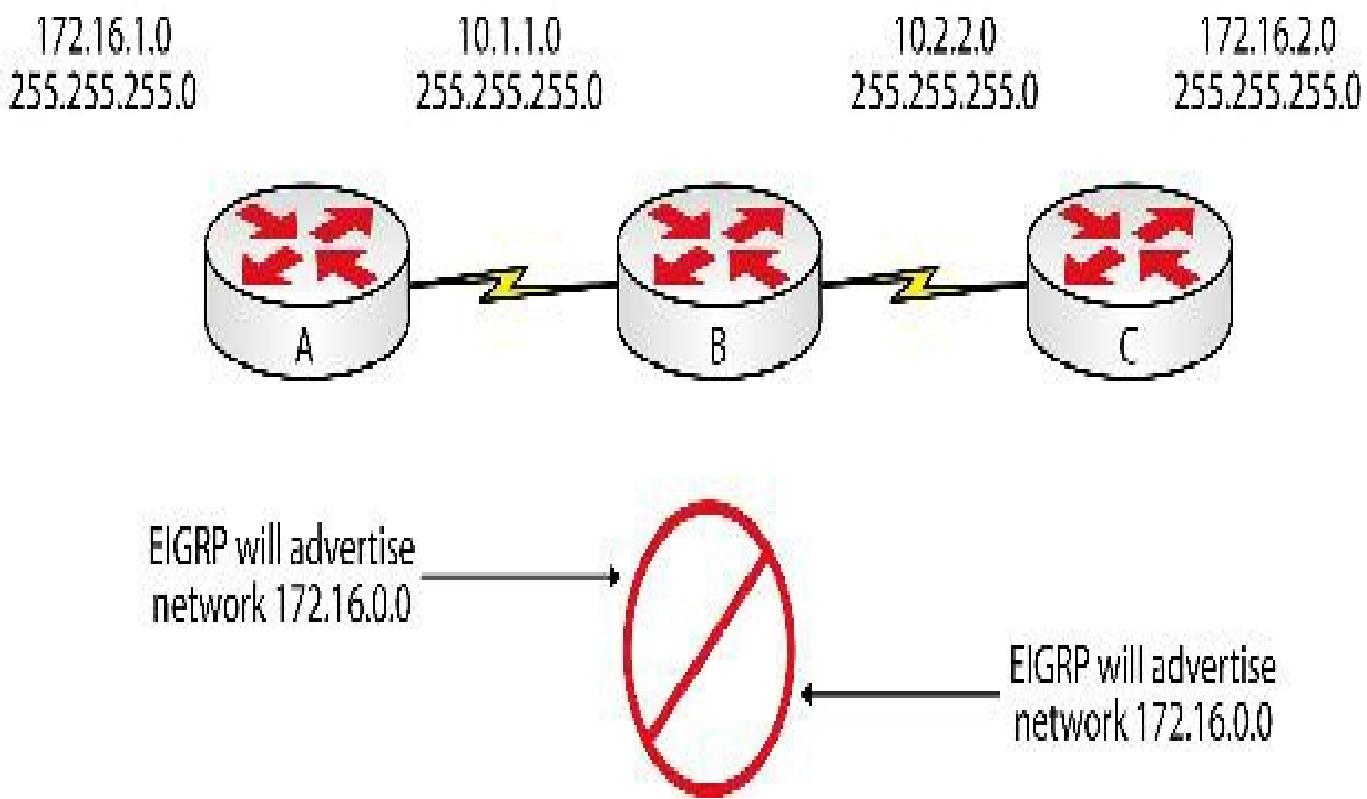


FIG 14.1 – EIGRP auto summarization

This is default behavior for the protocol. If you are using VLSM in your network, you may need to turn this feature off using the no auto-summary command (see the EIGRP

lab for an example of this).

This is IOS-dependent. Auto summary is disabled by default starting in release 15.0(1)M.



```
RouterA#config terminal  
RouterA(config)#router eigrp 20  
RouterA(config-router)#no auto-summary
```

EIGRP does not send periodic updates. Instead, when a metric for a route changes, a partial update is sent out. Periodic Hello packets are sent out at predetermined intervals. These Hello packets inform neighbor routers that the router is alive and available to receive and send routing updates.

EIGRP neighbors will form if both ends of the link agree on any authentication configured, have the same autonomous system number (ASN), see the Hello packet coming from an IP address in the same subnet, and have matching K values. This is very important to bear in mind for the exam, as you may be asked to fix a network running EIGRP where neighbors won't form!

EIGRP Terminology

- **Metric** – EIGRP uses a composite metric. An algorithm is used for metric calculation. Components of the composite metrics are bandwidth, delay (of the line), reliability, and load. EIGRP uses bandwidth and delay by default (metric = [bw + delay] x 256).
- **Topology table** – Each EIGRP router maintains a topology table for every configured protocol, such as IP, IPX, etc. All routes learned via EIGRP are held in this table. The topology table holds the metrics and feasible distances associated with the routes.
- **Neighbor table** – The neighbor table is a list of adjacent routers running EIGRP. The show ip eigrp neighbors command shows the neighbors.
- **Successor/feasible successor** – When EIGRP runs DUAL, it forms a loop-free topology of the network. Part of this process is to determine a successor and feasible successor to every route in the EIGRP routing table. The successor is

the primary path to the route and the feasible successor is the next-best route if the successor is not available. The feasible successor is a backup route based on and stored in the topology table.

- **Internal route** – Routes that originate from within an EIGRP AS are known as internal routes. This route is propagated within the entire AS.
- **External route** – These routes are learned from another routing protocol or AS or are shown in the routing table as static routes.

EIGRP Composite Metric Calculation

As stated above, EIGRP uses a composite metric to determine the best path to take from A to B. This metric can take four attributes into account—bandwidth, delay, load, and reliability. These attributes are used in a formula to determine the EIGRP metric used. You can influence the attributes used by adjusting some special constants in the formula. These values are referred to as K values.

Bandwidth is expressed in Kilobits. EIGRP will take the default bandwidth set for an interface; for example, on a T1 interface it is 1544 Kbps. You can also manually set the interface bandwidth with the bandwidth command.

Serial 0/0 is up, line protocol is up

Hardware is MCI Serial

Internet address is 192.168.10.203, subnet mask is 255.255.255.0

MTU 1500 bytes, **BW 1544 Kbit**, DLY 20000 usec, rely 255/255, load 1/255

Delay is expressed in microseconds (you can see it next to the bandwidth in the output above). It can be adjusted manually with the delay interface-level command. Cisco recommends using the delay command rather than the bandwidth command if you want to influence EIGRP routing decisions. Many junior engineers believe that adjusting these values actually changes the throughput of the interface but this is not the case; only the routing metric is affected. If it was affected you could add the bandwidth 1544 command on a 64 Kbps interface and dramatically increase its throughput capability for free!

Interface statistics and QoS can also be affected.



Reliability is a dynamic number ranging from 1 to 255. A value of 255 means the link is the most reliable. The value can be seen in the output above.

Load is also in the range of 1 to 255 and it represents the output load of an interface, with 1 being the least loaded and 255 being 100% loaded.

It's also worth noting the MTU value, which is the largest packet size the interface can transmit without having to fragment it.

The EIGRP composite metric can be calculated using the formula below:

$$[K1 * \text{bandwidth} + (K2 * \text{bandwidth}) / (256 - \text{load}) + K3 * \text{delay}] * [K5 / (\text{reliability} + K4)]$$

The default K values are $K1 = K3 = 1$ and $K2 = K4 = K5 = 0$. As such, the formula can be reduced to:

$$\text{Metric} = \text{bandwidth} + \text{delay} \times 256.$$

However, the bandwidth and delay used in the EIGRP calculation are not just taken from the interface. EIGRP uses the minimum bandwidth in a path (which makes sense since that is the bottleneck) and the sum of the delays of all the links in the path. The bandwidth and delay values are derived using the formula below:

$$Bw = 256 * 10^7 / \text{minimum bandwidth on path} \text{ (expressed in Kilobits)}$$

$$\text{Delay} = 256 * \text{sum of delays} \text{ (expressed in 10s of microseconds)}$$

Assuming that you have the bandwidth and delay in the right units, the default EIGRP metric can be expressed as:

$$[(10^7/\text{least bandwidth on path}) + (\text{sum of all delays})] \times 256$$

The calculation of the composite metric can be adjusted by changing the K values. This can be done using the metric weights [tos] k1 k2 k3 k4 k5 router configuration

command. The [tos] represents type of service and it is usually set to 0. The K values can be set to any value between 1 and 255. You can view the K values using the show ip protocols command as shown in the output below. You can see that only K1 and K3 are used by default.

```
Router#show ip protocols
Routing Protocol is eigrp 150
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Default networks flagged in outgoing updates
    Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    EIGRP maximum hopcount 100
    EIGRP maximum metric variance 1
    Redistributing: eigrp 150
    EIGRP NSF-aware route hold timer is 240s
    Automatic network summarization is not in effect
    Maximum path: 4
    Routing for Networks:
        192.168.1.0
    Routing Information Sources:
        Gateway      Distance      Last Update
        192.168.1.3      90      00:00:15
    Distance: internal 90 external 170
```

You should remember that K values have to match before two EIGRP routers can establish a neighbor relationship. As such, when changing the K values, you have to ensure that the change is made across the entire autonomous system.

NOTE: Adjusting the default K value settings is not recommended. It should be done only with the assistance of seasoned senior-level engineers who have a solid understanding of the implications of such actions within the network or based on the recommendation of a Cisco TAC.

EIGRP Neighbors

Because EIGRP doesn't advertise its entire routing table at set intervals (which older protocols such as RIP does), it needs a method of exchanging routing information with adjacent routers. EIGRP uses the concept of neighbor relationships in order to achieve this outcome.

Once a router has been configured to use EIGRP and the interfaces have been brought up with the no shut command, EIGRP attempts to find neighbors by sending Hello packets every five seconds using multicast address 224.0.0.10 (if it is using a broadcast media such as Ethernet or certain high-speed Serial interfaces such as T1). If the connection is NBMA or low-speed Serial (under T1 speeds), the Hello is unicast every 60 seconds.

Time	Source IP	Destination IP	Protocol	Action
69 149.847386	192.168.1.2	224.0.0.10	EIGRP	1 Hello
70 149.999886	c2:00:86:95:00:00	c2:00:86:95:00:00	LOOP	60 Reply
71 151.211877	c2:01:86:95:00:00	c2:01:86:95:00:00	LOOP	60 Reply

► Frame 65: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 ▷ Ethernet II, Src: c2:01:86:95:00:00 (c2:01:86:95:00:00), Dst: TPv4mcast_00:00:0a [01:00:5e:00:00:0a]
 ↳ Destination: TPv4mcast_00:00:0a [01:00:5e:00:00:0a]
 ↳ Source: c2:01:86:95:00:00 [c2:01:86:95:00:00]
 Type: IP (0x0800)
 ▷ Internet Protocol Version 4, Src: 192.168.1.2 [192.168.1.2], Dst: 224.0.0.10 [224.0.0.10]
 Version: 4
 Header length: 20 bytes
 Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not ECT (No. ECN Capable Transport))
 Total Length: 60
 Identification: 0x0000 (0)
 Flags...: 0x0000

FIG 14.2 – EIGRP Hello packet capture

Routers running EIGRP in the same network receive the multicast Hello and respond in order to form an adjacency. The adjacency is maintained unless a set number of Hello packets are not received; a hold timer starts and if this expires the route is marked as unreachable.

Reliable Transport Protocol

EIGRP uses Reliable Transport Protocol (RTP) to ensure that packets are delivered, received, ordered, and acknowledged. This is how EIGRP guarantees the delivery and reliability of its routing packets. It doesn't use TCP or UDP for this process. Whenever a multicast packet is received, the router must send a response in the form of a unicast message. Also, any updates must be sent with a sequence number to ensure that the correct order is maintained.

EIGRP uses five packet types (via IP protocol 88), which are Hello, Ack, Update, Query, and Reply. Each packet type has an opcode, which is a 4-bit field that specifies the EIGRP message. Type 1 = Update, 3 = Query, 4 = Reply, 5 = Hello, and 6 = IPX SA. There are other packet types but these are redundant.

▼ Cisco EIGRP

Version: 2

Opcode: Hello (5)

Checksum: 0xeeecb [correct]

▶ Flags: 0x00000000

Sequence: 0

Acknowledge: 0

Virtual Router ID: 0 (Address-Family)

Autonomous System: 1

▶ Parameters

▶ Software Version: EIGRP=12.4, TLV=1.2

FIG 14.3 – Capture of a Hello (Type 5) EIGRP packet

Understanding DUAL and Feasibility Condition

DUAL is used to determine the best path (the loop-free path with the lowest metric) to a destination network. The selected route is known as the successor route and its metric is known as feasible distance (FD). The next-hop router for the successor route (the router that advertised the route) is called the successor.

The successor (next-hop neighbor) advertises the route with its own metric, which is known as the reported distance (RD) or advertised distance. Feasible distance includes the reported distance and the cost to reach the successor. The successor route is placed in the IP routing table and the EIGRP topology table, with the next-hop neighbor as the successor.

If the same prefix is learned from other neighbors, and the reported distance of the route advertised by the neighbor is less than the feasible distance of the successor path, then the neighbors are determined to be loop-free and their routes are referred to as feasible successor routes. These routes are placed into the EIGRP topology table, but not into the IP routing table. In this way, it can quickly be accessed should the successor route fail.

for some reason (see below).

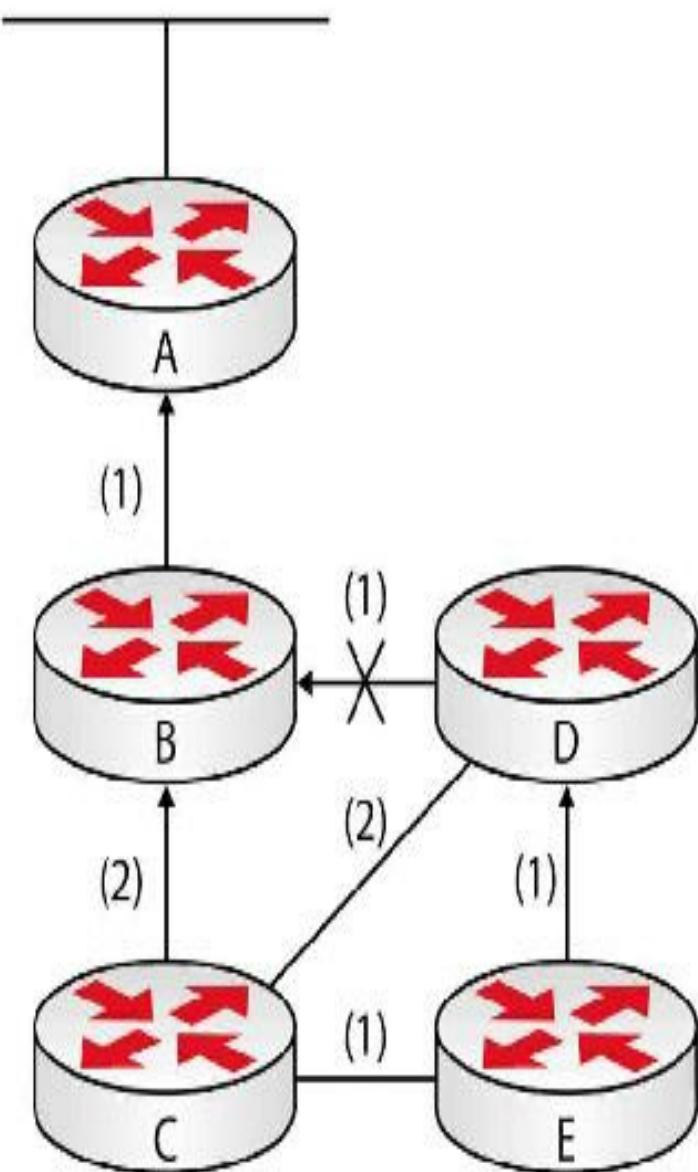
This requirement for the candidates for feasible successors to have a reported distance of the destination network, which is less than the feasible distance of the successor routes, is known as the feasibility condition and is used by EIGRP to prevent loops. If the reported distance is higher, then there is a probability that the path being advertised to the router is in fact going through the router itself.

When a neighbor changes a metric, or when a topology change occurs and the successor route is removed or changes, DUAL checks for feasible successors for the route and if one is found, DUAL uses it to avoid recomputing the route unnecessarily. This is referred to as local computation. Performing a local computation saves CPU power because the feasible successor has been chosen and already exists before the successor or primary route fails.

When no feasible successor for the destination network exists, the local router will send a query to neighbor routers asking if they have information on the destination network. If the information is available and another neighbor does have a route to the destination network, then the router performs a diffusing computation to determine a new successor.

Figure 14.4 below illustrates many of the above points. We will look at the various tables EIGRP uses throughout this section.

Network A



C	EIGRP	FD	RD	Topology
Network A				(FD)
		3	1	(Successor)
		via B	3	
		via D	4	(FS)
		via E	4	3

D	EIGRP	FD	RD	Topology
Network A				(FD)
		2	1	(Successor)
		via B	2	
		via C	5	3

E	EIGRP	FD	RD	Topology
Network A				(FD)
		3	2	(Successor)
		via D	3	
		via C	4	3

LEGEND

C	Destination
EIGRP	Protocol Type
FD	Feasible Distance
RD	Reported Distance as advertised by neighbor router
Successor	Primary Route to Destination
FS	Feasible Successor - Backup Route to Destination

FIG 14.4 – EIGRP topology calculation

EIGRP uses DUAL to create three tables—neighbor, topology, and routing.

The neighbor table maintains a list of routers an adjacency has been formed with. You can see this list with the show ip eigrp neighbors command.

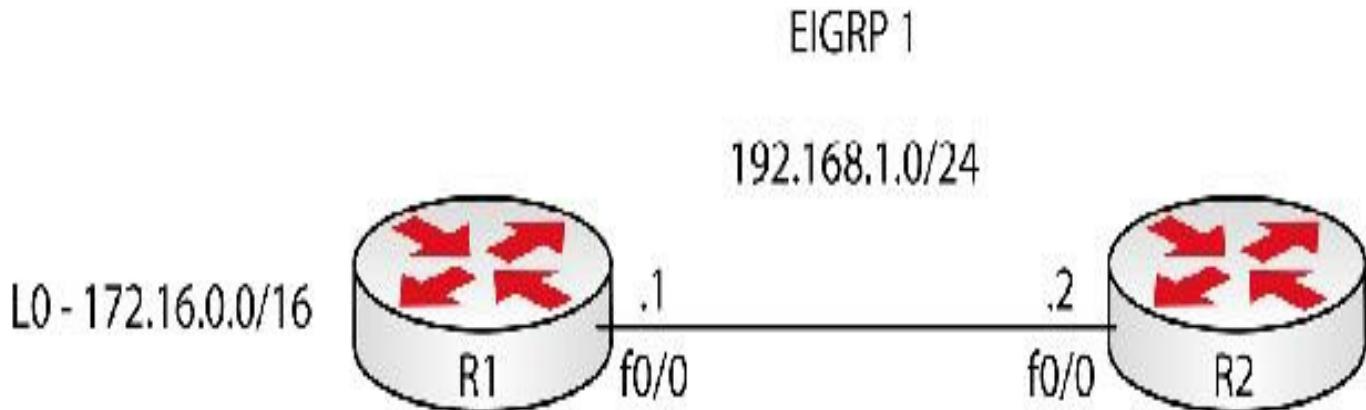


FIG 14.5 – EIGRP topology example

R2#show ip eigrp neighbors

IP-EIGRP neighbors for process 1

Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq Num
		(ms)	(ms)			Cnt	

192.168.1.1	Fa0/0	11	00:21:33	360	2160	0	7
-------------	-------	----	----------	-----	------	---	---

You can see that the process number is 1, which is the ASN configured. The address 192.168.1.1 is the peer IP address. Interface Fa0/0 is where the router received the Hello packets. Holdtime is the amount of seconds the router will wait until it declares the neighbor down. Uptime (hr:min:sec) is the time that has elapsed since the router first heard from the neighbor. SRTT is the smooth round-trip time, which is the number of milliseconds required for the EIGRP packet to be sent to the neighbor and acknowledged. RTO is the retransmission time out (ms), which is how long the router waits before resending a packet from the retransmission queue. Q is the queue count, which is the number of EIGRP packets waiting to be sent. Finally, the sequence number is the number of the last Update, Query, or Reply packet.

The topology table is where all learned routes can be found. All feasible distances and metrics are stored here. You can see the contents of the topology table with the show ip eigrp topology command. Figure 14.5 above shows the topology table.

R2# show ip eigrp topology

IP-EIGRP Topology Table for AS(1)/ID(192.168.1.2)

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

r - reply Status, s - sia Status

P 192.168.1.0/24, 1 successors, FD is 281600

via Connected, FastEthernet0/0

P 172.16.0.0/16, 1 successors, FD is 409600

via 192.168.1.1 (409600/128256), FastEthernet0/0

In the output above, the codes represent the following:

- Passive – no EIGRP computations are being performed for this destination
- Active – EIGRP computations are being performed
- Update – an Update packet was sent for this destination
- Query – a Query packet was sent
- Reply – a Reply packet was sent
- SIA – refers to stuck in active, which means that EIGRP hasn't received a reply to a query from the neighbor for the allotted time (approximately three minutes)

Successors is the number of successors and corresponds to the number of hops in the routing table. FD is feasible distance, which we will discuss shortly. The number 409600 refers to the cost to the destination and 128256 is the metric advertised by the neighbor.

The routing table is where the routes with the lowest composite metric are placed. You can issue the show ip route to see the output of the routing table. If a route goes down, EIGRP can quickly replace it using the topology table to find the feasible successor for the route.

R2#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B – BGP,

D - EIGRP, EX - EIGRP external, O - OSPF,

IA - OSPF inter area

Gateway of last resort is not set

D 172.16.0.0/16 [90/409600] via 192.168.1.1, 00:22:07, F0/0

C 192.168.1.0/24 is directly connected, F0/0

EIGRP Router ID

EIGRP uses the Router ID feature to prevent routing loops. This is used differently from

OSPF, which uses the RID to identify the neighbor. In EIGRP, the RID is used to identify the source of external routes. External routes with the same RID as the local router are discarded, and this is done to prevent routing loops when external routes are injected into an EIGRP AS at multiple points.

The EIGRP Router ID process is similar to that in OSPF. The highest IP address configured on the router is chosen. If Loopback interfaces exist on the router, then they are preferred over physical interfaces (since they are more stable). The RID can be manually configured using the `eigrp router-id` command. The RID is always listed in the EIGRP topology table, as shown below.

When the EIGRP Router ID of a router is manually changed using the `eigrp router-id [address]` router configuration command, the EIGRP neighbors are updated with the new RID and a new adjacency is formed. This is shown in the output below:

```
R1#show ip eigrp topology
```

```
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
```

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

```
[output truncated]
```

A RID of 1.1.1.1 is now configured on the router as follows:

```
R1(config)#router eigrp 150
```

```
R1(config-router)#eigrp router-id 1.1.1.1
```

```
*Mar 1 05:50:13.642: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 150: Neighbor  
150.1.1.2 (Serial0/0) is down: route configuration changed
```

```
*Mar 1 05:50:16.014: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 150: Neighbor  
150.1.1.2 (Serial0/0) is up: new adjacency
```

After the change, the EIGRP neighbor relationship is reset and the new RID is reflected immediately in the EIGRP topology table as shown below:

```
R1#show ip eigrp topology
```

```
IP-EIGRP Topology Table for AS(150)/ID(1.1.1.1)
```

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

```
[output truncated]
```

Mini-lab – EIGRP Passive Interfaces

When EIGRP is enabled on an interface, Hello packets are sent out of that interface and these Hellos allow EIGRP to dynamically establish neighbor relationships. This is desired on physical interfaces where you would expect to establish neighbor relationships. However, this default behavior might be undesired on shared interfaces where you do not want the router to establish neighbor relationships.

In the Cisco IOS, you can make an interface passive in the EIGRP process by specifying that the interface is passive using the `passive-interface` router configuration command. This prevents the interface from sending Hello packets and as such stops the interface from establishing neighbor relationships. The example below shows how you can configure passive interfaces in EIGRP. You can configure these commands on any router because you aren't testing any EIGRP traffic in this example.

```
R1(config)#interface Loopback0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback1
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0/0
R1(config-if)#ip address 150.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#router eigrp 150
R1(config-router)#no auto-summary
R1(config-router)#network 150.1.1.0 0.0.0.255
R1(config-router)#network 10.0.0.0 0.0.0.255
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#passive-interface Loopback0
R1(config-router)#passive-interface Loopback1
R1(config-router)#exit
```

In the configuration above, the Loopback interfaces are enabled for EIGRP (since the network statements cover those interfaces) but they will not send any Hellos (since they are passive). This ensures that other neighbors learn about the networks connected to R1 while suppressing Hello packets on interfaces that are unneeded.

You can view the interfaces that are configured as passive in the output of the `show ip protocols` command as shown below:

```
R1#show ip protocols
```

Routing Protocol is eigrp 150

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 150

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

10.0.0.0/24

10.1.1.0/24

150.1.1.0/24

Passive Interface(s):

Loopback0

Loopback1

Routing Information Sources:

Gateway	Distance	Last Update
---------	----------	-------------

Distance: internal 90 external 170

What if you wanted to make all the interfaces on a router passive in EIGRP? To do this you would use the default keyword. The passive-interface default command suppresses Hellos from being sent on all interfaces on a router. To establish neighbors, you can use the no passive-interface [name] command to allow Hellos to be sent on the interfaces that you desire. The following output illustrates the use of the passive-interface default command:

```
R1(config)#interface Loopback0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback1
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback3
R1(config-if)#ip address 10.3.3.1 255.255.255.0
R1(config-if)#exit
```

```
R1(config)#interface Loopback2
R1(config-if)#ip address 10.2.2.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0/0
R1(config-if)#ip address 150.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#router eigrp 150
R1(config-router)#network 10.0.0.1 255.255.255.0
R1(config-router)#network 10.1.1.1 255.255.255.0
R1(config-router)#network 10.3.3.1 255.255.255.0
R1(config-router)#network 10.2.2.1 255.255.255.0
R1(config-router)#network 150.1.1.1 255.255.255.0
R1(config-router)#passive-interface default
R1(config-router)#no passive-interface Serial0/0
R1(config-router)#exit
```

In this case, Hellos would only be sent out of interface Serial0/0.

[END OF MINI-LAB]

EIGRP Load Balancing

The EIGRP protocol allows unequal-cost load balancing. This means that EIGRP allows traffic to be spread across links with different speeds. By default, EIGRP allows load balancing across four links. You can see this in the output of the show ip protocols command below:

```
R2#show ip protocols
Routing Protocol is eigrp 150
```

[Output truncated]

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

 150.1.1.2/32

The network in Figure 14.6 below shows that it is running EIGRP 1, and R2 has two equal paths it can take to reach the 172.16.0.0 network:

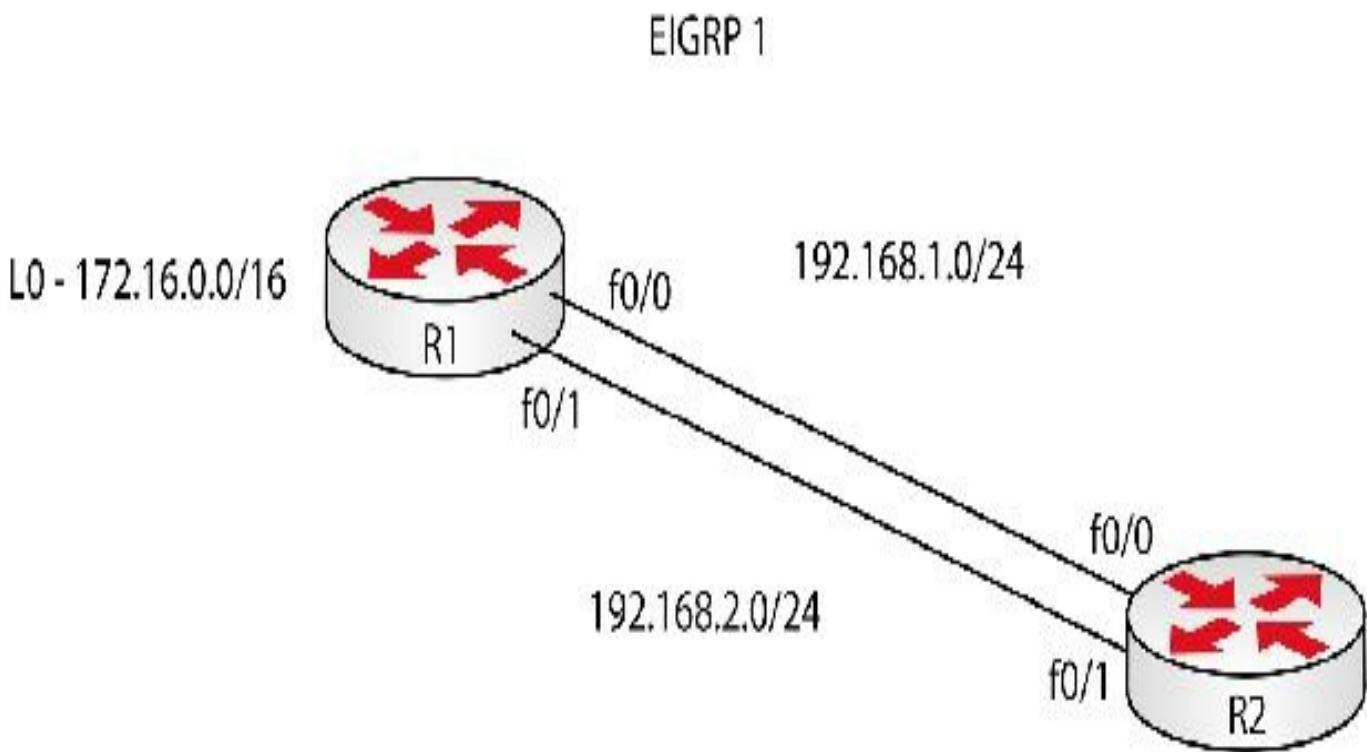


FIG 14.6 – EIGRP load balancing

To edit the amount of paths that is considered in the EIGRP load balancing algorithm, you can use the `maximum-paths [1-6]` router configuration command. You can view the ratio of the traffic-sharing between the links in the output of the `show ip route [network]` command. When performing equal-cost load balancing, the traffic-share values are set to 1 on all the links as illustrated in the following output:

```
R2#show ip route 172.16.0.0
Routing entry for 172.16.0.0/16
Known via "eigrp 1", distance 90, metric 409600, type internal
Redistributing via eigrp 1
Last update from 192.168.2.1 on FastEthernet0/1, 00:02:02 ago
Routing Descriptor Blocks:
192.168.2.1, from 192.168.2.1, 00:02:02 ago, via FastEthernet0/1
```

Route metric is 409600, **traffic share count is 1**

Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 1

* **192.168.1.1**, from 192.168.1.1, 00:02:02 ago, via **FastEthernet0/0**

Route metric is 409600, **traffic share count is 1**

Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 1

The * in front of the 192.168.1.1 indicates that the next packet will take this route. After that, the asterisk will appear before the 192.168.2.1 route.

By default, EIGRP performs equal-cost load balancing. You can enable unequal-cost load balancing using the variance [multiplier] router configuration command.

The multiplier keyword is an integer that indicates that the router can load balance across links that have a metric less than the minimum metric multiplied by the multiplier. This means that any route in the topology table with a metric less than the minimum metric multiplied by the multiplier will be added to the routing table.

Note that for a route to be installed in the topology table, the route must meet the feasibility condition. EIGRP would never install routes that do not meet the feasibility condition, regardless of the variance configured.

To determine the variance multiplier that should be configured under the EIGRP process, you need to take a ratio of the highest metric to the lowest metric. If you change the bandwidth on f0/1 on both routers from the default 10000 to 1000 with the bandwidth 1000 interface command, the route via 192.168.2.1 will no longer be shown.

```
R2(config)#int f0/1
R2(config-if)#band 1000
R2(config-if)#end
*Mar 1 00:10:40.551: %SYS-5-CONFIG_I: Configured from console by console

R2#show ip route 172.16.0.0
Routing entry for 172.16.0.0/16
Known via "eigrp 1", distance 90, metric 409600, type internal
Redistributing via eigrp 1
Last update from 192.168.1.1 on FastEthernet0/0, 00:00:07 ago
Routing Descriptor Blocks:
* 192.168.1.1, from 192.168.1.1, 00:00:07 ago, via FastEthernet0/0
Route metric is 409600, traffic share count is 1
Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 1
```

Now you have to check the topology table to find the route:

```
R2#show ip eigrp topology 172.16.0.0 255.255.0.0
```

IP-EIGRP (AS 1): Topology entry for 172.16.0.0/16

State is Passive, Query origin flag is 1, 1 Successor(s), FD is 409600

Routing Descriptor Blocks:

192.168.1.1 (**FastEthernet0/0**), from **192.168.1.1**, Send flag is 0x0

Composite metric is (**409600**/128256), Route is Internal

Vector metric:

Minimum bandwidth is 10000 Kbit

Total delay is 6000 microseconds

Reliability is 255/255

Load is 1/255

Minimum MTU is 1500

Hop count is 1

192.168.2.1 (**FastEthernet0/1**), from **192.168.2.1**, Send flag is 0x0

Composite metric is (**2713600**/128256), Route is Internal

Vector metric:

Minimum bandwidth is 1000 Kbit

Total delay is 6000 microseconds

Reliability is 255/255

Load is 1/255

Minimum MTU is 1500

Hop count is 1

The first number (409600 for the 192.168.1.1 peer) is the EIGRP metric that represents the cost to the destination. The second number (128256) is the EIGRP metric that this peer advertised.

To determine the variance multiplier value, you can use the formula below:

Multiplier = Highest metric of a feasible successor/Metric of the successor route

Based on this, the variance multiplier for R2 would be:

Multiplier = 2713600/ 409600

Multiplier = 6.625

Since the EIGRP process accepts only whole number values, the value for the multiplier must be rounded up to the nearest integer, which would be 7 in the example above. The output below shows the configuration on R2:

```
R2(config)#router eigrp 1
R2(config-router)#variance 7
```

```
R2(config-router)#exit
```

Now, let's examine the routing table for the 172.16.0.0/16 prefix:

```
R2#show ip route 172.16.0.0 255.255.0.0
```

Routing entry for 172.16.0.0/16

Known via "eigrp 1", distance 90, metric 409600, type internal

Redistributing via eigrp 1

Last update from 192.168.2.1 on FastEthernet0/1, 00:00:27 ago

Routing Descriptor Blocks:

192.168.2.1, from 192.168.2.1, 00:00:27 ago, via FastEthernet0/1

Route metric is 2713600, **traffic share count is 3**

Total delay is 6000 microseconds, minimum bandwidth is 1000 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 1

* 192.168.1.1, from 192.168.1.1, 00:00:27 ago, via FastEthernet0/0

Route metric is 409600, **traffic share count is 20**

Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 1

The traffic share shows that for every three packets sent via f0/1, 20 would be sent via f0/0. This is a ratio of the metric and it ensures that traffic is balanced based on the EIGRP metric.

Mini-lab – Configuring EIGRP

EIGRP can be enabled using the router eigrp [ASN] global configuration command. The ASN keyword represents the autonomous system number, which can range between 1 and 65535. EIGRP neighbors must have the same ASN before a neighbor relationship can be established.

Once EIGRP has been enabled with the router eigrp [ASN] command, you can then specify the interfaces that you want to enable EIGRP on using the network [subnet] [mask] router configuration command.

If an interface has an IP address that falls within the network range, then:

1. EIGRP is enabled on that interface and Hello messages are sent out;
2. Hello messages are received on the interface, and a neighbor relationship can be formed;

3. The subnet information of that interface is sent out in EIGRP updates;
4. Routes received on that interface are considered in the DUAL process and can be installed in the topology or routing table; and
5. The interface subnets are added to the EIGRP topology table.

You can view EIGRP protocol configuration using the `show ip protocols` command (or `sh ip prot` for short):

```
R1#show ip protocols
Routing Protocol is eigrp 150
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
...
...
```

If you do not use a mask in the `network` statement, EIGRP will default to the classful mask and enable all the interfaces in the classful range to use EIGRP. For example, consider a router with the following interfaces:

- Loopback 0 – IP Address 10.0.0.1/24
- Loopback 1 – IP Address 10.1.1.1/24
- Loopback 2 – IP Address 10.2.2.1/24
- Loopback 3 – IP Address 10.3.3.1/24

Enabling EIGRP using the `network 10.0.0.0` command will enable EIGRP on all four Loopback interfaces, and you can see that the 10.0.0.0/8 network is enabled for EIGRP in the output of the `show ip protocols` command below:

```
R1#show ip protocols
Routing Protocol is eigrp 150
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
```

Redistributing: eigrp 150

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks:

10.0.0.0

Routing Information Sources:

Gateway	Distance	Last Update
Distance: internal 90 external 170		

Also, the subnets of the Loopback interfaces are placed into the EIGRP topology table and you can see them using the show ip eigrp topology command, as shown below:

R1#show ip eigrp topology

IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)

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

P 10.3.3.0/24, 1 successors, FD is 128256

 via Connected, Loopback3

P 10.2.2.0/24, 1 successors, FD is 128256

 via Connected, Loopback2

P 10.1.1.0/24, 1 successors, FD is 128256

 via Connected, Loopback1

P 10.0.0.0/24, 1 successors, FD is 128256

 via Connected, Loopback0

Note that this behavior is consistent regardless of the network number specified. As long as a mask is not specified, EIGRP will use the classful subnet mask. Consider the output below:

R1(config)#router eigrp 150

R1(config-router)#network 10.1.1.0

R1(config-router)#network 10.3.3.0

R1(config-router)#exit

Although the network statements refer to specific networks, the show ip protocols command reveals the following:

R1#show ip protocols

Routing Protocol is eigrp 150

Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 150

EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is in effect
Maximum path: 4

Routing for Networks:

10.0.0.0

Routing Information Sources:

Gateway	Distance	Last Update
Distance: internal 90 external 170		

The way to enable EIGRP on a subset of the classful network is to enable EIGRP with a wildcard mask. You can modify the previous example to include the wildcard masks as shown below:

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#network 10.3.3.0 0.0.0.255
R1(config-router)#exit
```

EIGRP will only be enabled on the interfaces covered by the 10.1.1.0/24 and 10.3.3.0/24 networks. You can verify this using the show ip protocols command as follows:

```
R1#show ip protocols
Routing Protocol is eigrp 150
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
```

EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:

10.1.1.0/24
10.3.3.0/24

Routing Information Sources:

Gateway	Distance	Last Update
Distance: internal 90 external 170		

The show ip eigrp interfaces command shows only the Lo1 and Lo3 interfaces:

```
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 150
```

Xmit	Mean	Pacing	Multicast	Pending	Flow	Timer
Queue	Un/Reliable	Time	Un/Reliable			
Interface	Peers	SRTT				
Routes						
Lo1	0	0/0	0	0/10	0	0
Lo3	0	0/0	0	0/10	0	0

Note that you can use the subnet mask directly in the network statement when configuring EIGRP and the router will convert it to the wildcard mask for you. For example, if you enter the configuration below:

```
R1(config-router)#router eigrp 150
R1(config-router)#network 10.1.1.0 255.255.255.0
R1(config-router)#network 10.3.3.0 255.255.255.0
R1(config-router)#exit
```

The router converts the subnet masks to the appropriate wildcard masks by inverting the 0s and 1s. The configuration file now looks like this:

```
R1#show running-config | begin router eigrp
router eigrp 150
network 10.1.1.0 0.0.0.255
network 10.3.3.0 0.0.0.255
auto-summary
```

[END OF MINI-LAB]

As you can imagine, there is far more to EIGRP than we have covered here, but we need to stay on track for the CCNA exam. I recommend that you pursue the CCNP RS if you want to discover more on this subject.

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 14 exam.

Chapter 14 Labs

Lab 1: EIGRP

The physical topology is shown in Figure 14.7 below:

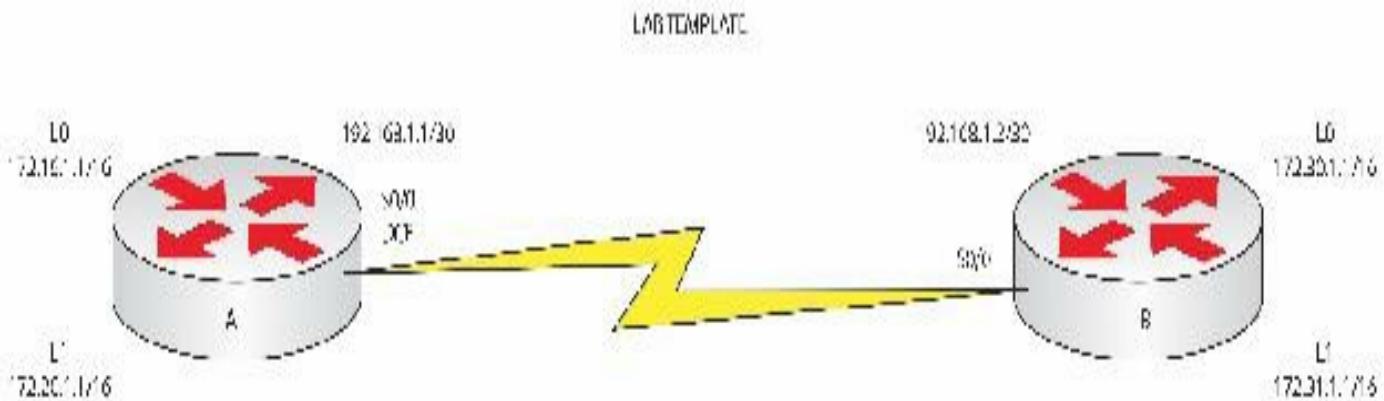


FIG 14.7 – EIGRP Lab

Lab Exercise

Your task is to configure the network above to allow full connectivity using the EIGRP routing protocol. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

EIGRP is a very popular routing protocol and is in wide use today. You will need to have a good working knowledge of it for the CCNA exam and as a Cisco engineer.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 14.7 above. Router A needs to configure a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Configure the EIGRP routing protocol to advertise all networks attached to the router.
5. Ensure that the routing information is correct by checking the routing table for entries of neighbors' addresses.
6. Finally, try to ping all the Loopback interfaces of the neighbors, and then try to access the neighbor routers via Telnet.

Lab Walk-through

1. To set the IP addresses on an interface, you will need to do the following:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#
```

Ping across the Serial link now.

2. To set Telnet access, you need to configure the VTY lines to allow Telnet access. To do this, type the following (in configuration mode):

```
RouterA(config)#line vty 0 4
RouterA(config-line)#login local
RouterA(config-line)#exit
RouterA(config)#username banbury password ccna
```

Router B:

```
RouterB(config)#line vty 0 4  
RouterB(config-line)#login local  
RouterB(config-line)#exit  
RouterB(config)#username banbury password ccna
```

3. To set an enable password, do the following:

```
RouterA(config)#enable secret cisco
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. To configure EIGRP on a router, there are two steps: first, enable the routing protocol; and second, specify the networks to be advertised by EIGRP.

```
RouterA(config)#router eigrp 20
```

RouterA(config-router)#network 192.168.1.0 *Specifies the networks for EIGRP to advertise; one network statement is needed for every network advertised.*

```
RouterA(config-router)#network 172.16.0.0
```

```
RouterA(config-router)#network 172.20.0.0
```

RouterA(config-router)#no auto-summary *This command prevents the router from summarizing the networks*

Router B:

```
RouterB(config)#router eigrp 20  
RouterB(config-router)#network 192.168.1.0  
RouterB(config-router)#network 172.30.0.0  
RouterB(config-router)#network 172.31.0.0  
RouterB(config-router)#no auto-summary
```

Use the show ip route command to determine whether the networks being advertised by the neighbors' EIGRP processes are in your routing table.

Without the no auto-summary command, the router will automatically summarize at the major subnet boundary. It is not so important to do this for this lab. However, if you were using 10.0.0.0 addressing, the network details would be summarized to 10.0.0.0. Make sure that you are aware of the no auto-summary command and when you would want to use it.

```
RouterA#show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP,

M - mobile, B – BGP, D - EIGRP, EX - EIGRP external,
O - OSPF, IA - OSPF inter area,
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2,
E1 - OSPF external type 1, E2 - OSPF external type 2,
E – EGP, i - IS-IS, L1 - IS-IS level-1,
L2 - IS-IS level-2, ia - IS-IS inter area,
* - candidate default, U - per-user static route,
o – ODR, P - periodic downloaded static route

Gateway of last resort is not set

- C 172.16.0.0/16 is directly connected, Loopback0
- C 172.20.0.0/16 is directly connected, Loopback1
- D 172.31.0.0/16 [90/2297856] via 192.168.1.2, 00:00:03, Serial0/0
- D 172.30.0.0/16 [90/2297856] via 192.168.1.2, 00:00:03, Serial0/0
192.168.1.0/30 is subnetted, 1 subnets
- C 192.168.1.0 is directly connected, Serial0/0

RouterA#

You may also have a summary route in your routing table.

RouterA#show ip protocols

Routing Protocol is eigrp 20

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 20

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

172.16.0.0

172.20.0.0

192.168.1.0

Routing Information Sources:

Gateway	Distance	Last Update
(this router)	90	00:03:23

```
192.168.1.2      90      00:02:08  
Distance: internal 90 external 170
```

5. To test connectivity, you will need to use the ping command, and to log in to neighbor routers, you need to use the telnet command:

```
RouterA#ping 172.30.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

```
RouterA#
```

```
RouterA#telnet 172.30.1.1
```

```
Trying 172.30.1.1 ... Open
```

```
User Access Verification
```

```
Username: banbury
```

```
Password:
```

```
RouterB#exit
```

```
[Connection to 172.30.1.1 closed by foreign host]
```

```
RouterA#
```

Do the same on router B:

```
RouterB#ping 172.20.1.1
```

```
RouterB#ping 172.16.1.1
```

```
RouterB#telnet 172.20.1.1
```

Other commands to try:

```
debug ip eigrp
```

```
show ip eigrp neighbors
```

```
show ip eigrp topology
```

```
show ip eigrp interfaces
```

6. Now please enter reload at the Router# prompt and type yes.

Show Runs

```
RouterA#show run
```

```
Building configuration...
```

```
Current configuration : 838 bytes
```

```
!
```

```
version 15.1
```

```
!
hostname RouterA
!
enable secret 5 $1$rujI$BJ8GgiK8U9p5cdfXyApPr/
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Loopback1
ip address 172.20.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.1 255.255.255.252
clockrate 64000
!
router eigrp 20
network 172.16.0.0
network 172.20.0.0
network 192.168.1.0
no auto-summary
!
end
```

RouterA#

```
RouterB#show run
Building configuration...
Current configuration : 824 bytes
!
version 15.1
!
hostname RouterB
!
```

```
enable secret 5 $1$ydeA$MyfRKeV0ckjm7w/0ornnB1
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
!
router eigrp 20
network 172.30.0.0
network 172.31.0.0
network 192.168.1.0
no auto-summary
no eigrp log-neighbor-changes
!
```

Chapter 15 — Advanced IP Services

What You Will Learn in This Chapter

HSRP

VRRP

GLBP

Syslog

SNMP

Syllabus Topics Covered

3.0 IP Services

3.1 Recognize high availability (FHRP)

3.1.a VRRP

3.1.b HSRP

3.1.c GLBP

3.2 Configure and verify syslog

3.2.a Utilize syslog output

3.3 Describe SNMP v2 and v3

You have learned that hosts will typically use DHCP to establish the IP address of the default gateway they should use. Having a single default gateway leaves your network vulnerable in the event that the gateway (router) is no longer available. This can happen if the device or physical interface fails, as shown in Figure 15.1 below:

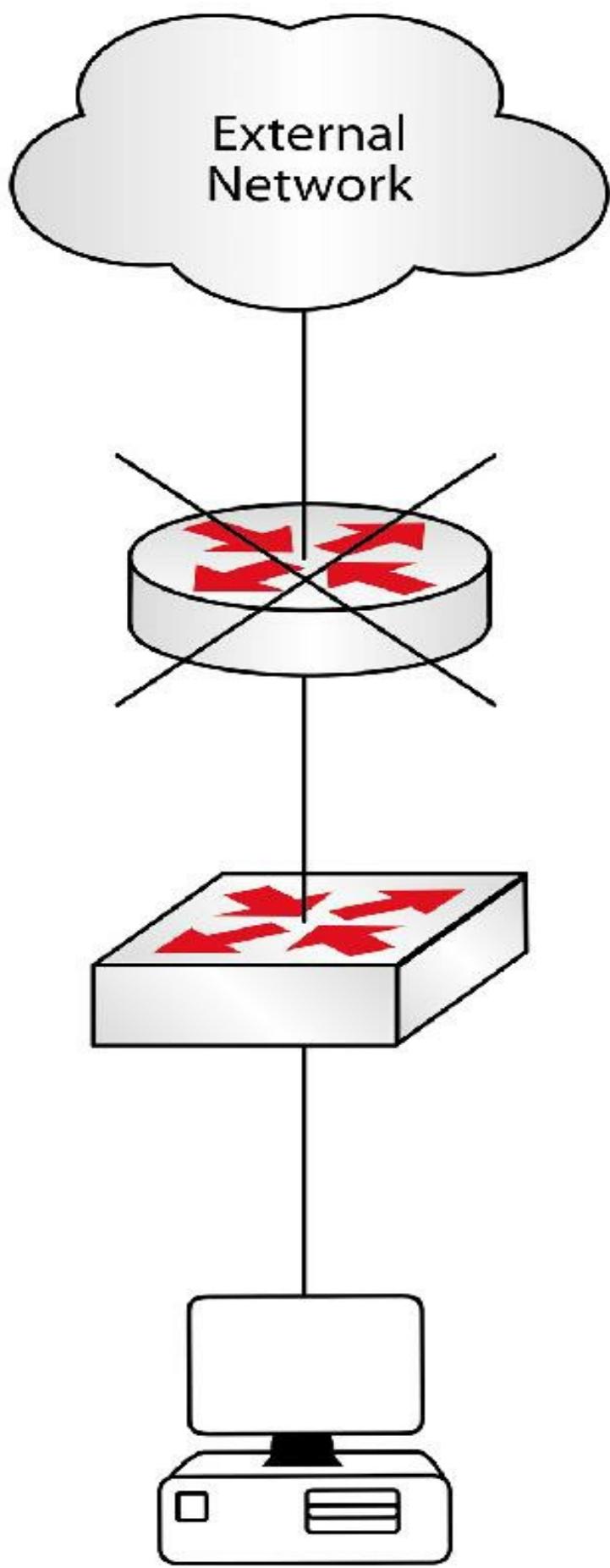
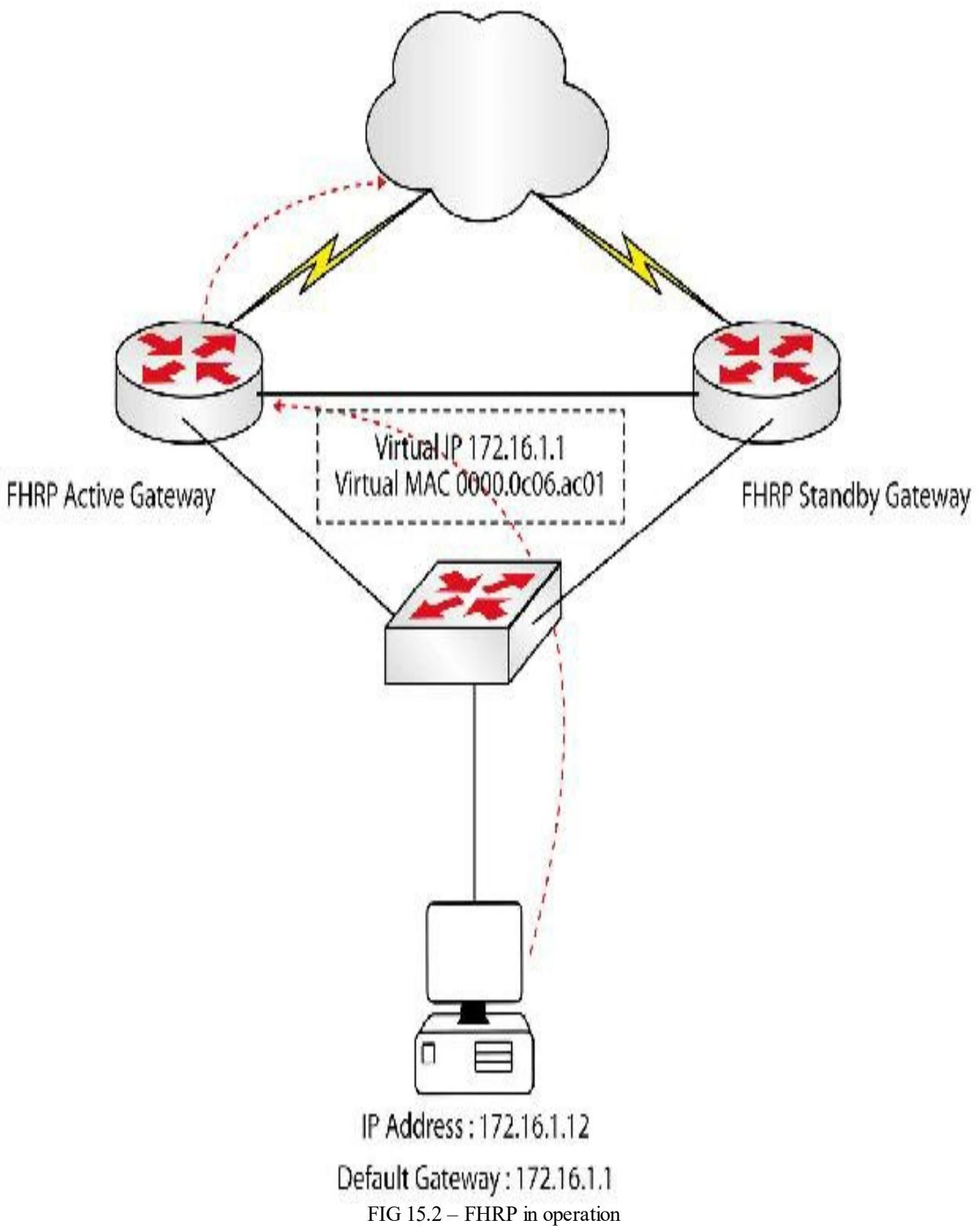


FIG 15.1 – Single gateway = single point of failure

First Hop Redundancy Protocols (FHRPs) overcome this issue by allowing multiple routers to share a virtual IP and MAC address so that in the event that one device fails, another assumes the role as default gateway instantly. This process is transparent to network hosts. You can see FHRP in action in Figure 15.2 below. The host is sending all traffic to a virtual IP address shared by both gateway routers. Should one go down, the traffic will divert to the standby gateway.



Using FHRP isn't the only option open to you of course. There are many ways to achieve first-hop (workstation-to-router) redundancy. Some of these include:

- Proxy ARP on routers
- Explicit configuration

You learned in the ICND1 section that Proxy ARP involves a router responding on behalf of a remote client. This happens when a workstation tries to reach a device that is not directly connected on its subnet. The workstation sends an ARP request for the host and the default router receives this request, realizes that it can service that request (knows it can reach the client), and responds on behalf of the client using Proxy ARP. The router actually pretends to be the host so the workstation can encapsulate the frame with the next-hop address and send traffic destined to that specific client to the router. Multiple routers can provide this service on the same subnet, providing a form of redundancy. The drawback of this approach is the amount of broadcasts that would be sent on the subnet.

Explicit configuration is the most common way of accomplishing workstation-to-router redundancy because some of the operating systems allow multiple default gateway configuration. The problem with this is the increase in latency while the device is trying to figure out which of the configured gateways is the active one. Another drawback of the explicit configuration of multiple default gateways is that not all operating systems support this feature.

The preferred solution is a technology that does not place any burden on the hosts and that is completely transparent to them. The hosts just need to configure a single default gateway because the entire redundancy configuration is made on the routers. The protocols that can be used to accomplish this are generically called First Hop Redundancy Protocols. You can think of them as standby default gateways and they include:

- HSRP (Hot Standby Router Protocol)
- VRRP (Virtual Router Redundancy Protocol)
- GLBP (Gateway Load Balancing Protocol)

HSRP is a Cisco proprietary protocol that inspired IEEE to create the open standard protocol VRRP. The functionality of both protocols is almost identical. GLBP, the most recent protocol of the three, is another Cisco proprietary protocol and it provides more features than both HSRP and VRRP.

HSRP

Hot Standby Router Protocol allows a group of routers (or layer 3 switches) to share one consistent virtual IP and MAC address, even in the event of a gateway device failure. Analyzing Figure 15.3 below, the network has two gateway routers that connect into one layer 2 switch that connects to the network hosts:

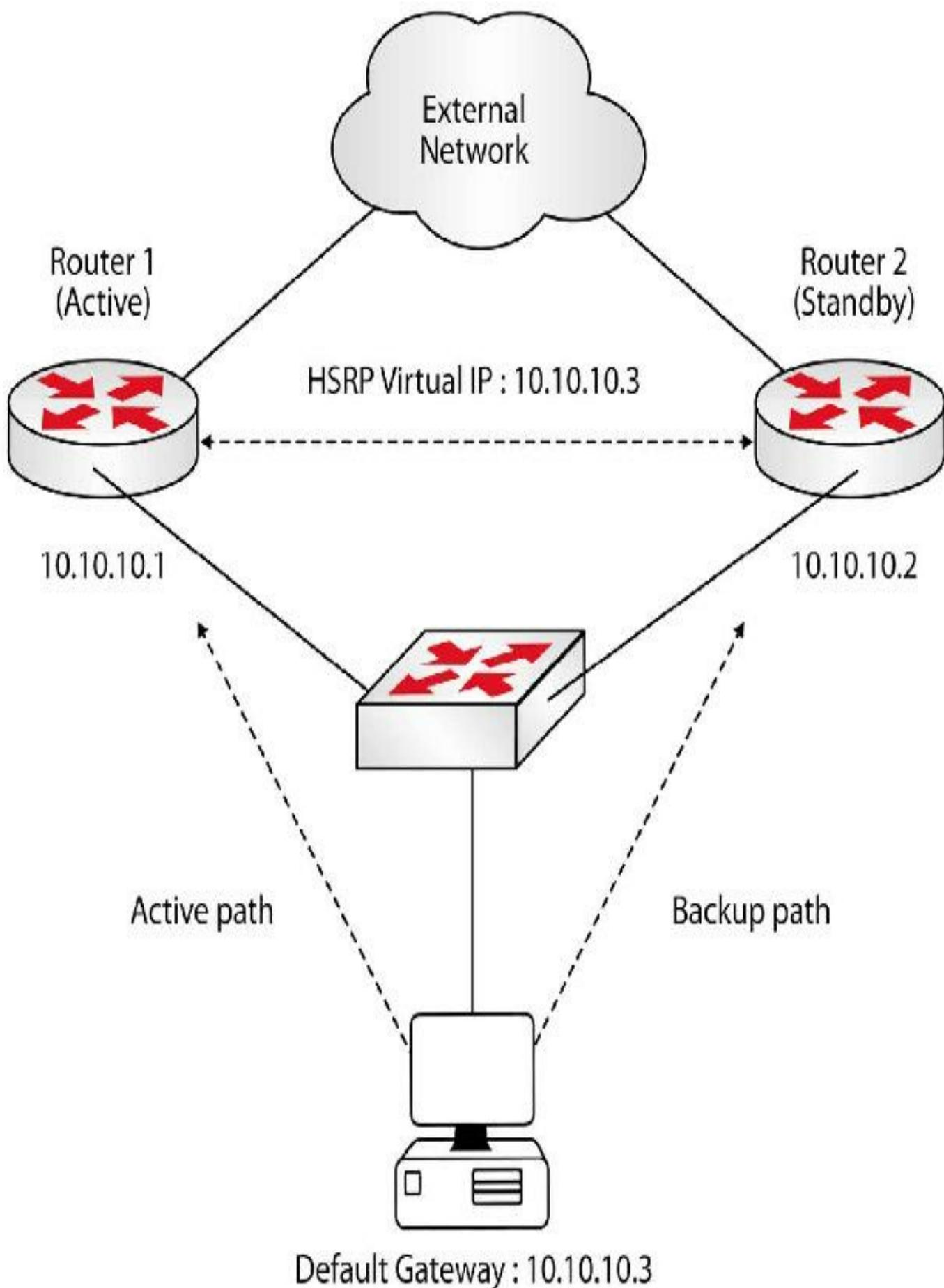
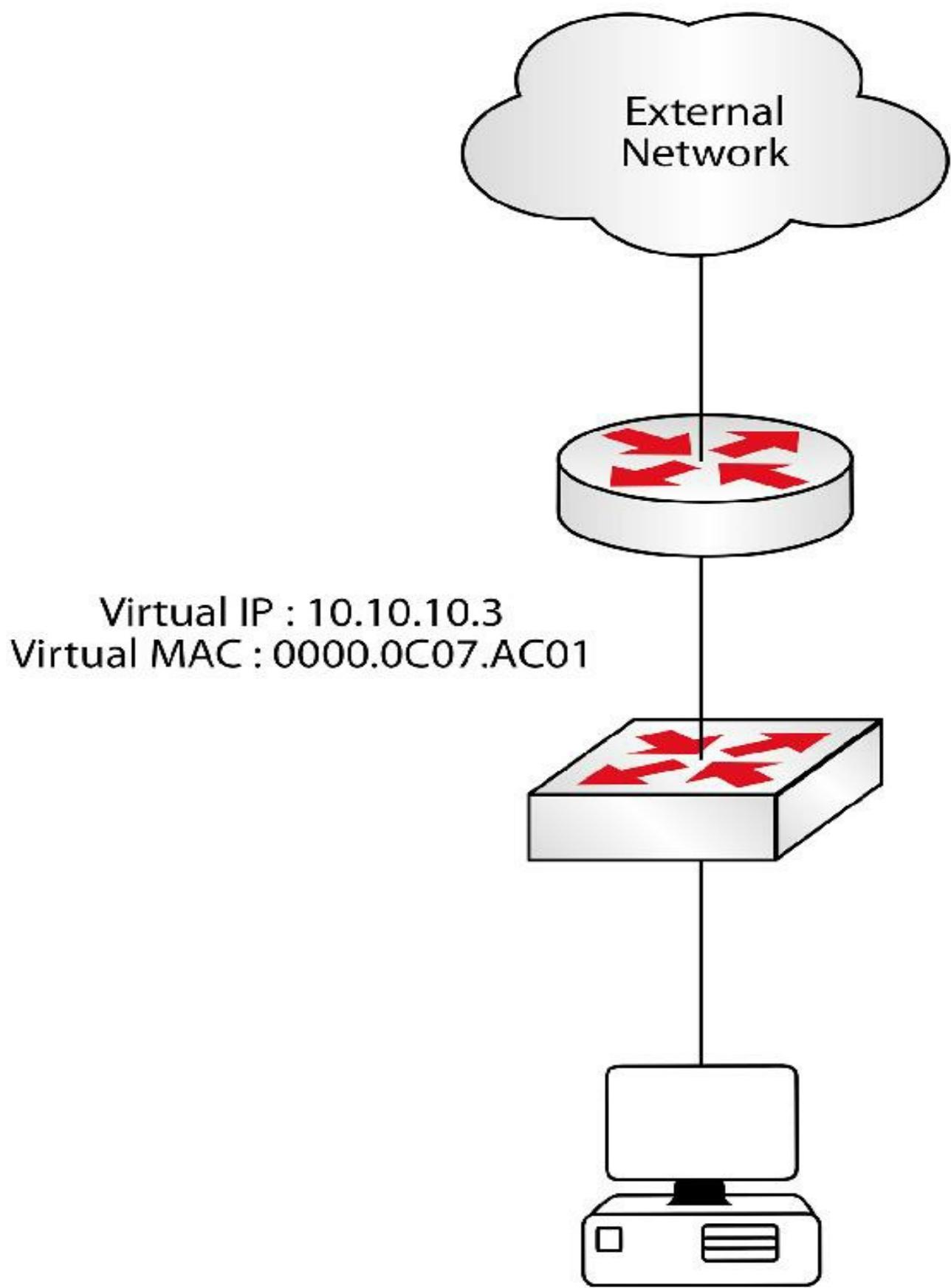


FIG 15.3 – Hot Standby Router Protocol

Router 1 has one physical interface address (10.10.10.1) and Router 2 has another physical interface address (10.10.10.2) in the same network. The two routers are configured in an HSRP group and they present to the clients a virtual default gateway address of 10.10.10.3. This address is configured as the host's default gateway address, although it is not assigned to any router's physical interface because it's just a virtual address.

One of the two routers is the Active device (Router 1 in this example) and it is the one that is actually forwarding traffic for the 10.10.10.3 virtual address. Router 2 is the standby HSRP device. The two routers exchange HSRP Hello messages in order to check on each other's health status. For instance, if Router 2 no longer hears from Router 1, it realizes that Router 1 is down and it will take over as the active HSRP device. The default Hello interval is three seconds and there is a 10-second Dead interval timer.

As mentioned, this process is transparent to the host device, which only sees this:



Default Gateway : 10.10.10.3

FIG 15.4 – FHRPs are transparent to networks hosts

Note the virtual MAC address, which we will address shortly.

Although we will cover configuration shortly, now would be a good time to see the Hello intervals displayed with the show standby command. In the example below, I've used layer 3 switches (3560 models), which can perform in much the same way as routers do. The output doesn't relate to any of the figures above.

Sw2#show standby

Vlan172 - Group 100 (version 2)

State is Standby

3 state changes, last state change 00:20:15

Virtual IP address is 172.16.31.254

Active virtual MAC address is unknown

Local virtual MAC address is 0000.0C9F.F064 (v2 default)

Hello time 3 sec, hold time 10 sec

Next Hello sent in 2.113 secs

Preemption disabled

Active router is 172.16.31.1

Standby router is local

Priority 100 (default 100)

Group name is hsrp-V11-100 (default)

These devices are transparently providing access for the clients by serving up the virtual default gateway address. When the clients want to send packets to the default gateway, they will send out an ARP request, asking for the MAC address of the configured gateway (10.10.10.3). The ARP request will be broadcast in the network and the routers who receive this will reply with the MAC address of the primary gateway, as per the HSRP configuration. When the primary gateway router fails, the router(s) will reply with the MAC address of the newly elected HSRP primary device.

HSRP has two versions available. It is doubtful that the CCNA exam will go into detail on this but version 2 does offer several enhancements over version 1, including millisecond timers instead of whole second timers and improved management and troubleshooting.

HSRPv2 uses a new MAC address range of 0000.0C9F.F000 to 0000.0C9F.FFFF for the virtual gateway address. You can see an example of this in the output above. A partial output from a router running HSRPv1 is shown below. Note the virtual MAC address. This is an important exam topic.

Switch#show standby

FastEthernet0/0 - Group 1

State is Active

8 state changes, last state change 00:13:07

Virtual IP address is 192.168.1.254

Active virtual MAC address is **0000.0c07.ac01**

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

In HSRPv1, the layer 2 address that is used by the virtual IP address will be a virtual MAC address composed of 0000.0C07.ACxx, where xx is the HSRP group number in hexadecimal value and is based on the respective interface. I mention this because it is a typical type of exam question about HSRP. You can see from the ac01 in the output above that the HSRP group number is 1.

You can influence the HSRP primary gateway election by configuring a higher HSRP priority on the router or switch you want to act as the primary gateway. The default priority value is 100 and can go up to 255. If both routers use the same priority, the election will be won by the router with the higher IP address.

In the output below, Sw1 has been configured with a priority of 105 to force it to become the active gateway. Sw2 has been left at the default of 100. You can see in the show commands below that the IP address for Sw1 is 172.16.31.1 and for Sw2 it's 172.16.31.2.

Sw1#show standby

Vlan172 - Group 100 (version 2)

State is Active

5 state changes, last state change 00:19:06

Virtual IP address is 172.16.31.254

Active virtual MAC address is 0000.0C9F.0000

Local virtual MAC address is 0000.0C9F.F064 (**v2** default)

Hello time 3 sec, hold time 10 sec

Next Hello sent in 2.467 secs

Preemption disabled

Active router is local

Standby router is **172.16.31.2**

Priority 105 (configured 105)

Group name is hsrp-VI1-100 (default)

Sw2#show standby

Vlan172 - Group 100 (version 2)

State is Standby

3 state changes, last state change 00:20:15

Virtual IP address is 172.16.31.254

Active virtual MAC address is unknown

Local virtual MAC address is 0000.0C9F.F064 (**v2** default)

Hello time 3 sec, hold time 10 sec

Next Hello sent in 1.75 secs

Preemption disabled

Active router is 172.16.31.1

Priority 100

Group name is hsrp-VI1-100 (default)

Let's assume that the primary gateway is configured with HSRP priority 150 and the backup gateway is left with the default HSRP priority 100. If the primary gateway fails, the backup one assumes the role of active gateway. After a while, if the previous active gateway (configured with priority 150) comes up again, it would not assume the role as primary gateway unless it was configured with a feature called HSRP preemption. This feature allows a gateway with higher priority to assume active gateway functionality when a primary gateway is already present in an HSRP group.

In the output below the HSRP group is 100:

Sw1(config)#int vlan 172

Sw1(config-if)#standby 100 preempt

If you issued the show standby command, you would see that preemption is now enabled:

Next Hello sent in 0.783 secs

Preemption enabled

HSRP message exchange can be authenticated in one of two ways:

- Plain text authentication – not recommended, as the keys are exchanged in plain text
- MD5 authentication – recommended, because of the high level of encryption

HSRP Interface Tracking

HSRP allows administrators to track the status of interfaces on the current active gateway so that when that interface fails, the gateway decrements its priority by a specified value, the default being 10, allowing another gateway to assume the role of active gateway for the HSRP group. This concept is illustrated below in Figure 15.5:

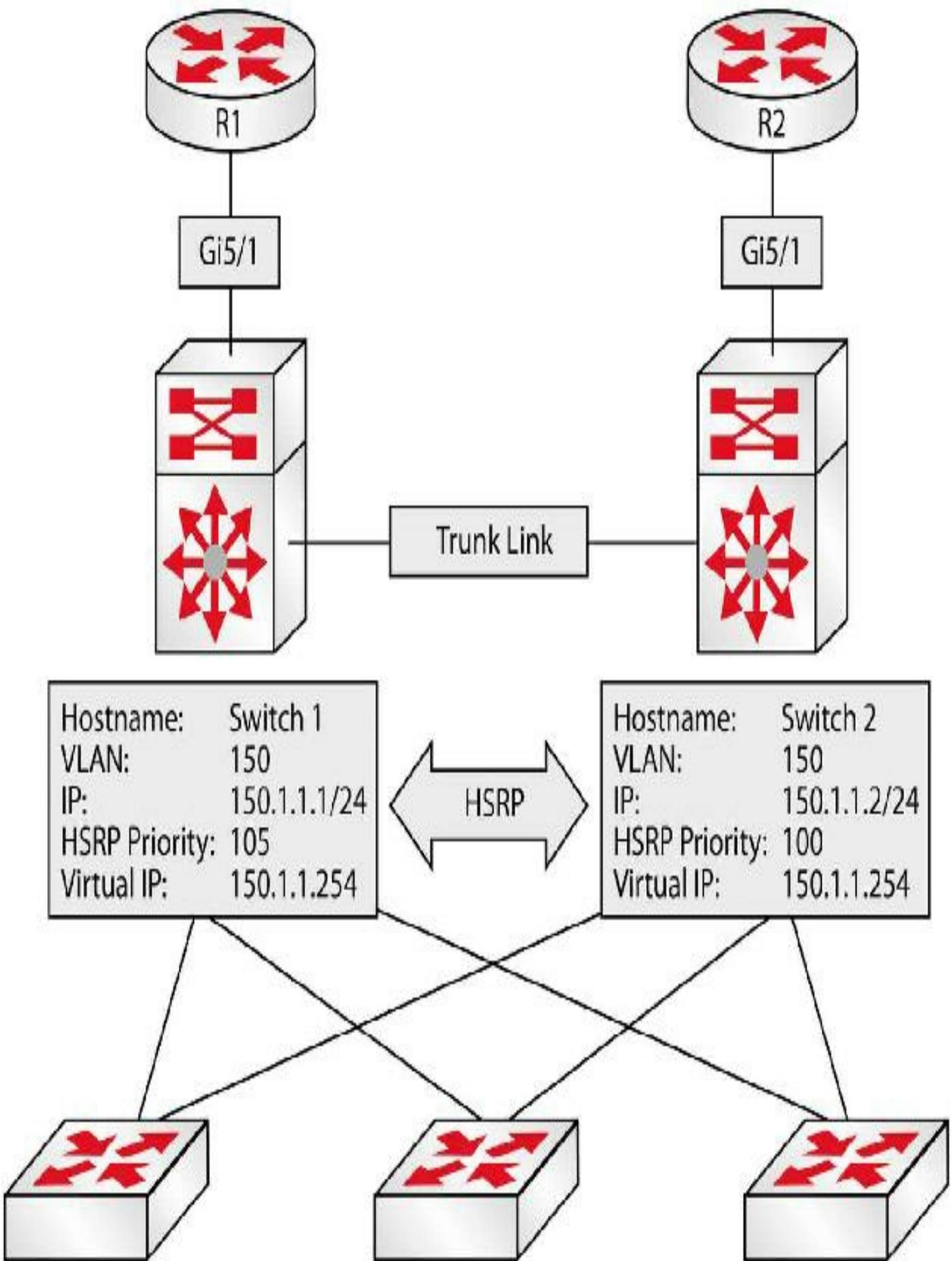


FIG 15.5 – HSRP interface tracking

Referencing Figure 15.5 above, HSRP has been enabled on Switch 1 and Switch 2 for VLAN 150. Based on the current priority configuration, Switch 1, with a priority value of 105, has been elected as the primary switch for this VLAN. Both Switch 1 and Switch 2 are connected to two routers via their Gigabit Ethernet 5/1 interfaces. It is assumed that these two routers peer with other external networks, such as the Internet.

Without HSRP interface tracking, if the Gigabit Ethernet 5/1 interface between Switch 1 and R1 failed, Switch 1 would retain its primary gateway status. It would then have to forward any received packets destined for the Internet, for example, over to Switch 2 using the connection between itself and Switch 2. The packets would be forwarded out via R2 toward their intended destination. This results in a suboptimal traffic path within the network.

HSRP interface tracking allows the administrators to configure HSRP to track the status of an interface and decrement the active gateway priority by either a default value of 10 or a value specified by the administrators. Referencing Figure 15.5, if HSRP interface tracking was configured using the default values on Switch 1, allowing it to track the status of interface Gigabit Ethernet 5/1, and that interface failed, Switch 1 would decrement its priority for the HSRP group by 10, resulting in a priority of 95.

Assuming that Switch 2 was configured to preempt, which is mandatory in this situation, it would realize that it had the higher priority (100 versus 95) and perform a coup, assuming the role of active gateway for this HSRP group.

Configuring HSRP Interface Tracking

In the following output, Switch 1 (a layer 3 switch) is configured to track the state of interface Gigabit Ethernet 5/1, which is connected to an imaginary WAN router. In the event that the state of that interface transitions to down, the gateway will decrement its priority value by 10 (which is the default). The configurations don't relate to any of the figures so far:

```
Switch1(config)#interface vlan 172
Switch1(config-if)#standby 1 track GigabitEthernet5/1
```

This configuration may be validated using the show standby [interface] command. This is illustrated in the following output:

```
Switch#show standby vlan 172
Vlan172 - Group 1
```

State is Active

5 state changes, last state change 00:33:22

Virtual IP address is 172.16.31.254

Active virtual MAC address is 0000.0c07.ac01

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 3 sec, hold time 10 sec

Next Hello sent in 1.085 secs

Preemption enabled

Active router is local

Standby router is 172.16.31.2, priority 100 (expires in 7.616 sec)

Priority 105 (configured 105)

IP redundancy name is “hsrp-Vl172-1” (default)

Priority tracking 1 interfaces or objects, 1 up:

Interface or object Decrement State

GigabitEthernet5/1 10 Up

To configure the gateway to decrement its priority value by 50, for example, the standby [name] track [interface] [decrement value] command can be issued as shown in the following output:

```
Switch1(config)#interface vlan 172
```

```
Switch1(config-if)#standby 1 track GigabitEthernet5/1 50
```

This configuration may be validated using the show standby [interface] command. This is illustrated in the following output:

```
Switch1#show standby vlan 172
```

Vlan172 - Group 1

State is Active

5 state changes, last state change 00:33:22

Virtual IP address is 172.16.31.254

Active virtual MAC address is 0000.0c07.ac01

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 3 sec, hold time 10 sec

Next Hello sent in 1.085 secs

Preemption enabled

Active router is local

Standby router is 172.16.31.2, priority 100 (expires in 7.616 sec)

Priority 105 (configured 105)

IP redundancy name is “hsrp-Vl172-1” (default)

Priority tracking 1 interfaces or objects, 1 up:

Interface or object	Decrement	State
---------------------	-----------	-------

GigabitEthernet5/1	50	Up
--------------------	----	----

You can debug the process with the debug standby command, which I recommend that you try during your labs.

Mini-lab – HSRP Configuration

Referring to the same network presented at the beginning of the HSRP section, Router 1 and Router 2 will act as edge routers toward the external network (you can consider this to be the Internet) and an internal host will be configured to use the HSRP address as the gateway to all external networks. This is a very basic configuration. You will go into more detail on this subject if you progress to the CCNP RS after passing the CCNA exam. You can swap the PC for a router if you want and use the Fast Ethernet interface.

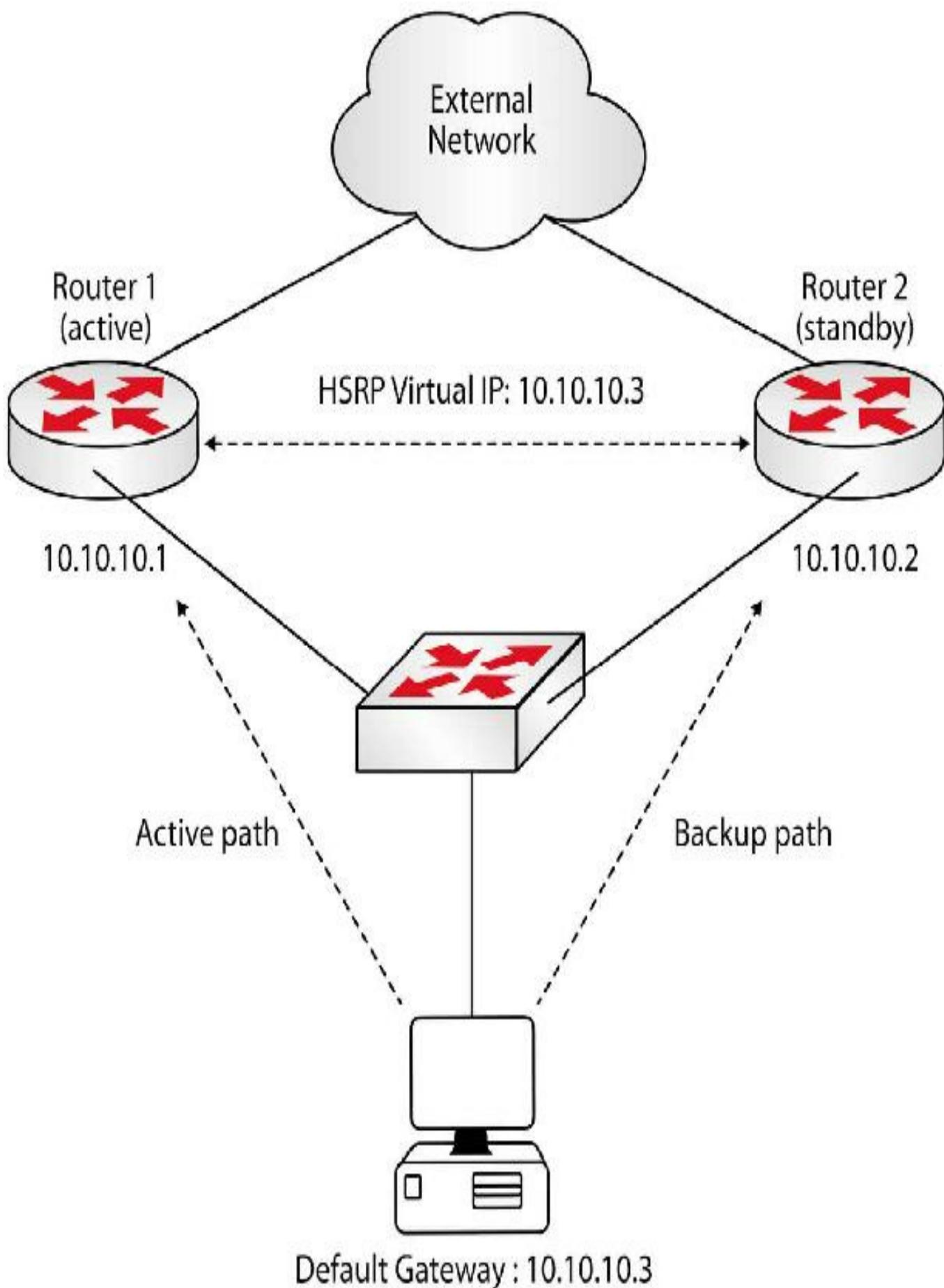


FIG 15.6 – Mini-lab: HSRP Configuration

Start by defining the IP addresses on both routers:

```
R1(config)#int fa0/0
R1(config-if)#ip address 10.10.10.1 255.255.255.0
R1(config-if)#no shut
R2(config)#int fa0/0
R2(config-if)#ip address 10.10.10.2 255.255.255.0
R2(config-if)#no shut
R1#ping 10.10.10.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:

.!!!!

At this point you have connectivity between the routers. Proceed with configuring the HSRP virtual IP. This is done using the `standby [group_id] ip [virtual_ip_address]` command on the interfaces of both routers. Make Router 1 primary, configuring a priority of 120. Router 2 will have the default priority of 100.

```
R1(config)#int fa0/0
R1(config-if)#standby 1 ip 10.10.10.3
R1(config-if)#standby 1 priority 120
R2(config)#int fa0/0
R2(config-if)#standby 1 ip 10.10.10.3
*Mar 1 00:23:01.071: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state
Standby -] Active
R2(config-if)#
*Mar 1 00:12:23.859: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state
Speak -] Standby
```

Next, authenticate the HSRP peering session using an MD5 password in order to secure the HSRP connection. This isn't likely to be a CCNA topic but it's worth knowing.

```
R1(config)#int fa0/0
R1(config-if)#standby 1 authentication md5 key-string CCNA
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#standby 1 authentication md5 key-string CCNA
```

You can also define some additional parameters, such as assigning an HSRP group name and adjusting the timers. By default, HSRP Hello packets are transmitted every three seconds, with a Dead timer of 10 seconds. In addition, enable both routers for preemption.

```
R1(config)#int fa0/0
```

```
R1(config-if)#standby 1 name CCNA
```

```
R1(config-if)#standby 1 timers 1 3
```

```
R1(config-if)#standby 1 preempt
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#standby 1 name CCNA
```

```
R2(config-if)#standby 1 timers 1 3
```

```
R1(config-if)#standby 1 preempt
```

Next, check the status of the HSRP group on each router:

```
R1#show standby
```

```
FastEthernet0/0 - Group 1
```

State is Active

2 state changes, last state change 00:06:08

Virtual IP address is 10.10.10.3

Active virtual MAC address is 0000.0c07.ac01

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 1 sec, hold time 3 sec

Next Hello sent in 0.524 secs

Authentication MD5, key-string CCNA

Preemption enabled

Active router is local

Standby router is 10.10.10.2, priority 100 (expires in 2.340 sec)

Priority 120 (configured 120)

Group name is CCNA (cfgd)

R2#show standby

FastEthernet0/0 - Group 1

State is Standby

6 state changes, last state change 00:03:59

Virtual IP address is 10.10.10.3

Active virtual MAC address is 0000.0c07.ac01

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 1 sec, hold time 3 sec

Next Hello sent in 0.624 secs

Authentication MD5, key-string CCNA

Preemption enabled

Active router is 10.10.10.1, priority 120 (expires in 2.908 sec)

Standby router is local

Priority 100 (default 100)

Group name is CCNA (cfgd)

At this point the internal host can use the HSRP virtual address of 10.10.10.3 as the default gateway in order to access external networks.

As a last configuration step, enable interface tracking so that Router 2 will assume primary gateway functionality when Router 1's uplink fails.

R1(config)#int fa0/0

R1(config-if)#standby 1 track FastEthernet0/1 ?

[1-255] Decrement value

[cr]

R1(config-if)#standby 1 track FastEthernet0/1 30

This command decrements Router 1's HSRP priority by 30 if the uplink interface (Fast Ethernet 0/1) fails. This makes the priority value equal to 90, which is lower than Router 2's priority of 100, so Router 2 will be used as the primary gateway toward the external destination.

[END OF MINI-LAB]

VRP

Virtual Router Redundancy Protocol works in a way similar to HSRP, with a few small differences. The main difference is that VRRP is an IETF protocol, which means that it can be implemented on multiple vendor equipment. One of the other differences is that the two routers are configured in a VRRP group; one router is called the master device (instead of the active router), which does all the forwarding, while the other is called the backup device (instead of the standby router). The main differences are highlighted in Table 15-1 below:

Table 15-1: Differences between HSRP and VRRP

Hot Standby Router Protocol (HSRP)	Virtual Redundancy Router Protocol (VRRP)
Cisco proprietary	Industry standard
Uses multicast address 224.0.0.2 or 224.0.0.102	Uses multicast address 224.0.0.18
Uses virtual Mac address 0000.0c07.acxx	Uses virtual MAC Address 0000.5e00.01xx
Described in RFC 2281	Described in RFC 5798
Preemption disabled by default	Preemption enabled by default
Hello timer is three seconds	Hello timer is one second

As with the xx in the HSRP address, this will be replaced with the group number (in hexadecimal) when configured. We will explore VRRP using a similar network topology:

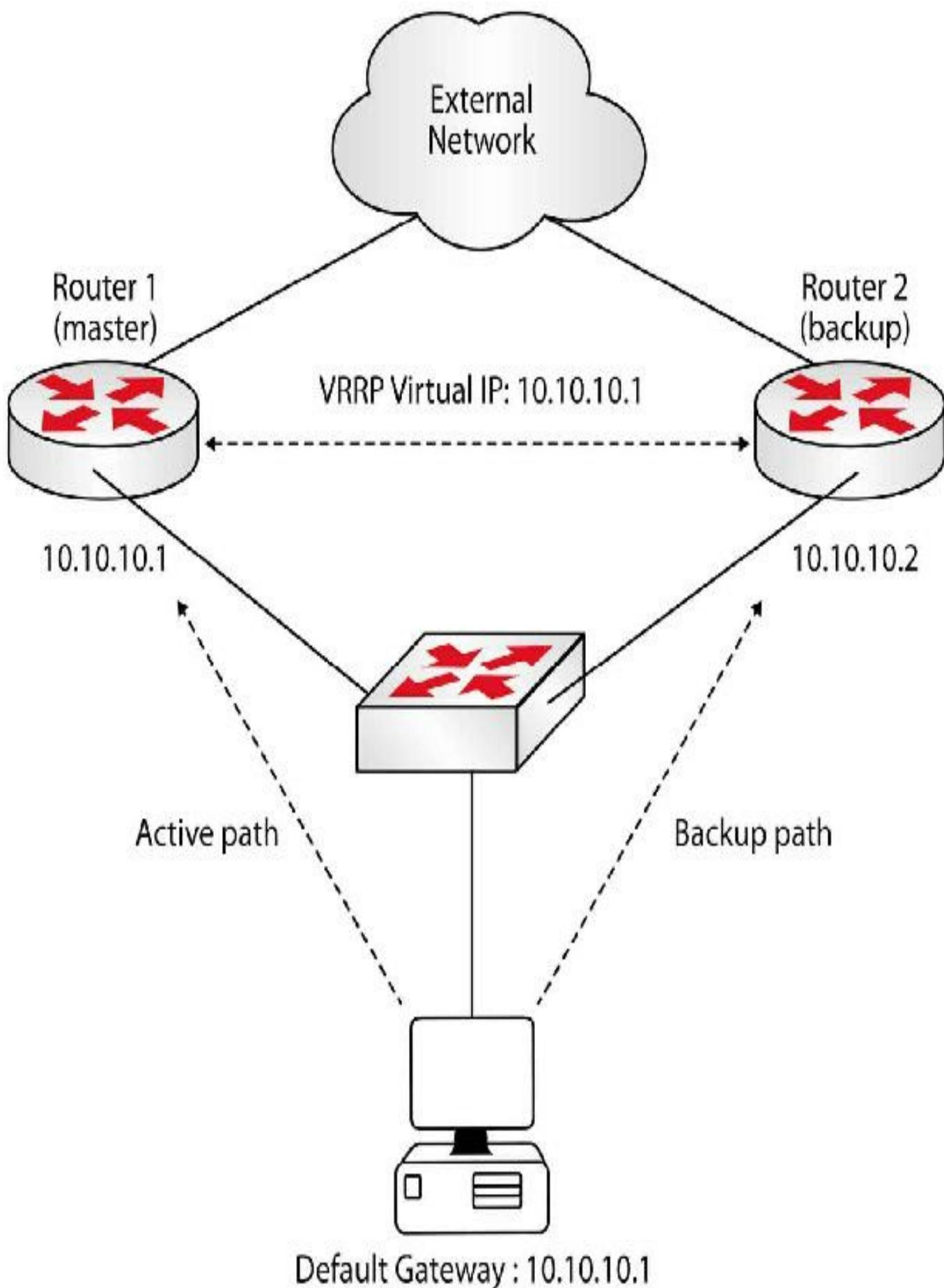


FIG 15.7 – Virtual Router Redundancy Protocol

As was the case for HSRP, the VRRP group presents a virtual IP address to the clients. An interesting aspect about VRRP is that you can utilize the virtual IP address using the same address that is allocated to the master device. In this case, the virtual address is configured as 10.10.10.1, identical to the address on the Router 1 interface.



This is very useful in real-world implementations when VRRP is used in public segments (Internet edge, DMZ, etc.)—it needs only two public IPs, while HSRP needs three public IPs.

The rest of the VRRP functionality details are the same as with HSRP, including authentication, interface tracking, etc. The configuration differences will be presented in detail in the next section.

Mini-lab – VRRP Configuration

Referring to Figure 15.7 above, Router 1 and Router 2 will act as edge routers toward the external network (you can consider this to be the Internet), and an internal host will be configured to use the VRRP address as the gateway to all external networks.

Start by defining the IP addresses on both routers:

```
R1(config)#int fa0/0
R1(config-if)#ip address 10.10.10.1 255.255.255.0
R1(config-if)#no sh
R2(config)#int fa0/0
R2(config-if)#ip address 10.10.10.2 255.255.255.0
R2(config-if)#no sh
R1#ping 10.10.10.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:

.!!!!

At this point you have connectivity between the routers. Proceed with configuring the VRRP virtual IP. This is done using the vrrp [group_id] ip [virtual_ip_address] command on the interfaces of both routers. As opposed to HSRP, VRRP allows you to use a virtual IP identical to the interface-level IP. Make Router 1 primary, configuring a priority of 120. Router 2 will have the default priority of 100.

```
R1(config)#int fa0/0
```

```
R1(config-if)#vrrp 1 ip 10.10.10.1
```

```
R1(config-if)#vrrp 1 priority 120
```

```
*Mar 1 01:01:55.643: %VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -] Master
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#vrrp 1 ip 10.10.10.1
```

```
R2(config-if)#{
```

```
*Mar 1 00:48:01.467: %VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -] Backup
```

Next, authenticate the VRRP peering session using an MD5 password in order to secure the VRRP connection.

```
R1(config)#int fa0/0
```

```
R1(config-if)#vrrp 1 authentication md5 key-string CCNA
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#vrrp 1 authentication md5 key-string CCNA
```

You can also define some additional parameters, such as a VRRP description. VRRP preemption is enabled by default, as opposed to HSRP.

```
R1(config)#int fa0/0
```

```
R1(config-if)#vrrp 1 description CCNA
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#vrrp 1 description CCNA
```

Next, check the status of the VRRP group on each router:

```
R1#sho vrrp
```

```
FastEthernet0/0 - Group 1
```

```
CCNA
```

```
State is Master
```

Virtual IP address is 10.10.10.1

Virtual MAC address is 0000.5e00.0101

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 120

Authentication MD5, key-string CCNA

Master Router is 10.10.10.1 (local), priority is 255

Master Advertisement interval is 1.000 sec

Master Down interval is 3.003 sec

R2#sho vrrp

FastEthernet0/0 - Group 1

CCNA

State is Backup

Virtual IP address is 10.10.10.1

Virtual MAC address is 0000.5e00.0101

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 100

Authentication MD5, key-string CCNA

Master Router is 10.10.10.1, priority is 255

Master Advertisement interval is 1.000 sec

Master Down interval is 3.609 sec (expires in 3.593 sec)

At this point the internal host can use the VRRP virtual address of 10.10.10.1 as the default gateway in order to access external networks.

[END OF MINI-LAB]

Configuring VRRP Interface Tracking

VRRP offers a facility referred to as object tracking (HSRP offers only interface tracking). VRRP cannot directly track an interface but it can track an object, which can be anything but most commonly is an interface. When tracked, the priority of the device

can be altered to allow the best VRRP router to take over as master.

In order to configure VRRP to track an interface (for example), a tracked object must be created in global configuration mode using the `track [object number] interface [line-protocol|ip routing]` global configuration command for interface tracking or the `track [object number] ip route [address/prefix] [reachability | metric threshold]` command for IP prefix tracking. Up to 500 objects may be tracked on the switch, depending on the software and platform. Tracked objects are then tracked by VRRP using the `vrrp [number] track [object]` interface configuration command.

The following output shows how to configure tracking for VRRP, referencing tracked object 1, which tracks the line protocol of the Loopback 0 interface:

```
Switch(config)#track 1 interface Loopback0 line-protocol
Switch(config-track)#exit
Switch(config)#interface vlan 192
Switch(config-if)#vrrp 1 track 1
Switch(config-if)#exit
```

The following output shows how to configure tracking for VRRP, referencing tracked object 2, which tracks the reachability of the 1.1.1.1/32 prefix. A tracked IP route object is considered to be up and reachable when a routing table entry exists for the route and the route is not inaccessible (i.e., has a route metric of 255), in which case the route is removed from the Routing Information Base (RIB) anyway:

```
Switch(config)#track 2 ip route 1.1.1.1/32 reachability
Switch(config-track)#exit
Switch(config)#interface vlan 192
Switch(config-if)#vrrp 1 track 2
```

VRRP tracking configuration is verified using the `show vrrp interface [name]` command. This is illustrated in the following output:

```
Switch#show vrrp interface vlan 192
Vlan192 - Group 1
“SWITCH-VRRP-Example”
State is Master
Virtual IP address is 192.168.1.254
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 0.100 sec
Preemption enabled
```

Priority is 105

 Track object 1 state Up decrement 10

 Track object 2 state Up decrement 10

Authentication MD5, key-string

Master Router is 192.168.1.1 (local), priority is 105

Master Advertisement interval is 0.100 sec

Master Down interval is 0.889 sec

To view the parameters of the tracked objects, use the show track [number][brief] [interface] [ip] [resolution][timers] command. The output of the show track command is illustrated as follows:

Switch#show track

Track 1

 Interface Loopback0 line-protocol

 Line protocol is Up

 1 change, last change 00:11:36

 Tracked by:

 VRRP Vlan192 1

Track 2

 IP route 1.1.1.1 255.255.255.255 reachability

 Reachability is Up (connected)

 1 change, last change 00:08:48

 First-hop interface is Loopback0

 Tracked by:

 VRRP Vlan192 1

NOTE: Tracked objects can also be used in conjunction with HSRP and GLBP.

GLBP

Gateway Load Balancing Protocol is Cisco proprietary and is the most unique of the First Hop Redundancy Protocols. With GLBP, you not only have the ability to achieve gateway redundancy but also the ability to load balance, and it is a lot easier to use more than two devices.

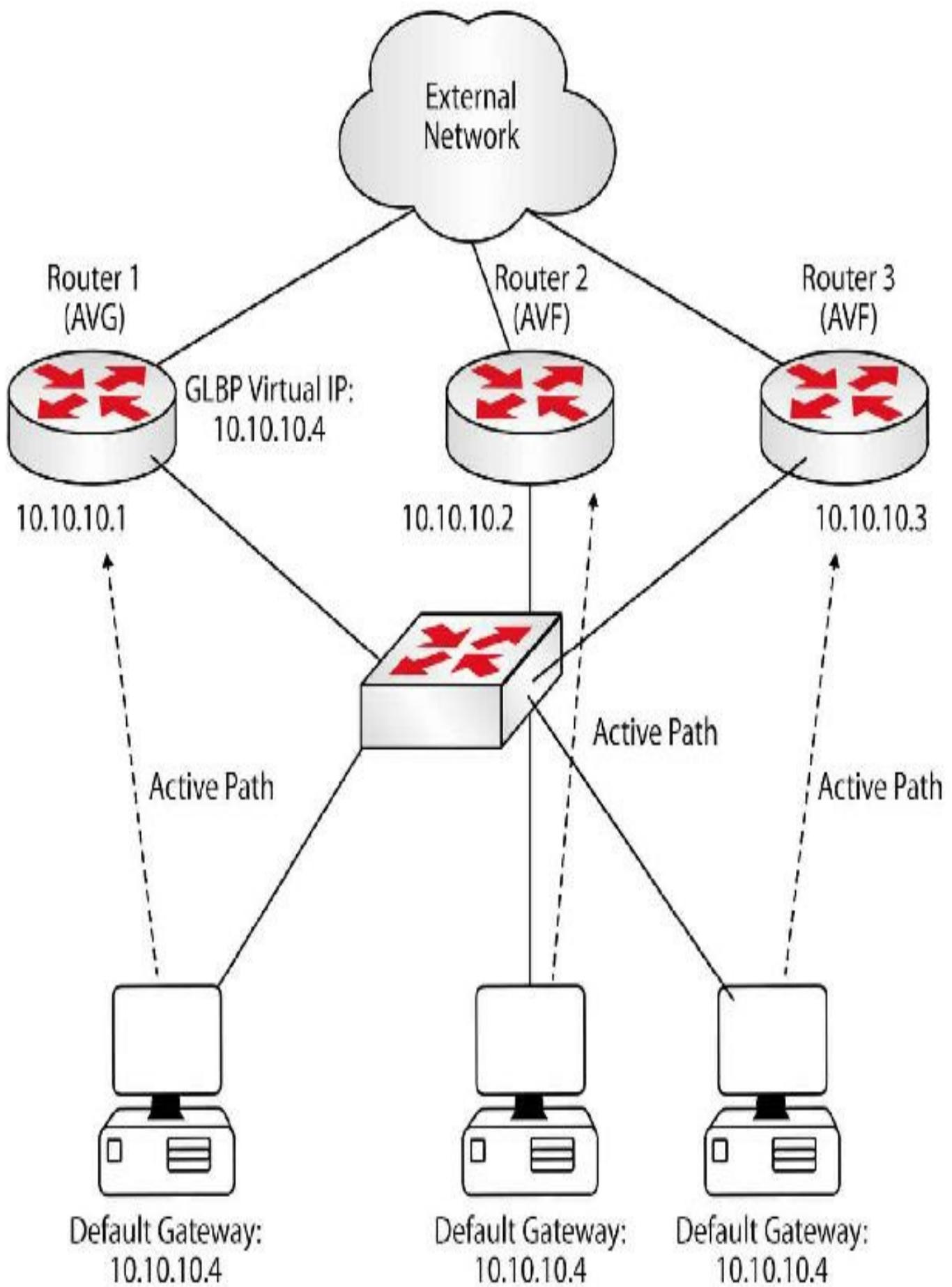


FIG 15.8 – Gateway Load Balancing Protocol

Let's consider an example in which you have three routers configured in a GLBP group that will be assigned a virtual default gateway address (10.10.10.4) also configured on the clients. One of the devices (Router 1 in this example) is elected AVG (Active Virtual Gateway) and the other devices are in the state of AVF (Active Virtual Forwarder). There can be up to four AVFs load-sharing simultaneously. In addition, GLBP supports up to 1,024 virtual routers (GLBP groups).

When the hosts ARP for the 10.10.10.4 MAC address, the AVG responds to the ARP requests and it can round-robin with the virtual MAC addresses of the AVF machines. Router 1 responds to the first ARP it receives with its own virtual MAC address, then it responds to the second ARP it receives with the second router's virtual MAC address, and then to the third ARP with the third router's virtual MAC address. In this way, the AVG can round-robin the traffic over the available AVF devices. This simplistic round-robin balancing approach can be changed in the configuration of other load-balancing techniques for GLBP.

NOTE: The AVG can also function as an AVF and it usually does so.

GLBP uses weights to determine the forwarding capacity of each group member. The assigned weight will determine the proportion of the total traffic that will be served by each AVF. The default weight value is 100. In addition, GLBP uses a number of load-balancing algorithms:

- Host-dependent – each client will be assigned to a unique AVF
- Round-robin – traffic is equally distributed across all AVFs by default
- Weighted – traffic is distributed based on the weight values; a higher weight value means the specific AVF MAC address will be used more frequently in ARP replies toward clients

Just as with HSRP and VRRP, GLBP can use plain text or MD5 authentication. The MD5 option offers a higher level of security so it is recommended over plain text authentication.

GLBP Configuration

Referring to Figure 15.8, Router 1 and Router 2 will act as edge routers toward the external network (you can consider this to be the Internet) and an internal host will be configured to use the GLBP address as the gateway to all external networks.

Start by defining the IP addresses on both routers:

```
R1(config)#int fa0/0
```

```
R1(config-if)#ip address 10.10.10.1 255.255.255.0
```

```
R1(config-if)#no sh
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#ip address 10.10.10.2 255.255.255.0
```

```
R2(config-if)#no sh
```

```
R1#ping 10.10.10.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:

.!!!!

At this point you have connectivity between the routers. Proceed with configuring the GLBP virtual IP. This is done using the `glbp [group_id] ip [virtual_ip_address]` command on the interfaces of both routers. Make Router 1 primary, configuring a priority of 120. Router 2 will have a priority of 100.

```
R1(config)#int fa0/0
```

```
R1(config-if)#glbp 1 ip 10.10.10.4
```

```
R1(config-if)#glbp 1 priority 120
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#glbp 1 ip 10.10.10.4
```

```
R2(config-if)#glbp 1 priority 100
```

Next, authenticate the GLBP peering session using an MD5 password in order to secure the GLBP connection.

```
R1(config)#int fa0/0
```

```
R1(config-if)#glbp 1 authentication md5 key-string CCNA
```

```
R2(config)#int fa0/0
```

```
R2(config-if)#glbp 1 authentication md5 key-string CCNA
```

You can also define some additional parameters, such as assigning a GLBP group name and adjusting the timers. By default, GLBP Hello packets are transmitted every three seconds, with a Dead timer of 10 seconds. Enable both routers for preemption (disabled by default).

```
R1(config)#int fa0/0
```

```
R1(config-if)#glbp 1 name CCNA
```

```
R1(config-if)#glbp 1 timers 1 3  
R1(config-if)#glbp 1 preempt  
R2(config)#int fa0/0  
R2(config-if)#glbp 1 name CCNA  
R2(config-if)#glbp 1 timers 1 3  
R2(config-if)#glbp 1 preempt
```

Next, adjust the AVF weights and the load-balancing mechanism so that Router 1 can forward twice as much traffic as Router 2 can. In order to do this, set the value on Router 1 at a value that's double the weight value on Router 2. You also need to configure the load-balancing mechanisms to weighted as opposed to the default behavior of round-robin.

```
R1(config)#int fa0/0  
R1(config-if)#glbp 1 weighting 200  
R1(config-if)#glbp 1 load-balancing weighted  
R2(config)#int fa0/0  
R2(config-if)#glbp 1 weighting 100  
R2(config-if)#glbp 1 load-balancing weighted
```

Next, check the status of the GLBP group on each router:

```
R1#show glbp  
FastEthernet0/0 - Group 1
```

State is Active

2 state changes, last state change 00:07:35

Virtual IP address is 10.10.10.4

Hello time 1 sec, hold time 3 sec

Next Hello sent in 0.292 secs

Redirect time 600 sec, forwarder timeout 14400 sec

Authentication MD5, key-string CCNA

Preemption enabled, min delay 0 sec

Active is local

Standby is 10.10.10.2, priority 100 (expires in 2.472 sec)

Priority 120 (configured)

Weighting 200 (configured 200), thresholds: lower 1, upper 200

Load balancing: weighted

IP redundancy name is CCNA

Group members:

c201.0f00.0000 (10.10.10.1) local

c202.2d6c.0000 (10.10.10.2) authenticated

There are 2 forwarders (1 active)

Forwarder 1

State is Active

1 state change, last state change 00:07:25

MAC address is 0007.b400.0101 (default)

Owner ID is c201.0f00.0000

Redirection enabled

Preemption enabled, min delay 30 sec

Active is local, weighting 200

Forwarder 2

State is Listen

4 state changes, last state change 00:04:17

MAC address is 0007.b400.0102 (learned)

Owner ID is c202.2d6c.0000

Redirection enabled, 599.880 sec remaining (maximum 600 sec)

Time to live: 14399.880 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.10.10.2 (primary), weighting 100 (expires in 2.880 sec)

R2#show glbp

FastEthernet0/0 - Group 1

State is Standby

3 state changes, last state change 00:05:20

Virtual IP address is 10.10.10.4

Hello time 1 sec, hold time 3 sec

Next Hello sent in 0.580 secs

Redirect time 600 sec, forwarder timeout 14400 sec

Authentication MD5, key-string CCNA

Preemption enabled, min delay 0 sec

Active is 10.10.10.1, priority 120 (expires in 2.508 sec)

Standby is local

Priority 100 (default)

Weighting 100 (default 100), thresholds: lower 1, upper 100

Load balancing: weighted

IP redundancy name is CCNA

Group members:

c201.0f00.0000 (10.10.10.1) authenticated

c202.2d6c.0000 (10.10.10.2) local

There are 2 forwarders (1 active)

Forwarder 1

State is Listen

MAC address is 0007.b400.0101 (learned)

Owner ID is c201.0f00.0000

Time to live: 14398.960 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.10.10.1 (primary), weighting 200 (expires in 1.956 sec)

Forwarder 2

State is Active

3 state changes, last state change 00:05:09

MAC address is 0007.b400.0102 (default)

Owner ID is c202.2d6c.0000

Preemption enabled, min delay 30 sec

Active is local, weighting 100

At this point the internal host can use the GLBP virtual address of 10.10.10.4 as the default gateway in order to access external networks. Traffic will be load balanced between Router 1 and Router 2 based on the configured weights.

You can see from the output above that the unique MAC address for GLBP follows the format of 0007.b400.xxxy, with xx as the GLBP group and yy as a different number for each router. When hosts ARP for the IP address (virtual) the AVG will reply with one of the virtual MAC addresses, thus achieving load balancing as opposed to using just one active router.

Syslog

Syslog is a standard used to generate and collect informational messages; to use the vernacular, it's a system for logging messages.

Routers and switches (as well as many other network devices) maintain timestamped logs reporting events that may be of interest to us as network administrators. Depending on the device, it can report these events in real time (using debugging commands), send the logs to another network device, or record them in NVRAM to be examined at a convenient time. To enable the router to record the logs in NVRAM, you would use the logging buffered command.

Syslog is defined in RFC 5424 and it is supported on a wide range of devices and platforms. Because of this, syslog can be used to integrate log data from many different types of systems into a central repository. It is a lightweight client/server event notification protocol that initially worked over UDP (User Datagram Protocol) but now uses TCP (Transmission Control Protocol).

Syslog servers (also called syslog daemons) are entities that listen to syslog messages generated by network devices. Usually, data is sent in clear text but mechanisms have been developed that secure syslog data using Transport Layer Security (TLS), which is the successor of Secure Sockets Layer (SSL) encryption. Syslog uses port number 514. The advantages of logging to an actual server (rather than a router) include saving router resources (and space) and the ability to interrogate historic logs and alerts in detail.

RFC 5424 ruled that syslog over UDP is now obsolete and that syslog must now support TLS using [port number](#) 6514 by default, so firewalls should be configured to pass packets for this port.

The syslog standard defines eight severity levels. On Cisco routers the default is Level 7 (Debug):

- 0 (Emergency) – This is a very urgent notification that requires immediate

attention. The system is unusable when such a message is generated.

- 1 (Alert) – This is an urgent notification, announcing to the system administration that action must be taken immediately to correct a problem.
- 2 (Critical) – This message indicates a failure that should be corrected as soon as possible but is usually associated with a secondary system, such as the loss of a backup connection.
- 3 (Error) – This message indicates a non-urgent failure.
- 4 (Warning) – Warning messages are related to issues that might lead to an error if action is not taken.
- 5 (Notice) – This indicates an event that is unusual but is not an error condition. No immediate action required.
- 6 (Informational) – These are normal operational status messages.
- 7 (Debug) – This information is useful for developers or for network engineers during a troubleshooting process.

It's worth noting that whichever level you choose to monitor, the levels above it will also be included; for example, if you choose Level 3 (Error), Levels 2 through 0 will also be monitored.

Mini-lab – Cisco IOS Syslog Configuration

When configuring syslog in a network environment, the first thing you should do is make sure that you have a consistent time/clock setting on all network devices. This will be very useful when trying to correlate log messages from multiple devices when analyzing an event that happened in the network at a specific time.

As discussed earlier, on a Cisco device you have two ways of configuring the clock. Manual clock configuration was demonstrated earlier in the guide. You can also configure the time zone:

- Manual clock configuration
- NTP (Network Time Protocol)

R1(config)#clock timezone CST -4

Or use NTP:

R1(config)#ntp server 10.10.10.100

After you have made sure that the correct time and clock settings are configured on all network devices involved in the log collecting process, the next step is to enable the timestamps service on the devices. This permits the system to associate a timestamp with each generated syslog message.



The timestamp feature should be on by default but here is how you enable it if it's been deactivated by another network administrator:

```
R1(config)#service timestamps
```

As with all the commands in this guide, there are several configuration options but these fall outside the CCNA exam. If you want to know more about the other permutations, you can find this information at Cisco.com. Next, specify the host that will collect the log messages; in other words, the syslog server. You need to install the syslog server on a server or workstation with a fixed IP address.

```
R1(config)#logging host 10.9.9.10
```

Next, the router needs to know what kinds of messages should be exported to the syslog server. This is accomplished by defining the severity level (configured severity level + all higher severity level messages are sent):

```
R1(config)#logging trap ?
```

[0-7]	Logging severity level
alerts	Immediate action needed (severity=1)
critical	Critical conditions (severity=2)
debugging	Debugging messages (severity=7)
emergencies	System is unusable (severity=0)
errors	Error conditions (severity=3)
informational	Informational messages (severity=6)
notifications	Normal but significant conditions (severity=5)
warnings	Warning conditions (severity=4)

[cr]

```
R1(config)#logging trap informational
```

You can also specify a syslog facility to be used. In syslog, the facility is used to represent the source that generated the message. This source can be a process on the local device, an application, or even an operating system. By default, Cisco IOS devices use facility local7 (Level 7 – Debug) to send syslog messages.

```
R1(config)#logging facility ?
```

```
auth Authorization system
```

```
cron Cron/at facility
```

```
daemon System daemons
```

```
kern Kernel
```

```
local0 Local use
```

```
local1 Local use
```

```
local2 Local use
```

```
local3 Local use
```

```
local4 Local use
```

```
local5 Local use
```

```
local6 Local use
```

```
local7 Local use
```

```
lpr Line printer system
```

```
mail Mail system
```

```
news USENET news
```

```
sys10 System use
```

```
sys11 System use
```

```
sys12 System use
```

```
sys13 System use
```

```
sys14 System use
```

```
sys9 System use
```

```
syslog Syslog itself
```

```
user User process
```

```
uucp Unix-to-Unix copy system
```

```
R1(config)#logging facility local1
```

You can verify the syslog configuration using the following command:

```
R1#show logging
```

Syslog logging: enabled (12 messages dropped, 0 messages rate-limited, 0 flushes, 0

overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: level debugging, 55 messages logged,
xml disabled, filtering disabled

Monitor logging: level debugging, 0 messages logged,
xml disabled, filtering disabled

Buffer logging: disabled, xml disabled,
filtering disabled

Logging Exception size (4096 bytes)

Count and timestamp logging messages: disabled

Persistent logging: disabled

No active filter modules.

ESM: 0 messages dropped

Jun 1 10:26:14.467: %SYS-6-LOGGINGHOST_STARTSTOP: **Logging to host 10.9.9.10 port 514 started - CLI initiated**

Trap logging: level informational, 58 message lines logged

Logging to 10.9.9.10 (udp port 514, audit disabled,
authentication disabled, encryption disabled, link up),
2 message lines logged,
0 message lines rate-limited,
0 message lines dropped-by-MD,
xml disabled, sequence number disabled
filtering disabled

Please do try all of these commands on a router. Reading about them is simply not enough to get them to sink in.

[END OF MINI-LAB]

SNMP

Simple Network Management Protocol is an application layer protocol that runs over

UDP ports 161/162 and is used by network administrators to gather statistics and control network devices. SNMP is used to share management information between network devices, usually between a management workstation and routers, switches, or other devices. SNMP has two components:

- The SNMP server (manager), which is usually a dedicated workstation
- The SNMP agent, which is usually a service on the managed network device

The manager is usually a workstation or server running free or commercial Network Management Station (NMS) software, which in the case of Cisco is CiscoWorks or one of Cisco's newer network management software solutions.



Look into Cisco Prime Infrastructure if you are interested in this area.

The SNMP agent is a piece of software that resides on network devices, such as servers, routers, or switches. The agent provides data to the manager. The manager may send requests from any available source port to port 161 on the agent. The agent will respond to the relevant source port on the manager. The manager receives notifications (Traps and Inform Requests) on port 162.

SNMP has evolved over the years and has now reached version 3. SNMPv3 provides authentication and privacy, which makes it more secure than the earlier SNMP versions (1 and 2). SNMP is used by network administrators and engineers to:

- Monitor network performance
- Troubleshoot
- Plan scalable enterprise solutions and intelligent services
- Configure the network remotely

SNMP accesses detailed information in MIBs (Management Information Bases) and it uses SNMP agents. The MIB is an object-oriented hierarchical database system stored locally on the network device. An MIB entry example is 1.3.6.1.2.1.2.1.20.0, with 1 being the root of the MIB tree and 0 being the final leaf.

The SNMP agent is used to send and receive information from the device to the NMS and the other way around. In order to do that, different types of SNMP messages are

used. The NMS will run network management software that retrieves and displays the SNMP information in a GUI (graphical user interface) format. The information displayed is used for control, troubleshooting, and planning. Several companies provide SNMP GUI-based software.

Using SNMP, the administrator can gather reports from the network device regarding parameters like CPU utilization, memory utilization, or interface bandwidth utilization. The managed device contains the SNMP agent and the MIB that stores all the information. Different types of messages are used in order to get information from the NMS to/from the managed device (the monitored device).

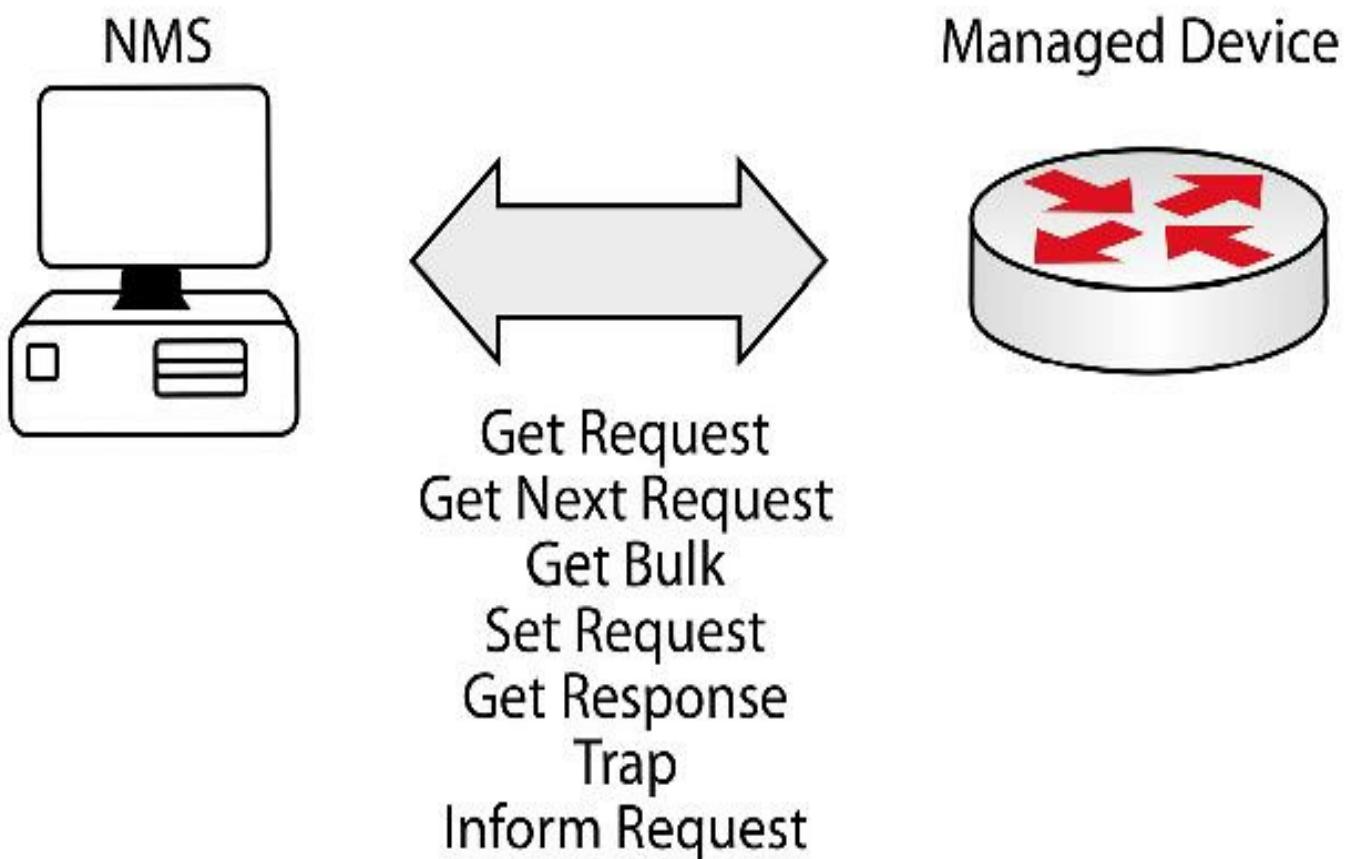


FIG 15.9 – SNMP messages

The first message is called the Get Request. This is sent to the managed device when the NMS wants to get a specific MIB variable from the SNMP agent that runs on that device. The Get Next Request information is used to return the next object in the list after the Get Request message returned a value. The Get Bulk message was added in SNMPv2 and still works in SNMPv3 environments; it can be used to retrieve a big chunk of data (an entire table) and it reduces the need to have to use many Get Request and Get Next Request messages. This reduces the overhead on the bandwidth utilization on the link.

The Set Request message is also sent by the NMS and is used to set an MIB variable on

the agent. The Get Response message is the response from the SNMP agent to the NMS Get Request, Get Next Request, or Get Bulk messages.

A Trap is used by the SNMP agent to transmit unsolicited alarms to the NMS when certain conditions occur (e.g., device failure, state change, or parameter modifications). Different thresholds can be configured on the managed device for different parameters (e.g., disk space, CPU utilization, memory utilization, or bandwidth utilization), and Traps are sent when the defined thresholds are reached. SNMPv3 introduced another message called the Inform Request. This is similar to a Trap message and is what a managed device will send to the NMS as an acknowledgment to other messages.

Multiple SNMP versions have been developed since SNMP was created, and you will need to know the differences for the CCNA exam:

- SNMPv1 – the initial implementation of SNMP, offering limited security via community strings
- SNMPv2c – the revised SNMP version, which includes multiple enhancements but still offers limited security functionality via community strings
- SNMPv3 – the latest SNMP version, which offers advanced security features, including authentication (using HMAC-SHA-2 Authentication Protocol), message integrity (prevents packet tampering), and encryption; the preferred and most secure version

SNMP v1 and v2 were both criticized because the only security available was authentication using a password (community string) sent in clear text between a manager and an agent. This was addressed in SNMPv3, which provides three security levels:

- NoAuthNoPriv – no authentication and no privacy mechanisms
- AuthNoPriv – authentication (MD5, SHA) but no privacy mechanisms
- AuthPriv – the highest level, uses authentication (MD5, SHA) and privacy (DES, for example)

Let's examine a few examples of the parameters monitored by an NMS (graphical view) via SNMP. Figure 15.10 below illustrates a sample report for interface utilization on a network device:

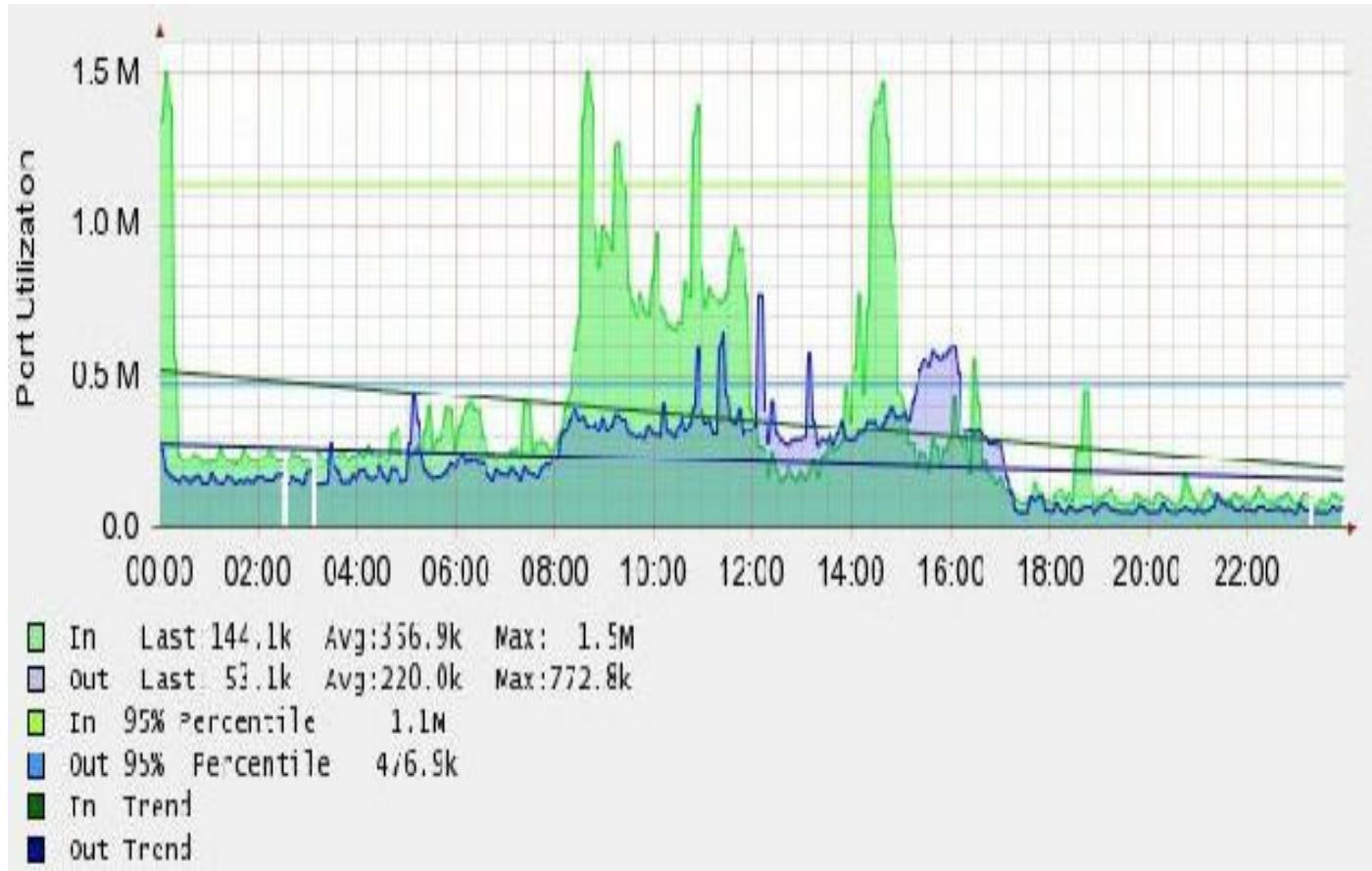


FIG 15.10 – SNMP interface utilization example

Figure 15.11 below illustrates a sample report for CPU parameters (utilization, temperature, and fan) on a network device:

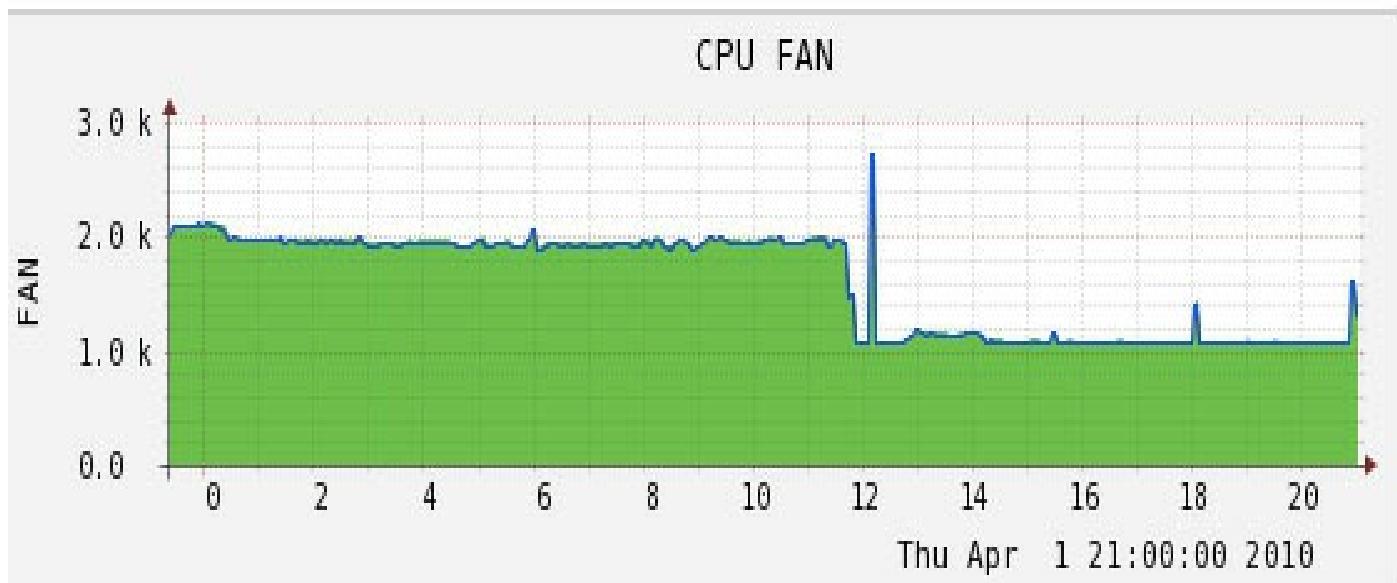
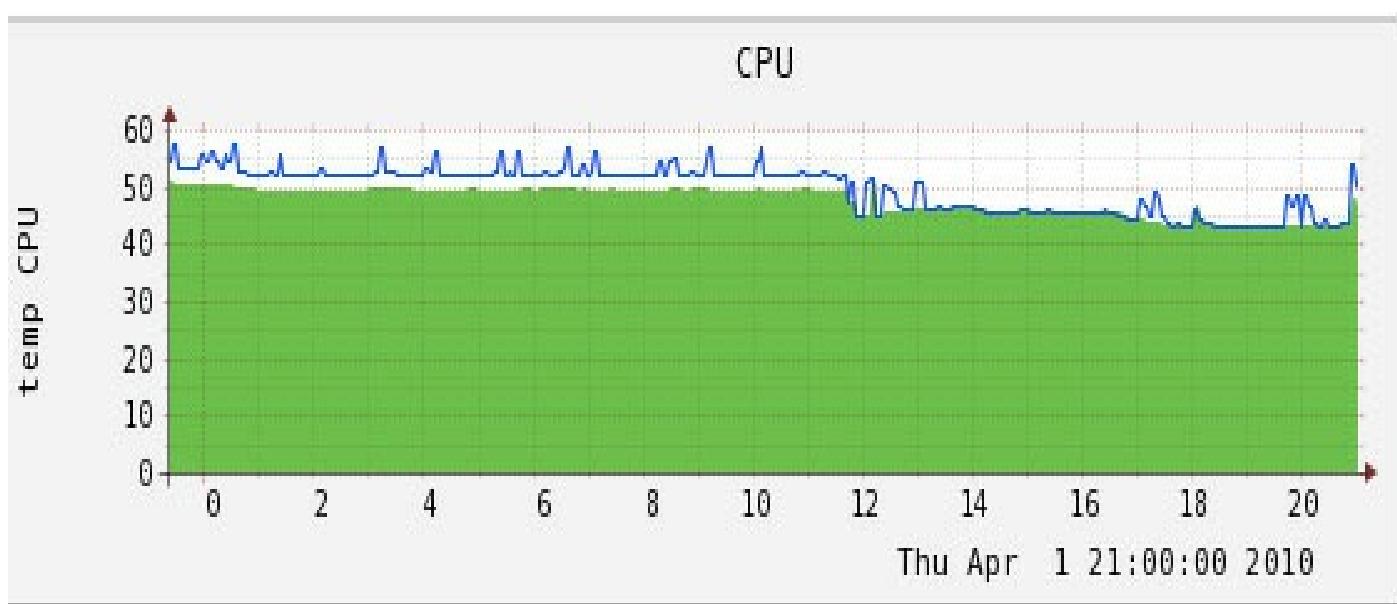
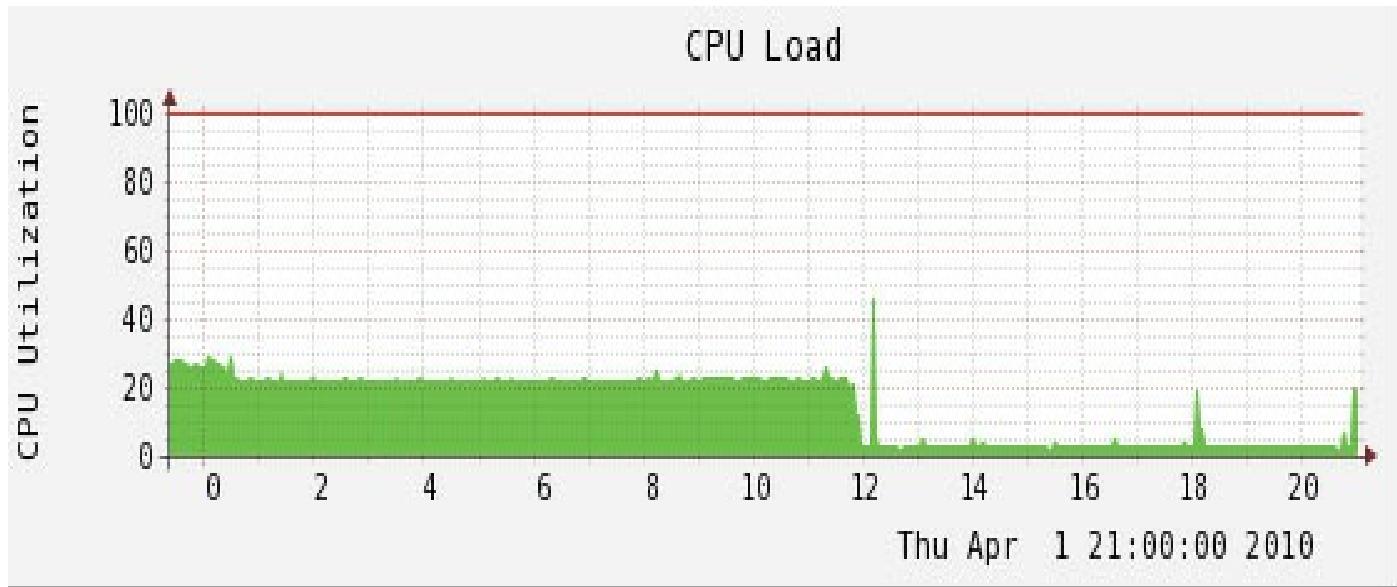


FIG 15.11 – SNMP CPU monitoring example

MIB browsers are dedicated software tools that allow the management of SNMP-enabled network devices. They allow administrators to load standard and proprietary MIBs and issue SNMP requests to retrieve data or make changes to an agent. A sample interface example of an MIB browser is presented below:

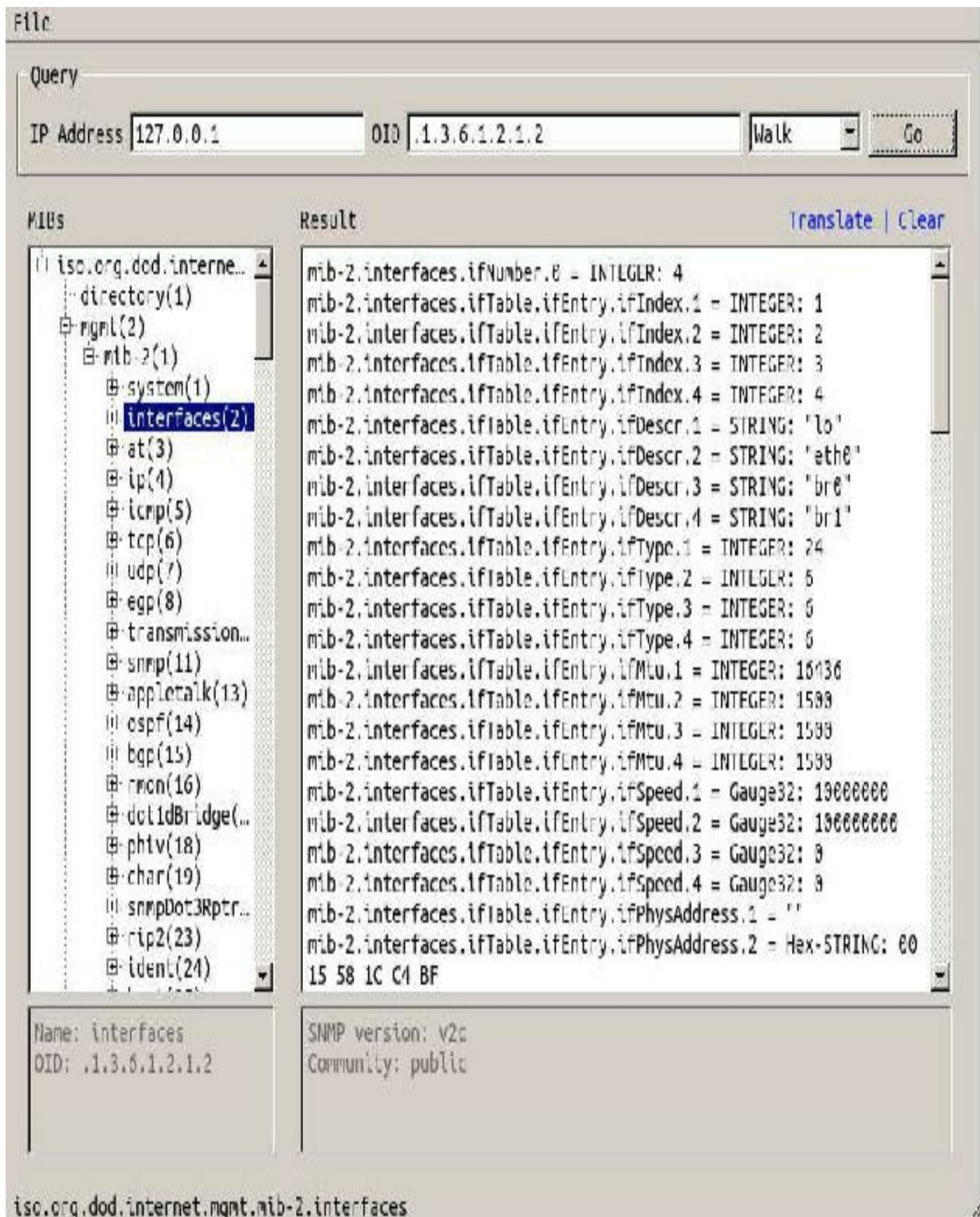


FIG 15.12 – MIB browser example

Cisco IOS SNMP Configuration

SNMPv2 can be configured on Cisco devices by defining read-only and/or read-write communities on the device. Communities can be limited to a specific address range trying to get access. The address range can be defined with an access list:

```
R1(config)#snmp-server community public RO  
R1(config)#snmp-server community private RW 10  
R1(config)#access-list 10 permit 10.0.0.0 0.255.255.255
```

In the output above, the public community is defined as read-only, which is available to anyone, and the private community is defined as read-write, which is available only to users within the range defined in access list 10.

In order to enable SNMP Traps so that the Cisco device will actively export information to a specific host, you need to define the SNMP host that will collect the information:

```
R1(config)#snmp-server host 10.0.0.99 traps
```

Then, you have to enable the SNMP Traps by using the following command to enable all available Traps:

```
R1(config)#snmp-server enable traps
```

You can fine-tune the Trap types that you want to enable by associating the command with additional parameters:

```
R1(config)#snmp-server enable traps ?  
atm           Enable SNMP atm traps  
bgp           Enable BGP traps  
bulkstat      Enable Data-Collection-MIB Collection notifications  
ccme          Enable SNMP ccme traps  
cnpd          Enable NBAR Protocol Discovery traps  
config        Enable SNMP config traps  
config-copy   Enable SNMP config-copy traps  
cpu           Allow cpu related traps  
dial          Enable SNMP dial control traps  
dnis          Enable SNMP DNIS traps  
ds0-busyout   Enable ds0-busyout traps
```

[output omitted]

When configuring SNMPv3, you need to decide which hosts should be allowed to query the network device via SNMP. You need to configure an access list to define this:

```
R1(config)#access-list 10 permit 10.0.0.0 0.255.255.255
```

Next, create an SNMP view that restricts what data the SNMP user will be able to access. For this example, include mib-10 as a sample MIB:

```
R1(config)#snmp-server view CCNA_VIEW mib-10 included
```

You also need to create an SNMP group and assign it the view you just created:

```
R1(config)#snmp-server group CCNA_GROUP ?
```

v1 group using the v1 security model

v2c group using the v2c security model

v3 group using the User Security Model (SNMPv3)

```
R1(config)#snmp-server group CCNA_GROUP v3 ?
```

auth group using the authNoPriv Security Level

noauth group using the noAuthNoPriv Security Level

priv group using SNMPv3 authPriv security level

```
R1(config)#snmp-server group CCNA_GROUP v3 priv ?
```

access specify an access-list associated with this group

context specify a context to associate these views for the group

match context name match criteria

notify specify a notify view for the group

read specify a read view for the group

write specify a write view for the group

[cr]

```
R1(config)#snmp-server group CCNA_GROUP v3 priv read ?
```

WORD read view name

```
R1(config)#snmp-server group CCNA_GROUP v3 priv read CCNA_VIEW
```

The last step is to create an SNMPv3 user and associate it with the SNMP group created previously:

```
R1(config)#snmp-server user CCNA_USER ?
```

WORD Group to which the user belongs

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP ?
```

remote Specify a remote SNMP entity to which the user belongs

v1 user using the v1 security model

v2c user using the v2c security model

v3 user using the v3 security model

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP v3 ?
```

access specify an access-list associated with this group

auth authentication parameters for the user

encrypted specifying passwords as MD5 or SHA digests

[cr]

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP v3 encrypted ?
```

access specify an access-list associated with this group

auth authentication parameters for the user

[cr]

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP v3 encrypted auth ?
```

md5 Use HMAC MD5 algorithm for authentication

sha Use HMAC SHA algorithm for authentication

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP v3 encrypted auth md5 ?
```

WORD authentication password for user

```
R1(config)#snmp-server user CCNA_USER CCNA_GROUP v3 encrypted auth md5  
PSSWD ?
```

Additional optional SNMP parameters that can be configured on the network device include the SNMP location and contact, which are just informative aspects sent in plain text:

```
R1(config)#snmp-server location CCNA_LAB
```

```
R1(config)#snmp-server contact ccna@cisco.com
```

It's worth noting that adding SNMP to network devices will increase CPU load, so you should test it or get advice from Cisco TAC before implementing it on your network.

“The reason behind the High CPU usage can be caused by the Network Management Server (SNMP Server) like HP OpenView querying for the

Routing Tables and ARP tables to learn about other networks or querying for certain MIBs which can be resource intensive” (© DebianAdmin.com).

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 15 exam.

Chapter 15 Labs

Lab 1: HSRP

The physical topology is shown in Figure 15.13 below:

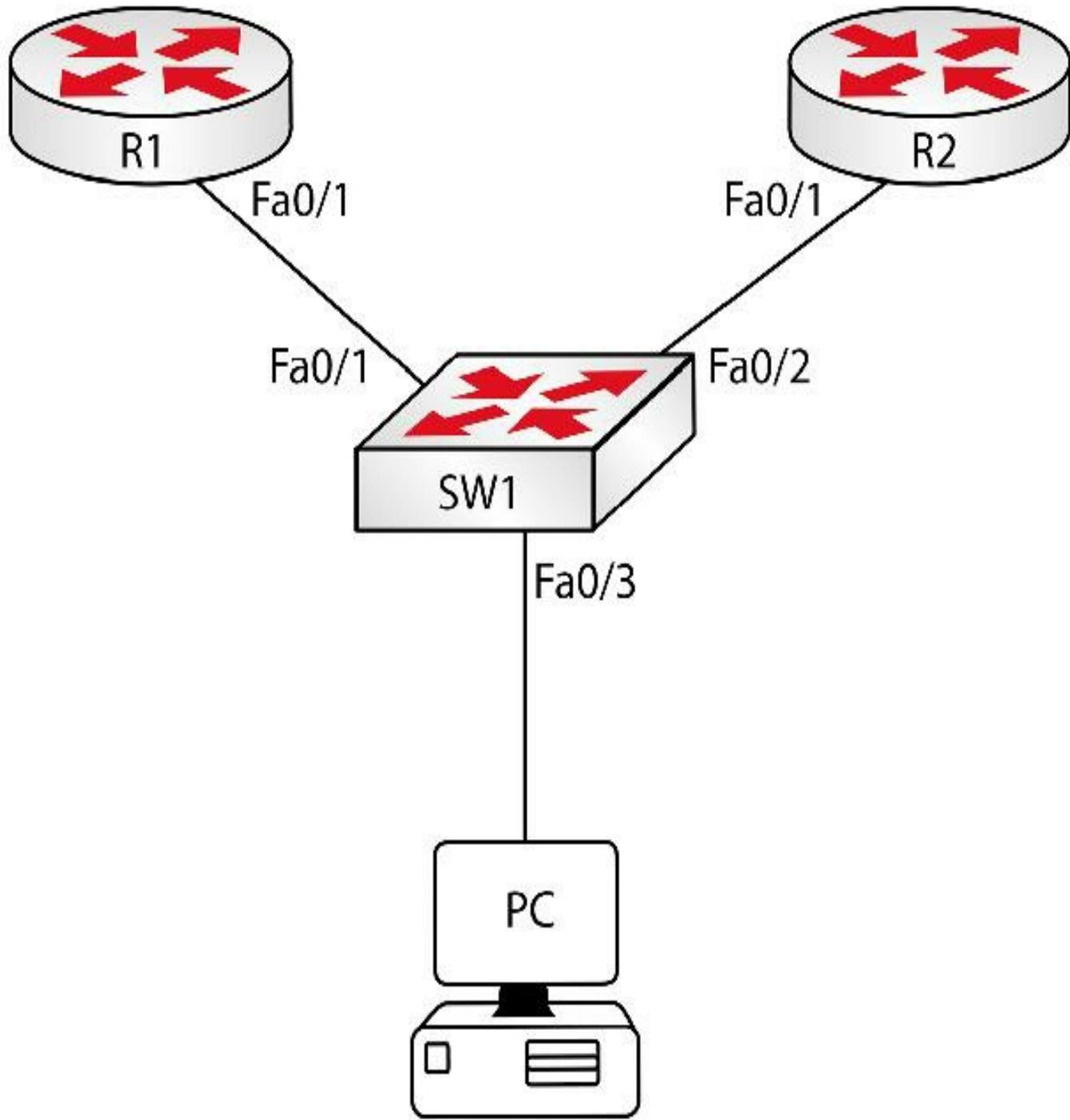


FIG 15.13 – HSRP Lab

Lab Exercise

Your task is to configure the network in Figure 15.13 to allow the workstation to have connectivity to the HSRP group created on the two routers.

Purpose

HSRP is a very popular FHRP protocol and is in wide use today. You will need to have a good working knowledge of it for the CCNA exam and as a Cisco engineer.

Lab Objectives

1. Configure IP addresses on the router interfaces.
2. Configure HSRP on the routers.
3. Fine-tune the HSRP configuration as per the configuration guidelines presented in this chapter.
4. Verify workstation connectivity and HSRP configuration.

Lab Walk-through

1. Configure the IP address 192.168.0.100/24 and the gateway address 192.168.0.10/24 (HSRP group address) on the workstation:

Internet Protocol Version 4 (TCP/IPv4) Properties



General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically

Use the following IP address:

IP address:

192 . 168 . 0 . 100

Subnet mask:

255 . 255 . 255 . 0

Default gateway:

192 . 168 . 0 . 10

Obtain DNS server address automatically

Use the following DNS server addresses:

Preferred DNS server:

[Empty text box]

Alternate DNS server:

[Empty text box]

Validate settings upon exit

Advanced...

OK

Cancel

2. Configure IP addressing on the routers: 192.168.0.1/24 and 192.168.0.2/24 on the switch-facing interfaces.

```
R1(config)#int fa0/1
R1(config-if)#ip add 192.168.0.1 255.255.255.0
R1(config-if)#no sh
*Mar 1 00:07:38.915: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:07:39.915: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

```
R2(config)#int fa0/1
R2(config-if)#ip add 192.168.0.2 255.255.255.0
R2(config-if)#no sh
*Mar 1 00:07:50.647: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:07:51.647: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

3. Configure HSRP group 10 on the switch-facing interfaces using the 192.168.0.10 address. Name the HSRP group CCNA. Control the election of the primary HSRP gateway using priority 110 on R1 and 100 on R2.

```
R1(config)#int fa0/1
R1(config-if)#standby 10 ip 192.168.0.10
R1(config-if)#standby 10 name CCNA
R1(config-if)#standby 10 priority 110
*Mar 1 00:09:34.987: %HSRP-5-STATECHANGE: FastEthernet0/1 Grp 10 state Standby -] Active
```

```
R2(config)#int fa0/1
R2(config-if)#standby 10 ip 192.168.0.10
R2(config-if)#standby 10 name CCNA
R2(config-if)#standby 10 priority 100
R2(config-if)#
*Mar 1 00:10:16.719: %HSRP-5-STATECHANGE: FastEthernet0/1 Grp 10 state Speak -] Standby
```

4. Configure HSRP preemption on both routers:

```
R1(config-if)#standby 10 preempt
R2(config-if)#standby 10 preempt
```

5. Adjust HSRP timers to 1 and 3 seconds:

```
R1(config-if)#standby 10 timers 1 3  
R2(config-if)#standby 10 timers 1 3
```

6. Configure MD5 HSRP authentication between the routers:

```
R1(config-if)#stand 10 authentication md5 key-string CCNA  
R2(config-if)#stand 10 authentication md5 key-string CCNA
```

7. Verify HSRP configuration on the routers:

```
R1(config-if)#do sho standby  
FastEthernet0/1 - Group 10  
  State is Active  
    2 state changes, last state change 00:03:49  
    Virtual IP address is 192.168.0.10  
    Active virtual MAC address is 0000.0c07.ac0a  
      Local virtual MAC address is 0000.0c07.ac0a (v1 default)  
    Hello time 1 sec, hold time 3 sec  
      Next Hello sent in 0.032 secs  
    Authentication MD5, key-string CCNA  
    Preemption enabled  
    Active router is local  
    Standby router is 192.168.0.2, priority 100 (expires in 1.976 sec)  
    Priority 110 (configured 110)  
    Group name is CCNA (cfgd)
```

```
R2(config-if)#do sho standby  
FastEthernet0/1 - Group 10  
  State is Standby  
    4 state changes, last state change 00:00:53  
    Virtual IP address is 192.168.0.10  
    Active virtual MAC address is 0000.0c07.ac0a  
      Local virtual MAC address is 0000.0c07.ac0a (v1 default)  
    Hello time 1 sec, hold time 3 sec  
      Next Hello sent in 0.468 secs  
    Authentication MD5, key-string CCNA  
    Preemption enabled  
    Active router is 192.168.0.1, priority 110 (expires in 2.516 sec)  
    Standby router is local  
    Priority 100 (default 100)
```

Group name is CCNA (cfgd)

8. Test PC to HSRP group connectivity:

C:\Users\ccna]ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:

Reply from 192.168.0.10: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.10:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 1ms, Average = 1ms

Lab 2: Syslog

The physical topology is shown in Figure 15.14 below:



FIG 15.14 – Syslog Lab

Lab Exercise

Your task is to configure syslog functionality on a single router (R1).

Purpose

You need to learn how to configure router logging functionality. This is a very commonly used tool for monitoring network devices.

Lab Objectives

1. Configure the logging facility.
2. Globally enable logging and select severity.

3. Configure a logging server.
4. Verify logging functionality.

Lab Walk-through

R1(config)#logging facility local2

R1(config)#logging 192.168.0.10

R1(config)#logging trap informational

R1(config)#logging buffered informational

R1(config)#do show logging

Syslog logging: enabled (12 messages dropped, 0 messages rate-limited, 0 flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: level debugging, 16 messages logged, xml disabled, filtering disabled

Monitor logging: level debugging, 0 messages logged, xml disabled, filtering disabled

Buffer logging: level informational, 0 messages logged, xml disabled, filtering disabled

Logging Exception size (4096 bytes)

Count and timestamp logging messages: disabled

Persistent logging: disabled

No active filter modules.

ESM: 0 messages dropped

Trap logging: level informational, 20 message lines logged

Logging to 192.168.0.10 (udp port 514, audit disabled,
authentication disabled, encryption disabled, link down),
0 message lines logged,
0 message lines rate-limited,
0 message lines dropped-by-MD,
xml disabled, sequence number disabled

filtering disabled

Log Buffer (4096 bytes):

Lab 3: SNMP

The physical topology is shown in Figure 15.15 below:



FIG 15.15 – SNMP Lab

Lab Exercise

Your task is to configure SNMP functionality on a single router (R1).

Purpose

You need to learn how to configure router SNMP functionality. This is a very commonly used tool for monitoring and managing network devices.

Lab Objectives

1. Configure the SNMP server on the router.
2. Configure RO and RW communities.

Lab Walk-through

```
R1(config)#snmp-server host 192.168.0.10 CCNA
```

```
R1(config)#snmp-server community ro CCNA_RO
```

```
R1(config)#snmp-server community rw CCNA_RW
```

Now please type the below two show commands and note the output:

```
R1#show snmp
```

```
R1#show snmp community
```


Chapter 16 — WAN Technologies

What You Will Learn in This Chapter

PPP and PPPoE

Frame Relay

Metro Ethernet

VSAT

T1/E1

ISDN

DSL

Cable

Cellular Networks

VPN Technologies

MPLS

Syllabus Topics Covered

5.0 WAN Technologies

5.1 Identify different WAN technologies

5.1.a Metro Ethernet

5.1.b VSAT

5.1.c Cellular 3G/4G

5.1.d MPLS

5.1.e T1/E1

5.1.f ISDN

5.1.g DSL

5.1.h Frame Relay

5.1.i Cable

5.1.j VPN

5.2 Configure and verify a basic WAN Serial connection

5.3 Configure and verify a PPP connection between Cisco routers

5.4 Configure and verify Frame Relay on Cisco routers

5.5 Implement and troubleshoot PPPoE

Much the same as with routing protocols, we tend to configure WAN connections and

then forget about them. We usually only need to address them if there is a connection issue or the need to add a remote office connection.

For the CCNA exam, Cisco expects you to understand the common WAN protocols available. Unfortunately, they also expect you to understand some legacy technologies such as Frame Relay and ISDN, which were dropped from the CCIE exam some time ago because they are no longer relevant.

WAN Technologies

Wide Area Networks (WANs) are used to connect Local Area Networks (LANs) together. Because of the long distances involved, you will normally have to use a third-party company known as a service provider/telephone company (telco) to provide this service.

When I started working at Cisco Systems in the UK in 2002, the leading WAN connection types supported were ISDN, T1, and Frame Relay. Of course, technology has since improved and now you have a range of options open to you depending on your required bandwidth, security requirements, budget, and location. Even entry-level Cisco routers for branch offices now feature WAN support for multimode VDSL2/ADSL2/2+, multimode G.SHDSL, ISDN, xDSL, Ethernet, 3G and 4G, and fiber.

Common WAN Networking Terms

You will hear many terms when discussing WAN technologies, such as the following:

Customer Premise Equipment

CPE is any equipment owned and maintained internally and located on your premises. If the CPE breaks, it will be your responsibility to resolve the problem as the network engineer.

Data Terminal Equipment

This is normally the interface on your side of the WAN link that connects to the telcos network. The DTE interface uses clocking signals generated by the DCE interface to synchronize traffic. Your network router will almost always be the DTE side of the network.

Data Communications Equipment

DCE interfaces provide connections to the service provider's network. Here traffic is forwarded, data is synchronized, and clocking signals are provided. When practicing networking with routers at home, you will have to configure your own DCE interface because one end of the cable will be DTE and the other end will be DCE. On the DCE

end of the cable, you simply add the clock rate command and give a clocking number. You can normally tell which end of the cable is DCE because it will have the letters DCE stamped on it (see [Chapter 1](#)). A DCE interface is normally defined by the cable and not the actual interface. You can check which type of cable you have attached with the show controllers serial x command, where x is the number of the interface. If you have a DCE cable attached to the interface, you need to add the clock rate # command:

```
RouterA#config t  
RouterA(config)#interface Serial0/0  
RouterA(config-interface)#clock rate 56000 i Sets the speed to 56,000 bps
```

Point of Demarcation

Normally, inside a switching closet in the communications room is where the CPE meets the local loop, which is the point of demarcation, or demarc for short. This is usually installed by the service provider as the termination of a digital service line, such as T1, T3, E1, E3, etc.

Local Loop

This is the cabling and connectivity that extends from the demarc to the nearest local telco switch or exchange. It can sometimes cause confusion that the local loop includes only the interface, trunk, or line card of the telco device connected to the other end of the circuit. Figure 16.1 below illustrates the local loop and the telco CO, which we will cover next:

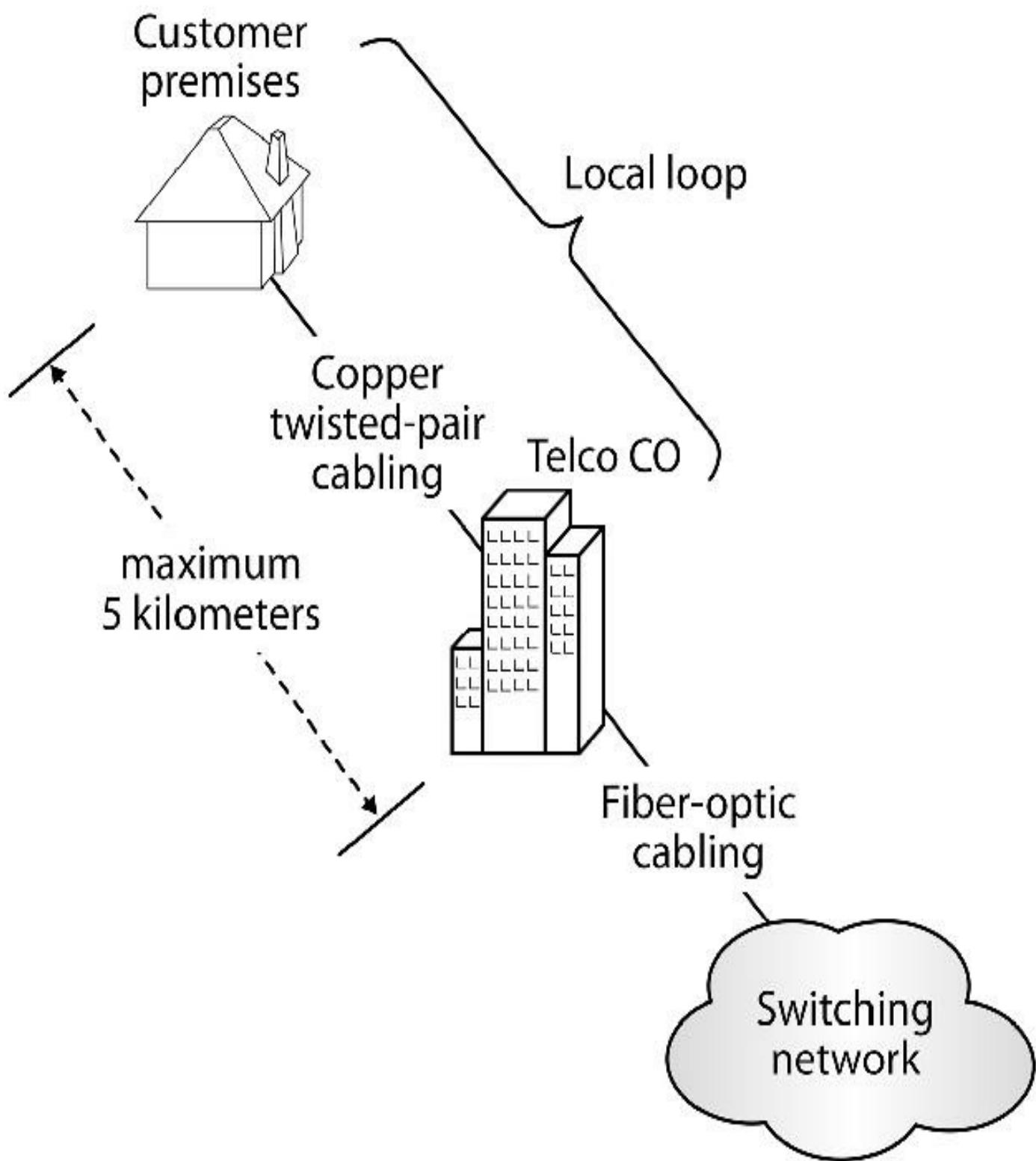


FIG 16.1 – The local loop

Telco Central Office

The telco CO is the main point of presence for the telco's WAN service to the end-user. This is also referred to as simply the central office.

Channel Service Unit/Data Service Unit

DCE equipment also includes a utility referred to as a channel service unit/data service unit (CSU/DSU). This device provides the conversion from your LAN data format to one compatible with your telco's requirements. Although you might think that a CSU/DSU works in the same way as a modem, they actually do different things entirely. A CSU/DSU converts digital signals from a router to a leased line (the local loop), while a modem converts digital signals from a router to a phone line (the local loop).

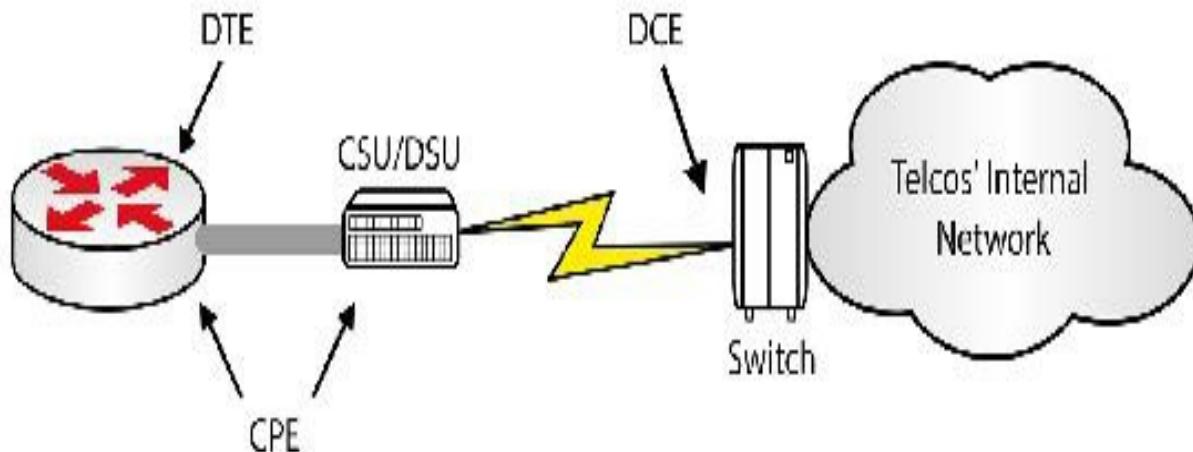


FIG 16.2 – Common WAN terms

Modems

A modem converts digital signals to analog and back again. It MODulates data over frequencies outbound and DEModulates the signal received.

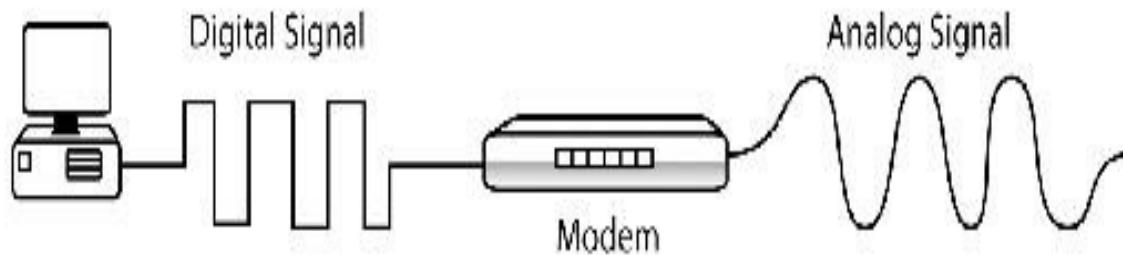


FIG 16.3 – A modem converts digital signals to analog signals

The correct name for a modem is modulator/demodulator. The purpose of a modem is to

convert a digital signal to an analog signal for use across Plain Old Telephone Service (POTS) lines. It is used for small bandwidth requirements or more commonly as a backup solution. Note that, as shown in Figure 16.3 above, the modem will terminate an analog line (local loop) coming into your network.

WAN Connection Types

There are three main connection types for WANs:

1. Leased lines
2. Circuit switching
3. Packet switching

Leased Line

A leased line is a dedicated connection between your site and another site. It can also be referred to as a point-to-point link. The link is not shared with any other company and is available 24 hours a day. Leased lines can be very expensive, depending on bandwidth and distance, but they do eliminate some of the security and traffic engineering problems associated with connections to your remote site over the Internet or with shared connections.



Leased Line (point-to-point)

FIG 16.4 – A point-to-point leased line

Leased lines are usually created for point-to-point connections. They typically result in a high-quality connection but they offer limited flexibility.

Circuit Switching

Just like for a telephone call, for a circuit-switched connection to take place, a dedicated temporary connection has to be made between the end-devices. Once the session is no longer required between the two end-devices, the connection is normally

torn down. Circuit switching can be very cost-effective but the speeds are slow.

All packets traveling along the WAN take the same path in a circuit-switched connection. Integrated Services Digital Network (ISDN) is an example of a circuit-switched network.

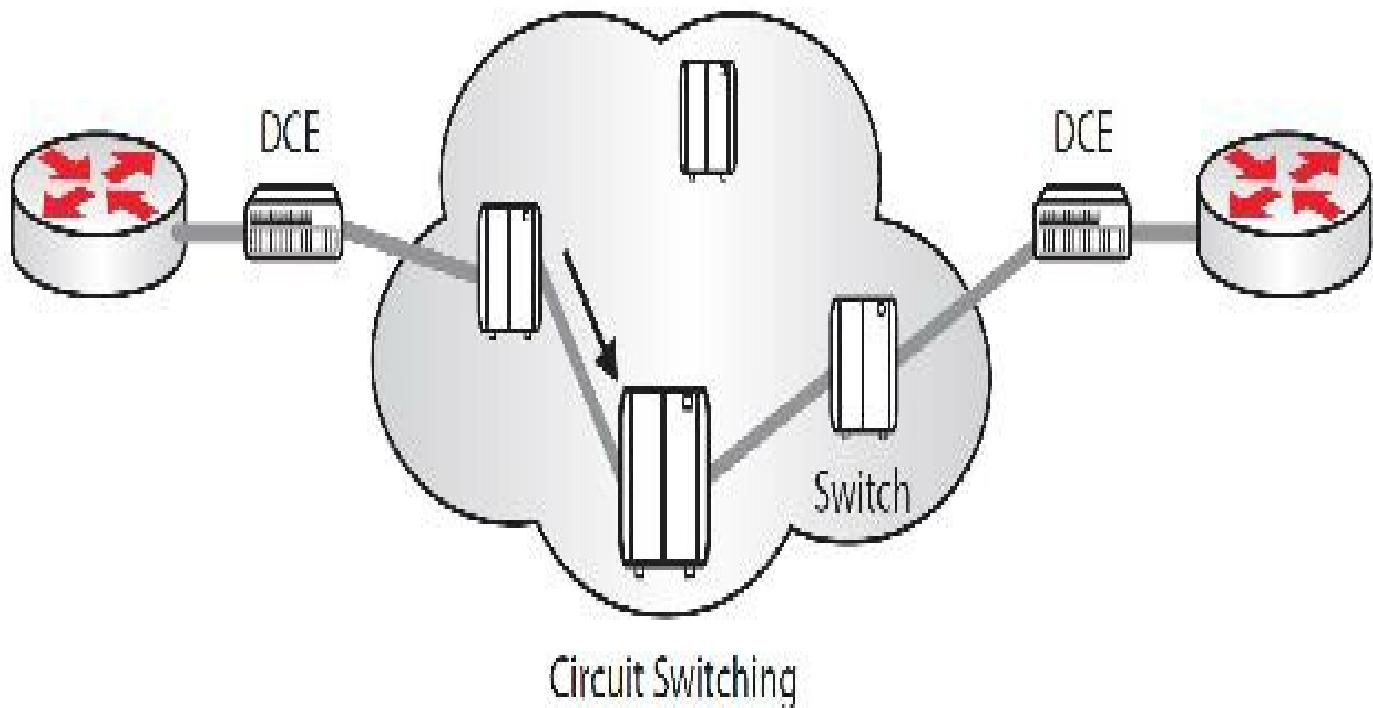


FIG 16.5 – A circuit-switched network

Packet Switching

In packet-switched connections, users may share a connection with other networks. The cost is generally less for users since the telco can make more efficient use of their bandwidth. End-to-end connectivity in packet-switched networks is known as having virtual circuits (VCs). Common examples of packet-switched networks are Frame Relay, ATM, and X.25.

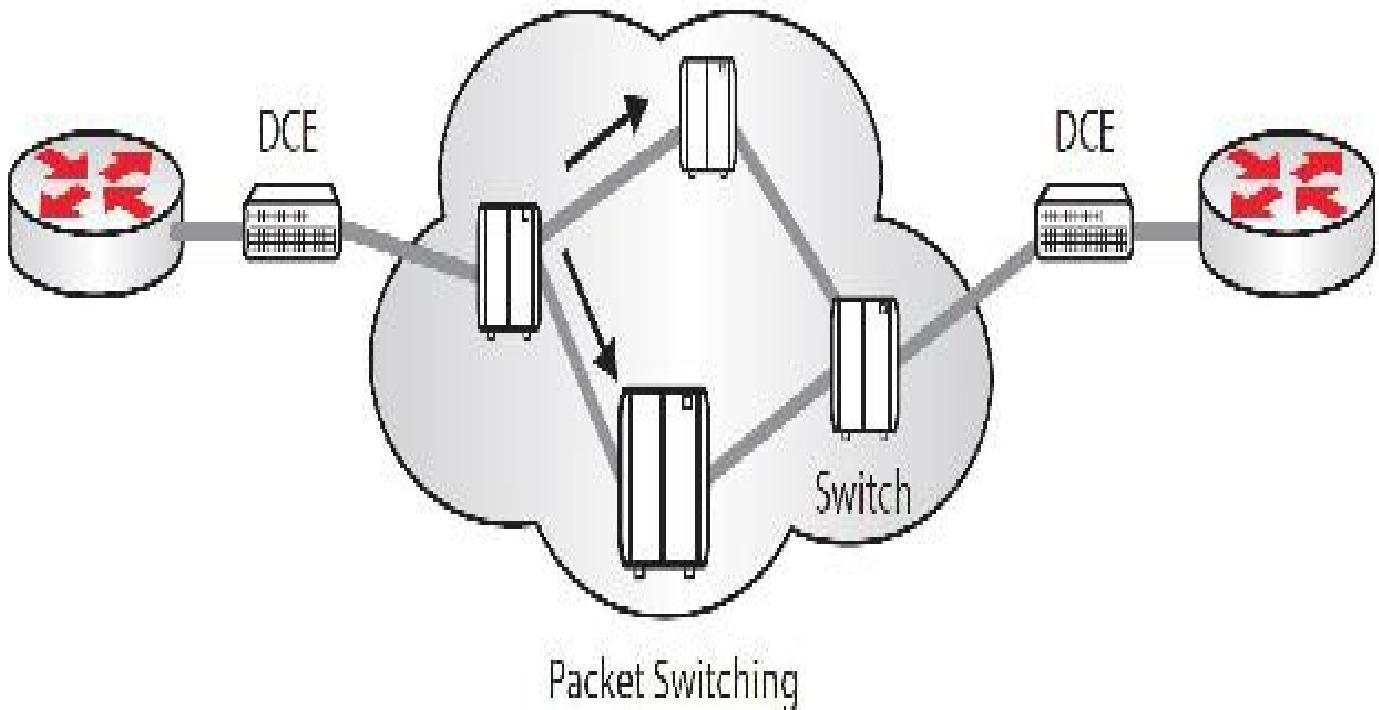


FIG 16.6 – A packet-switched network

With a packet-switched connection, you have no choice in which path your data takes. Typically, the service provider's policy will allow for an optimal path, which is decided depending on how much traffic is saturating their connections. When your data arrives at the other end, it is reassembled and put into the correct order.

Packet switching is very efficient but can be complex to configure, especially for large networks spanning multiple locations.

Point-to-Point Protocols

There are several protocols you can use when connecting over a WAN. Some are compulsory when you use a certain service and some you can choose from. When you pay for a leased line for a point-to-point connection, you will normally choose HDLC or PPP.

High-Level Data Link Control

HDLC is a layer 2 protocol used for WAN connectivity. It is based primarily on IBM's Synchronous Data Link Control (SDLC) protocol. HDLC uses keepalives to monitor connectivity with the remote end-device.

The DCE side of the connection sends the DTE side a keepalive packet containing a sequence number. The DTE side echoes this sequence number back to the DCE, proving connectivity. If three consecutive packets are not received, the link is declared down.

You can monitor the keepalives on an HDLC link with the debug serial interface

command. You can test this command on any lab where you are using Serial interfaces. Although HDLC is a widely used protocol, Cisco has created their own proprietary version, so if you are connecting a Cisco device to a non-Cisco device you will not be able to use it. Configuring it on an interface is very straightforward. Remember, though, that it is on by default on Cisco Serial interfaces, so you don't need to configure the encapsulation.

```
Router#config t  
Router(config)#interface Serial0/0  
Router(config-if)#encapsulation hdlc i Sets the encapsulation type  
Router(config-if)#ip address 192.168.1.1 255.255.255.0  
Router(config-if)#no shutdown  
Router(config-if)#^Z  
Router#
```

You can check your interface protocol settings (and many other interface settings) by typing show interface serial 0/0. You would normally never need to set the encapsulation type to HDLC on a Cisco router since it is the default.

```
Router#show interface Serial0/0  
Serial0 is up, line protocol is up  
Hardware is HD64570  
Internet address is 192.168.1.1/24  
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation HDLC, Loopback not set i Encapsulation setting  
Keepalive set (10 sec)
```



FIG 16.7 – The same encapsulation type must be on each side of the connection

HDLC uses 10-second keepalives to verify the integrity of the connection. The DTE and DCE ends increment a set of sequence numbers that you should see increment with a

debug serial packet command. Three missed keepalives will cause the link to be deactivated.

R1#debug serial packet

Serial network interface debugging is on

*Mar 1 00:21:52.727: Serial0/0: HDLC myseq 0, mineseen 0, yourseen 0, line up

Mar 1 00:22:02.727: Serial0/0: HDLC myseq 1, mineseen 1, yourseen 4, line up

Mar 1 00:22:12.727: Serial0/0: HDLC myseq 2, mineseen 2, yourseen 5, line up

You can debug the actual interface with the debug serial interface command.

Another important command to know and use is show interface serial 0/0 (or whatever your interface number is). The output below is truncated:

R1#show interface s0/0

Serial0/0 is up, line protocol is up

Hardware is GT96K Serial

Internet address is 192.168.1.1/24

MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255

Encapsulation HDLC, Loopback not set

Keepalive set (10 sec)

0 carrier transitions

DCD=up DSR=up DTR=up RTS=up CTS=up

It's worth checking Cisco documentation for the meaning of the fields, many of which have been discussed throughout this manual; however, you should see that the interface, line protocol, correct IP address and subnet, correct encapsulation type, and DCD through CTS are all up.

Point-to-Point Protocol

PPP is very popular for use over dedicated and circuit-switched links, as well as when you are connecting to non-Cisco equipment. PPP is specified in RFC 1661. You would have to use PPP if you were connecting your Cisco router to a non-Cisco router over a Serial line.

PPP is popular because it is vendor-neutral and it can work over many different connection types, including synchronous (clocks on both sides agree), asynchronous

(clocks differ), ISDN, Digital Subscriber Line (DSL), and High Speed Serial Interface (HSSI) links. In addition, PPP has built-in error detection and data compression and it supports authentication with CHAP and PAP, as well as network-layer address negotiation.

PPP is made up of two main components—NCP and LCP:

- **Network Control Protocol (NCP)** – a family of independent protocols that encapsulate network layer protocols, such as TCP/IP
- **Link Control Protocol (LCP)** – negotiates, sets up, and tears down control options for the data link connection to the WAN

PPP Authentication

PPP offers optional authentication and has two ways of authenticating the calling router —PAP and CHAP:

- Password Authentication Protocol (PAP) – This protocol uses a two-way handshake, allowing the remote host to authenticate itself. The password is sent in clear text so it can easily be captured and read.
- Challenge Handshake Authentication Protocol (CHAP) – This protocol uses a three-way handshake and never sends the password over the link in clear text. Instead, a hashed value made from the password is sent. This hashed value can only be read by a host with the appropriate key to the MD5 algorithm, which is a very strong level of encryption.

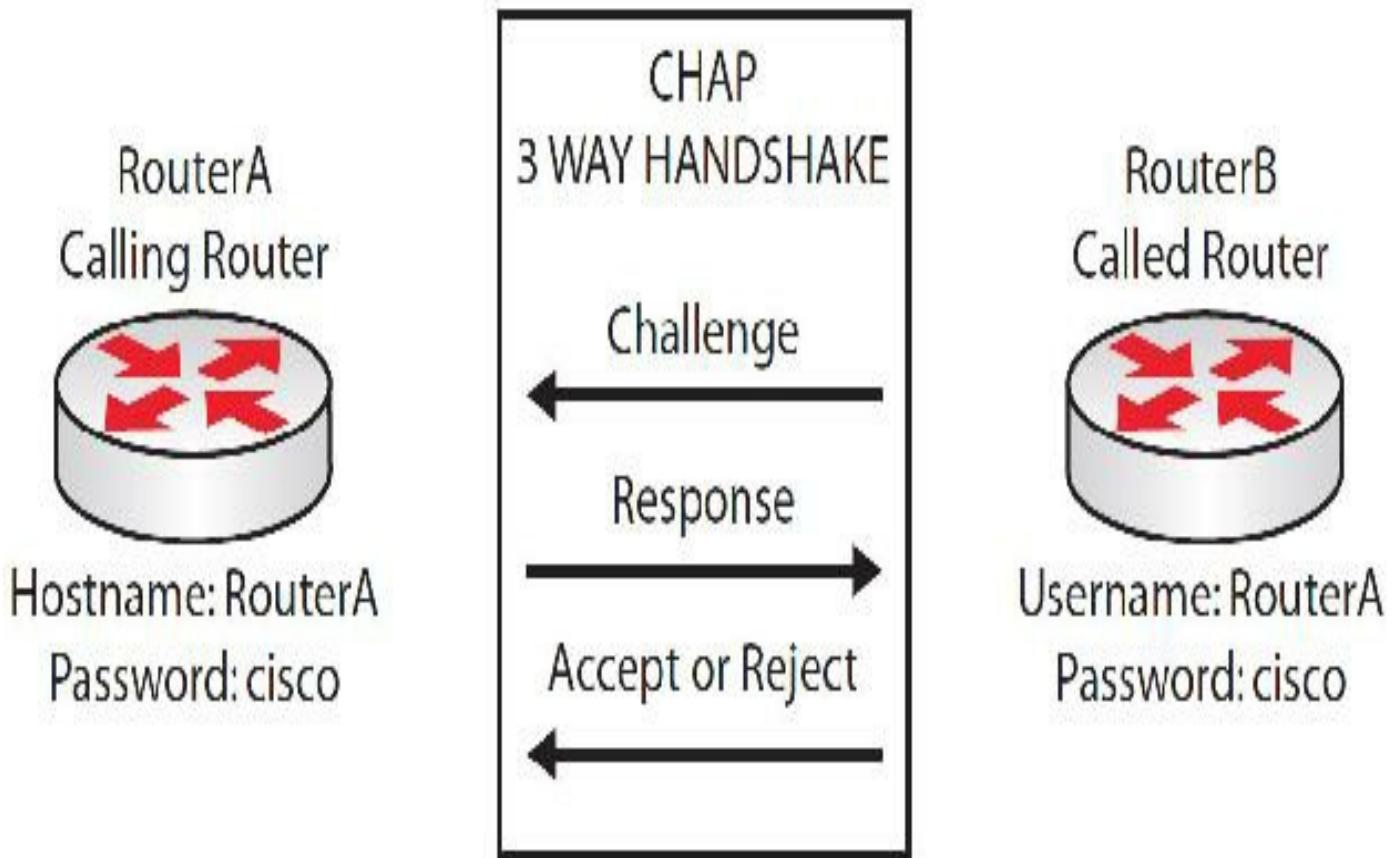


FIG 16.8 – CHAP uses a three-way handshake

On the calling router, a hostname and username/password must be added. Of course, encapsulation must be set to PPP and authentication type to CHAP. CHAP will continue to carry out authentication on the line once the connection is established using the three-way handshake.

On the called or authenticating router, a hostname and username/password of which routers will be calling has to be configured. AAA security can also be used with PPP, but this is outside the CCNA syllabus.

LCP Configuration Options

Cisco routers offer several configuration options to use with some of the features LCP offers, as shown In Table 16-1 below:

Table 16-1: LCP Options

Feature	Operation	Protocol	Command
Authentication	Requires a password, performs challenge handshake	PAP, CHAP	ppp authentication pap ppp authentication chap
Compression	Compresses data at the source and decompresses	Stacker, Predictor	ppp compress stacker ppp compress predictor

	data at the destination		
Error Detection	Monitors dropped data, avoids frame looping	Quality, Magic Number	ppp quality [number 1-100]
Multilink	Performs multiple-link load balancing	Multilink Protocol	ppp multilink

Mini-lab – Configuring PPP

PPP can easily be configured by changing the encapsulation type from HDLC to PPP. Optionally, you can add many other features, including authentication, compression, link quality, and a raft of ISDN options. For the CCNA exam, you should be comfortable configuring PPP with CHAP authentication.

We will do a full lab at the end of this chapter, but here is a brief demonstration of the configuration commands you need to enable PPP with CHAP. You can optionally add a second method of authentication to be used if the first methods fails, so CHAP and then PAP, or vice versa.

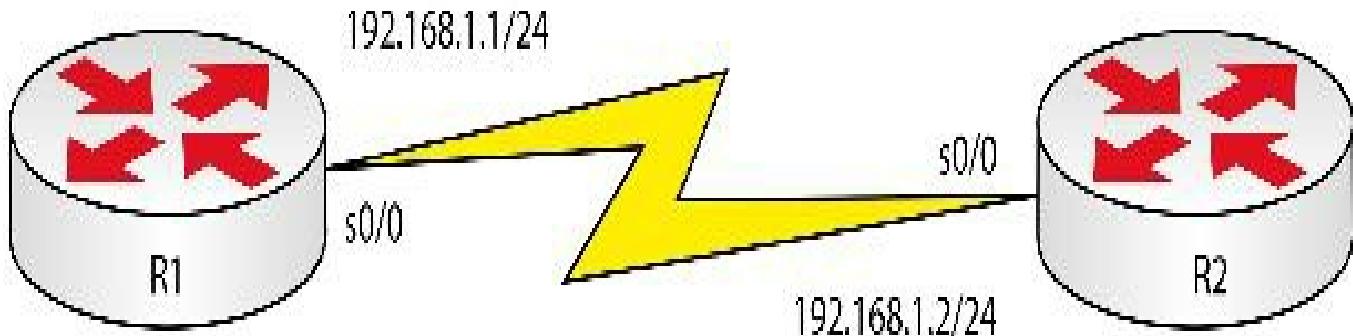


FIG 16.9 – Mini-lab: Configuring PPP

```
R1(config)#username R2 password howtonetwork
```

```
R1(config)#int s0/0
```

```
R1(config-if)#ip add 192.168.1.1 255.255.255.0
```

```
R1(config-if)#encapsulation ppp
```

```
R1(config-if)#ppp authentication chap pap
```

```
R1(config-if)#no shut
```

```
R1(config-if)#end
```

NOTE: The `ppp authentication chap pap` command may not work on some IOS versions, so stick to `chap` only if this is the case.

```
R2(config)#int s0/0
```

```
R2(config-if)#ip add 192.168.1.2 255.255.255.0
R2(config-if)#clock rate 64000
R2(config-if)#no shut
R2(config-if)#encap ppp
R2(config-if)#ppp authentication chap pap
R2(config-if)#exit
R2(config)#username R1 password howtonetwork
R2(config)#exit
R2#sh int s0/0
Serial0/0 is up, line protocol is up
Hardware is GT96K Serial
Internet address is 192.168.1.2/24
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
[END OF MINI-LAB]
```

There are other ways to configure CHAP and PAP authentication; however, this is the easiest.

PPPoE

The Point-to-Point Protocol over Ethernet is a network protocol that allows encapsulating PPP packets inside Ethernet frames. It is often used in the context of DSL connections as a solution for tunneling packets over the DSL connection to the ISP's IP network. This solution allows authentication, encryption, and traffic compression, which makes it a very attractive option from an ISP's point of view.

The PPP session authenticates the user based on a username or password via the PAP or CHAP protocols. PPPoE has two distinct stages:

- PPPoE Discovery
- PPP Session

The PPPoE Discovery stage allows the MAC addresses of the endpoints to be known before the PPP control packets are exchanged so a connection can be established over Ethernet. Once the MAC addresses of the two peers are known and the session has been

established, the Session stage will start.

Mini-lab – PPPoE Configuration

A PPPoE session needs special configuration both on the ISP (server) side and on the customer (client) side. The server device is configured by creating a broadband aggregation (BBA) group, which will be associated with a virtual template. The virtual template will be configured with a pool of IP addresses to be assigned to PPPoE clients. You can authenticate the connection using two methods: PAP or CHAP. For this mini-lab, use CHAP authentication, which is more secure.

The configuration on the server side is as follows:

```
ISP(config)#bba-group pppoe ISP
ISP(config-bba-group)#virtual-template 1
ISP(config-bba-group)#exit
ISP(config)#interface virtual-template 1
ISP(config-if)#ip address 10.0.0.1 255.255.255.0
ISP(config-if)#peer default ip address pool POOL
ISP(config-if)#ppp authentication chap
ISP(config-if)#exit
ISP(config)#username CCNA_USER password CCNA_PW
ISP(config)#ip local pool POOL 10.0.0.2 10.10.10.254
ISP(config)#interface FastEthernet0/1
ISP(config-if)#no ip address
ISP(config-if)#pppoe enable group ISP
```

On the customer side (PPPoE client), you need to define a dialer interface that will be associated to the physical interface connecting to the ISP via a dialer pool ID. Secure the connection using CHAP. The configuration on the client side is as follows:

```
R1(config)#interface dialer1
R1(config-if)#dialer pool 1
R1(config-if)#encapsulation ppp
R1(config-if)#ip address negotiated
R1(config-if)#ppp authentication chap
```

```
R1(config-if)#interface FastEthernet0/1
R1(config-if)#no ip address
R1(config-if)#pppoe-client dial-pool-number 1
R1(config-if)#exit
R1(config)#username CCNA_USER password CCNA_PW
```

You can see the connected customers on the PPPoE server as follows: (you won't have an output because you aren't actually connected to an ISP):

```
ISP#show pppoe session
```

1 client session

Uniq ID	PPPoE	RemMAC	Port	Source	VA	State
SID	LocMAC				VA-st	
N/A	16	ca00.4843.0008	Fa0/0	Di1	Vi1	UP
		ca01.4843.0008				UP

[END OF MINI-LAB]

Frame Relay

Frame Relay is a very popular packet-switched layer 2 WAN protocol that is used internationally; however, it is becoming less popular due to the increase in the availability of broadband. Frame Relay can provide connection speeds from 56 Kbps up to 2 Mbps. The nature of Frame Relay is that the connection is normally shared with other companies, which brings down the cost.

Frame Relay (along with ATM and X.25) is a non-broadcast multiple access (NBMA) network, which means that multiple hosts are attached but data is transmitted only directly from one computer to another single host over a virtual circuit. NBMA networks don't support multicast or broadcast traffic without additional configuration (this is why it can cause problems for protocols such as EIGRP and OSPF).

Frame Relay service providers guarantee a minimum amount of bandwidth. If the line is quiet, customers may be able to use bandwidth belonging to other users, providing they are not using their bandwidth allocation.

Frame Relay is based on an older standard called X.25. As service provider lines advanced in quality, the error correction and subsequent overhead related to X.25 was no longer needed. Changes were made and all windowing and retransmission systems

were taken out, leaving the upper OSI layers to control these functions. This made it a lot more efficient than its predecessor, with a higher payload and less overhead per packet.

The default encapsulation type on Cisco Frame Relay interfaces is CISCO, while the alternative is IETF:

- Cisco – this is used when connecting to other Cisco devices
- IETF – this is used when connecting to non-Cisco devices

```
R1(config-if)#encapsulation frame-relay ?
```

```
    ietf Use RFC1490/RFC2427 encapsulation  
<cr>
```

Virtual Circuits

One feature of Frame Relay is the establishment of virtual circuits (VCs). This is a logical connection made between two DTE devices. Frame Relay virtual circuits come in two varieties, permanent virtual circuits (PVCs) and switched virtual circuits (SVCs). SVCs are less common, as they are temporary connections between two devices used for occasional communications. PVCs are permanent connections used when regular data transmission is taking place.

Forward Explicit Congestion Notification

Part of the Frame Relay header consists of the Forward Explicit Congestion Notification (FECN) field. When the DTE device sends frames into the network, if the DCE end detects congestion in the network it sets the FECN bit value to 1. The receiving DTE device, upon receiving the frame, can then see that the packets received from the sending device experienced congestion on the path there.

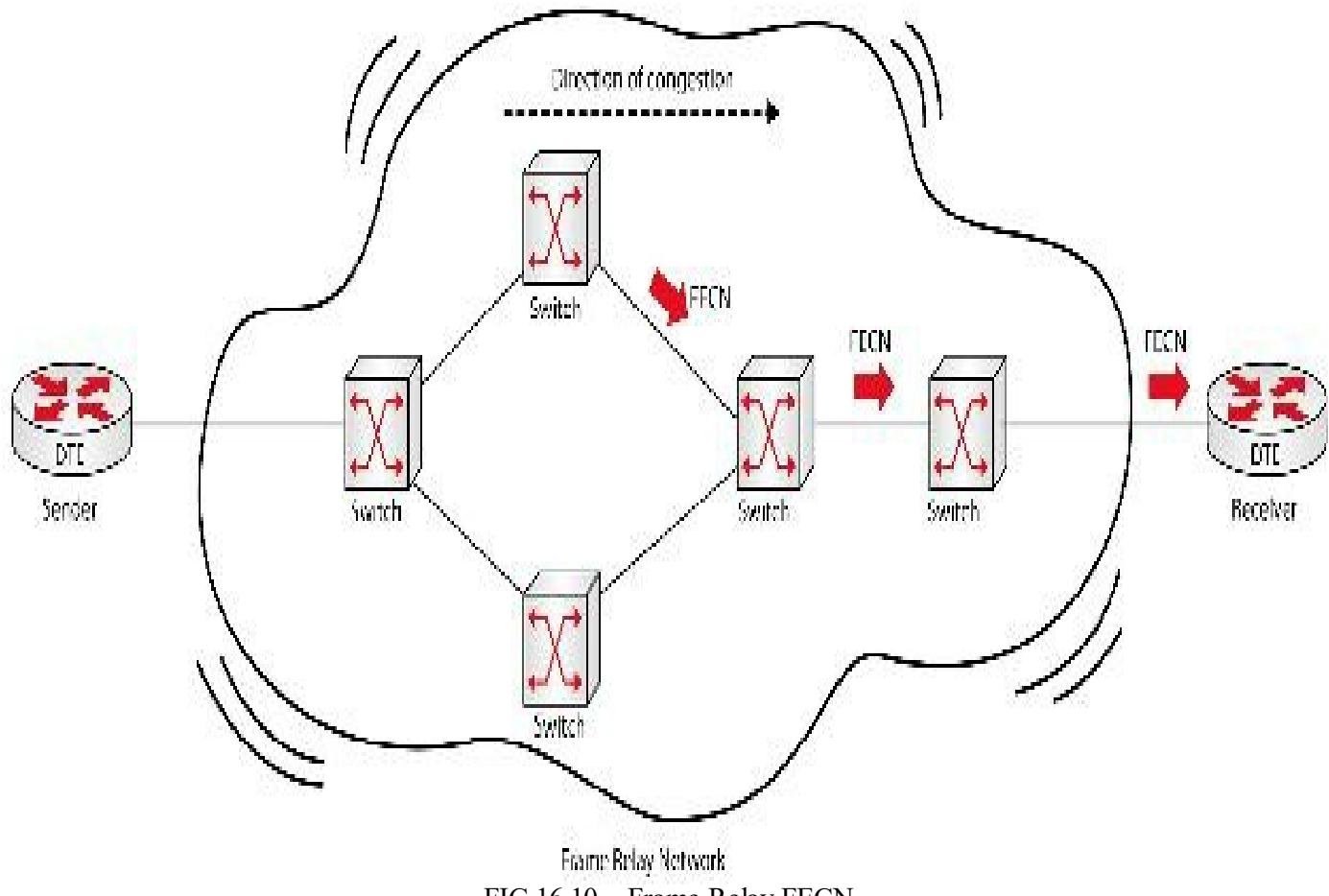


FIG 16.10 – Frame Relay FECN

Backward Explicit Congestion Notification

DCE devices will set the Backward Explicit Congestion Notification (BECN) bit to 1 for returning frames that have had their FECN bit set. This will inform the original sender that congestion was experienced on the path between the devices.

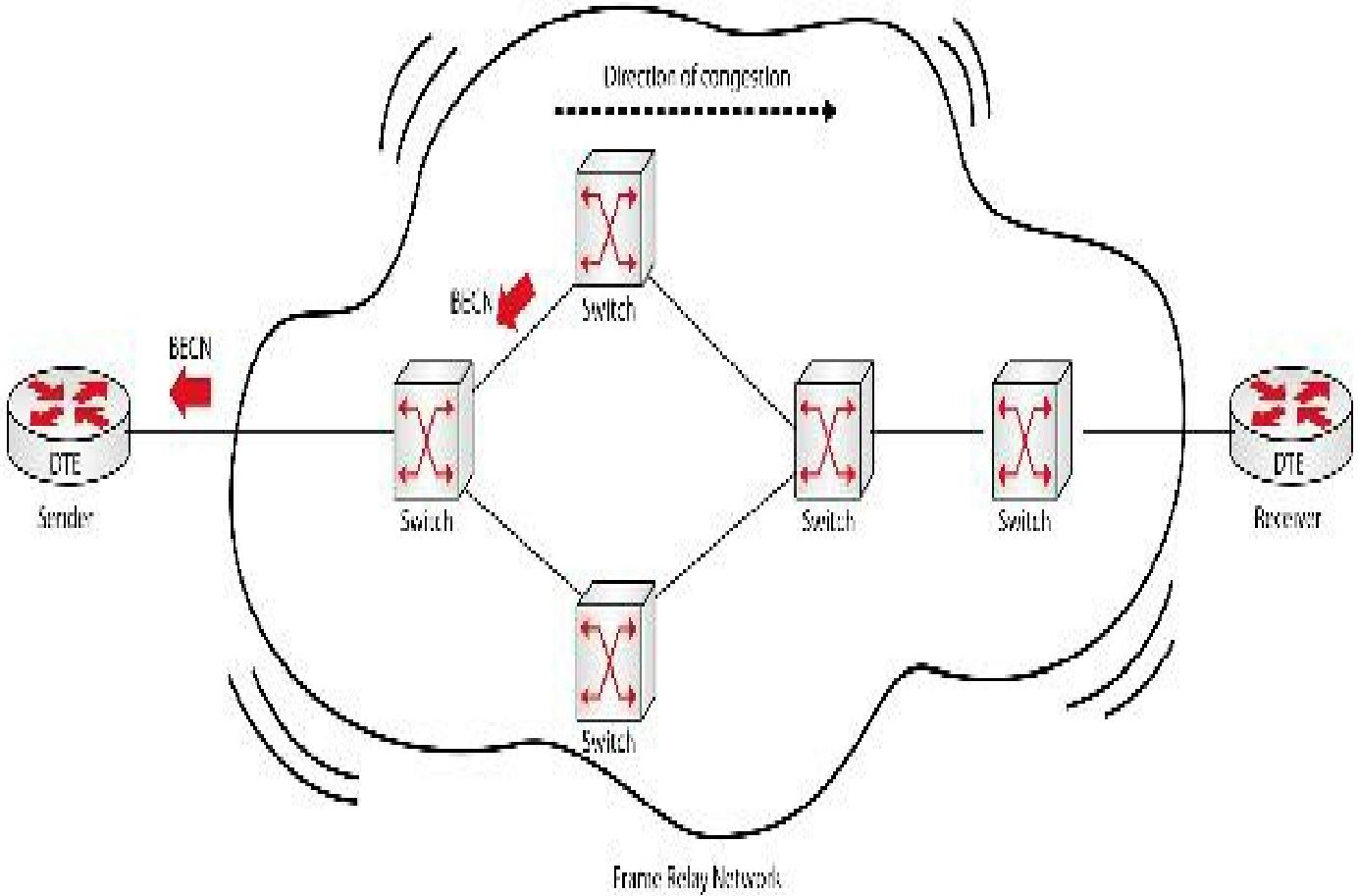


FIG 16.11 – Frame Relay BECN

The show frame-relay pvc command will show any FECN or BECN bits:

RouterA#show frame PVC

[output truncated]

DLCI = 200, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 0	output pkts 0	in bytes 0
out bytes 0	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	

Data Link Connection Identifier

DLCIs are used in Frame Relay networks for addressing between the DTE and the DCE device that belongs to the Frame Relay provider. A router at one end of the connection could have a different DLCI from the router at the other end. They are only significant to the local connection to the Frame Relay switch that belongs to the provider.

When you apply for a Frame Relay connection, you will be given a DLCI number by the Frame Relay provider. So, if you are given DLCI number 300, you will configure this on your interface. Your traffic will go into the Frame Relay switch at the provider's end and then be directed to the destination (see [Figure 16.15](#)).

You have two choices for using a DLCI with an IP address—dynamically or statically. With dynamic mapping, Frame Relay uses ARP to get the remote interface's IP address and then maps this to the DLCI used to connect to the local Frame Relay switch. Frame Relay ARP is normally referred to as Inverse ARP. Static mapping means that you have to configure the IP details yourself. We will look at both in the Configuring Frame Relay section.

The DLCI can be any number from 16 to 1007, inclusive:

```
R1(config-if)#frame-relay interface-dlci ?
```

```
<16-1007> Define a switched or locally terminated DLCI
```

Frame Relay Inverse ARP

Frame Relay Inverse ARP is a mechanism that allows the remote layer 3 addresses to be associated with the local layer 2 DLCI. When the Frame Relay circuit is initialized, the interface sends an Inverse ARP request out on each local DLCI defined. The remote router replies to the request with its IP address.

Frame Relay Inverse ARP is known as dynamic address mapping. It is enabled by default on a physical interface. In Figure 16.12 below the HQ router has two DLCIs configured. Frame Relay sends out an Inverse ARP request on each DLCI in an attempt to resolve the known DLCI (layer 2) address to the IP (layer 3) address.

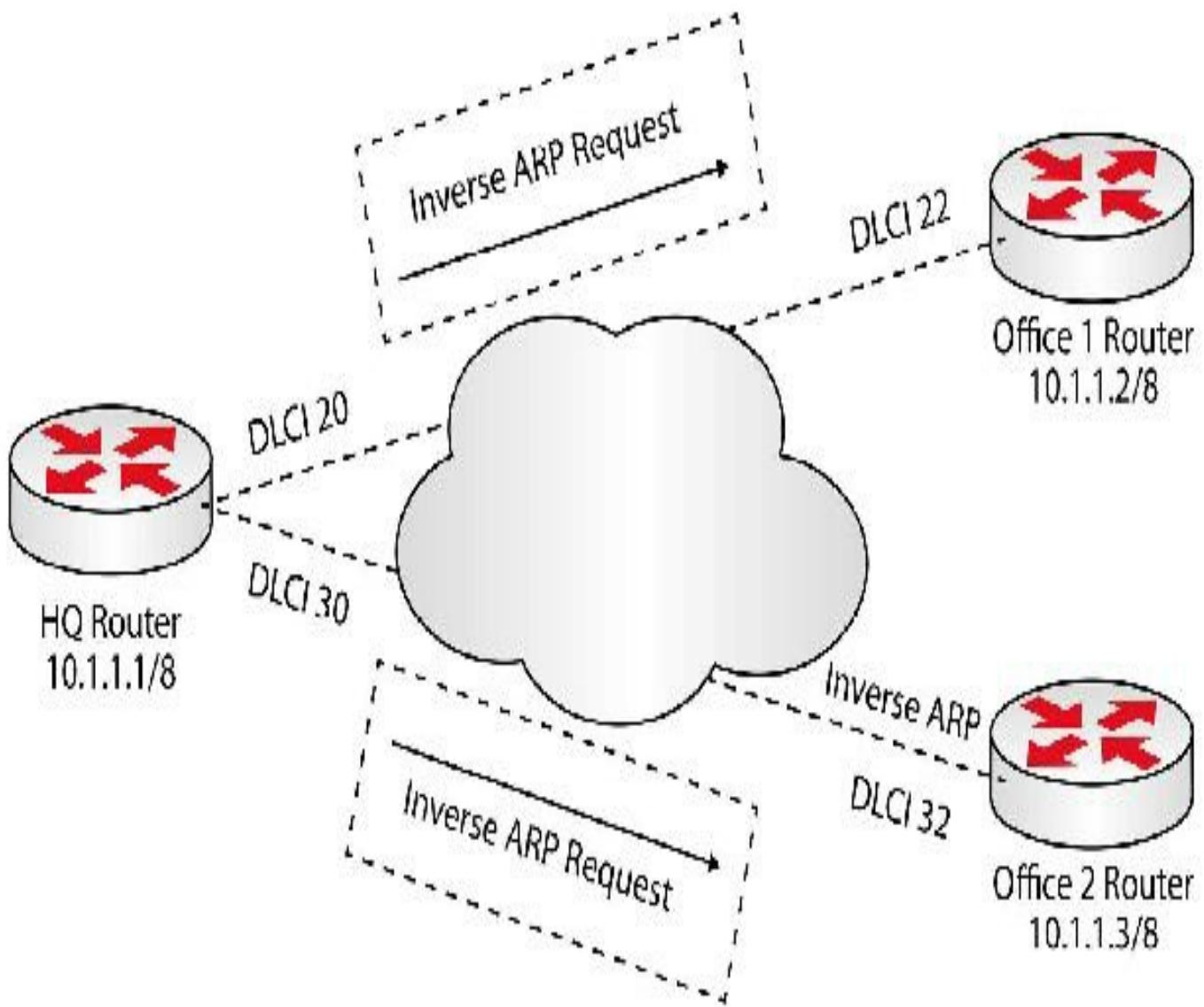


FIG 16.12 – Frame Relay Inverse ARP request

The remote routers will respond and the HQ router will map 10.1.1.2 for DLCI 20 and 10.1.1.3 for DLCI 30.

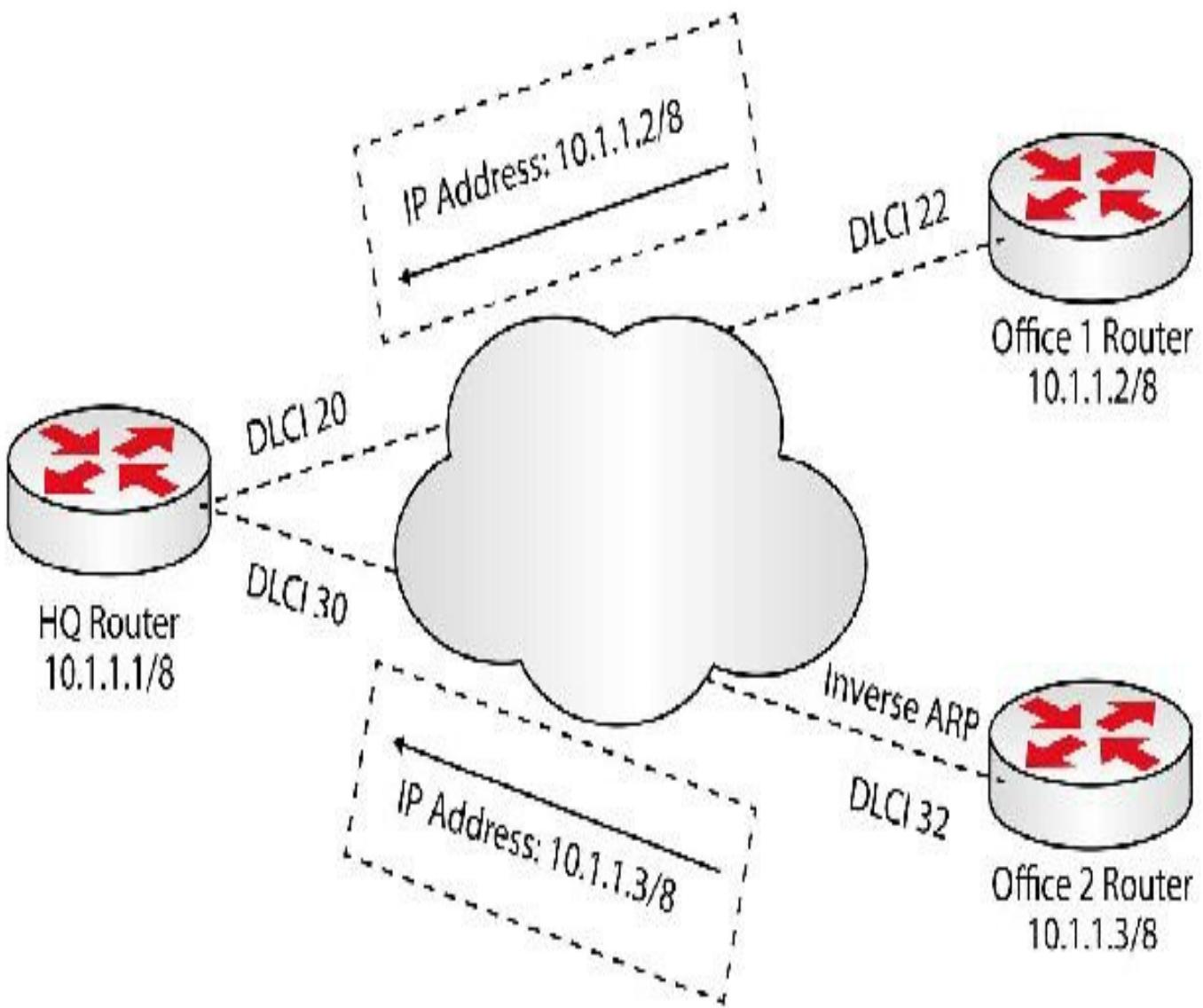


FIG 16.13 – Frame Relay Inverse ARP response

You can see whether your router learned the remote address with a static configuration or dynamically via Inverse ARP with the show frame-relay map command:

```
HQ_Router#show frame-relay map
Serial0/0 (up): ip 10.1.1.2 dlci 20(0x64,0x1840), dynamic, broadcast,
status defined, active
Serial0/1 (up): ip 10.1.1.3 dlci 30(0x64,0x1840), dynamic, broadcast,
status defined, active
```

Frame Relay Address Mapping

Frame Relay can discover the IP address at the other end of the connection using the frame-relay map command, the frame-relay interface-dlci command, or Inverse ARP. You would use static mapping if the other end of the circuit does not support Inverse ARP or does not support Inverse ARP for a specific protocol you want to use across the

Frame Relay link.

Which commands you use for the configuration depends entirely on how you have the Frame Relay set up, which remote devices you are connecting to, and whether they support Frame Relay Inverse ARP.

Putting Frame Relay encapsulation on an interface will allow it to use Frame Relay Inverse ARP to discover the IP address at the remote end. If you use Frame Relay subinterfaces, you will have to choose one of the commands below:

- On a multipoint interface, you would use the frame-relay map command.
- On a point-to-point interface, you would use the frame-relay interface dlci command.

Because Frame Relay is inherently a non-broadcast technology, some routing protocols require additional configuration parameters in order to operate correctly. One such parameter is adding the broadcast keyword to the end of frame-relay map statements. This is required to forward specific broadcasts used by protocols such as OSPF, RIP, and EIGRP.

```
Router(config)#int s0.1 multipoint  
Router(config-subif)#frame-relay map ip 172.16.1.3 120  
broadcast
```

Discard Eligibility

The Discard Eligibility (DE) bit is marked on frames that have a lower importance than other frames. The DE bit is part of the Frame Relay address header. If the DE bit is set to 1, the DCE device can discard those frames as opposed to frames without the DE bit set. This bit is usually activated when you go over your allotted bandwidth allowance.

Figure 16.14 below shows a capture of a Frame Relay packet. You can see the DE bit as well as the BECN and FECN fields.

▽ Frame Relay

- ▽ First address octet: 0x00
 - 0000 00.. = Upper DLCI: 0x00
 -0. = CR: Response
 -0 = EA: More Follows
- ▽ Second address octet: 0x01
 - 0000 = Second DLCI: 0x00
 - 0.... = FECN: False
 -0.. = BECN: False
 -0. = DE: False 
 -1 = EA: Last Octet

DLCI: 0

- ▷ Control field: U, func=UI (0x03)

FIG 16.14 – DE, BECN, and FECN fields in a Frame Relay packet capture

Frame Relay Data Rate Metrics

Each Frame Relay connection comes with service-level agreements that vary depending on the speed you have paid for. You need to be familiar with some data rate metrics terms used by Frame Relay providers. This concept is referred to as Frame Relay traffic shaping.

- Committed burst (Bc) – the number of bits committed to accept and transmit at the CIR
- Committed Information Rate (CIR) – the maximum permitted level of traffic per PVC; when this is exceeded, the DE bit is set to inform the Frame Relay switches that the frame can be dropped if the capacity of the link is reached
- Excess Burst (Be) – the number of bits to transmit after the Bc value is reached
- Max Data Rate (MaxR) – measured by CIR x (Bc+Be/Bc) in bits per second

Cyclic Redundancy Check

The CRC mechanism is a 2-bit check that compares two values in the frame to determine whether an error occurred on the frame during transit from source to destination. This process is known as error detection; error correction is taken care of by higher levels of the OSI model in the case of Frame Relay.

Local Management Interface

LMI is a standard used for signaling between the DTE and the Frame Relay switch. The connection is continually monitored in the same fashion that a keepalive is used on Serial or Ethernet connections.

There are three types of LMI connections to choose from—Cisco, Q933-A, and ANSI. On Cisco routers the default is Cisco; you would only change this if your Frame Relay provider tells you to or if you are connecting to non-Cisco equipment. The LMI frame encompasses a frame check sequence (FCS), which verifies the integrity of the data transmitted.

```
R1(config-if)#frame-relay lmi-type ?  
cisco  
ansi  
q933a
```

You can monitor the passing of the Frame Relay LMIs between the router and the frame switch. If you experience problems with your Frame Relay connection, the first thing you would look at are the LMI statistics.

LMI exchanges between the router and the Frame Relay switch can be monitored with the debug frame-relay lmi command, which we will cover in the Troubleshooting Frame Relay section.

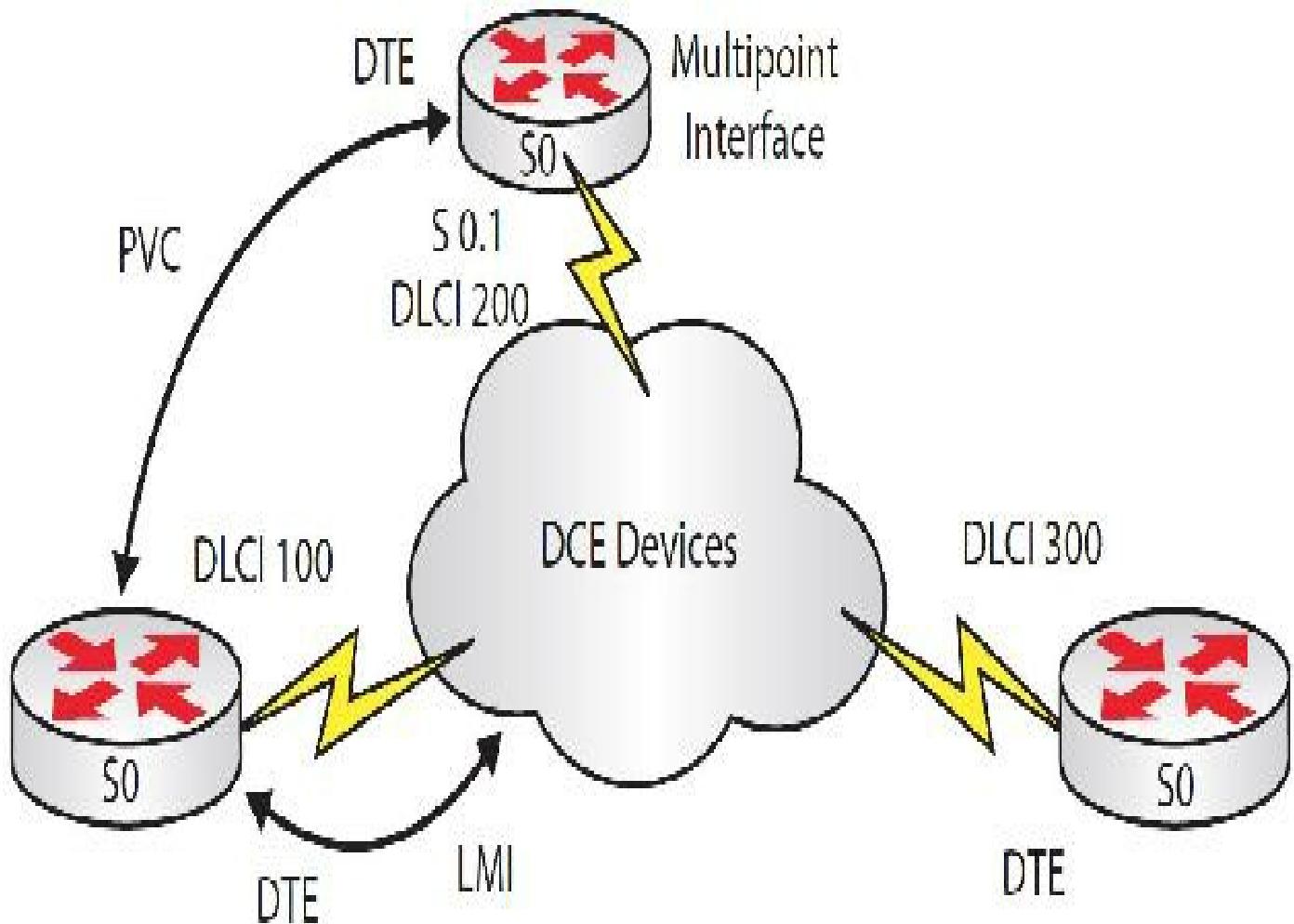


FIG 16.15 – Frame Relay PVC and LMI

Frame Relay Subinterfaces

There are many situations where you will have one router at the core of your network and three or four remote sites want to connect to the core. In this circumstance, you would have to purchase a router with a physical interface for every connection it needs to reach.

An alternative to this is to use a topology known as hub-and-spoke. This topology uses one physical interface but creates one or more subinterfaces on that one physical interface, which is a logical (i.e., exists in software) division of a physical interface. In this case the IP address will be configured on the subinterfaces, not on the physical interface.

Subinterfaces can be:

- point-to-point
- multipoint

Point-to-point subinterfaces are used when you have one-to-one connections from one

router to another. Point-to-point subinterfaces can use Inverse ARP to discover the remote address after you configure the DLCI value. Alternatively, you can statically configure the mapping.

Different point-to-point subinterfaces have to be in different subnets. This is the main reason for using them. This allows several routers to connect to the hub router when you have only one physical interface.

Multipoint interfaces are used when you have two or more other sites connecting to this one physical interface. All multipoint addresses on the same interface have to be in the same subnet. For multipoint subinterfaces, you have to statically map the remote IP address to the DLCI value.

Figure 16.16 below shows Router A as the headquarters router. There are two networks in operation, 10.10.10.0 and 20.20.20.0. The two routers on the 10.10.10.0 network must terminate on a multipoint network. The 20.20.20.0 network can terminate on a point-to-point network.

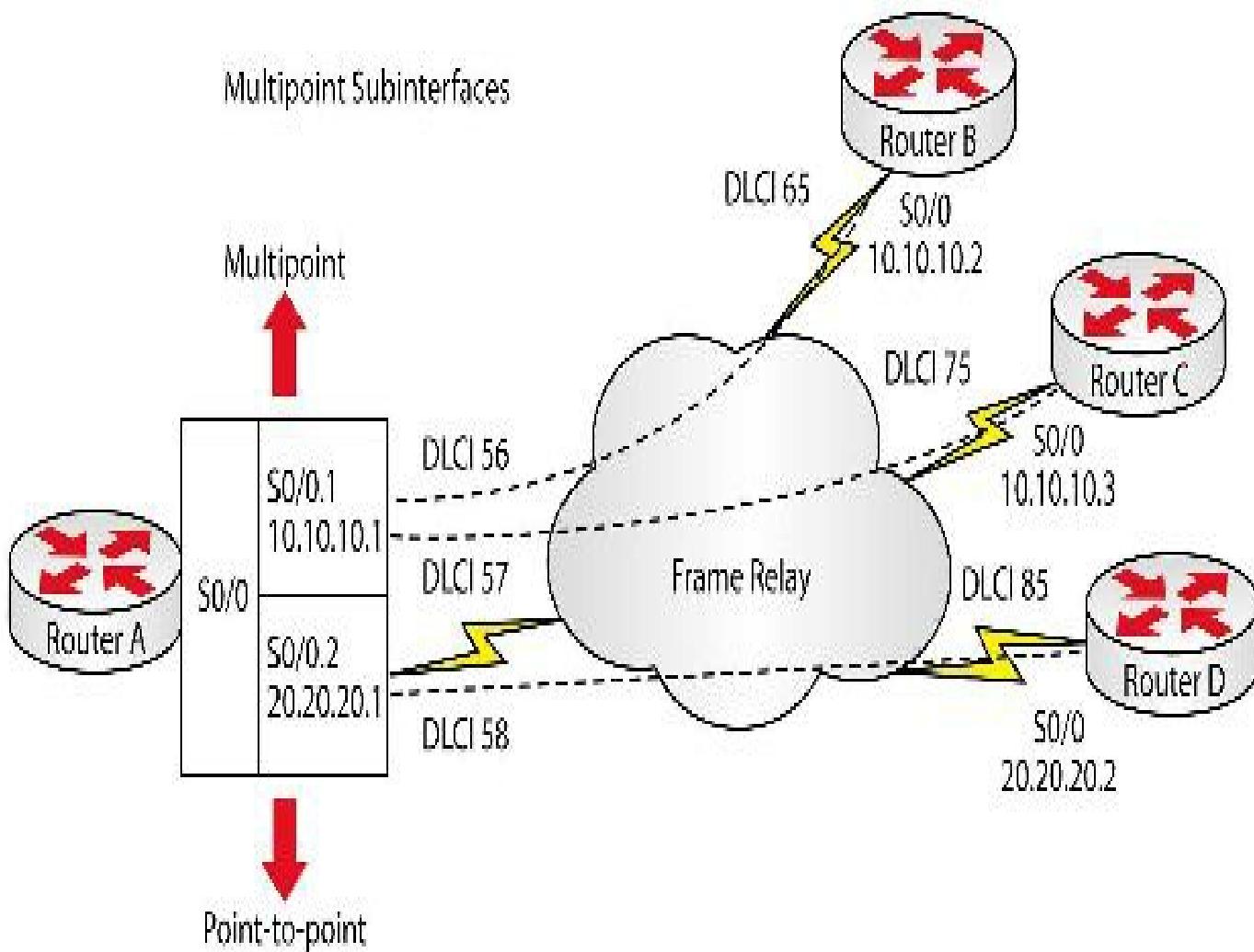


FIG 16.16 – Multipoint and point-to-point interfaces

Using subinterfaces allows you to overcome the split horizon issue, where an interface cannot advertise a route out of the same interface it was received on. Split horizon is on by default for IGRP, RIP, and EIGRP networks and prevents routing updates from being passed on multipoint interfaces.

To turn off split horizon for networks using RIP or IGRP, use the `no ip split-horizon` command. To turn it off for EIGRP networks, use the `no ip split-horizon eigrp 20`, where 20 is the ASN. Although it's not covered in the CCNA syllabus, each routing protocol requires specific commands in order to operate correctly over Frame Relay networks. As mentioned, Frame Relay has been dropped from the latest CCIE syllabus.

If you are using point-to-point interfaces, each subinterface will be in a different subnet so you will not need to address any split horizon issues. This type of connection will act in much the same way as a standard leased line. Figure 16.17 below is a sample Frame Relay network using point-to-point subinterfaces. We will configure the top router, which is the HQ router I have named Router A in the configuration.

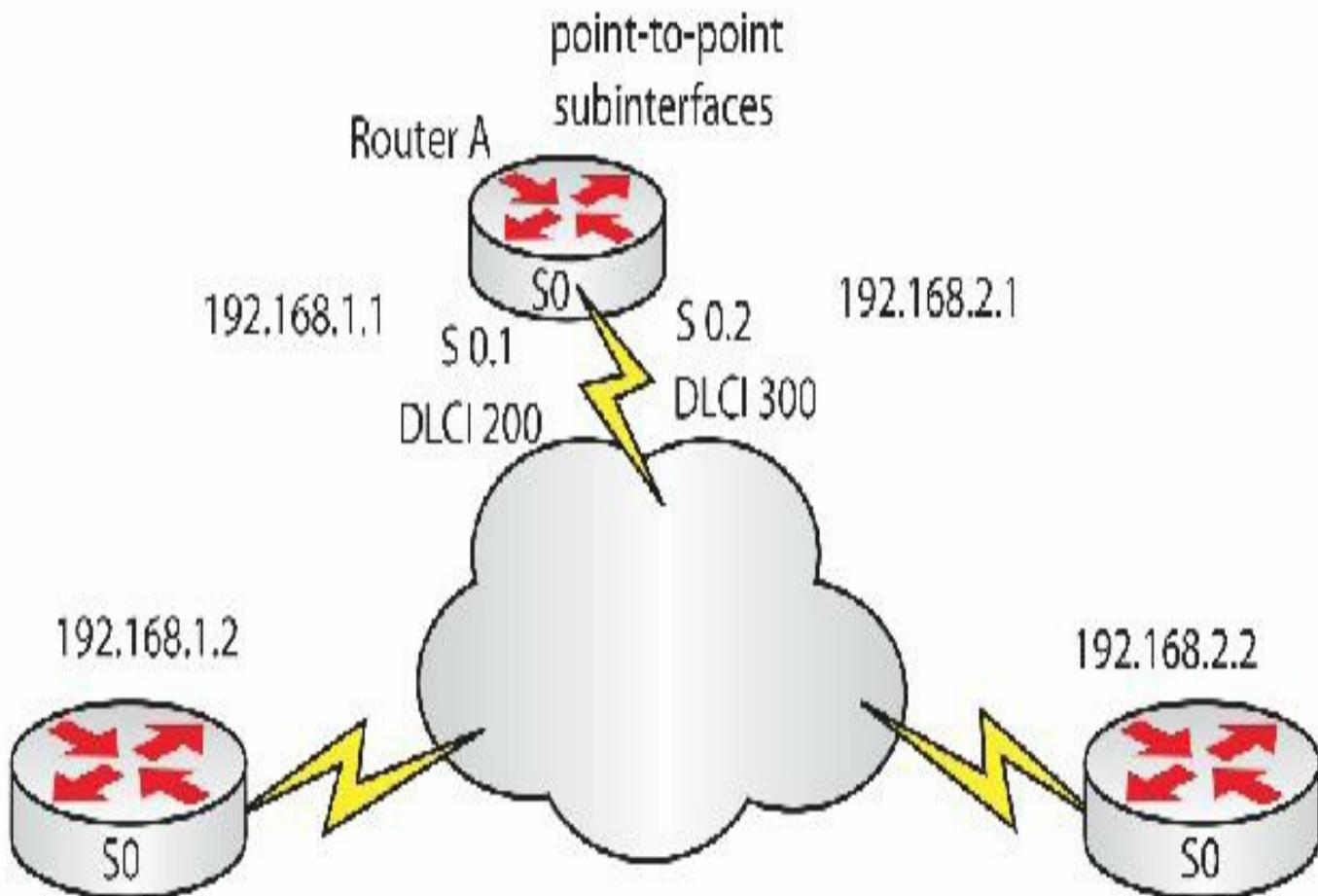


FIG 16.17 – Frame Relay network using point-to-point subinterfaces

When using subinterfaces, leave the IP address off the physical interface and add the configuration values to the subinterface. Router A is the hub router at the top.

```

RouterA#config t
RouterA(config)#interface Serial0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#exit
RouterA(config)#interface serial 0.1 point-to-point i Subinterface
RouterA(config-subif)#ip address 192.168.1.1 255.255.255.0
RouterA(config-subif)#frame-relay interface-dlci 200 i Dynamic mapping
RouterA(config-fr-dlci)#^Z i Your prompt line may look different
RouterA#config t
RouterA(config)#interface Serial0.2 point-to-point
RouterA(config-subif)#ip address 192.168.2.1 255.255.255.0
RouterA(config-subif)#frame-relay interface-dlci 300
RouterA(config-fr-dlci)#^Z

```

Note that on point-to-point subinterfaces, each address has to be in a different subnet or network.

Figure 16.18 below shows a multipoint configuration for the HQ router (RouterA) at the top.

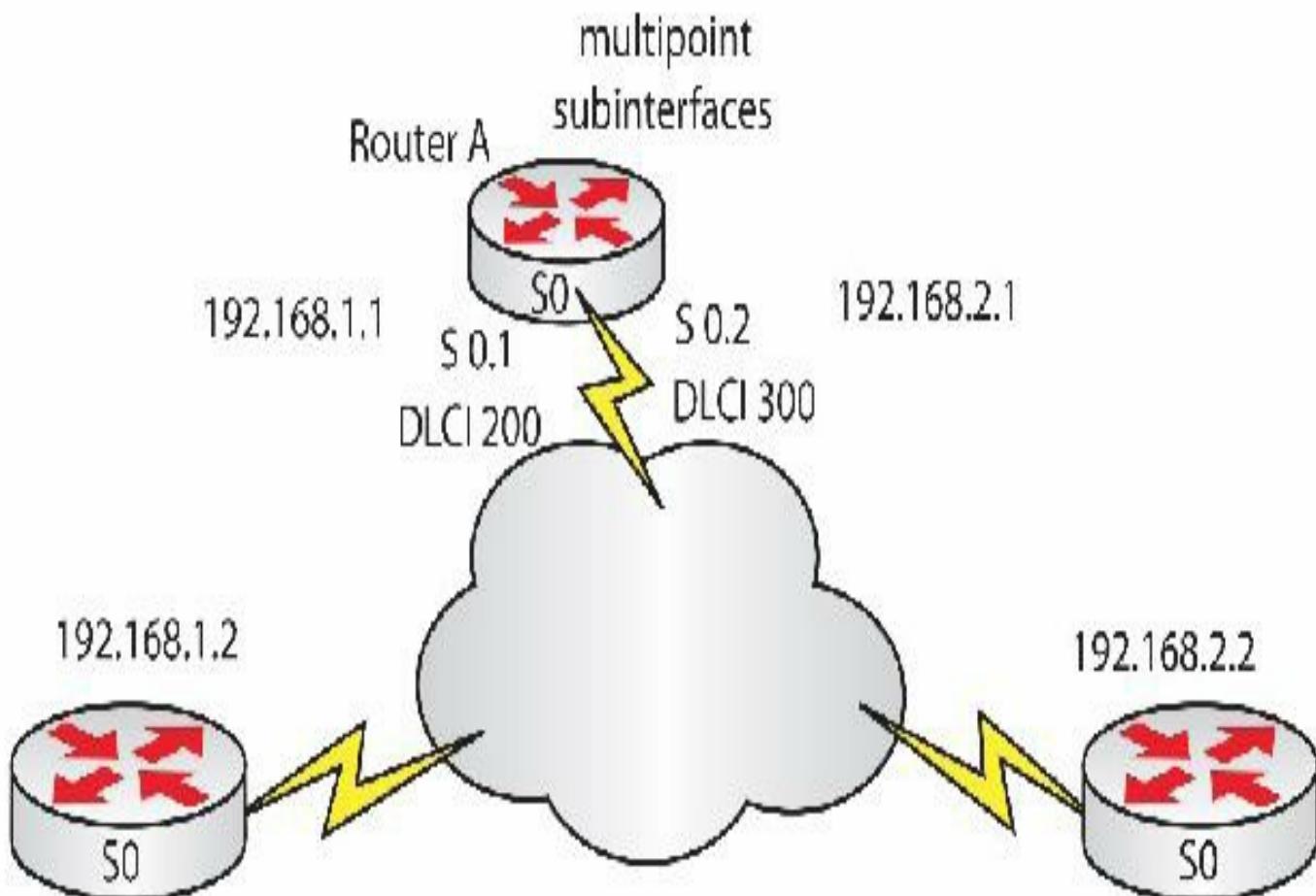


FIG 16.18 – Multipoint configuration

```
RouterA#config t
RouterA(config)#interface Serial0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#exit
RouterA(config)#interface s0.1 multipoint
RouterA(config-subif)#ip address 192.168.1.1 255.255.255.0
RouterA(config-subif)#frame-relay map ip 192.168.1.2 200 broadcast i Static mapping
RouterA(config-subif)#frame-relay map ip 192.168.1.3 300 broadcast i Static mapping
RouterA(config-if)#no ip split-horizon i Turns off split horizon (optional)
RouterA(config-subif)#^Z
```

The frame-relay map ip statement tells the router which DLCI to use to get to a particular network address. The broadcast parameter allows protocols such as OSPF and RIPv2 to multicast across the DLCI.

Configuring Frame Relay

Here are a few Frame Relay configuration commands to try on a single router. We will cover more configuration commands in the Labs section.

```
RouterA(config)#interface Serial0/0
RouterA(config-if)#encapsulation frame-relay ? i Just press Enter here
      ietf Use RFC1490/RFC2427 encapsulation i Add ietf if connecting to non-Cisco devices
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#frame-relay lmi-type ? i Leave this out if both are Cisco routers
      cisco
      ansi
      q933a

RouterA(config-if)#frame-relay lmi-type ansi
RouterA(config-if)#frame-relay map ip 192.168.1.2 200 broadcast i Static mapping
RouterA(config-if)#^Z
```

Troubleshooting Frame Relay

If your network users report problems with the Frame Relay connection, there are several commands that will give you a lot of information. Under the interface command you can check the encapsulation type as well as the LMI values. First check that your interface is up.

RouterA#show interface Serial0 i **Or sh int s0 for short**

Serial0 is up, line protocol is up

Hardware is HD64570

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255

Encapsulation FRAME-RELAY, Loopback not set

Keepalive set (10 sec)

LMI enq sent 1133, LMI stat recv 1133, LMI upd recv 0, DTE LMI down

LMI enq recv 0, LMI stat sent 0, LMI upd sent 0

LMI DLCI 1023 LMI type is CISCO frame relay DTE

Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0

You can check whether the router is seeing LMIs coming from the Frame Relay switch or sending them with the following command:

RouterA#debug frame-relay lmi

Frame Relay LMI debugging is on

Displaying all Frame Relay LMI data

00:46:38: Serial0(out): StEnq, **myseq 53, yourseen 52**, DTE up

00:46:38: datagramstart = 0xE3EEA4, datagramsize = 13

00:46:38: FR encapsulation = 0xFCF10309

00:46:38: 00 75 01 01 01 03 02 35 34

00:46:38:

00:46:38: Serial0(in): Status, myseq 53

00:46:38: RT IE 1, length 1, type 1

00:46:38: KA IE 3, length 2, yourseq 53, myseq 53

00:46:48: Serial0(out): StEnq, **myseq 54, yourseq 53**, DTE up

00:46:48: datagramstart = 0xE3EEA4, datagramsize = 13

00:46:48: FR encapsulation = 0xFCF10309

00:46:48: 00 75 01 01 01 03 02 36 35

00:46:48:

00:46:48: Serial0(in): Status, myseq 54

00:46:48: RT IE 1, length 1, type 1

00:46:48: KA IE 3, length 2, yourseq 54, myseq 54

00:46:58: Serial0(out): StEnq, **myseq 55, yourseq 54**, DTE up

00:46:58: datagramstart = 0xE3EEA4, datagramsize = 13

00:46:58: FR encapsulation = 0xFCF10309

00:46:58: 00 75 01 01 00 03 02 37 36

00:46:58:

```

00:46:58: Serial0(in): Status, myseq 55
00:46:58: RT IE 1, length 1, type 0
00:46:58: KA IE 3, length 2, yourseq 55, myseq 55
00:46:58: PVC IE 0x7 , length 0x6 , dlc1 100, status 0x2 , bw 0
RouterA#un all
All possible debugging has been turned off

```

The myseq and yourseq incrementing status 0x2 means that the PVC is operational. Out of every six LMI messages, one should report status 0x2.

Next, look at the PVC statistics. For the exam, you must know what the status outputs below mean:

- ACTIVE indicates a successful end-to-end circuit.
- INACTIVE indicates a successful connection to the Frame Relay switch (DTE to DCE) but you can't connect to the remote DTE on the other end of the PVC. This often happens due to a faulty configuration by the ISP on their switch.
- DELETED means that the router (DTE) is configured for a DLCI that the Frame Relay switch does not recognize as valid for that interface. You need to double-check your configuration and ensure that the ISP has given you the correct DLCI number.
- STATIC is rarely seen because it indicates that the keepalives have been disabled.

```
RouterA#show frame pvc
```

PVC Statistics for interface Serial0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 200, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 0	output pkts 0	in bytes 0
out bytes 0	dropped pkts 0	in FECN pkts 0
in BECN pkts 0	out FECN pkts 0	out BECN pkts 0
in DE pkts 0	out DE pkts 0	
out bcast pkts 0	out bcast bytes 0	
5 minute input rate 0 bits/sec, 0 packets/sec		
5 minute output rate 0 bits/sec, 0 packets/sec		

pvc create time 00:00:10, last time pvc status changed 00:00:10

How many LMI packets have been sent and replied to? If you are sending more packets than you have received a reply to, then you may have a problem.

RouterA#show frame lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CISCO

Invalid Unnumbered info 0	Invalid Prot Disc 0
Invalid dummy Call Ref 0	Invalid Msg Type 0
Invalid Status Message 0	Invalid Lock Shift 0
Invalid Information ID 0	Invalid Report IE Len 0
Invalid Report Request 0	Invalid Keep IE Len 0
Num Status Enq. Sent 1133	Num Status msgs Rcvd 1133 iLMIs
Num Update Status Rcvd 0	Num Status Timeouts 0

Have you mapped the correct DLCI and is it active? If the map is dynamic you can also see the DLCI. You can also check whether the encapsulation is CISCO or IETF with the command below. You can also see the status of the PVC.

RouterA#show frame-relay map

Serial0 (administratively down): ip 192.168.1.2

**dlci 200(0xC8,0x3080), static, i Address mapped statically
broadcast,
CISCO, status defined, active i CISCO encapsulation**

If you want to configure a Frame Relay lab, you will need to add a third router and configure that to be a Frame Relay switch. Configuring a Frame Relay switch is outside the CCNA syllabus, but you can find out how to do it by doing the Frame Relay labs in the Labs section.

Metro Ethernet

Metro Ethernet is a technology that uses Carrier Ethernet technology in MANs (Metropolitan Area Networks) to offer connectivity services. Common Metro Ethernet attributes include:

- Cost-effective connectivity
- Reliable connection
- Increased scalability
- Flexible bandwidth management

Metro Ethernet can connect LANs to a WAN or to the Internet. Multisite organizations can use this technology to connect their branches to an Intranet or to the Internet.

A typical Metro Ethernet system has a star network or a mesh network topology, with individual nodes connected through fiber-optic media. Using Ethernet in a MAN environment is relatively inexpensive compared with pure SHD (Smallest Hamming Distance) or MPLS (Multi-Protocol Label Switching) systems of similar bandwidth.

Ethernet on the MAN can be used with the following technologies:

- Simple Ethernet (cheapest)
- Ethernet over SHD
- Ethernet over MPLS (most reliable)
- Ethernet over DWDM (dense wavelength division multiplexing)

MPLS

Multi-Protocol Label Switching provides a mechanism for forwarding packets for any network protocol. It was originally developed in the late 1990s to provide faster packet forwarding for IP routers. Since then, its capabilities have expanded massively, for example, to support service creation (VPNs), traffic engineering, network convergence, and increased resiliency.

MPLS is now the de facto standard for many carrier and service provider networks and its deployment scenarios are continuing to grow.

Traditional IP networks are connectionless, meaning that when a packet is received, the router determines the next hop using the destination IP address on the packet with information from its own forwarding table. The router's forwarding tables contain information in the network topology obtained via an IP routing protocol, such as OSPF, IS-IS, BGP, or RIP, or static configuration, which keeps that information synchronized with changes in the network.

MPLS similarly uses IP addresses, either IPv4 or IPv6, to identify endpoints and intermediate switches and routers. This makes MPLS networks IP-compatible and easily integrated with traditional IP networks. However, unlike traditional IP, MPLS flows are connection-oriented and packets are routed along preconfigured Label Switched Paths (LSPs).

MPLS works by tagging the traffic (packets) with an identifier (a label) to distinguish the LSPs. When a packet is received, the router uses this label (and sometimes also the link over which it was received) to identify the LSP. It then looks up the LSP in its own forwarding table to determine the best link over which to forward the packet and the label to use on this next hop.

A different label is used for each hop, and it is chosen by the router or switch performing the forwarding operation. This allows the use of very fast and simple

forwarding engines, which are often implemented in hardware.

Ingress routers at the edge of the MPLS network classify each packet potentially using a range of attributes, not just the packet's destination address, to determine which LSP to use. Inside the network, the MPLS routers use only the LSP labels to forward the packet to the egress router.

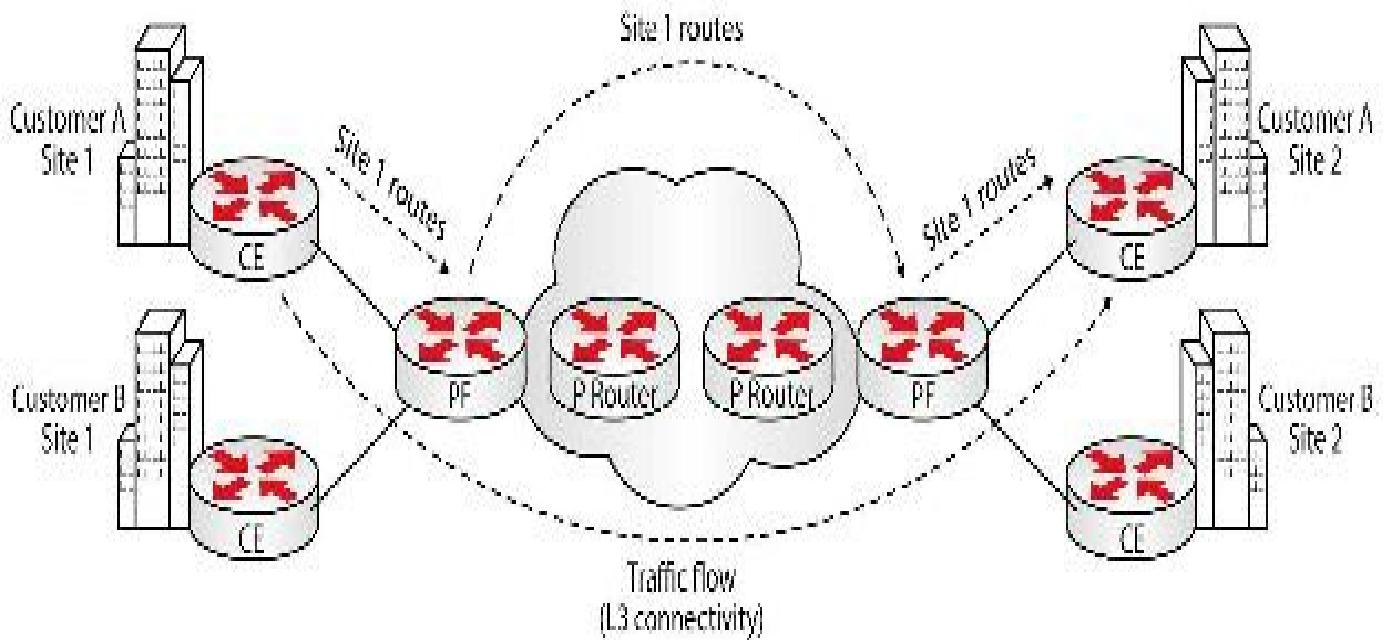


FIG 16.19 – MPLS VPN topology

In Figure 16.19 above, the top-left CE (Customer Edge) Router advertises Customer A Site 1 routes to the left-side PE (Provider Edge) Router, which injects the routes into the MPLS network, assigning each of them an MPLS label. When the packet arrives at the first P (Provider) Router, the label is switched to a locally assigned label for that specific route and then the packet is forwarded to the next P Router. At this point the procedure is exactly the same: the label is switched and then forwarded to the outbound (right-side) PE Router. The PE Router strips the label from the packet and forwards the pure IP packet to the top-right CE Router.

This way, routes from Site 1 are advertised to Site 2 and the switching in the ISP network is achieved based on the MPLS label, thus accomplishing the process faster than standard IP routing. Each customer's traffic is also tagged with specific RD (Route Distinguisher) values associated with that specific customer, so the end-to-end layer 3 path is often referred to as an MPLS VPN. Customers can even advertise prefixes from overlapping ranges to the ISP MPLS cloud. They will be treated differently, however, because of the associated RD value.

VSAT

A very small aperture terminal (VSAT) is a small telecommunication earth station that

transmits and receives real-time data via satellite. The VSAT transmits signals to orbital satellites, which relay the information to other systems in other locations around the globe.

The CPE (Customer Premises Equipment) for VSAT users is generally a box that acts as an interface between the local network and the external antenna or satellite dish transceiver. The antenna sends data to the satellite to be received in another location and the data received is sent to other locations. The satellite acts as a hub for the system and receives signals from each earth station in a star topology.

VSAT data rates are typically between 56 Kbps and a few Mbps. VSATs are generally used to transmit:

- Narrowband data, such as point of sale (credit card transactions)
- Broadband data, for the provision of Internet satellite access in remote areas requiring voice or video (or both)

VSATs are also used for maritime communications that need to be mobile, as well as on-the-move communications (using phased array antennas).

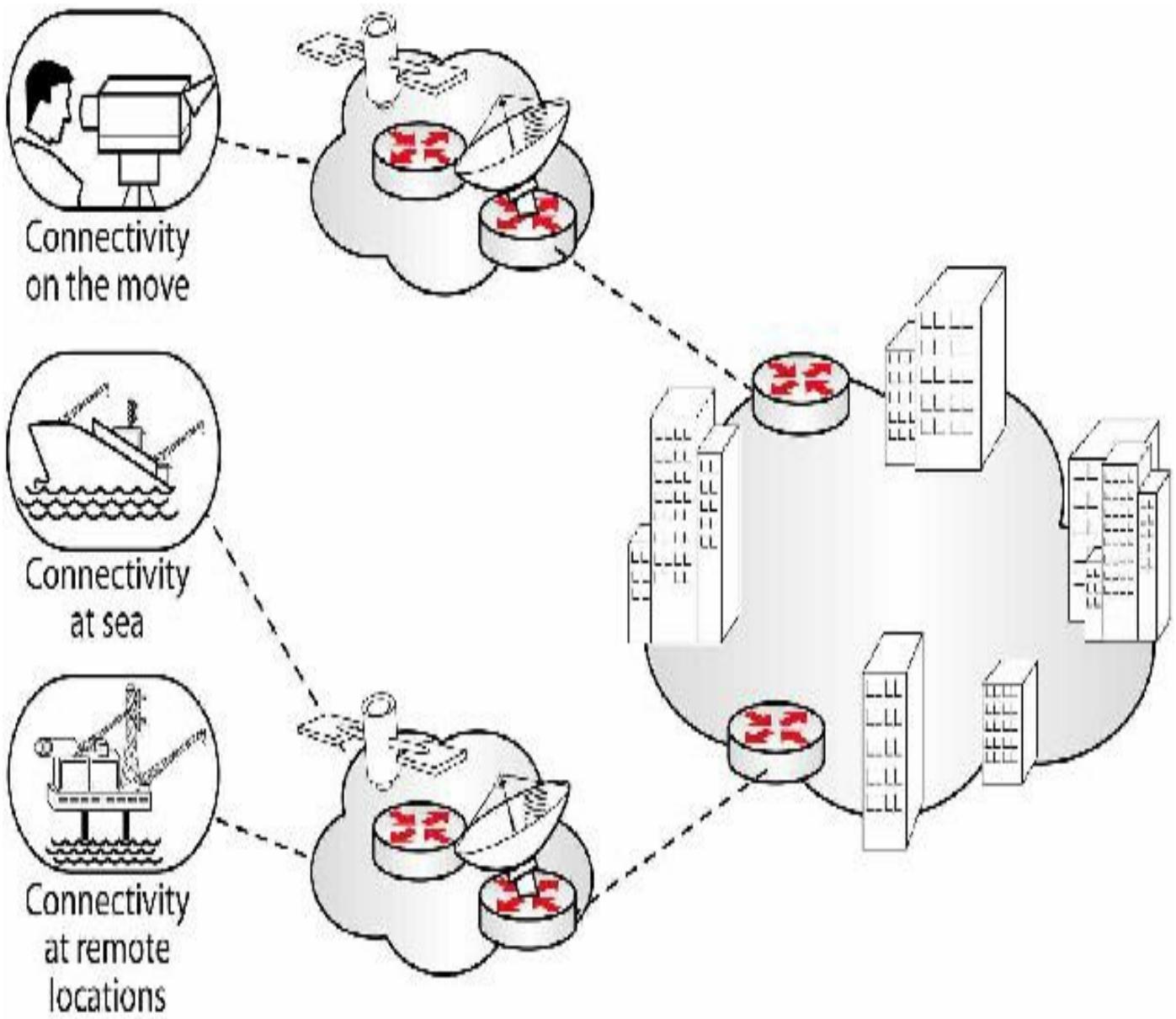


FIG 16.20 – VSAT connections

Cellular Networks

Cellular networks work via signals that carry voice, text, and digital data. These signals are transmitted via radio waves from one device to another. The data is transmitted through a global network of transmitters and receivers.

The cellular design in these kinds of networks involves dividing the overall structure into a multitude of overlapping geographical areas called cells. These cells overlap to ensure continuous transmission for roaming users. Each cell is served by a base station, which functions as a hub for the specific area. RF (radio frequency) signals are transmitted by an individual phone and received by the base station. The base station transmits the signal to another base station or directly to the receiving phone.

The entire cellular system is controlled by a mobile switching center (MSC), which

coordinates the actions of all base stations, providing overall control and acting as a switch and connection to external networks. As such, it has a variety of communication links into it that include fiber-optic links as well as some microwave links and some copper wire cables. The MSC might contain many backups and duplicate circuits to ensure that it does not fail.

When a mobile phone is turned on, it needs to be able to communicate with the cellular telecommunications network. Even if a call is not made instantly, the network needs to be able to communicate with the mobile phone to know where it is. In this way, the network can route any calls through the relevant base station, as the network would soon be overloaded if the notification of an incoming call had to be sent via several base stations.

There are a variety of tasks that need to be undertaken when a phone is turned on. This can be seen in the few seconds it takes before the phone is ready for use after turning it on. Part of this process is the software start-up for the phone, but a majority of it involves the registration process with the cellular network. There are several aspects to the registration: first, it must make contact with the base station; and second, the mobile phone has to be registered to allow it to have access to and use the network.

In order to make contact with the base station, the mobile phone uses a paging or control channel. The name of this channel and the exact way in which it works will vary from one cellular standard to the next, but it is a channel that the mobile phone can access to indicate its presence. The message sent is often called the attach message. Once this has been achieved, it is necessary for the mobile phone to register with the cellular network and to be accepted into it.

It is necessary to have a register or database of users allowed to register with a given network. With mobile phones often being able to access all the channels available in a country, methods of ensuring that the mobile phone registers with the correct network and its account is valid are required. Additionally, it is required for billing purposes. To achieve this, an entity in the network often known as the Authentication Center (AuC) is used. The network and the mobile phone communicate, with numbers giving the identity of the subscriber. Next, the user's information is checked to provide authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call, protecting users and network operators from fraud.

Once accepted into the network, two further registers are normally required—the Home Location Register (HLR) and the Visitor's Location Register (VLR). These two registers are required to keep track of the mobile phone so that the network knows where it is at any time and that calls can be routed to the correct base station or general area of the network. These registers are used to store the last known location of the

mobile phone. Thus, at registration the register is updated and then periodically the mobile phone updates its position.

When the mobile phone is switched off, it sends a detach message. This informs the network that it is switching off and enables the network to update the last known position of the mobile phone.

Based on their capabilities in terms of data transmissions, cellular networks are classified as follows:

- 2G networks (GSM and CDMA) – limited data rate
- HSPA+ (High Speed Packet Access) – download rates up to 84 Mbps and upload rates of up to 22 Mbps
- LTE (Long Term Evolution) – download rates up to 300 Mbps and upload rates up to 75 Mbps
- 4G networks – transmission rates that exceed 100 Mbps

T1/E1

T1/E1 are specifications for telecommunications standards. They work using time-division multiplexing, meaning they use multiple transmission channels to carry digital signals, with a different channel being served at a different moment of time.

The T1 standard offers a data rate of 1.544 Mbps and it contains 24 digital channels. The E1 standard is similar to the T1 standard, except that it offers a data rate around 2 Mbps and it can serve up to 32 channels.

T1 and E1 connections are specific to different geographical regions. T1 is used in North America, Japan, and South Korea, while E1 is used in Europe.

T3 and E3 standards offer higher bandwidth than the T1 and E1 standards. T3 connections offer around 44 Mbps, while E3 connections offer a total line rate of around 33 Mbps.

ISDN

ISDN (Integrated Services Digital Network) is a set of standards for digital transmission over ordinary telephone copper wire as well as over other media. ISDN is not used very much nowadays, as it has been replaced by technologies like DSL and cable modems, which will be described in the next few sections.

ISDN offers two levels of service:

- BRI (Basic Rate Interface)
- PRI (Primary Rate Interface)

Both rates include a number of B-channels and D-channels. The difference between the

two types of channels is that B-channels are used to carry data, voice, and other services, while D-channels are used for control and signaling.

BRI is the ISDN service most people use to connect to the Internet. An ISDN BRI connection supports two 64 Kbps B-channels and one 16 Kbps D-channel over a standard phone line; thus, a BRI user can have up to 128 Kbps service. BRI is often called 2B+D, referring to its two B-channels and one D-channel. The D-channel on a BRI line can even support low-speed (9.6 Kbps) X.25 data; however, this is not a very popular application in the United States.

ISDN PRI service is used primarily by large organizations with intensive communication needs. An ISDN PRI connection supports 23 64 Kbps B-channels and one 64 Kbps D-channel (or 23B+D) over a high-speed DS1 (or T-1) circuit. The European PRI configuration is slightly different, supporting 30B+D. These services are illustrated in Figure 16.21 below:

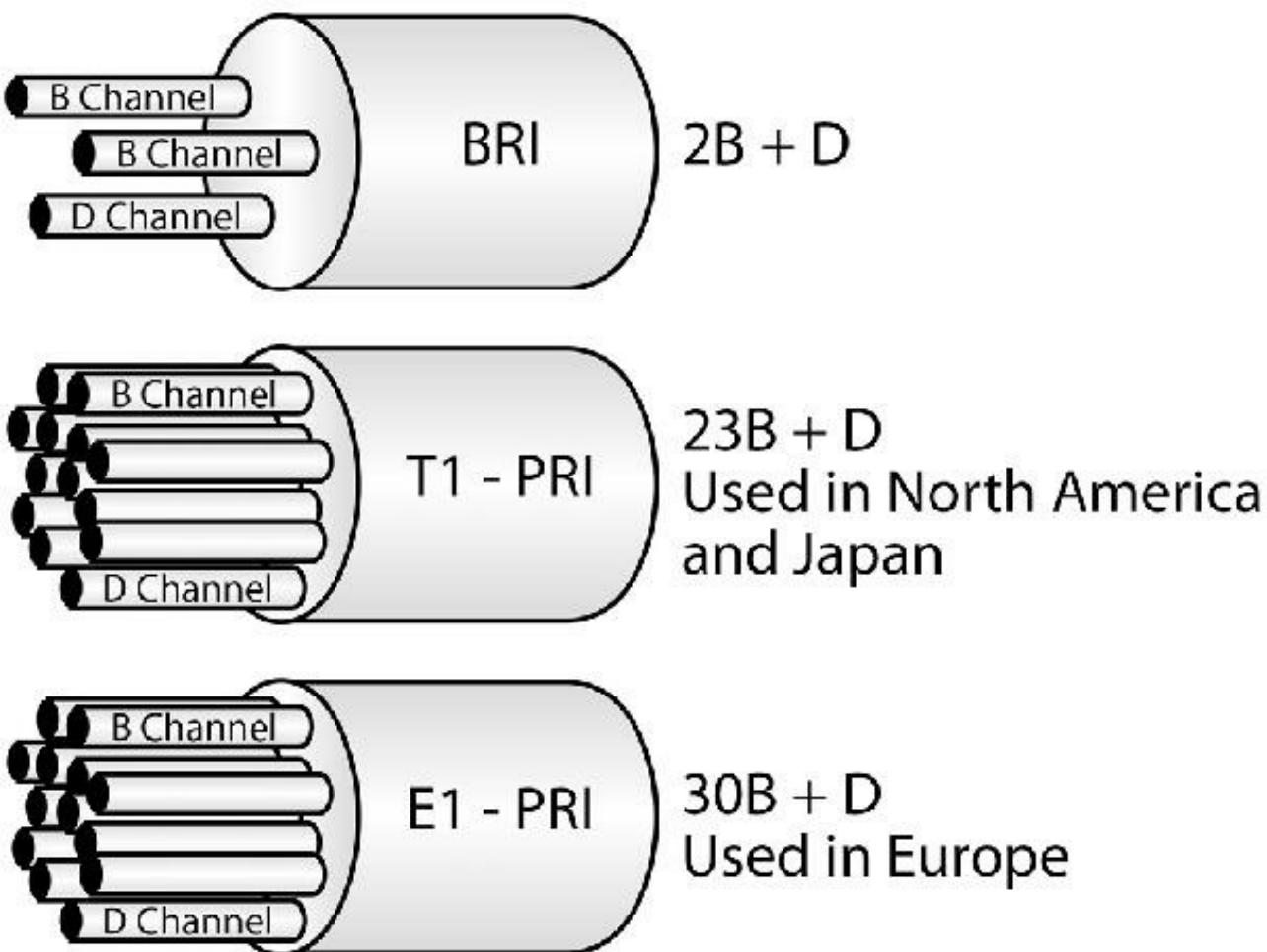


FIG 16.21 – ISDN, BRI, and PRI

ISDN in concept is the integration of both analog or voice data together with digital data over the same network. Although ISDN integrates these on a medium designed for

analog transmission, broadband ISDN (BISDN) is intended to extend the integration of both services throughout the rest of the end-to-end path using fiber-optic and radio media. Broadband ISDN encompasses Frame Relay service for high-speed data that can be sent in large bursts, the Fiber Distributed-Data Interface (FDDI), and the Synchronous Optical Network (SONET).

DSL

Digital Subscriber Line is a type of high-speed Internet access. DSL Internet access is delivered across the telephone network backbone. Not all phone companies offer DSL service in every residential area, so even if phone service is available, it does not necessarily mean that DSL will also be available.

Compared with a dial-up connection, where a modem is used to connect to the Internet over the phone line, DSL is always on. There is no need to dial in or disconnect. DSL is generally much faster than a dial-up connection, which is limited to 56 Kbps.

DSL speed and bandwidth are usually somewhat lower than cable, which is available wherever cable TV service is available; however, cable Internet access is a shared media. What this means is that if cable is available in your area, all users who are connected to the cable hub share a fixed amount of bandwidth. The more devices that are connected, the less bandwidth each user gets. With DSL, each user has a dedicated circuit and doesn't share bandwidth on that circuit with any other users.

Even though with DSL you don't have to share access with other users, the closer your home is located to a telephone company's central office switch, the better. This is a physical building where the local switching equipment is located. Distance to the switch is a determining factor in whether or not DSL service is available and what speed will be available.

DSL service introduces interference on your phone line. It is necessary to eliminate this noise using a filter supplied by your DSL provider, as demonstrated in Figure 16.22 below:

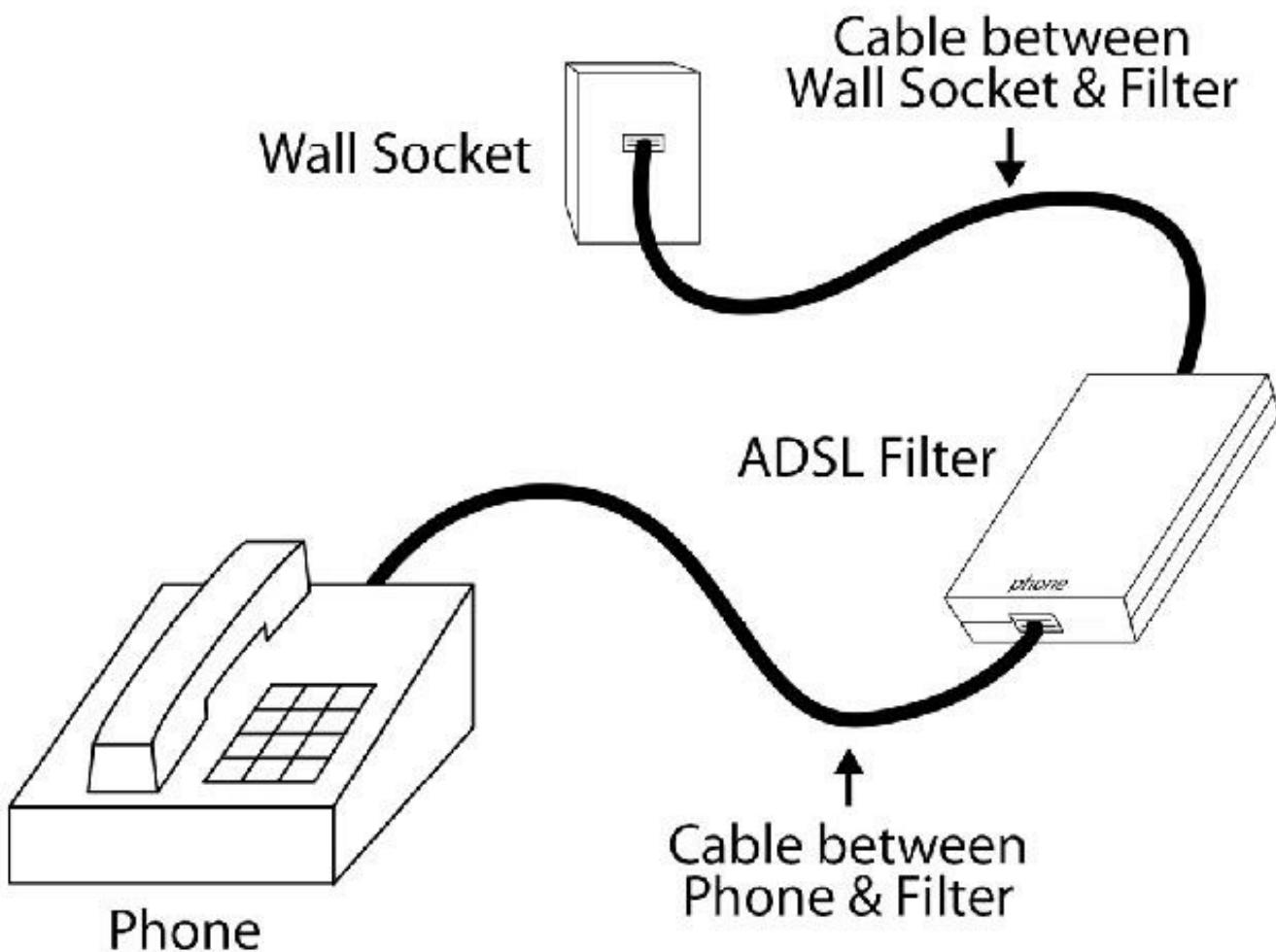


FIG 16.22 – Home connection to the DSL circuit using a filter

The most common DSL types are detailed in the following sections.

ADSL

The variation called ADSL (Asymmetric Digital Subscriber Line) is the form of DSL that is most familiar to home and small business users. ADSL is called asymmetric because most of its two-way or duplex bandwidth is devoted to the downstream direction, sending data to the user. Only a small portion of bandwidth is available for upstream or user-interaction messages. However, most Internet and especially graphics- or multimedia-intensive web data need lots of downstream bandwidth, but user requests and responses are small and require little upstream bandwidth.

Using ADSL, up to 6.1 Mbps of data can be sent downstream and up to 640 Kbps upstream. The high downstream bandwidth means that your telephone line will be able to bring motion video, audio, and 3-D images to your computer or hooked-in TV set. In addition, a small portion of the downstream bandwidth can be devoted to voice rather than data, and you can hold phone conversations without requiring a separate line.

HDSL

The earliest variation of DSL to be widely used was HDSL (High bit-rate DSL), which is used for wideband digital transmission within a corporate site and between the telephone company and a customer. The main characteristic of HDSL is that it is symmetrical: an equal amount of bandwidth is available in both directions. For this reason, the maximum data rate is lower than for ADSL. HDSL can carry as much on a single wire of twisted-pair as can be carried on a T1 line in North America or an E1 line in Europe (2,320 Kbps).

IDSL

IDSL (ISDN DSL) is somewhat of a misnomer since it's really closer to ISDN data rates and service at 128 Kbps than to the much higher rates of DSL.

RADSL

RADSL (Rate-Adaptive DSL) is an ADSL technology in which software is able to determine the rate at which signals can be transmitted on a given customer phone line and adjust the delivery rate accordingly. Westell's FlexCap2 system uses RADSL to deliver data from 640 Kbps to 2.2 Mbps downstream and from 272 Kbps to 1.088 Mbps upstream over an existing line.

VDSL

VDSL (Very high data rate DSL) provides much higher data rates over relatively short distances (between 51 and 55 Mbps over lines up to 1,000 feet or 300 meters in length).

Cable

The cable TV network can be used to connect a local computer or network to the Internet, competing directly with DSL technology.

This type of network often uses both fiber optics and coaxial cables. The connection between the cable TV company and the distribution points is made using fiber optics, while the connection between the distribution points and the users' homes is made using coaxial cables. Each distribution node typically serves between 500 and 2,000 clients. In the coaxial network, amplifiers can be used to regenerate the signal and expand the maximum lengths of the coaxial network. As a result, cable networks do not suffer from the same problems that DLS networks do in terms of electromagnetic interferences and cable length issues.

The most common system used by cable TV companies to offer Internet access is DOCSIS (Data Over Cable Services Interface Specification). The latest DOCSIS

version is 3.1 and it allows some interesting features, including channel bonding.

The coaxial cable used by the cable TV allows the transmission of several channels using different frequencies. Typically, each channel is 6 MHz wide and a whole channel is used for downstream transmissions, with a maximum transfer rate of 42.88 Mbps. For upstream transmissions, a 6.4 MHz channel is used in DOCSIS 3.0, which offers a maximum transfer rate of 30.72 Mbps.

DOCSIS 1.0 uses time-division multiple access (TDMA), while DOCSIS 2.0 and 3.0 also allow the use of code-division multiple access (CDMA). DOCSIS 3.0 also allows the use of more than one channel at the same time, a feature called channel bonding. This increases the transfer rates; for example, if four channels are used for downstream transmissions, the maximum bandwidth can reach 171.52 Mbps.

The actual transfer rate achieved using cable TV networks is related to the number of users connected to the optical node at the same time, as this system is based on the fact that not all users will be accessing the Internet at the same time.

VPN

A virtual private network (VPN) is a data network that uses a public telecommunications infrastructure, maintaining privacy through the use of a tunneling protocol and security procedures. The main purpose of a VPN is to give companies the same capabilities as private leased lines at a much lower cost through the use of the shared public infrastructure.

Some of the most important features that need to be incorporated into a virtual private network include:

- Security
- Reduced costs
- Reliability
- Scalability
- Network management
- Policy management

The main reason that companies use secure VPNs is to inexpensively transmit sensitive information over the Internet without the risk of the information being compromised. Everything that goes over a secure VPN is encrypted to such a level that even if someone captured a copy of the traffic, they could not read the content. Using a secure VPN ensures that an attacker cannot alter the contents of a company's transmissions. Secure VPNs are particularly valuable for remote access, where a user is connected to the Internet at a location not controlled by the network administrator, such

as from a hotel room, airport kiosk, or home.

Companies that use VPN technologies do so because they want to ensure that their data is moving over a set of paths that has specified properties and is controlled by one ISP or a trusted confederation of ISPs. This allows customers to use their own private IP addressing schemes and possibly handle their own routing. The customer trusts that the paths will be maintained according to an agreement and that people the customer does not trust (such as an attacker) cannot change the paths of any part of the VPN or insert traffic into the VPN. Typically, Internet Protocol Security (IPSec) is used to protect data flows over VPNs.

VPNs can broadly be classified as follows:

- Site-to-site VPNs – Permanent connections are established between different sites of the same company. Traffic is encrypted between these locations and the end-users see the other sites as directly connected. Figure 16.23 below shows a typical site-to-site VPN connection:

SITE-TO-SITE VPN

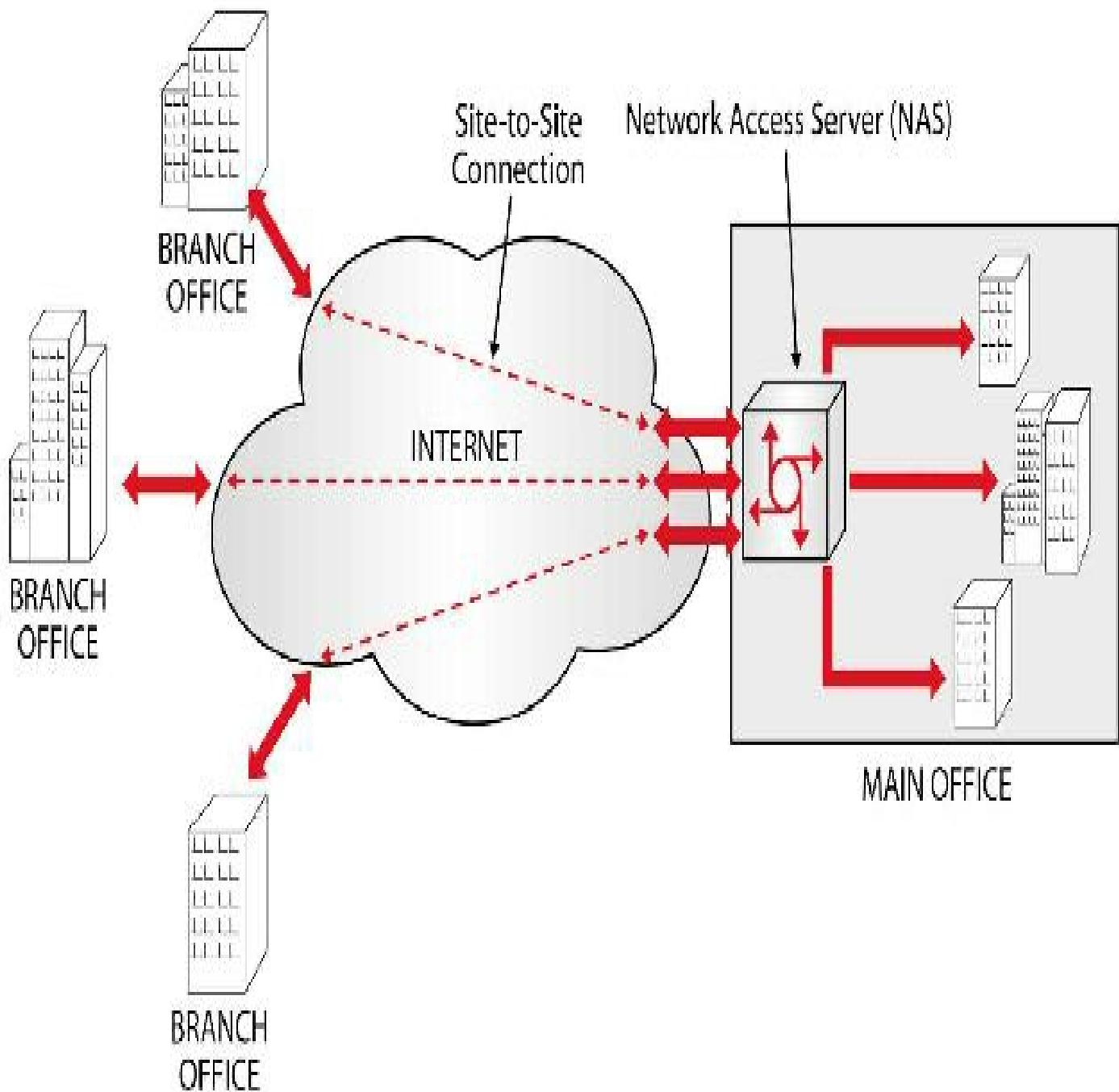


FIG 16.23 – Site-to-site VPN

- Remote access VPNs – These are dynamic secure connections between small sites or mobile workers and the company headquarters. Figure 16.24 below shows a typical remote access VPN connection:

REMOTE ACCESS VPN

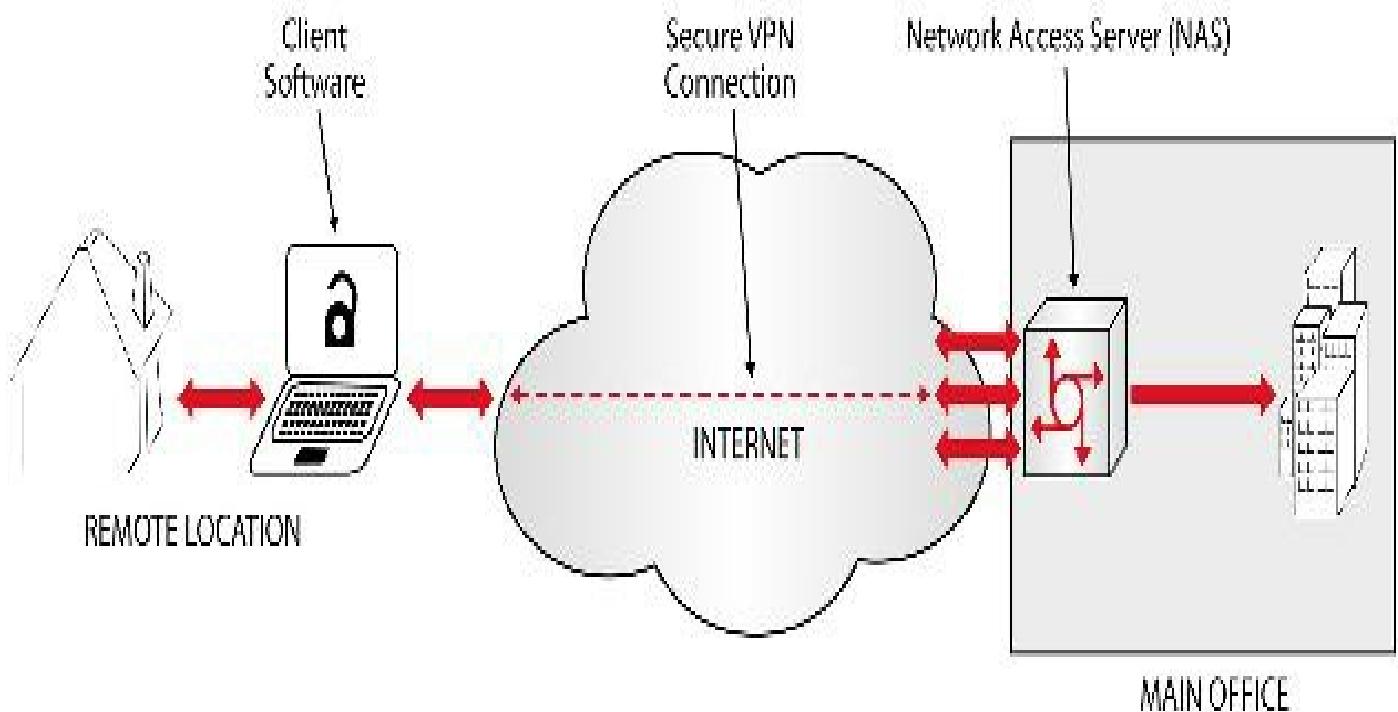


FIG 16.24 – Remote access VPN

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 16 exam.

Chapter 16 Labs

Lab 1: WAN Lab – Point-to-Point Protocol

The physical topology is shown in Figure 16.25 below:

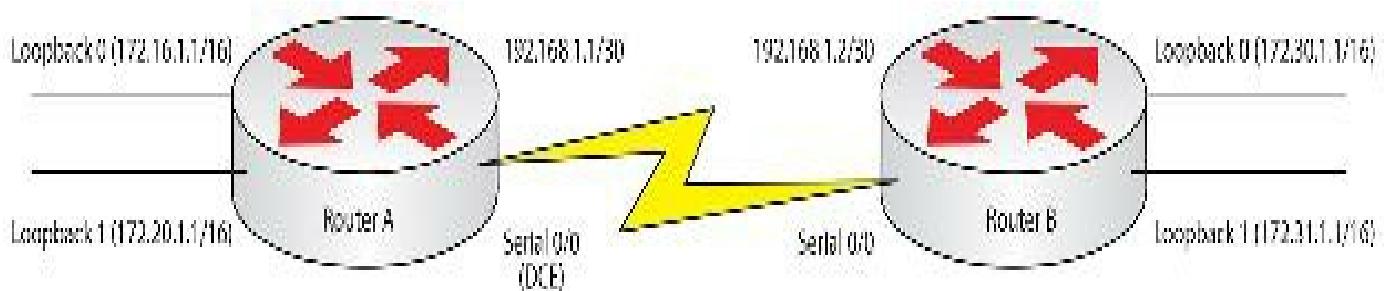


FIG 16.25 – PPP Lab

Lab Exercise

Your task is to configure the network in Figure 16.25 to allow full connectivity using PPP in a WAN. Please feel free to try the lab without following the Lab Walk-through section.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Not all networks run the default encapsulation of HDLC. Many companies use PPP, especially for ISDN connections. PPP is popular due to improved security features.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 16.25. Router A needs to have a clock rate on interface Serial 0/0: set this to 64000.
2. Set Telnet access for the router to use the local login permissions for username banbury and the password ccna.
3. Configure the enable password to be cisco.
4. Configure PPP on the Serial interface to provide connectivity to the neighbor.
5. Enable CHAP authentication.
6. Configure a default route to allow full IP connectivity.
7. Finally, test that the PPP link is up and working by sending a ping across the link.

Lab Walk-Through

1. To set the IP addresses on an interface, you will need to do the following:
Router#config t

```
Router(config)#hostname RouterA
RouterA(config)#
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip address 192.168.1.1 255.255.255.252
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shutdown
RouterA(config-if)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
RouterA(config-if)#interface Loopback1
RouterA(config-if)#ip address 172.20.1.1 255.255.0.0
RouterA(config-if)#^Z
RouterA#
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.252
RouterB(config-if)#no shutdown
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
RouterB(config-if)#interface Loopback1
RouterB(config-if)#ip address 172.31.1.1 255.255.0.0
RouterB(config-if)#^Z
RouterB#
```

To set the clock rate on a Serial interface (DCE connection only), you need to use the `clock rate #` command on the Serial interface, where # indicates the speed:

```
RouterA(config-if)#clock rate 64000
```

Ping across the Serial link now.

2. To set PPP CHAP authentication, you need to set a username and password on each router. The username must match the hostname of the calling router exactly:

```
RouterA(config)#username RouterB password cisco
```

Router B:

```
RouterB(config)#username RouterA password cisco
```

3. To set the enable password, do the following:

```
RouterA(config)#enable secret cisco
```

Router B:

```
RouterB(config)#enable secret cisco
```

4. You now need to configure PPP as the WAN link for this lab. To enable PPP, you will need to do the following:

```
RouterA(config)#interface Serial0/0
```

```
RouterA(config-if)#encapsulation ppp
```

```
RouterA(config-if)#ppp authentication chap i Use CHAP to authenticate
```

Router B:

```
RouterB(config)#interface Serial0/0
```

```
RouterB(config-if)#encapsulation ppp
```

```
RouterB(config-if)#ppp authentication chap
```

5. To configure a default route, there is one simple step (in configuration mode):

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

Router B:

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0
```

6. To test the PPP connection, you will need to first check that the link is up. To do this, use the show interface command:

```
RouterA#show interface Serial0/0
```

```
Serial0/0 is up, line protocol is up
```

```
Hardware is HD64570
```

```
Internet address is 192.168.1.1/30
```

```
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  
reliability 255/255, txload 1/255, rxload 1/255
```

```
Encapsulation PPP, Loopback not set
```

[output truncated]

Router B:

```
RouterB#show interface Serial0/0
```

```
Serial0/0 is up, line protocol is up
```

Hardware is HD64570

Internet address is 192.168.1.2/30

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255

Encapsulation PPP, Loopback not set

LCP Open

Open: IPCP, CDPCP

Make sure that Serial 0/0 is up and the line protocol is up.

Next, ping the neighbor Serial interface; this will test whether the link is up:

RouterA#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms

Router B:

RouterB#ping 192.168.1.1

If everything is OK, you will receive five replies and have a 100 percent success rate.

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/32 ms

7. To test the PPP negotiation, you can shut the Serial interface and then no shut it with the debug ppp authentication and debug ppp negotiation commands. You can see the CHAP challenge taking place and the line coming up as you read the debug.

RouterB#config t

RouterB(config)#do debug ppp authentication

RouterB(config)#do debug ppp negotiation

RouterB(config)#interface s0/0

RouterB(config-if)#shut

RouterB(config-if)#

01:41:37: %LINK-5-CHANGED: Interface Serial0, changed state to administratively down

RouterB(config-if)#

01:41:37: Se0 IPCP: Remove link info for cef entry 192.168.1.1
01:41:37: Se0 IPCP: State is Closed
01:41:37: Se0 CDPCP: State is Closed
01:41:37: Se0 PPP: Phase is TERMINATING
01:41:37: Se0 LCP: State is Closed
01:41:37: Se0 PPP: Phase is DOWN
01:41:37: Se0 IPCP: Remove route to 192.168.1.1
RouterB(config-if)#
01:41:38: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
RouterB(config-if)#no shut
RouterB(config-if)#^Z
01:41:46: %SYS-5-CONFIG_I: Configured from console by console
01:41:46: %LINK-3-UPDOWN: Interface Serial0, changed state to up
01:41:46: Se0 PPP: Treating connection as a dedicated line
01:41:46: Se0 PPP: Phase is ESTABLISHING, Active Open
01:41:46: Se0 PPP: Authorization NOT required
01:41:46: Se0 LCP: O CONFREQ [Closed] id 184 len 15
01:41:46: Se0 LCP: AuthProto CHAP (0x0305C22305)
01:41:46: Se0 LCP: MagicNumber 0x093B9E12 (0x0506093B9E12)
01:41:48: Se0 LCP: State is Open
01:41:48: Se0 PPP: Phase is AUTHENTICATING, by both
01:41:48: Se0 CHAP: O CHALLENGE id 180 len 28 from RouterB
01:41:48: Se0 CHAP: I CHALLENGE id 180 len 28 from RouterA
01:41:48: Se0 PPP: Sent CHAP SENDAUTH Request to AAA
01:41:48: Se0 CHAP: I RESPONSE id 180 len 28 from RouterA
01:41:48: Se0 PPP: Phase is FORWARDING, Attempting Forward
01:41:48: Se0 PPP: Phase is AUTHENTICATING, Unauthenticated User
01:41:48: Se0 PPP: Sent CHAP LOGIN Request to AAA
01:41:48: Se0 PPP: Received SENDAUTH Response from AAA = PASS
01:41:48: Se0 CHAP: O RESPONSE id 180 len 28 from RouterB
01:41:48: Se0 PPP: Received LOGIN Response from AAA = PASS
01:41:48: Se0 PPP: Phase is FORWARDING, Attempting Forward
01:41:48: Se0 PPP: Phase is AUTHENTICATING, Authenticated User
01:41:48: Se0 CHAP: O SUCCESS id 180 len 4
01:41:48: Se0 CHAP: I SUCCESS id 180 len 4
01:41:48: Se0 PPP: Phase is UP
01:41:48: Se0 IPCP: O CONFREQ [Closed] id 2 len 10

```
01:41:48: Se0 IPCP: Address 192.168.1.2 (0x0306C0A80102)
01:41:48: Se0 IPCP: I CONFACK [ACKsent] id 2 len 10
01:41:48: Se0 IPCP: Address 192.168.1.2 (0x0306C0A80102)
01:41:48: Se0 IPCP: State is Open
01:41:48: Se0 IPCP: Install route to 192.168.1.1
01:41:48: Se0 IPCP: Add link info for cef entry 192.168.1.1
01:41:49: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,
changed state to up
RouterB#un all
All possible debugging has been turned off
```

Show Runs

```
RouterA#show run
```

```
Building configuration...
```

```
Current configuration : 739 bytes
```

```
!
```

```
version 15.1
```

```
!
```

```
hostname RouterA
```

```
!
```

```
enable secret 5 $1$jjQo$YJXxLo.EZm9t6Sq4UYeCv0
```

```
!
```

```
username RouterB password cisco
```

```
!
```

```
ip subnet-zero
```

```
!
```

```
interface Loopback0
```

```
ip address 172.16.1.1 255.255.0.0
```

```
!
```

```
interface Loopback1
```

```
ip address 172.20.1.1 255.255.0.0
```

```
!
```

```
interface Serial0/0
```

```
ip address 192.168.1.1 255.255.255.252
```

```
encapsulation ppp
```

```
ppp authentication chap
```

```
clockrate 64000
```

```
!
```

```
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0/0
!
end
```

RouterA#

```
RouterB#show run
Building configuration...
Current configuration : 721 bytes
!
version 15.1
!
hostname RouterB
!
enable secret 5 $1$HrXN$ThplDHEZdnCbbeA/Ie67E1
!
username RouterA password cisco
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Loopback1
ip address 172.31.1.1 255.255.0.0
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
encapsulation ppp
ppp authentication chap
!
ip classless
```

```
ip route 0.0.0.0 0.0.0.0 Serial0/0
no ip http server
!
end
```

RouterB#

Lab 2: Basic Frame Relay

Lab Exercise

Your task is to configure the network in Figure 16.26 below to allow full connectivity using Frame Relay. In order to complete the lab, you will have to use three routers—two as hosts and one as the Frame Relay router/switch. Configuring a Frame Relay switch can be a little tricky, but you will never be expected to do this for the CCNA exam. It is purely for use in a lab environment.

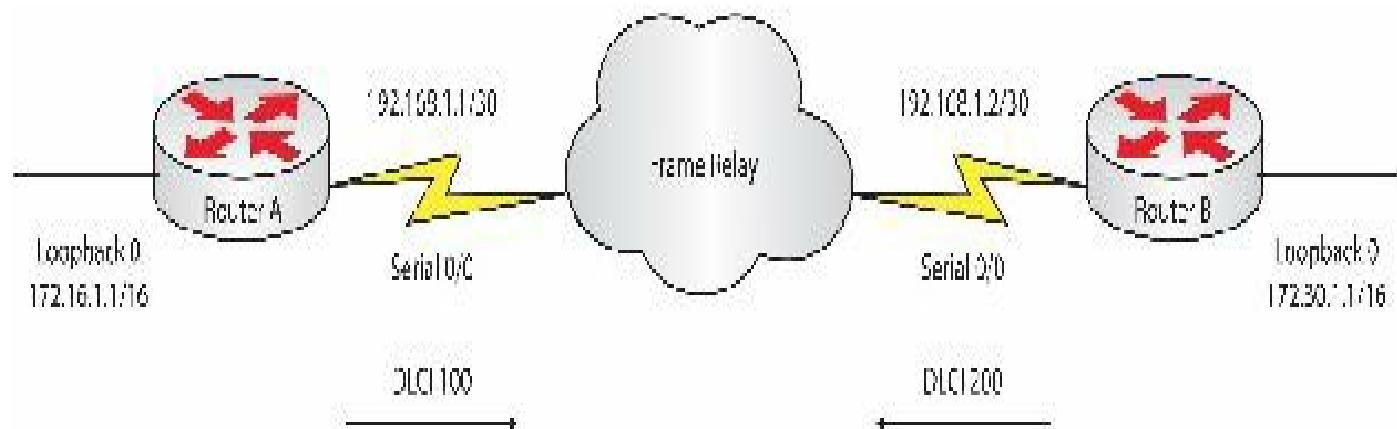


FIG 16.26 – Basic Frame Relay

If you do not have a third router to use as a Frame Relay switch, then just practice inputting the commands on a router without being able to test it. **The DCE cables are always plugged into the Frame Relay switch.**

One last thing to bear in mind is that you may have plugged your cables into different interfaces than the one shown in Figure 16.26. You need to draw out your own lab diagram marking your own ports.

Text in Courier New font indicates commands that can be entered on the router.

Purpose

Frame Relay is not widely used all around the world and is still an area but you could be tested on in the CCNA exam.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 16.26. The Frame Relay

switch will have the DCE interfaces, so you will need to have the clock rate command added.

2. Configure Frame Relay on the Serial interfaces of Routers A and B.
3. Configure the Frame Relay switch.
4. Configure RIP on Routers A and B to allow for end-to-end connectivity.
5. Test the link by pinging across it.

Lab Walk-through

1. Set the IP address and encapsulation type on the routers:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#ip add 192.168.1.1 255.255.255.252
RouterA(config-if)#no shut
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#frame-relay interface-dlci 100
RouterA(config-if)#interface Loopback 0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
```

*NOTE: Your router may show a different prompt at the DLCI input.

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#ip add 192.168.1.2 255.255.255.252
RouterB(config-if)#no shut
RouterB(config-if)#encapsulation frame-relay
RouterB(config-if)#frame-relay interface-dlci 200
RouterB(config-if)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
```

2. You need to configure RIP to allow all networks to see each other. You can choose another protocol if you want to.

```
RouterA#config t
RouterA(config)#router rip
RouterA(config-router)#version 2
RouterA(config-router)#network 192.168.1.0
```

```
RouterA(config-router)#network 172.16.0.0
```

Router B:

```
RouterB#config t  
RouterB(config)#router rip  
RouterB(config-router)#version 2  
RouterB(config-router)#network 192.168.1.0  
RouterB(config-router)#network 172.30.0.0
```

3. You need to configure the Frame Relay switch. Just copy and paste the configuration and change the interface if you are not using the same ones shown here. Make sure that you get the DLCI interfaces on the correct side, facing the correct router.

```
Router#config t  
Router(config)#hostname FrameSwitch  
FrameSwitch(config)#frame-relay switching i Makes the router a Frame Relay switch  
FrameSwitch(config)#interface Serial0  
FrameSwitch(config-if)#clock rate 64000  
FrameSwitch(config-if)#encapsulation frame-relay  
FrameSwitch(config-if)#frame-relay intf-type dce i Interface is the DCE  
FrameSwitch(config-if)#frame-relay route 100 interface Serial1 200 i Sends traffic from dlc1 100 out of interface Serial1 as dlc1 200  
FrameSwitch(config-if)#no shut  
FrameSwitch(config-if)#interface Serial1  
FrameSwitch(config-if)#clock rate 64000  
FrameSwitch(config-if)#encapsulation frame-relay  
FrameSwitch(config-if)#frame-relay intf-type dce  
FrameSwitch(config-if)#frame-relay route 200 interface Serial0 100  
FrameSwitch(config-if)#no shut
```

4. Ping from Router A to the Loopback on Router B.

```
RouterA#ping 172.30.1.1  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms  
RouterA#
```

If you cannot ping, then go to the relevant troubleshooting section in Chapter 17.
Check the following:

- The interfaces are up/up
- Clock rate is configured on the Frame Relay switch DCE interfaces
- The encapsulation is set to Frame Relay on all interfaces
- The interfaces have been no shut
- You have put the correct Frame Relay route statements on the correct interfaces
- You have configured the correct RIP routes

5. Check for Frame Relay connectivity.

RouterA#show frame-relay map

Serial0/0 (up): ip 192.168.1.2 dlci 100(0x64,0x1840), dynamic,
broadcast, status defined, active

RouterA#show frame-relay pvc

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
Serial0

input pkts 20 output pkts 20 in bytes 1838
out bytes 1898 dropped pkts 0 in FECN pkts 0
in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
in DE pkts 0 out DE pkts 0
out bcast pkts 10 out bcast bytes 858

pvc create time 00:07:57, last time pvc status changed 00:03:23

RouterA#show frame-relay lmi

LMI Statistics for interface Serial0/0 (Frame Relay DTE) LMI TYPE = CISCO

Invalid Unnumbered info 0	Invalid Prot Disc 0
Invalid dummy Call Ref 0	Invalid Msg Type 0
Invalid Status Message 0	Invalid Lock Shift 0
Invalid Information ID 0	Invalid Report IE Len 0
Invalid Report Request 0	Invalid Keep IE Len 0
Num Status Enq. Sent 47	Num Status msgs Rcvd 48
Num Update Status Rcvd 0	Num Status Timeouts 0

RouterA#debug frame-relay lmi

Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
RouterA#

```
00:46:38: Serial0/0(out): StEnq, myseq 53, yourseen 52, DTE up
00:46:38: datagramstart = 0xE3EEA4, datagramsize = 13
00:46:38: FR encapsulation = 0xFCF10309
00:46:38: 00 75 01 01 01 03 02 35 34
00:46:38:
00:46:38: Serial0/0(in): Status, myseq 53
00:46:38: RT IE 1, length 1, type 1
00:46:38: KA IE 3, length 2, yourseq 53, myseq 53
00:46:48: Serial0/0(out): StEnq, myseq 54, yourseen 53, DTE up
00:46:48: datagramstart = 0xE3EEA4, datagramsize = 13
00:46:48: FR encapsulation = 0xFCF10309
00:46:48: 00 75 01 01 01 03 02 36 35
00:46:48:
00:46:48: Serial0/0(in): Status, myseq 54
00:46:48: RT IE 1, length 1, type 1
00:46:48: KA IE 3, length 2, yourseq 54, myseq 54
00:46:58: Serial0/0(out): StEnq, myseq 55, yourseen 54, DTE up
00:46:58: datagramstart = 0xE3EEA4, datagramsize = 13
00:46:58: FR encapsulation = 0xFCF10309
00:46:58: 00 75 01 01 00 03 02 37 36
00:46:58:
00:46:58: Serial0/0(in): Status, myseq 55
00:46:58: RT IE 1, length 1, type 0
00:46:58: KA IE 3, length 2, yourseq 55, myseq 55
00:46:58: PVC IE 0x7, length 0x6, dlci 100, status 0x2, bw 0
```

RouterA#un all

All possible debugging has been turned off

RouterA#

Watch the myseq and yourseq incrementing; status 0x2 means that the PVC is operational.

RouterA#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B - BGP, D - EIGRP, EX - EIGRP external,
O - OSPF, IA - OSPF inter area,

N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2,
E1 - OSPF external type 1, E2 - OSPF external type 2,
E – EGP, i - IS-IS, L1 - IS-IS level-1,
L2 - IS-IS level-2, ia - IS-IS inter area,
* - candidate default, U - per-user static route,
o – ODR, P - periodic downloaded static route

Gateway of last resort is not set

C 172.16.0.0/16 is directly connected, Loopback0
R 172.30.0.0/16 [120/1] via 192.168.1.2, 00:00:12, Serial0/0
 192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Serial0/0
RouterA#

Try all of these commands on Router B also.

Show Runs

```
RouterA#show run
Building configuration...
Current configuration : 726 bytes
!
version 15.1
service timestamps debug uptime
no service password-encryption
!
hostname RouterA
!
enable secret 5 $1$jjQo$YJXxLo.EZm9t6Sq4UYeCv0
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Serial0/0
```

```
ip address 192.168.1.1 255.255.255.252
encapsulation frame-relay
frame-relay interface-dlci 100
!
router rip
version 2
network 172.16.0.0
network 192.168.1.0
!
end
```

RouterA#

```
RouterB#show run
Building configuration...

Current configuration : 633 bytes
!
version 15.1
service timestamps debug uptime
no service password-encryption
!
hostname RouterB
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Serial0/0
ip address 192.168.1.2 255.255.255.252
encapsulation frame-relay
frame-relay interface-dlci 200
!
router rip
version 2
network 172.30.0.0
```

```
network 192.168.1.0
```

```
!
```

```
end
```

```
RouterB#
```

```
---
```

```
FrameSwitch#show run
```

```
Building configuration...
```

```
Current configuration : 685 bytes
```

```
!
```

```
version 15.1
```

```
service timestamps debug uptime
```

```
no service password-encryption
```

```
!
```

```
hostname FrameSwitch
```

```
!
```

```
ip subnet-zero
```

```
!
```

```
frame-relay switching
```

```
!
```

```
interface Ethernet0
```

```
no ip address
```

```
shutdown
```

```
!
```

```
interface Serial0
```

```
no ip address
```

```
encapsulation frame-relay
```

```
clockrate 64000
```

```
frame-relay intf-type dce
```

```
frame-relay route 100 interface Serial1 200
```

```
!
```

```
interface Serial1
```

```
no ip address
```

```
encapsulation frame-relay
```

```
clockrate 64000
```

```
frame-relay intf-type dce
```

```
frame-relay route 200 interface Serial0 100
!
end
```

Lab 3: Frame Relay Subinterfaces

Lab Exercise

Your task is to configure the network in Figure 16.27 below to allow full connectivity using Frame Relay. In order to complete the lab, you will have to use three routers—two as hosts and one as the Frame Relay router/switch. Configuring a Frame Relay switch can be a little tricky, but you will never be expected to do this for the CCNA exam. It is purely for use in a lab environment.

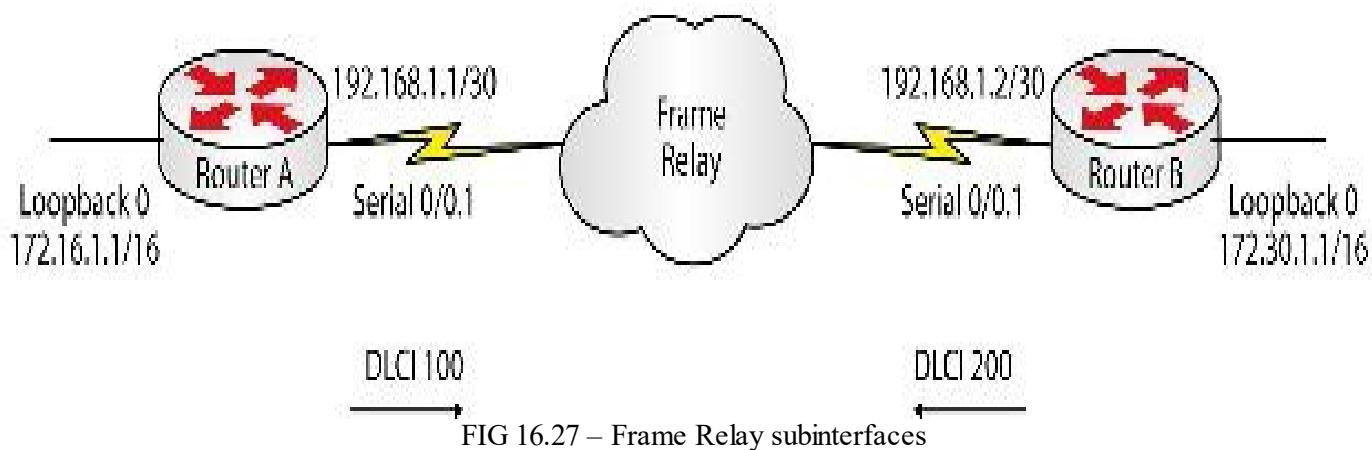


FIG 16.27 – Frame Relay subinterfaces

Frame Relay subinterfaces overcome split horizon issues, allowing more than one network to be attached to one physical interface. The physical interface can be divided into one or more subinterfaces.

Text in Courier New font indicates commands that can be entered on the router.

Lab Objectives

1. Use the IP addressing scheme depicted in Figure 16.27. The Frame Relay switch will have the DCE interfaces, so you will need to have the clock rate command added.
2. Configure Frame Relay on the Serial interfaces of Routers A and B.
3. Configure the Frame Relay switch.
4. Configure RIP on Routers A and B to allow for end-to-end connectivity.
5. Test the link by pinging across it.

Purpose

Frame Relay subinterfaces are in very common use. Subinterfaces allow multiple PVCs

to terminate on one physical interface. You will only have one PVC per interface in this lab, but you will practice using subinterfaces.

Lab Walk-through

1. Set the IP address and encapsulation type on the routers:

```
Router#config t
Router(config)#hostname RouterA
RouterA(config)#interface Serial0/0
RouterA(config-if)#encapsulation frame-relay
RouterA(config-if)#no shut
RouterA(config)#interface Serial0/0.1 point-to-point
RouterA(config-subif)#ip add 192.168.1.1 255.255.255.252
RouterA(config-subif)#frame-relay interface-dlci 100 i You may need to exit to config-if after this command
RouterA(config-fr-dlci)#interface Loopback0
RouterA(config-if)#ip address 172.16.1.1 255.255.0.0
```

Router B:

```
Router#config t
Router(config)#hostname RouterB
RouterB(config)#interface Serial0/0
RouterB(config-if)#encapsulation frame-relay
RouterB(config-if)#no shut
RouterB(config-if)#interface Serial0/0.1 point-to-point
RouterB(config-subif)#ip add 192.168.1.2 255.255.255.252
RouterB(config-subif)#frame-relay interface-dlci 200
RouterB(config-fr-dlci)#interface Loopback0
RouterB(config-if)#ip address 172.30.1.1 255.255.0.0
```

2. You need to configure RIP to allow all networks to see each other.

```
RouterA#config t
RouterA(config)#router rip
RouterA(config-router)#version 2
RouterA(config-router)#network 192.168.1.0
RouterA(config-router)#network 172.16.0.0
```

Router B:

```
RouterB#config t
RouterB(config)#router rip
```

```
RouterB(config-router)#version 2
RouterB(config-router)#network 192.168.1.0
RouterB(config-router)#network 172.30.0.0
```

3. You need to configure the Frame Relay switch. Just copy the configuration and change the interface if you are not using the same ones shown here.

```
Router#config t
Router(config)#hostname FrameSwitch
FrameSwitch(config)#frame-relay switching i Make the router a frame switch
FrameSwitch(config)#interface Serial0
FrameSwitch(config-if)#no ip address
FrameSwitch(config-if)#clock rate 64000
FrameSwitch(config-if)#encapsulation frame-relay
FrameSwitch(config-if)#frame-relay intf-type dce i Make the interface DCE
FrameSwitch(config-if)#frame-relay route 100 interface serial1 200 i Sends traffic from dlc1 100 out of interface Serial1 as dlc1 200
FrameSwitch(config-if)#no shut
FrameSwitch(config-if)#interface Serial1
FrameSwitch(config-if)#no ip address
FrameSwitch(config-if)#encapsulation frame-relay
FrameSwitch(config-if)#frame-relay intf-type dce
FrameSwitch(config-if)#frame-relay route 200 interface Serial0 100
FrameSwitch(config-if)#clock rate 64000
FrameSwitch(config-if)#no shut
```

4. Ping from Router A to the Loopback on Router B.

```
RouterA#ping 172.30.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/60/60 ms
RouterA#
```

5. Check for Frame Relay connectivity.

```
RouterA#show frame-relay map
Serial0 (up): ip 192.168.1.2 dlci 100(0x64,0x1840), dynamic,
               broadcast, status defined, active
RouterA#show frame-relay pvc
PVC Statistics for interface Serial0/0 (Frame Relay DTE)
```

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 20 output pkts 20 in bytes 1838
 out bytes 1898 dropped pkts 0 in FECN pkts 0
 in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
 in DE pkts 0 out DE pkts 0
 out bcast pkts 10 out bcast bytes 858
 pvc create time 00:07:57, last time pvc status changed 00:03:23

RouterA#show frame-relay lmi

LMI Statistics for interface Serial0/0 (Frame Relay DTE) LMI TYPE = CISCO

Invalid Unnumbered info 0	Invalid Prot Disc 0
Invalid dummy Call Ref 0	Invalid Msg Type 0
Invalid Status Message 0	Invalid Lock Shift 0
Invalid Information ID 0	Invalid Report IE Len 0
Invalid Report Request 0	Invalid Keep IE Len 0
Num Status Enq. Sent 47	Num Status msgs Rcvd 48
Num Update Status Rcvd 0	Num Status Timeouts 0

RouterA#

RouterA#debug frame-relay lmi

Frame Relay LMI debugging is on

Displaying all Frame Relay LMI data

RouterA#

00:46:38: Serial0(out): StEnq, myseq 53, yourseen 52, DTE up

00:46:38: datagramstart = 0xE3EEA4, datagramsize = 13

00:46:38: FR encapsulation = 0xFCF10309

00:46:38: 00 75 01 01 01 03 02 35 34

00:46:38:

00:46:38: Serial0(in): Status, myseq 53

00:46:38: RT IE 1, length 1, type 1

00:46:38: KA IE 3, length 2, yourseq 53, myseq 53

00:46:48: Serial0(out): StEnq, myseq 54, yourseen 53, DTE up

00:46:48: datagramstart = 0xE3EEA4, datagramsize = 13

00:46:48: FR encapsulation = 0xFCF10309

00:46:48: 00 75 01 01 01 03 02 36 35

00:46:48:
00:46:48: Serial0(in): Status, myseq 54
00:46:48: RT IE 1, length 1, type 1
00:46:48: KA IE 3, length 2, yourseq 54, myseq 54
00:46:58: Serial0(out): StEnq, myseq 55, yourseen 54, DTE up
00:46:58: datagramstart = 0xE3EEA4, datagramsize = 13
00:46:58: FR encapsulation = 0xFCF10309
00:46:58: 00 75 01 01 00 03 02 37 36
00:46:58:
00:46:58: Serial0(in): Status, myseq 55
00:46:58: RT IE 1, length 1, type 0
00:46:58: KA IE 3, length 2, yourseq 55, myseq 55
00:46:58: PVC IE 0x7 , length 0x6 , dlci 100, status 0x2 , bw 0
RouterA#un all
All possible debugging has been turned off
RouterA#
RouterA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - mobile, B – BGP, D - EIGRP, EX - EIGRP external,
O - OSPF, IA - OSPF inter area,
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2,
E – EGP, i - IS-IS, L1 - IS-IS level-1,
L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route,
o – ODR, P - periodic downloaded static route
Gateway of last resort is not set
C 172.16.0.0/16 is directly connected, Loopback0
R 172.30.0.0/16 [120/1] via 192.168.1.2, 00:00:12, Serial0/0
192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Serial0/0
RouterA#

Try all of these commands on Router B also.

Show Runs

RouterA#show run

Building configuration...

```
Current configuration : 781 bytes
!
version 15.1
service timestamps debug uptime
no service password-encryption
!
hostname RouterA
!
enable secret 5 $1$jjQo$YJXxLo.EZm9t6Sq4UYeCv0
!
username banbury password 0 ccna
!
ip subnet-zero
!
interface Loopback0
ip address 172.16.1.1 255.255.0.0
!
interface Serial0/0
no ip address
encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
ip address 192.168.1.1 255.255.255.252
frame-relay interface-dlci 100
!
router rip
version 2
network 172.16.0.0
network 192.168.1.0
!
end
```

RouterA#

RouterB#show run
Building configuration...

```
Current configuration : 688 bytes
!
version 15.1
service timestamps debug uptime
no service password-encryption
!
hostname RouterB
!
ip subnet-zero
!
interface Loopback0
ip address 172.30.1.1 255.255.0.0
!
interface Serial0/0
no ip address
encapsulation frame-relay
!
interface Serial0/0.1 point-to-point
ip address 192.168.1.2 255.255.255.252
frame-relay interface-dlci 200
!
router rip
version 2
network 172.30.0.0
network 192.168.1.0
!
RouterB#
```

```
FrameSwitch#show run
Building configuration...

Current configuration : 685 bytes
!
version 15.1
service timestamps debug uptime
!
hostname FrameSwitch
```

```
!
ip subnet-zero
!
frame-relay switching
!
interface Ethernet0
no ip address
shutdown
!
interface Serial0
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 100 interface Serial1 200
!
interface Serial1
no ip address
encapsulation frame-relay
clockrate 64000
frame-relay intf-type dce
frame-relay route 200 interface Serial0 100
!
end
FrameSwitch#
```

Chapter 17 — Advanced Network Troubleshooting

What You Will Learn in This Chapter

Troubleshooting Using NetFlow Data

Troubleshooting STP

Troubleshooting InterVLAN Routing

Troubleshooting Routing Issues

Troubleshooting OSPF

Troubleshooting EIGRP

Troubleshooting WAN Connectivity

Troubleshooting EtherChannels

Syllabus Topics Covered

4.0 Troubleshooting

- 4.1 Identify and correct common network problems
- 4.2 Utilize NetFlow data
- 4.9 Monitor NetFlow statistics
- 4.3 Troubleshoot and resolve Spanning Tree operation issues
 - 4.3.a Verify root switch
 - 4.3.b Verify priority
 - 4.3.c Verify that mode is correct
 - 4.3.d Verify port states
- 4.4 Troubleshoot and resolve routing issues
 - 4.4.a Verify that routing is enabled (sh ip protocols)
 - 4.4.b Verify that routing table is correct
 - 4.4.c Verify correct path selection
- 4.5 Troubleshoot and resolve OSPF problems
 - 4.5.a Verify neighbor adjacencies
 - 4.5.b Verify Hello and Dead timers
 - 4.5.c Verify OSPF area
 - 4.5.d Verify interface MTU
 - 4.5.e Verify network types

- 4.5.f Verify neighbor states
- 4.5.g Review OSPF topology table
- 4.6 Troubleshoot and resolve EIGRP problems
 - 4.6.a Verify neighbor adjacencies
 - 4.6.b Verify ASN
 - 4.6.c Verify load balancing
 - 4.6.d Verify split horizon

- 4.7 Troubleshoot and resolve interVLAN routing problems
 - 4.7.a Verify connectivity
 - 4.7.b Verify encapsulation
 - 4.7.c Verify subnet
 - 4.7.d Verify native VLAN
 - 4.7.e Verify port mode trunk status

- 4.8 Troubleshoot and resolve WAN implementation issues
 - 4.8.a Serial interfaces
 - 4.8.b Frame Relay
 - 4.8.c PPP

4.10 Troubleshoot EtherChannel problems

We covered troubleshooting ICND1/CCENT issues in the first part of this manual. As you can imagine, ICND2 troubleshooting will be more complex because of the technologies involved. Initial configurations are more difficult, the commands are more granular, and you will find that your network could partially work or intermittently work, which will make your task all the more difficult.

In the exam you can expect your troubleshooting to fall into the categories below:

- General theory questions about the likely cause of an issue
- A network diagram and questions about the likely cause of an issue
- Logging into equipment and using show commands to determine the cause of an issue
- Having to log in to a network, diagnose, and resolve configuration problems

The last issue will be the most difficult because you did not complete the initial configurations, so you will have to quickly familiarize yourself with the topology, use the relevant show commands to check outputs and configurations, and quickly resolve the issue so you can move on to the next question.

Randomly typing show commands or typing show run on every device hoping to stumble

on the root cause is a recipe for disaster. Bear in mind that there may well be multiple issues on multiple OSI layers, such as physical, data link, and network on both routers and switches, so you will have your work cut out for you!

The secret sauce is really understanding the technology and doing lots of hands-on labs throughout this guide. You will make mistakes as you go along and discover which show command will reveal the root cause of the issue to you. Once you get comfortable with my labs, start making your own up using different IP addresses and interfaces. Like any master, you get there by doing tasks over and over until they become second nature. If you go into the exam not knowing which command will show you the Spanning Tree root (for example), then failure is inevitable because the show run command won't tell you.

Troubleshooting Using NetFlow

NetFlow is an IOS-based monitoring and measurement technology solution created by Cisco. It offers the ability to collect IP traffic as it either enters or exits an interface. NetFlow provides far more detail on the data than other monitoring protocols such as SNMP. It also scales to a large number of interfaces and this makes it a great enterprise and service provider solution.

Cisco has actually renamed NetFlow as Cisco Flexible NetFlow and they explain its purpose on their website, along with how to purchase, install, and configure it:

“It optimizes the network infrastructure, reducing operation costs and improving capacity planning and security incident detection with increased flexibility and scalability. The ability to characterize IP traffic and identify its source, traffic destination, timing, and application information is critical for network availability, performance, and troubleshooting. The monitoring of IP traffic flows increases the accuracy of capacity planning and ensures that resource allocation supports organizational goals. Flexible NetFlow helps Cisco customers determine how to optimize resource usage, plan network capacity, and identify the optimal application layer for Quality of Service (QoS). It plays a vital role in network security by detecting Denial of Service (DoS) attacks and network-propagated worms.”

By analyzing the data the network administrator can see the source and destination of traffic, the class or service, and the causes of network congestion.

An IP flow is characterized by a unidirectional sequence of packets with a matching set of attributes for layer 2, IPv4, or IPv6, including:

- Source IP
- Destination IP

- Source port
- Destination port
- IP protocol type
- Type of service
- Router ingress interface

If there is any change in any of the above fields it is considered a new flow.

NetFlow is also a great solution for service providers because it supports customer service programs and uses popular data warehousing and data mining solutions that are critical for competitive vendor offerings, such as flexible accounting and billing that can consider application usage, the time of day, bandwidth utilization, or Quality of Service elements. NetFlow is also a great tool for network scalability planning and overall analysis, and it can help lower the organization's TCO (total cost of ownership).

Figure 17.1 below shows NetFlow in operation on an enterprise network:

Different Flow Monitors for detecting different information

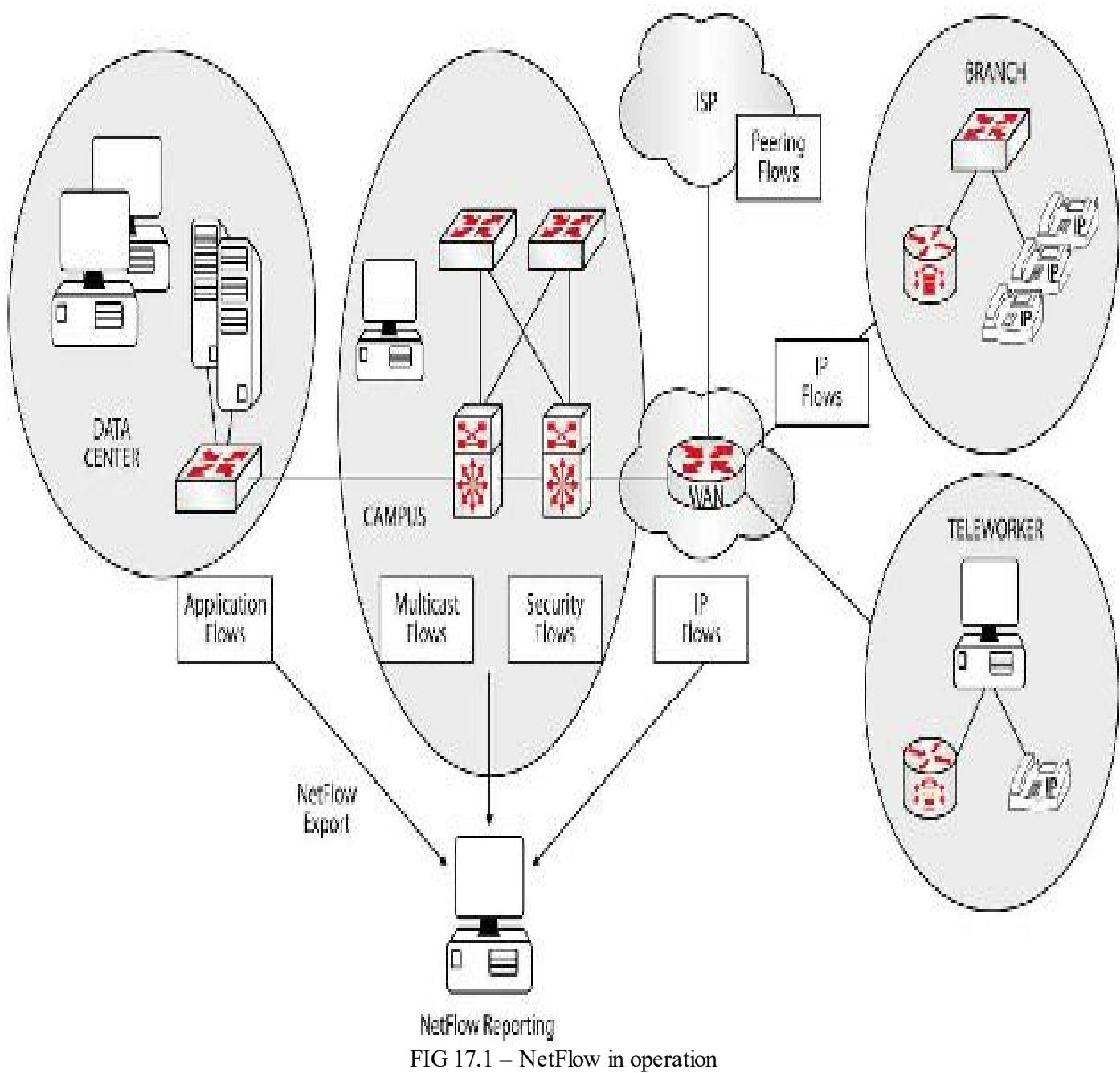


FIG 17.1 – NetFlow in operation

The NetFlow management architecture consists of three components:

- NetFlow data export service
- NetFlow flow collector service
- NetFlow data analysis

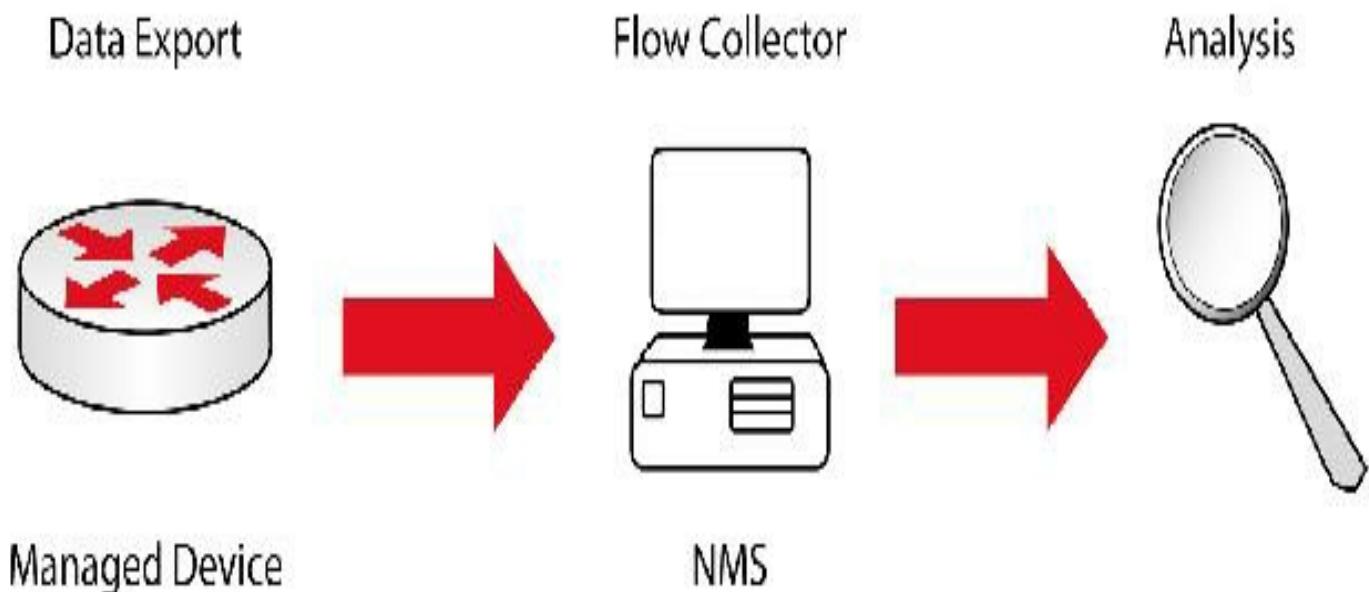


FIG 17.2 – NetFlow management architecture

At the top of the three-tier NetFlow architecture there is the NetFlow data export service. This is where the data warehousing and data mining solutions occur. It captures the accounting statistics for traffic on the monitored networking devices and it uses the UDP protocol to export data. This is a three-part process:

- Data switching
- Data export
- Data aggregation

The data is exported to the second tier—the NetFlow flow collector service. At this level, using servers and workstations you would complete tasks such as data collection, data filtering, aggregation, data storage, and file system management using the existing or third-party file systems.

At the lowest tier, at the access layer, you will find the NetFlow data analysis. At this level you can use network planning tools, overall network analysis tools, and accounting and billing tools, as well as export data to various database systems or Excel spreadsheets.

Some of the many usage scenarios NetFlow offers are:

- Network planning
- Network accounting and usage billing
- User monitoring
- Troubleshooting a slow network/performance issues
- Traffic accounting
- Security analysis and enforcement
- Network monitoring

- Data warehousing and mining
- Enforcement of network policies

Configuring NetFlow on a Cisco router requires a number of steps. First, you need to configure a specific interface to capture traffic flows. This is done using the following command:

```
Router(config)#interface FastEthernet0/1
```

```
Router(config-if)#ip flow ingress
```

Either ip flow ingress or ip route-cache flow command can be used depending on the Cisco IOS Software version. The ip flow ingress command is available in Cisco IOS Software Release 12.2(15)T or above. This is the flow monitor. According to Cisco:

“Flow monitors are the Flexible NetFlow component that is applied to interfaces to perform network traffic monitoring. Flow monitors consist of a record and a cache. You add the record to the flow monitor after you create the flow monitor. The flow monitor cache is automatically created at the time the flow monitor is applied to the first interface. Flow data is collected from the network traffic during the monitoring process based on the key and nonkey fields in the record, which is configured for the flow monitor and stored in the flow monitor cache.”

```
R1#debug flow ?
```

exporter Flow exporter information

monitor Flow monitor information

record Flow Record configuration and operation

```
R1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R1(config)#flow ?
```

exporter Define a Flow Exporter

monitor Define a Flow Monitor

record Define a Flow Record

```
R1(config)#flow monitor CCNA
```

```
R1(config-flow-monitor)#
```

Next, (if you are exporting the NetFlow cache to a reporting server), you need to configure the NetFlow version. This depends on the NetFlow collecting software you

are using and what NetFlow version it supports. Version 9 is the latest Cisco NetFlow version and is the most flexible version. Let's configure this on the router. You also want to configure the NetFlow collector address and the associated port number. NetFlow data is transmitted using UDP by default.

```
Router(config)#ip flow-export version 5
```

```
Router(config)#ip flow-export destination 150.1.1.254 5000
```

If you want to see the data statistics collected on the router, you can issue the show ip cache flow command on the device:

```
R1#show ip cache flow
```

IP packet size distribution (45 total packets):

1-32	64	96	128	160	192	224	256	288	320	352	384	416
.000	.000	.000	1.00	.000	.000	.000	.000	.000	.000	.000	.000	.000

IP Flow Switching Cache, 278544 bytes

1 active, 4095 inactive, 1 added

IP Sub Flow Cache, 34056 bytes

0 active, 1024 inactive, 0 added, 0 added to flow

last clearing of statistics never

Protocol	Total Flows	Packets Bytes	Packets Active(Sec)	Idle(Sec)
----------	-------------	---------------	---------------------	-----------

Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
-------	------	-------	------	------	-------	-------

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP
-------	--------------	-------	--------------	----	------

[output truncated]

NetFlow information collected by the dedicated software can be used to generate useful traffic statistics that allow the administrator to troubleshoot different processes in the network. Traffic can be examined based on application, user, protocol, etc. Network administrators can observe device and link utilization at different times of the day when investigating network capacity issues. They can also create a traffic baseline based on normal traffic patterns and detect anomalies like Denial of Service attacks or other security breaches.

With this information you can identify bandwidth hogs, avoid expensive and unnecessary upgrades, spot security or other anomalies, analyze the loads created by new applications, and identify peak use periods.

Before adding NetFlow to your network, you should consider the impact on device CPU

utilization, where you will send the information you've collected, and the number of devices exporting data. Cisco sales experts can assist with this process.

Troubleshooting STP

This section touches on common STP problems and ways to troubleshoot them. The steps given here apply to both 802.1D and 802.1W running different STP process on each VLAN.

STP is a very maintenance-free protocol and generally does not require troubleshooting. STP will mostly have the following problems:

- Incorrect root bridge
- Incorrect root port
- Incorrect designated port

Let's look at each of the problems and ways to troubleshoot them.

Incorrect Root Bridge

The root bridge is selected based on the bridge ID (BID), which consists of priority and the base MAC address of the switch. The `show spanning-tree vlan [vlan#]` command will show the current root bridge. Note the MAC address and the priority of the root bridge and compare it with those of the switch that you want to make the root bridge. Decreasing the priority of the correct switch should resolve the problem. This can be done using the `spanning-tree vlan [vlan#] priority [priority]` or the `spanning-tree [vlan #] root primary` command.

You can glean specific root bridge information with the `show spanning-tree bridge` command, which displays the local bridge information. This isn't likely to work on the exam simulator though.

ALS1#`show spanning-tree bridge`

Hello Max Fwd

Vlan	Bridge ID	Time	Age	Dly	Protocol
------	-----------	------	-----	-----	----------

VLAN0001	32769 (32768, 1)	0011.9247.db00	2	20	15 ieee
----------	------------------	----------------	---	----	---------

Incorrect Root Port

The root port is the fastest path from a switch to the root bridge. The cost is the cumulative cost of all the links in the path, with each port adding its cost to the VLAN. For example, if there are two 100 Mbps links between a switch and a root bridge, then

the cost is 38. The show spanning-tree vlan [vlan #] command will show the current root port and its cost. Compare that with the cost of the desired path. The cost of the desired path can be changed using the spanning-tree cost [cost] interface command.

In Figure 17.3 below, Switch 0 is the root bridge for VLAN 20 and all switches are connected by Fast Ethernet cables. You would expect Switch 2 to show a cost of 38. Fa0/1 points to the root bridge so it should be the root port and Fa0/2 points away so it should be the designated port.

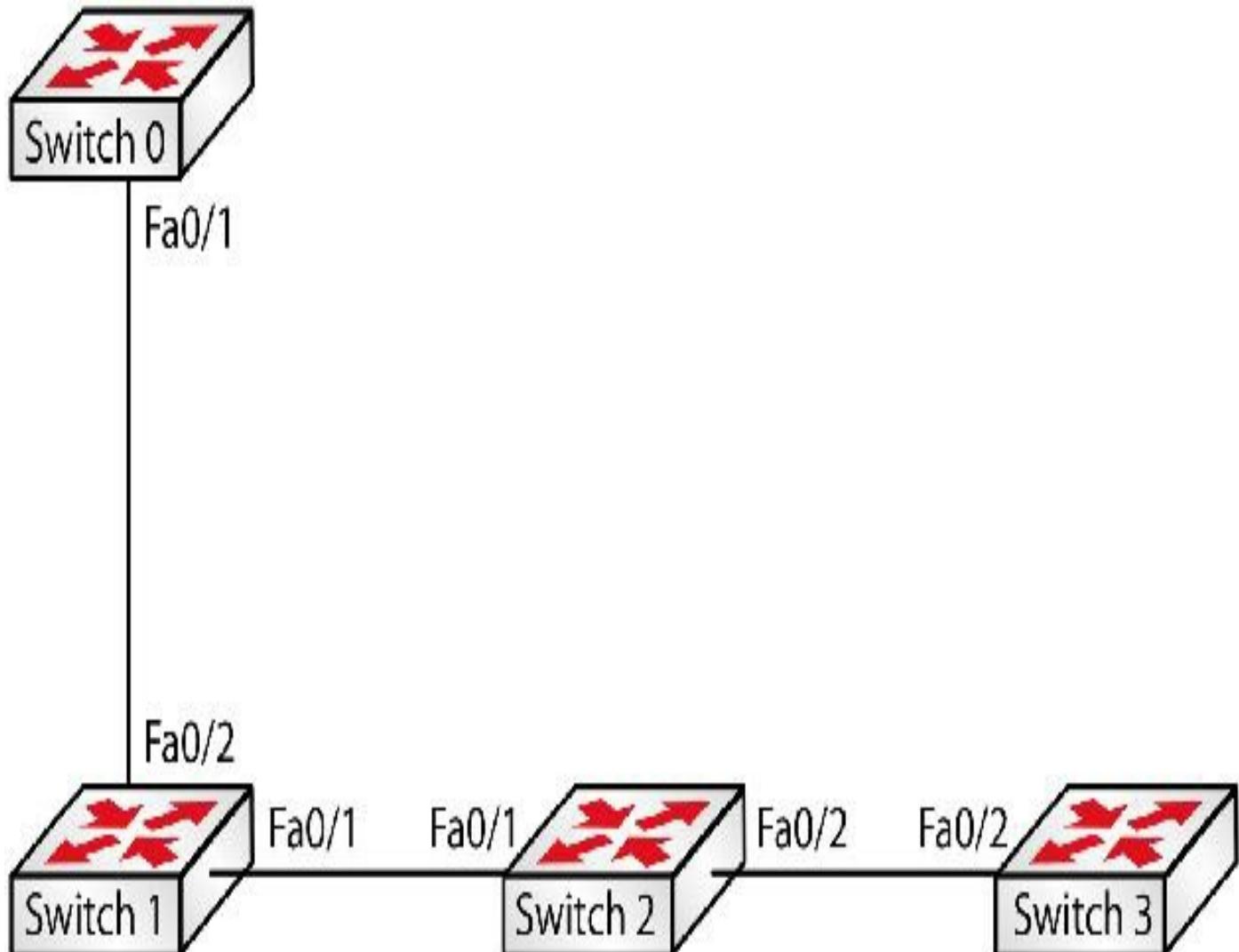


FIG 17.3 – Incorrect root port example

```
Switch2#show spanning-tree vlan 20
```

```
VLAN0020
```

```
Spanning tree enabled protocol ieee
```

```
Root ID Priority 24596
```

Address 0003.E469.8D9B
Cost 38
Port 1(FastEthernet0/1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32788 (priority 32768 sys-id-ext 20)
Address 0002.4A05.849B
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 20

Interface Role Sts Cost Prio.Nbr Type

Fa0/1 Root FWD 19 128.1 P2p
Fa0/2 Desg FWD 19 128.2 P2p

Switch 3 should add another 19 to the cost and have one root port pointing toward Switch 2:

Switch3#show spanning-tree vlan 20

VLAN0020

Spanning tree enabled protocol ieee

Root ID Priority 24596

Address 0003.E469.8D9B

Cost 57

Port 2(FastEthernet0/2)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32788 (priority 32768 sys-id-ext 20)

Address 0009.7CD4.6A95

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 20

Interface Role Sts Cost Prio.Nbr Type

Fa0/2 Root FWD 19 128.2 P2p

If you wanted to alter this behavior you could change the cost directly on a port with the above stated command.

Incorrect Designated Port

The designated port is the lowest cost port connecting a network segment to the rest of the network. The designated port cost can be seen and changed using the show spanning-tree vlan [vlan#] command and the spanning-tree cost [cost] command.

A summary of what you should see with a correctly configured Spanning Tree domain is shown in Figure 17.4 below. Make sure that you work out why each interface is in the relevant state and why the root bridge has become the root.

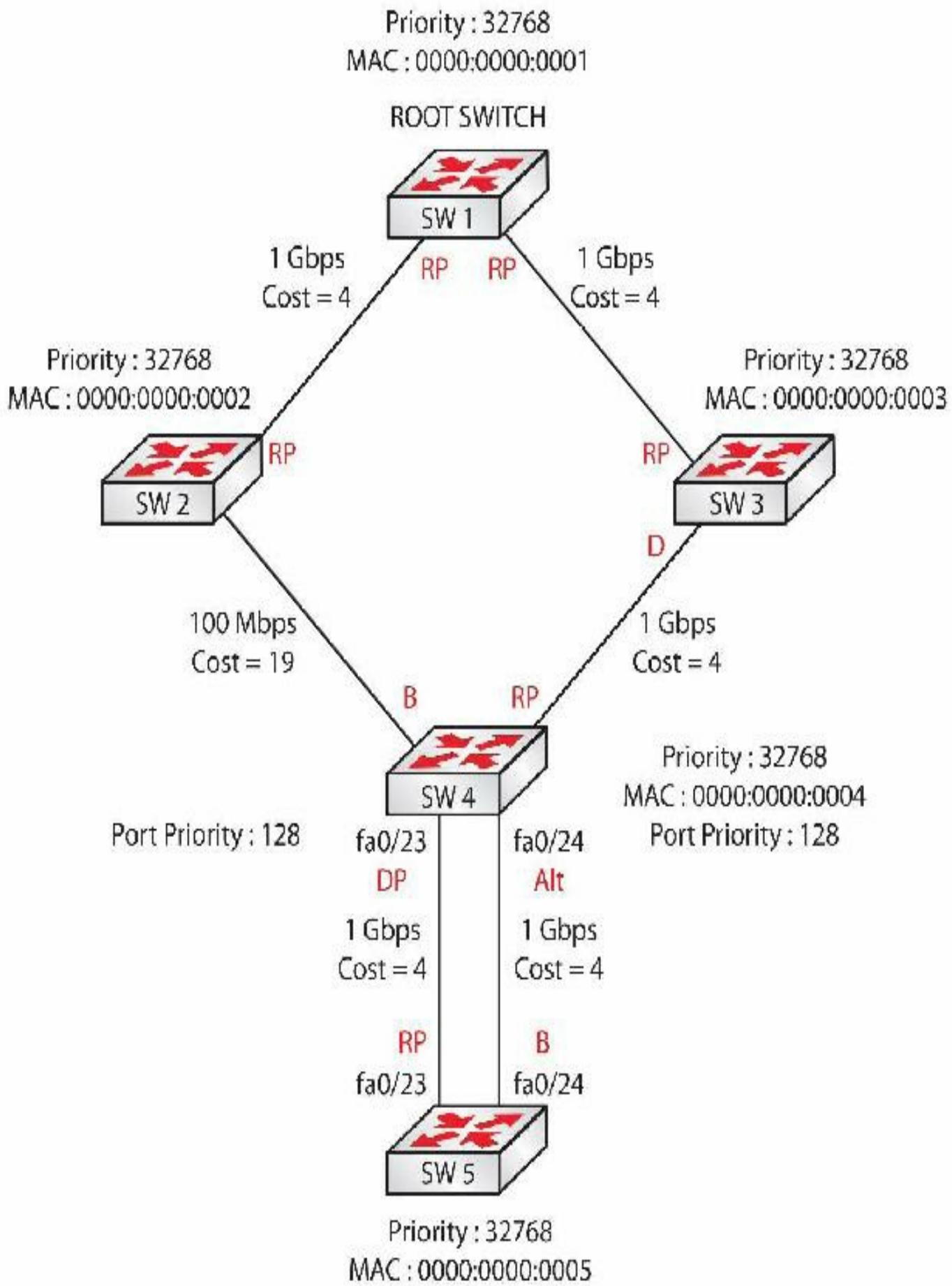


FIG 17.4 – STP operating correctly

To troubleshoot STP and VLAN issues, a very useful command to display a summary of interface settings, status, speed, and duplex settings is show interfaces status. As always, please type this out during any switching labs so the command is burned into your brain.

ALS1#show interfaces status

Port Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	notconnect	1	auto	auto	10/100BaseTX
Fa0/2	connected	1	a-full	a-100	10/100BaseTX
Fa0/3	connected	1	a-full	a-100	10/100BaseTX
Fa0/4	connected	1	a-full	a-100	10/100BaseTX
Fa0/5	notconnect	1	auto	auto	10/100BaseTX
Fa0/6	notconnect	1	auto	auto	10/100BaseTX
Fa0/7	connected	1	a-full	a-100	10/100BaseTX
Fa0/8	connected	1	a-full	a-100	10/100BaseTX

[output truncated]

Troubleshooting InterVLAN Routing Problems

When troubleshooting interVLAN routing, there are a few standard things you should check to ensure connectivity between hosts connected in different VLANs.

One of the first things to verify is layer 1 connectivity. Make sure that:

- User workstations are connected to the correct switch ports and the ports are active. Verify that the link is up, both on the workstations and on the switches.
- The links between the switches and the router(s) are functional and in up/up mode.

The next thing to do is verify that the user-facing switch ports are assigned to the correct VLANs. You can do this using the show vlan command on the switches:

Switch#show vlan

VLAN Name	Status	Ports
2 VLAN2	active	Fa1/1, Fa1/2

3 VLAN3 active Fa1/3, Fa1/4

It would be a good idea to also verify the trunk links between switches and between switches and the routers that perform the interVLAN routing to make sure that all relevant VLANs are carried on all trunk links up to the routers. You should also check the native VLAN match on both ends of the trunk. If it doesn't match, untagged traffic will not be transmitted as expected. You can check this using the show interfaces trunk command:

Switch#show interfaces trunk

Port Mode Encapsulation Status Native vlan

Fa1/11 on 802.1q trunking 1

Fa1/12 on 802.1q trunking 1

Port Vlans allowed on trunk

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

Port Vlans allowed and active in management domain

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

Port Vlans in spanning tree forwarding state and not pruned

Fa1/11 1,3,4,5

Fa1/12 1,3,4,5

If the trunk links are not forming, you have to check the encapsulation configured on them. A link configured as ISL on one end and dot1q on the other link will not work as expected (this is not an issue for the Cisco 2960 Switch model). The correct encapsulation can be analyzed with the show interfaces trunk command (see the output above). Another useful piece of information this command provides is the trunk mode (which is on in the output above).

Moving up to layer 3 verification, you need to make sure that correct IP addresses are configured on:

- User workstations
- Router subinterfaces or layer 3 switch SVIs

Each VLAN is usually associated with a dedicated IP subnet. You need to verify that all workstations and interfaces in a particular VLAN have a correct IP address and subnet mask configuration. You can check the IP addressing on the router subinterfaces using

the following command:

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	up	up
FastEthernet0/0.2	192.168.2.1	YES	manual	up	up
FastEthernet0/0.3	192.168.3.1	YES	manual	up	up

If you want to see the configured subnet, then of course use the show interface X command.

If all layer 1, layer 2, and layer 3 verifications successfully pass, you should have layer 3 connectivity between hosts in different VLANs. You can test this simply with a standard ping between workstations.

Troubleshooting Routing Issues

If you have access to network diagrams and documentation (which you will in the exam), you should be able to look at it and know what the configuration commands should be. For example, if you see a router with an OSPF area 0 interface showing IP address 192.168.1.1/30, you would expect to see the following configuration on the interface:

```
R1#show int f0/0
```

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c200.0700.0000 (bia c200.0700.0000)

Internet address is **192.168.1.1/30**

If it is anything but this you will have an issue. You would also expect the interface to be up physically at layer 2. If it isn't then there will likely be an issue with the clock rate or the no shut command having not been applied (clock rates are only a Serial line issue for DCE).

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.1	YES	manual	up	up

If this is all correct you would expect to see the route correctly advertised in the configuration.

```
R1#show ip protocols
```

Routing Protocol is “ospf 1”

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.1.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

192.168.1.0 0.0.0.255 area 0

However, there is an issue: the mask is /30 on the interface but the configuration is for a mask of /24. This will cause problems and you need to resolve this by removing the incorrect configuration and entering the correct one. If the interface had been down, you could have used other show commands to determine the cause of the issue. Once you believe that you have fixed the issue, you should check on a neighbor router to ensure that the routing table is showing the correct network.

As I mentioned, if you are randomly typing various show commands, you will either never find the cause of the issue above or it will take you so long that you run out of time in the exam and can't complete the other questions and labs.

There are only a few things you would find not working in the exam, and for routing protocol challenge labs these include:

- No clock rate on the DCE interface
- Interface hasn't had the no shut command applied
- Incorrect IP address or subnet mask on the interface
- Incorrect network advertised by the routing protocol
- Incorrect area or ASN configured for the network
- Auto summary turned off or on
- Access list blocking the routing protocol

The last entry is less likely in a routing troubleshooting challenge lab but it's worth bearing in mind.

We will cover the generic commands and then for each protocol the specific commands. On a live network you may use debug commands but the emulator in the exam will likely not support this. I always use the commands below until I find the cause of the problem:

- show ip route – on a neighbor router to check that the route is there
- show ip interface brief – is the interface up/up?
- show controllers s0/0 – which is the DCE interface/is there a clock rate added?

- show interface s0/0 – is the IP address/subnet mask and layer 2 protocol correct?
- show ip protocols – route correctly advertised/passive interface/auto summary?

You would use the commands above no matter what routing protocol you are troubleshooting. These should be more than enough for CCNA-level issues, but we will dig into protocol-specific troubleshooting below. A command that can come in handy is clear ip route *. This clears the routing table and forces the routing protocol to exchange routing tables.

When troubleshooting routing issues, the general recommendation is to follow the debugging process specific to each protocol (EIGRP, OSPF, etc.). We covered debug commands for these in the relevant chapters.

When using IPv6, you have to make sure that you configure the device for IPv6 routing. This is disabled by default on most devices.

Router(config)#ipv6 unicast-routing

The next step is to check whether the routing table is showing the correct entries learned by the specific OSPF protocol you configured and that the device is learning the necessary networks as advertised by the routing protocol. This is accomplished using the show ip route command:

R1#show ip route

[output truncated]

Gateway of last resort is not set

 10.0.0.0/24 is subnetted, 2 subnets

 C 10.0.12.0 is directly connected, FastEthernet0/0

 O IA 10.0.23.0 [110/20] via 10.0.12.2, 00:15:28, FastEthernet0/0

 192.168.1.0/32 is subnetted, 1 subnets

 C 192.168.1.1 is directly connected, Loopback0

In the output above, you can see that the router is learning about network 10.0.23.0 via OSPF. This means that the routing protocol is functioning, but if you want to make sure that it's functioning as it should (e.g., maybe you should have received multiple routes via OSPF or with other metrics), you should troubleshoot further, as explained in the next sections.

When learning the same prefix via multiple routing protocols, you might also need to check whether the correct route is installed into the routing table based on the

administrative distance of each protocol. For example, if you learn the same route via EIGRP and OSPF, only the EIGRP route will be installed in the routing table due to its lower AD value.

A truncated show ip route output with RIP advertising a remote network of 172.16.0.0 is shown below:

```
R 172.16.0.0/16 [120/1] via 192.168.1.2, 00:00:06, F0/0
C 10.0.0.0/8 is directly connected, Loopback0
C 192.168.1.0/24 is directly connected, F0/0
```

I added EIGRP on the remote router and even though RIP was still active, the route was replaced in the routing table due to the preferred AD:

```
D 172.16.0.0/16 [90/409600] via 192.168.1.2, 00:00:04, F0/0
C 10.0.0.0/8 is directly connected, Loopback0
C 192.168.1.0/24 is directly connected, F0/0
```

Troubleshooting OSPF

There are several commands you can use to verify and troubleshoot OSPF:

- show ip protocols – shows all routing protocol information on the router
- show ip route – shows the routing table
- show ip ospf interface – shows which interfaces are in which OSPF areas
- show ip ospf – shows the link state update interval and SPF details
- show ip ospf neighbor detail – shows a list of OSPF neighbors in detail
- show ip ospf database – shows the OSPF topological database

Of course, there are several OSPF debug commands available depending on which issue you are troubleshooting:

```
R2#debug ip ospf ?
adj          OSPF adjacency events
database-timer OSPF database timer
events        OSPF events
hello         OSPF Hello events
lsa-generation OSPF lsa generation
packet        OSPF packets
spf          OSPF spf
```

```
tree      OSPF database tree
```

[output truncated]

You can test these commands when you configure the OSPF labs in the Labs section.

Before you dive into troubleshooting, ensure that all the interfaces are up at layers 1 and 2. Also, check IP connectivity by pinging across all point-to-point links. Ensure that all subnet masks are correct and any layer 2 protocols such as PPP are operating correctly.

Bear in mind that access lists will block any routing traffic unless specifically permitted. Another gotcha is incomplete IPv6 parts in your configurations. For example, the command below will enable OSPF on the interface:

```
ip ospf 1 area 0
```

For OSPFv3, you need to type:

```
ipv6 ospf 1 area 0
```

I've made this mistake myself when hurriedly typing out configuration commands, and I've seen others do it and then spend a long time troubleshooting the issue.

When troubleshooting an OSPF implementation, you can approach the process in two ways:

1. Start by examining the OSPF adjacencies, LSDB, and then correct route installation
2. Start by examining whether the routes are present in the routing table, then the LSDB, and, finally, OSPF adjacencies

Either way, there are a few specific things that you need to verify depending on the topology. One of the first things to check is whether the OSPF neighbor adjacencies have been formed. If logging is enabled on your device, you should see a message similar to this one when two OSPF neighbors form an adjacency:

```
*Mar 1 00:03:24.687: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0  
from LOADING to FULL, Loading Done
```

This is a clear sign that the adjacency has been formed. If you want to dig a bit deeper into this, you can use one of the following commands to inspect neighbor relationships:

```
R2#show ip protocols
```

Routing Protocol is ospf 1

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 2.2.2.2

It is an area border router

Number of areas in this router is 2. 2 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

Routing on Interfaces Configured Explicitly (Area 0):

FastEthernet0/0

Routing on Interfaces Configured Explicitly (Area 23):

FastEthernet0/1

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway	Distance	Last Update
3.3.3.3	110	00:11:38
1.1.1.1	110	00:12:22

Distance: (default is 110)

R2#show ip ospf ?

[1-65535]	Process ID number
border-routers	Border and Boundary Router Information
database	Database summary
flood-list	Link state flood list
interface	Interface information
max-metric	Max-metric origination information
mpls	MPLS related information
neighbor	Neighbor list
request-list	Link state request list
retransmission-list	Link state retransmission list
rib	Routing Information Base (RIB)
sham-links	Sham link information
statistics	Various OSPF Statistics
summary-address	Summary-address redistribution Information
timers	OSPF timers information
traffic	Traffic related statistics
virtual-links	Virtual link information
	Output modifiers
[cr]	

R2#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/BDR	00:00:37	10.0.12.1	F0/0

```
3.3.3.3      1  FULL/DR  00:00:39  10.0.23.3  F0/1
```

The show ip ospf neighbor command gives you an explicit summary of the neighbor relationships. In the output above, you can see that R2 has two neighbors and you can also inspect the state of those neighbors, which is useful in broadcast networks in order to see which router has been selected as the DR.

If the adjacencies do not form, you might have a timer mismatch between the two neighbors. Remember, the default OSPF timers are 10/40 in broadcast networks. If you don't see OSPF adjacencies forming, one of the first debug commands you can issue is the debug ip ospf hello as shown in the output below:

```
R1#debug ip ospf hello
OSPF hello events debugging is on
*Mar 1 00:23:26.887: OSPF: Send Hello to 224.0.0.5 area 0 on FastEthernet0/0 from
10.0.12.1
R1#
*Mar 1 00:23:31.055: OSPF: Rcv Hello from 2.2.2.2 area 0 from FastEthernet0/0
10.0.12.2
*Mar 1 00:23:31.059: OSPF: Mismatched Hello parameters from 10.0.12.2
*Mar 1 00:23:31.059: OSPF: Dead R 40 C 7, Hello R 10 C 5 Mask R 255.255.255.0
C 255.255.255.0
*Mar 1 00:23:31.867: OSPF: Send Hello to 224.0.0.5 area 0 on FastEthernet0/0 from
10.0.12.1
```

In the output above, you are informed about a timer mismatch with neighbor 10.0.12.2 and you can see that the timers received from the neighbors (labeled R in the output) are 10 for Hello and 40 for Dead, while the locally configured timers (labeled C in the output) are 5 for Hello and 7 for Dead.

You can fix this by configuring timers with the same value at both ends:

```
R1(config)#int fa0/0
R1(config-if)#ip ospf hello-interval 10
R1(config-if)#ip ospf dead-interval 40
R1(config-if)#
*Mar 1 00:26:56.563: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0
from LOADING to FULL, Loading Done
```

The next thing you want to check if adjacencies are not formed is the OSPF area ID configured on each side of the connection. If the area ID differs, the adjacency will not

form. In this case, a warning message will appear directly on the console, without any debug command activated:

```
*Mar 1 00:43:09.931: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be virtual-link but not found from 10.0.12.2, FastEthernet0/0
```

You can also see this by debugging the OSPF adjacencies:

```
R2#deb ip ospf adj
OSPF adjacency events debugging is on
*Mar 1 00:46:07.083: OSPF: Rcv pkt from 10.0.12.1, FastEthernet0/0, area 0.0.0.0
      mismatch area 0.0.0.1 in the header
```

To correct this, simply configure the same area ID on both ends of the peers:

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fa0/0
R1(config-if)#no ip ospf 1 area 1
R1(config-if)#ip ospf 1 area 0
*Mar 1 00:46:32.019: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0
from LOADING to FULL, Loading Done
```

Another very commonly seen reason for OSPF adjacencies not forming is an MTU mismatch between the OSPF peers. When this happens, the OSPF adjacency is put in DOWN mode:

```
*Mar 1 00:58:13.975: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0
from EXSTART to DOWN, Neighbor Down: Too many retransmissions
```

The reason for this is that during the OSPF adjacency building process, the different MTUs do not allow OSPF DBD packets to be exchanged. The routers try to exchange DBDs over a period of time and if this does not work the neighbor is put in DOWN mode. You can see the warning about mismatched MTUs in the debugging output below:

```
R2#deb ip ospf adj
OSPF adjacency events debugging is on
*Mar 1 00:56:43.155: OSPF: Send DBD to 1.1.1.1 on FastEthernet0/0 seq 0x12D9 opt
0x52 flag 0x7 len 32
*Mar 1 00:56:43.155: OSPF: Retransmitting DBD to 1.1.1.1 on FastEthernet0/0 [6]
*Mar 1 00:56:43.859: OSPF: Rcv DBD from 1.1.1.1 on FastEthernet0/0 seq 0x1D31
```

opt 0x52 flag 0x7 len 32 mtu 1000 state EXSTART

*Mar 1 00:56:43.859: **OSPF: Nbr 1.1.1.1 has smaller interface MTU**

*Mar 1 00:56:43.863: OSPF: First DBD and we are not SLAVE

In order to see which MTU values are configured at both ends, you can use the show interface command:

R1#show int fa0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c201.2f70.0000 (bia c201.2f70.0000)

Internet address is 10.0.12.1/24

MTU 1000 bytes, BW 10000 Kbit/sec, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

[output truncated]

R2#show interface fa0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c202.09d0.0000 (bia c202.09d0.0000)

Internet address is 10.0.12.2/24

MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, Loopback not set

[output truncated]

The MTU is the maximum packet size that the interface can transmit in one instance and it differs per interface type and protocol (1500 for Ethernet). You can clearly see that R1's interface has an MTU of 1000 bytes, while R2's interface MTU is 1500. Let's configure both of them to 1500 to form an OSPF adjacency:

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#int fa0/0

R1(config-if)#mtu 1500

*Mar 1 01:02:27.491: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on

FastEthernet0/0 from **LOADING** to **FULL**, Loading Done

The MTU has to match on both sides in order for an OSPF adjacency to form. If you issued the debug ip ospf adj command, you would see something like the following:

OSPF: Nbr 192.168.100.2 has larger interface MTU

Another “fix” is to issue the command below, but beware because this can lead to other issues since there will still be the underlying MTU mismatch:

```
R1(config)#interface Gig1/0
R1(config-if)#ip ospf mtu-ignore
R1(config-if)#end
```

Another reason for OSPF adjacencies not forming might be an OSPF network type mismatch on the connecting interfaces. OSPF requires the same network type to be configured on the peers: broadcast, non-broadcast, point-to-multipoint, or point-to-point.

If there is a network type mismatch, the adjacency is down and the neighbor does not come up in the show ip ospf neighbor command:

```
R1#show ip ospf nei
```

You can check the configured network type using the following command:

```
R1#show ip ospf int fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.0.12.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 10
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State WAITING, Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured,Hello 30,Dead 120, Wait 120, Retransmit 5
```

[output truncated]

```
R2#show ip ospf int fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.0.12.2/24, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 10
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router [ID] 2.2.2.2, Interface address 10.0.12.2
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
```

[output truncated]

In the output above, R1's interface is configured as non-broadcast and R2's interface is configured as broadcast from an OSPF perspective. This prevents the neighbor adjacencies from forming. In order to fix this, let's configure both interfaces as broadcast (in the exam you will be told or will see in a network diagram which it should be):

```
R1(config)#int fa0/0
R1(config-if)#ip ospf network broadcast
*Mar 1 01:15:45.511: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0
from LOADING to FULL, Loading Done
```

To check the neighbor state on an interface, the simplest command to use is show ip ospf neighbor:

```
R2#show ip ospf neighbor
Neighbor ID Pri State    Dead Time Address      Interface
1.1.1.1    1 FULL/BDR 00:00:34 10.0.12.1    F0/0
3.3.3.3    1 FULL/DR 00:00:36 10.0.23.3    F0/1
```

In the output above, you can see that both adjacencies are in a FULL state. The neighbor on interface Fast Ethernet 0/0 acts as the BDR (which means R2 is the DR on that link, if the link is broadcast) and the neighbor on interface Fast Ethernet 0/1 acts as the DR. You can see that R2 is the DR on the link with R1 by also examining the output of the following command:

```
R2#sho ip ospf int fa0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.0.12.2/24, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 10
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 2.2.2.2, Interface address 10.0.12.2
  Backup Designated router (ID) 1.1.1.1, Interface address 10.0.12.1
  Timer intervals configured,Hello 10,Dead 40, Wait 40, Retransmit 5
```

[output truncated]

Let's see what happens on a point-to-point link. You will have to use a pipe to drill down to the interface type:

```
R1#show ip ospf int fa0/0 | i Type  
Process ID 1,Router ID 1.1.1.1, Network Type POINT_TO_POINT,Cost: 10
```

```
R2#show ip ospf int fa0/0 | i Type  
Process ID 1,Router ID 2.2.2.2, Network Type POINT_TO_POINT,Cost: 10
```

```
R1#sho ip ospf nei
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	0	FULL / -	00:00:32	10.0.12.2	F0/0

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	0	FULL / -	00:00:38	10.0.12.1	F0/0
3.3.3.3	1	FULL/DR	00:00:39	10.0.23.3	F0/1

In the output above, you can see that the adjacency on the point-to-point link is in a FULL state but you have no DR or BDR, as expected, because they aren't needed on this type of network.

As a last step, if the expected routes are still not installed in the routing table, you should check the OSPF link state database (LSDB) to analyze which LSAs (link state advertisements) are received and from which neighbors. You would do this using the following command:

```
R2#show ip ospf database
```

OSPF Router with ID (2.2.2.2) (Process ID 1)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	725	0x80000011	0x00CBF6	2
2.2.2.2	2.2.2.2	724	0x8000000D	0x00734E	2

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
10.0.23.0	2.2.2.2	1572	0x80000003	0x002DDA
192.168.1.1	1.1.1.1	386	0x80000005	0x00A622
192.168.3.3	2.2.2.2	1572	0x80000003	0x00C6F1

Router Link States (Area 23)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
2.2.2.2	2.2.2.2	1572	0x80000004	0x00DDF0	1
3.3.3.3	3.3.3.3	1583	0x80000006	0x00AD96	2

Net Link States (Area 23)

Link ID	ADV Router	Age	Seq#	Checksum
10.0.23.3	3.3.3.3	1583	0x80000003	0x006D86

Summary Net Link States (Area 23)

Link ID	ADV Router	Age	Seq#	Checksum
10.0.12.0	2.2.2.2	1574	0x80000003	0x00A66C
192.168.1.1	2.2.2.2	724	0x80000001	0x00F4C9

Don't be put off by most of the commands. At the CCNA level, Cisco is more likely to try to trick you by adding either the wrong area to the configuration or advertising the wrong network/wildcard. Trying to catch you out with hard-to-find timer mismatches is very unlikely as a hands-on troubleshooting lab.

Mini-Lab – Troubleshooting OSPF

Although I've already mentioned it, please do double-check that you are advertising the correct OSPF network. OSPF will only do what you ask it to, unlike EIGRP, which can in some circumstances work out what you intended to configure. Take the interface below, for example, in Figure 17.5, which you can easily configure:

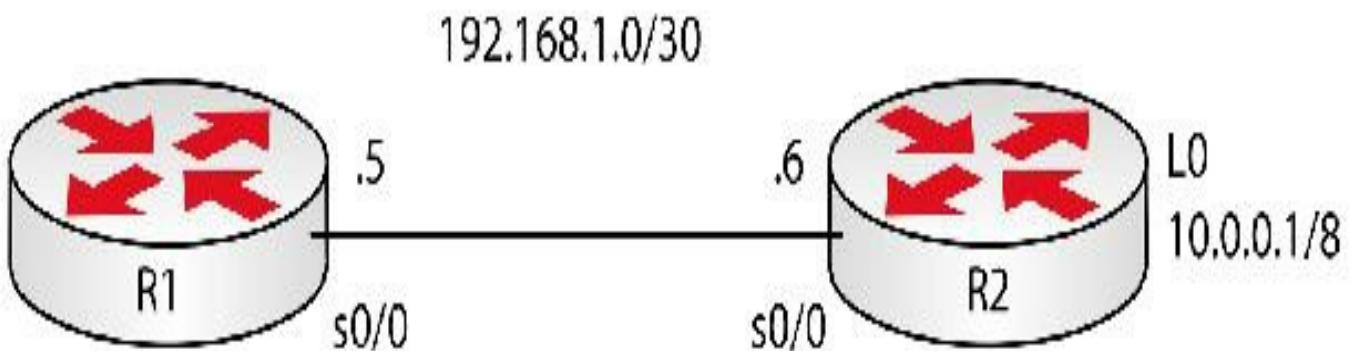


FIG 17.5 – Mini-lab: Troubleshooting OSPF

If you add the configuration below for OSPF, you might expect the neighbors to form an OSPF relationship and exchange LSAs, but this doesn't happen.

```
R1(config-router)#network 192.168.1.0 0.0.0.3 area 0
```

No neighbor relationship formed and if you started debugging the OSPF process, you

would see no outputs. Bear in mind that for the exam, debug commands will probably not be available.

```
R1#debug ip ospf hello  
OSPF Hello events debugging is on  
R1#debug ip ospf adj  
OSPF adjacency events debugging is on  
R1#debug ip ospf packet  
OSPF packet debugging is on
```

There is no OSPF neighbor present on the link, in fact.

```
R1#show ip ospf neighbor
```

```
R1#
```

But you have configured OSPF to advertise the hosts within the 192.168.1.0/30 network, which are (if you recall your subnetting) 192.168.1.1 and 192.168.1.2.

```
R1#show ip protocols  
Routing Protocol is "ospf 1"  
  Outgoing update filter list for all interfaces is not set  
  Incoming update filter list for all interfaces is not set  
  Router ID 192.168.1.1  
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa  
  Maximum path: 4
```

Routing for Networks:

192.168.1.0 0.0.0.3 area 0

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway	Distance	Last Update
192.168.1.1	110	00:10:27
192.168.1.2	110	00:10:03

Distance: (default is 110)

You can fix this by configuring OSPF to advertise the correct network on both sides (and removing the incorrect configuration).

```
R2(config-router)#no network 192.168.1.0 0.0.0.3 area 0
```

```
R2(config-router)#network 192.168.1.4 0.0.0.3 area 0
```

```
*Mar 1 00:27:24.755: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.5 on Serial0/0  
from LOADING to FULL, Loading Done
```

```
R2#show ip ospf nei
Neighbor ID Pri State      Dead Time Address      Interface
192.168.1.5 0   FULL/ -    00:00:32  192.168.1.5  Serial0/0
R2#
```

[END OF MINI-LAB]

Troubleshooting EIGRP

As with OSPF, the first place to start when troubleshooting an EIGRP problem is ensuring that layers 1 and 2 are operating correctly. Also ensure that the IP addresses and subnets have all been entered correctly and that neighbors are in the same subnet. You also need to check for any ACLs that could be blocking EIGRP traffic. Once this is all done, the first step in troubleshooting EIGRP is analyzing the output of the show ip protocols command:

```
R2#show ip protocols
```

Routing Protocol is eigrp 100

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 100

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks:

10.0.12.0/24

10.0.23.0/24

Routing Information Sources:

Gateway	Distance	Last Update
----------------	-----------------	--------------------

10.0.12.1 **90** **00:01:17**

10.0.23.3 **90** **00:00:56**

Distance: internal 90 external 170

In the output above, you can see the following things:

- R2 is part of EIGRP 100
- R2 is advertising two networks in EIGRP: 10.0.12.0/24 and 10.0.23.0/24
- R2 has two EIGRP neighbors: 10.0.12.1 and 10.0.23.3

You can illustrate what you now know (after using some of the show commands overleaf) with Figure 17.6 below. Other information isn't relevant for this example.

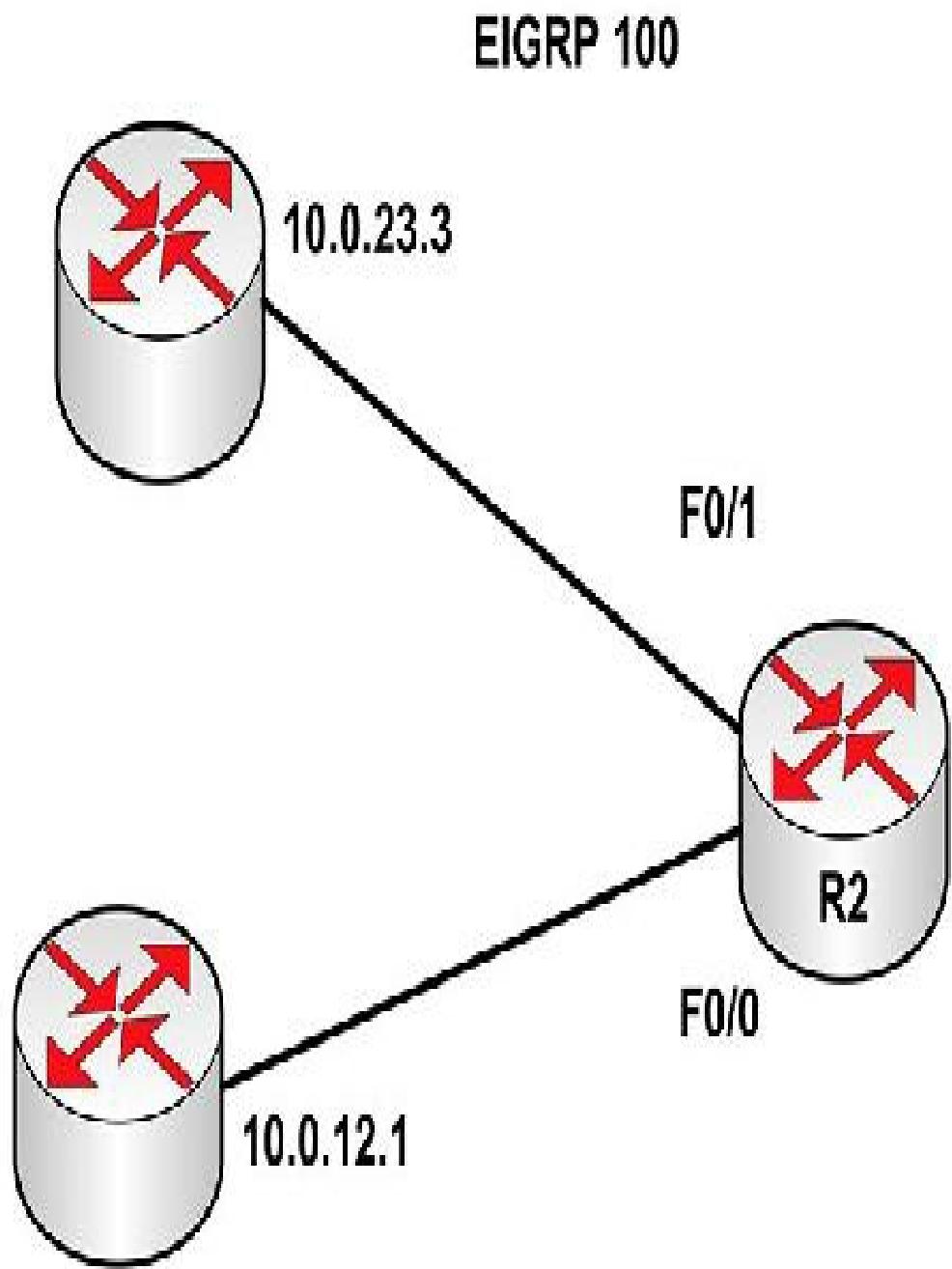


FIG 17.6 – Troubleshooting EIGRP

You can also see the EIGRP neighbor adjacencies using the `show ip eigrp neighbors` command. I'll show you all the options below first:

R2#`show ip eigrp ?`

- [1-65535] Autonomous System
- accounting IP-EIGRP accounting
- interfaces IP-EIGRP interfaces
- neighbors IP-EIGRP neighbors

topology IP-EIGRP Topology Table

traffic IP-EIGRP Traffic Statistics

vrf Select a VPN Routing/Forwarding instance

R2#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)	(ms)	Cnt	Num		
1	10.0.23.3	Fa0/1	10	00:05:23	102	612	0	5
0	10.0.12.1	Fa0/0	12	00:05:40	149	894	0	3

Another way to check EIGRP neighbor adjacencies is by issuing the following command:

R2#show ip eigrp interfaces

IP-EIGRP interfaces for process 100

Xmit	Mean	Pacing	Queue	Un/Reliable	Time	Multicast	Pending	Flow
Interface	Peers		SRTT	Un/Reliable		Timer		
Routes								
Fa0/0	0	0/0	0	0/2	624	0		
Fa0/1	1	0/0	104	0/2	432	0		

In the output above, you can see that R2 is configured with EIGRP on both Fa0/0 and Fa0/1 but it has only one peer, on Fa0/1. The most commonly seen reasons for a neighbor adjacency not forming after EIGRP is enabled on the interface are:

- Mismatched EIGRP ASN
- Passive interface configured

Let's check the EIGRP ASNs on both ends on that link:

R1#show ip protocols

Routing Protocol is eigrp 200

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

[output omitted]

R2#sho ip protocols

Routing Protocol is eigrp 100

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

[output omitted]

After modifying R1's configuration so that the EIGRP ASN is 100, the adjacency still does not form. Let's run a show ip protocols command on this router to further inspect the situation:

R1#sho ip prot

Routing Protocol is eigrp 100

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 100

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is in effect

Automatic address summarization:

10.0.0.0/8 for Loopback0

Summarizing with metric 281600

Maximum path: 4

Routing for Networks:

10.0.12.0/24

192.168.1.1/32

Passive Interface(s):

FastEthernet0/0

Routing Information Sources:

```
Gateway      Distance      Last Update  
(this router)      90      00:00:52
```

```
Gateway      Distance      Last Update  
10.0.12.2      90      00:01:02
```

Distance: internal 90 external 170

In the output above, you can see that R1's EIGRP process is configured as passive on the interface toward R2 (FastEthernet0/0). What this means is that the network configured on that specific interface is advertised into EIGRP but the router cannot form EIGRP adjacencies on that interface. Let's fix this:

```
R1#conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R1(config)#router eigrp 100
```

```
R1(config-router)#no passive-interface fa0/0
```

```
*Mar 1 02:27:21.871: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor  
10.0.12.2 (FastEthernet0/0) is up: new adjacency
```

Another thing you can inspect in the show ip protocols command is the unequal-cost load balancing configuration. As you know, as opposed to OSPF, EIGRP allows for such a configuration, where routes with different metrics may be installed in the routing table at the same time. They will route traffic in proportion to their metric, but in order for this to happen, a variance higher than 1 has to be configured on the router.

```
R1#sho ip protocols
```

Routing Protocol is eigrp 100

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 2

[output omitted]

In the output above, you can see that the router is configured with a variance of 2, which means that the EIGRP process will install in the routing table routes with a metric up to double in value when compared to the metric of the best route.

Another common issue in hub-and-spoke topologies is represented by routes advertised by a spoke that do not reach other spokes in the EIGRP domain. The reason for this might be split horizon being enabled on the hub interface. This prevents prefixes being received on the specific interface from being advertised back on that interface toward the other spokes. In order to fix this, you have to disable split horizon functionality on the hub interface. This is done on an interface level:

```
Hub(config)#int Serial0  
Hub(config-if)#no ip split-horizon eigrp 100
```

Another solution to this issue might be to configure the hub-and-spoke connections as point-to-point, using different dedicated IP subnets on each of them.

Troubleshooting WAN Connectivity

When investigating any reported problems in a WAN, there are several commands that you can use to locate the source of the problem. You will not be able to call your service provider to report a problem unless you can actually show that it is not your equipment that is at fault, rather it's theirs. Just as when you report an issue with your home network, your service provider help desk will want to blame you for any issues.

The first command used by most network engineers is ping. Try to ping the faulty host and ping the default gateway from the network segment. This will tell you whether you have IP connectivity. Traceroute can also be used to identify where the packet may be failing.

The `show ip interface brief` command gives a snapshot of all of your interfaces and whether they are up or down, which is a quick and easy way of seeing what is happening. If the interfaces are operating correctly at layers 1 and 2, they should be Status up and Protocol up. The output below shows that they are administratively down:

```
RouterB#show ip interface brief  
Interface    IP-Address  OK? Method Status      Protocol  
Fa0          unassigned  YES unset administratively down down  
Serial0/0    192.168.1.2 YES manual up        up  
Serial1/0    unassigned  YES unset administratively down down
```

The `show interface X` command provides a lot of useful information, such as statistics and the configuration. The default bandwidth on Serial interfaces is 1544 Kbits or 1.544 Mbps. This is purely for routing protocols to decide which path to take. It does not

affect the physical speed of the interface at all.

RouterB#show interface Serial0/0

Serial0 is up, line protocol is up

Hardware is HD64570

Internet address is 192.168.1.2/24

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

reliability 255/255, txload 1/255, rxload 1/255 **i Transmission and receive load**

Encapsulation HDLC, Loopback not set **i Encapsulation settings**

Keepalive set (10 sec)

Last input 00:00:07, output 00:00:00, output hang never

Last clearing of show interface counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: weighted fair

Output queue: 0/1000/64/0 (size/max total/threshold/drops)

Conversations 0/1/256 (active/max active/max total)

Reserved Conversations 0/0 (allocated/max allocated)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

18 packets input, 2813 bytes, 0 no buffer **i Traffic statistics**

Received 18 broadcasts, 0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

19 packets output, 1728 bytes, 0 underruns

0 output errors, 0 collisions, 4 interface resets **i Interface resets**

0 output buffer failures, 0 output buffers swapped out

2 carrier transitions

DCD=up DSR=up DTR=up RTS=up CTS=up

You should always issue a clear counters command when troubleshooting interface problems. Interface counters are cumulative, so you may be looking at errors that occurred some time ago.

RouterA#clear counters

Clear show interface counters on all interfaces [confirm]

Another useful command is debug serial interface.

Counters can be cleared on any specific interface you are troubleshooting; you don't have to reset the statistics on all the interfaces:

R1#clear counters FastEthernet0/0

Clear "show interface" counters on this interface [confirm]



RouterB#debug serial interface

Serial network interface debugging is on

RouterB#

02:33:41: Serial0: HDLC myseq 39, mineseen 39*, yourseen 39, line up

02:33:51: Serial0: HDLC myseq 40, mineseen 40*, yourseen 40, line up

02:34:01: Serial0: HDLC myseq 41, mineseen 41*, yourseen 41, line up

show controllers serial x is a very useful command and has been discussed in detail.

I've already covered specific WAN troubleshooting in relevant sections of this guide, such as Frame Relay.

Troubleshooting EtherChannels

In the CCNA exam, you may be presented with an EtherChannel diagram or the output of show commands and asked what the probable issue is. Ensure that you are familiar with which settings PAgP and LACP require in order to form an EtherChannel.

Remember that each port in the EtherChannel must have matching settings in order for the EtherChannel to form. If one site is full-duplex and the other half-duplex, for example, then the channel can't form.

When you follow my configuration commands for all the examples in the EtherChannel chapter, ensure that you check all the configurations and test out the relevant show commands. Also, try each of the commands below and note what each one reveals. We covered these in the theory section and EtherChannel labs.

Switch#show etherchannel ?

<1-64> Channel group number

detail Detail information

load-balance Load-balance/frame-distribution scheme among ports in
port-channel

port Port information
port-channel Port-channel information
protocol protocol enabled
summary One-line summary per channel-group
| Output modifiers
<cr>

End of Chapter Questions

Please visit www.howtonetwork.com/ccnasimplified to take the free Chapter 17 exam.

Chapter 18 — Advanced Labs

I've moved all the advanced labs online to save space and so I can add more than just the original three. They are all free at the URL below:

www.howtonetwork.com/ccnasimplified