CISCO.

Network Fundamentals

CCNA Exploration Companion Guide

Mark A. Dye • Rick McDonald • Antoon W. Rufi

Cisco | Networking Academy®
Mind Wide Open™

Network Fundamentals

CCNA Exploration Companion Guide

Mark A. Dye
Rick McDonald
Antoon W. Rufi

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

Network Fundamentals, CCNA Exploration Companion Guide

Mark A. Dye ▪ Rick McDonald ▪ Antoon W. Rufi

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

Printed in the United States of America

Tenth Printing February 2011

Library of Congress Cataloging-in-Publication Data

Dye, Mark A.

Network fundamentals : CCNA exploration companion guide/Mark A. Dye, Rick McDonald, Antoon W. Rufi.

p. cm.

ISBN-13: 978-1-58713-208-7 (hbk. : CD-ROM)

ISBN-10: 1-58713-208-7 (hbk. : CD-ROM) 1. Electronic data processing personnel—Certification. 2. Computer networks—Examinations—Study guides. I. McDonald, Rick. II. Rufi, Antoon W. III. Title. IV. Title: CCNA exploration companion guide.

QA76.3.D94 2007

004.6—dc22

2007038852

ISBN-13: 978-1-58713-208-7

ISBN-10: 1-58713-208-7

Trademark Acknowledgments

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

Publisher
Paul Boger

Associate Publisher
Dave Dusthimer

Cisco Representative
Anthony Wolfenden

Cisco Press Program Manager
Jeff Brady

Executive Editor
Mary Beth Ray

Managing Editor
Patrick Kanouse

Development Editor
Dayna Isley

Project Editor
Tonya Simpson

Copy Editor
Written Elegance, Inc.

Technical Editors
Martin S. Anderson, Gerlinde Brady

Editorial Assistant
Vanessa Evans

Book Designer
Louisa Adair

Composition
Mark Shirar

Indexer
Tim Wright

Proofreader
Gill Editorial Services

Warning and Disclaimer

This book is designed to provide information about the Cisco Network Fundamentals CCNA Exploration course. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

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

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

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:
U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States please contact:
International Sales international@pearsoned.com

Feedback Information

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

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

We greatly appreciate your assistance.

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Authors

Mark A. Dye was the technology manager and training manager for the Bevill Center at Gadsden State Community College, where he also managed and taught in the Cisco Academy program. He now works full time as an assessment and curriculum developer with Cisco. Mark also has maintained a private information technology consulting business since 1985. Mark's 30+-year career has included roles as biomedical instrumentation technician, field service engineer, customer service supervisor, network engineer, and instructor.

Rick McDonald teaches computer and networking courses at the University of Alaska Southeast in Ketchikan, Alaska. He is developing methods for delivering hands-on training via distance in Alaska using web-conferencing and NETLAB tools. Rick worked in the airline industry for several years before returning to full-time teaching. He taught CCNA and CCNP courses in the Cisco Networking Academy in North Carolina and was a CCNA instructor trainer.

Antoon "Tony" W. Rufi currently is the associate dean of computer and information science for all the ECPI College of Technology campuses. He also teaches the Cisco Networking Academy CCNA, CCNP, Network Security, Fundamentals of Wireless LAN, and IP Telephony curricula. Before becoming an instructor for ECPI, he spent almost 30 years in the United States Air Force, working on numerous electronic projects and computer programs.

About the Technical Reviewers

Martin S. Anderson is an instructor and program director for computer science technology at BGSU Firelands. BGSU Firelands, located in Huron, Ohio, is a regional branch college of Bowling Green State University. He has more than 30 years of experience in network computers, beginning with his family's small business in the mid-1970s. He has taught the CCNA curriculum at BGSU Firelands since 2002.

Gerlinde Brady has been teaching Cisco CCNA and CCNP courses at Cabrillo College, a Cisco Regional Networking Academy, since 1999. She holds a masters degree in education from the University of Hannover, Germany, and a masters degree in translation (English/German) from the Monterey Institute of International Studies. Her IT industry experience includes LAN design, network administration, technical support, and training.

Dedications

To my wonderful wife of more than 30 years. Frances, your zeal for life and compassion for people put a spark in this nerd's life. To my children, Jacob, Jonathan, Joseph, Jordan, Julianna, and Johannah, who share the many adventures of our lives. Also, to the young ladies that my sons have chosen to be my daughters, Barbie and Morgan. Finally, to my grandson Jacob Aiden; there truly is a reason why they are called grand. —*Mark Dye*

To my mother, Fran McDonald, who is an inspirational life-long learner. —*Rick McDonald*

To my wife, Linda. Without her understanding and support, I would not have been able to spend the amount of time required to produce something like this. —*Tony Rufi*

Acknowledgments

From Mark Dye:

I want to thank Mary Beth Ray and Dayna Isley with Cisco Press, whose unending patience made this book possible. I also want to thank the technical editors, Marty Anderson and Gerlinde Brady, for their insight. Additionally, I want to thank the other authors, Rick McDonald and Tony Rufi, who quietly and professionally made their portions of the book come together.

I want to say a special thanks to Telethia Wills with Cisco, who I have worked with for a number of years. Telethia has guided me through many different projects and allowed me to work with so many wonderful people.

From Rick McDonald:

I wish to thank my two talented coauthors, Mark and Tony, who were generous with their time and knowledge. They were under heavy workloads when the project began and held up throughout. Mary Beth did me a great kindness in bringing me to the table with you.

Gerlinde Brady and Marty Anderson did an excellent job as technical editors for my contributions. I am aware of the time and effort it took to cover so much so thoroughly, and your suggestions and corrections have had a direct impact on the quality of the content.

Mary Beth Ray has been the calming and guiding force behind the publication of this book. I wish to thank her for her vision and for her encouragement when I lost sight of it. Mary Beth's ability to guide the project and adapt to the ever-changing needs of students is to be loudly applauded. Mary Beth, thank you for your confidence and patience. You had to use both quite generously, and I appreciate it.

Dayna Isley once again amazed me with her ability to spot errors and clarify the unnecessarily complex sentences I would submit to her. There were times on the phone I was sure I could hear her eyes rolling, but she kept patiently guiding me through the processes of publishing with humor and kindness.

I wish to also thank Sarah Strickling at the University of Alaska Southeast for her helpful feedback and suggestions. Thanks also to Chris Lott and Christen Bouffard of the Center for Distance Education at the University of Alaska at Fairbanks for helping me understand some of the many ways imagination and technology are changing the way people think, learn, and work. I have tried to pass some of those ideas on within this book.

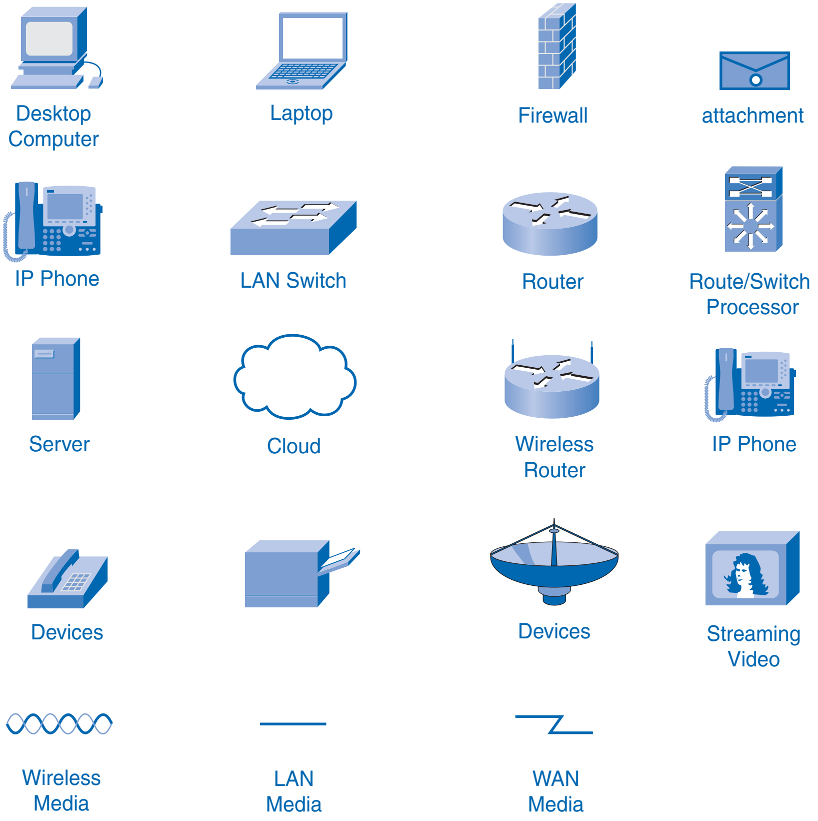
From Tony Rufi:

I would like to thank my coauthors, Mark Dye and Rick McDonald, for helping make writing this book such a joy. I would also like to thank ECPI College of Technology for all the support through the years, especially in reference to my quest for Cisco knowledge.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

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

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

Introduction

Cisco Networking Academy is a comprehensive e-learning program that delivers information technology skills to students around the world. The Cisco CCNA Exploration curriculum consists of four courses that provide a comprehensive overview of networking, from fundamentals to advanced applications and services. The curriculum emphasizes theoretical concepts and practical application, while providing opportunities for you to gain the skills and hands-on experience needed to design, install, operate, and maintain networks in small- to medium-size businesses, as well as enterprise and service provider environments. The Network Fundamentals course is the first course in the curriculum and is based on a top-down approach to networking.

Network Fundamentals, CCNA Exploration Companion Guide is the official supplemental textbook for the first course in v4.x of the CCNA Exploration online curriculum of the Networking Academy. As a textbook, this book provides a ready reference to explain the same networking concepts, technologies, protocols, and devices as the online curriculum.

This book emphasizes key topics, terms, and activities and provides many alternate explanations and examples as compared with the course. You can use the online curriculum as directed by your instructor and then use this Companion Guide's study tools to help solidify your understanding of all the topics.

Goal of This Book

First and foremost, by providing a fresh, complementary perspective of the online content, this book helps you learn all the required materials of the first course in the Networking Academy CCNA Exploration curriculum. As a secondary goal, individuals who do not always have Internet access can use this text as a mobile replacement for the online curriculum. In those cases, you can read the appropriate sections of this book, as directed by your instructor, and learn the topics that appear in the online curriculum. Another secondary goal of this book is to serve as your offline study material to help prepare you for the CCNA exam.

Audience for This Book

This book's main audience is anyone taking the first CCNA Exploration course of the Networking Academy curriculum. Many Networking Academies use this textbook as a required tool in the course, while other Networking Academies recommend the Companion Guides as an additional source of study and practice materials.

Book Features

The educational features of this book focus on supporting topic coverage, readability, and practice of the course material to facilitate your full understanding of the course material.

Topic Coverage

The following features give you a thorough overview of the topics covered in each chapter so that you can make constructive use of your study time:

- **Objectives:** Listed at the beginning of each chapter, the objectives reference the core concepts covered in the chapter. The objectives match the objectives stated in the corresponding chapters of the online curriculum; however, the question format in the Companion Guide encourages you to think about finding the answers as you read the chapter.

How To

- **"How-to" feature:** When this book covers a set of steps that you need to perform for certain tasks, the text lists the steps as a how-to list. When you are studying, the icon helps you easily refer to this feature as you skim through the book.
- **Notes, tips, cautions, and warnings:** These are short sidebars that point out interesting facts, timesaving methods, and important safety issues.
- **Chapter summaries:** At the end of each chapter is a summary of the chapter's key concepts. It provides a synopsis of the chapter and serves as a study aid.

Readability

The authors have compiled, edited, and in some cases rewritten the material so that it has a more conversational tone that follows a consistent and accessible reading level. In addition, the following features have been updated to assist your understanding of the networking vocabulary:

- **Key terms:** Each chapter begins with a list of key terms, along with a page-number reference from inside the chapter. The terms are listed in the order in which they are explained in the chapter. This handy reference allows you to find a term, flip to the page where the term appears, and see the term used in context. The Glossary defines all the key terms.
- **Glossary:** This book contains an all-new Glossary with more than 250 terms.

Practice

Practice makes perfect. This new Companion Guide offers you ample opportunities to put what you learn to practice. You will find the following features valuable and effective in reinforcing the instruction that you receive:

- **Check Your Understanding questions and answer key:** Updated review questions are presented at the end of each chapter as a self-assessment tool. These questions match the style of questions that you see in the online course. Appendix A, "Check Your Understanding and Challenge Questions Answer Key," provides an answer key to all the questions and includes an explanation of each answer.
- **(NEW) Challenge questions and activities:** Additional—and more challenging—review questions and activities are presented at the end of chapters. These questions are purposefully designed to be similar to the more complex styles of questions you might see on the CCNA exam. This section might also include activities to help prepare you for the exams. Appendix A provides the answers.

Packet Tracer
□ Activity

- **Packet Tracer activities:** Interspersed throughout the chapters you'll find many activities to work with the Cisco Packet Tracer tool. Packet Tracer allows you to create networks, visualize how packets flow in the network, and use basic testing tools to determine whether the network would work. When you see this icon, you can use Packet Tracer with the listed file to perform a task suggested in this book. The activity files are available on this book's CD-ROM; Packet Tracer software, however, is available through the Academy Connection website. Ask your instructor for access to Packet Tracer.

Labs and Study Guide

The supplementary book *Network Fundamentals, CCNA Exploration Labs and Study Guide*, by Cisco Press (ISBN: 1-58713-2036), contains all the labs from the curriculum plus additional challenge labs and study guide material. At the end of each chapter of this Companion Guide, icons indicate what hands-on activities, labs, and Packet Tracer activities are available in the Labs and Study Guide.

- **Lab and Activity references:** This icon notes the hands-on labs and other activities created for this chapter in the online curriculum. Within *Network Fundamentals, CCNA Exploration Labs and Study Guide*, you will also find additional labs and study guide material created by the authors of that book.

Packet Tracer
□ Companion

- **(NEW) Packet Tracer Companion activities:** Many of the hands-on labs include Packet Tracer Companion activities, where you can use Packet Tracer to complete a simulation of the lab. Look for this icon in *Network Fundamentals, CCNA Exploration Labs and Study Guide* for hands-on labs that have a Packet Tracer Companion.

Packet Tracer
□ Challenge

- **(NEW) Packet Tracer Skills Integration Challenge activities:** These activities require you to pull together several skills learned from the chapter to successfully complete one comprehensive exercise. Look for this icon in *Network Fundamentals, CCNA Exploration Labs and Study Guide* for instructions on how to perform the Packet Tracer Skills Integration Challenge for this chapter.

A Word About Packet Tracer Software and Activities

Packet Tracer is a self-paced, visual, interactive teaching and learning tool developed by Cisco. Lab activities are an important part of networking education. However, lab equipment can be a scarce resource. Packet Tracer provides a visual simulation of equipment and network processes to offset the challenge of limited equipment. Students can spend as much time as they like completing standard lab exercises through Packet Tracer, and have the option to work from home. Although Packet Tracer is not a substitute for real equipment, it allows students to practice using a command-line interface. This "e-doing" capability is a fundamental component of learning how to configure routers and switches from the command line.

Packet Tracer v4.x is available only to Cisco Networking Academies through the Academy Connection website. Ask your instructor for access to Packet Tracer.

The course includes essentially three different types of Packet Tracer activities. This book uses an icon system to indicate which type of Packet Tracer activity is available to you. The icons are intended to give you a sense of the purpose of the activity and the amount of time you need to allot to complete it. The three types of Packet Tracer activities follow:

Packet Tracer
□ Activity

- **Packet Tracer Activity:** This icon identifies straightforward exercises interspersed throughout the chapters where you can practice or visualize a specific topic. The activity files for these exercises are available on the book's CD-ROM. These activities take less time to complete than the Packet Tracer Companion and Challenge activities.

Packet Tracer
□ Companion

- **Packet Tracer Companion:** This icon identifies exercises that correspond to the hands-on labs of the course. You can use Packet Tracer to complete a simulation of the hands-on lab or complete a similar "lab." The Companion Guide points these out at the end of each chapter, but look for this icon and the associated exercise file in *Network Fundamentals, CCNA Exploration Labs and Study Guide* for hands-on labs that have a Packet Tracer Companion.

Packet Tracer
□ Challenge

- **Packet Tracer Skills Integration Challenge:** This icon identifies activities that require you to pull together several skills learned from the chapter to successfully complete one comprehensive exercise. The Companion Guide points these out at the end of each chapter, but look for this icon in *Network Fundamentals, CCNA Exploration Labs and Study Guide* for instructions on how to perform the Packet Tracer Skills Integration Challenge for this chapter.

How This Book Is Organized

This book covers the major topics in the same sequence as the online curriculum for the CCNA Exploration Network Fundamentals course. The online curriculum has 11 chapters for Network Fundamentals, so this book has 11 chapters with the same names and numbers as the online course chapters.

To make it easier to use this book as a companion to the course, the major topic headings in each chapter match, with just a few exceptions, the major sections of the online course chapters. However, the Companion Guide presents many topics in slightly different order inside each major heading. Additionally, the book occasionally uses different examples than the course. As a result, students get more detailed explanations, a second set of examples, and different sequences of individual topics, all to aid the learning process. This new design, based on research into the needs of the Networking Academies, helps typical students lock in their understanding of all the course topics.

Chapters and Topics

The book has 11 chapters, as follows:

- **Chapter 1, "Living in a Network-Centric World,"** presents the basics of communication and describes how networks support the way we live. This chapter introduces the concepts of data networks, scalability, quality of service (QoS), security issues, network collaboration tools, and Packet Tracer.
- **Chapter 2, "Communicating over the Network,"** introduces the devices, media, and protocols that enable network communication. This chapter introduces the OSI and TCP/IP models, the importance of addressing and naming schemes, and the process of data encapsulation. You also learn about the tools designed to analyze and simulate network functionality, such as Wireshark.
- **Chapter 3, "Application Layer Functionality and Protocols,"** introduces you to the top network model layer, the application layer. In this context, you will explore the interaction of protocols, services, and applications, with a focus on HTTP, DNS, DHCP, SMTP/POP, Telnet, and FTP.
- **Chapter 4, "OSI Transport Layer,"** focuses on the role of the transport layer as it provides the end-to-end transfer of data between applications. You learn how TCP and UDP apply to common applications.
- **Chapter 5, "OSI Network Layer,"** introduces the concepts of routing packets from a device on one network to a device on a different network. You learn important concepts related to addressing, path determination, data packets, and IP.

- **Chapter 6, “Addressing the Network: IPv4,”** focuses on network addressing in detail and describes how to use the address mask, or prefix length, to determine the number of subnetworks and hosts in a network. This chapter also introduces Internet Control Message Protocol (ICMP) tools, such as ping and trace.
- **Chapter 7, “OSI Data Link Layer,”** discusses how the OSI data link layer prepares network layer packets for transmission and controls access to the physical media. This chapter includes a description of the encapsulation processes that occur as data travels across the LAN and the WAN.
- **Chapter 8, “OSI Physical Layer,”** explores the functions, standards, and protocols associated with the physical layer (Layer 1). You discover how data sends signals and is encoded for travel across the network. You learn about bandwidth and also about the types of media and their associated connectors.
- **Chapter 9, “Ethernet,”** examines the technologies and operation of Ethernet. Topics include the evolution of Ethernet technologies, MAC, and Address Resolution Protocol (ARP).
- **Chapter 10, “Planning and Cabling Networks,”** focuses on designing and cabling a network. You will apply the knowledge and skills developed in the previous chapters to determine which cables to use, how to connect devices, and how to develop an addressing and testing scheme.
- **Chapter 11, “Configuring and Testing Your Network,”** describes how to connect and configure a small network using basic Cisco IOS commands for routers and switches.

This book also includes the following:

- **Appendix, “Check Your Understanding and Challenge Questions Answer Key,”** provides the answers to the Check Your Understanding questions that you find at the end of each chapter. It also includes answers for the challenge questions and activities that conclude most chapters.
- The **Glossary** provides a compiled list of all the key terms that appear throughout this book.

About the CD-ROM

The CD-ROM included with this book provides many useful tools and information to support your education:

Packet Tracer
□ **Activity**

- **Packet Tracer Activity files:** These are files to work through the Packet Tracer activities referenced throughout the book, as indicated by the Packet Tracer Activity icon.
- **Other files**: A couple files referenced in this book are on the accompanying CD-ROM:

  VLSM_Subnetting_Chart.pdf

  Exploration_Supplement_Structured_Cabling.pdf
- **Taking Notes:** This section includes a .txt file of the chapter objectives to serve as a general outline of the key topics of which you need to take note. The practice of taking clear, consistent notes is an important skill not only for learning and studying the material but for on-the-job success as well. Also included in this section is “A Guide to Using a Networker’s Journal” PDF booklet providing important insight into the value of the practice of using and organizing a professional journal and some best practices on what, and what not, to take note of in your journal.
- **IT Career Information:** This section includes a Student Guide to applying the toolkit approach to your career development. Learn more about entering the world of Information Technology as a career by reading two informational chapters excerpted from *The IT Career Builder’s Toolkit:* “Information Technology: A Great Career” and “Breaking into IT.”
- **Lifelong Learning in Networking:** As you embark on a technology career, you will notice that it is ever changing and evolving. This career path provides exciting opportunities to learn new technologies and their applications. Cisco Press is one of the key resources to plug into on your quest for knowledge. This section of the CD-ROM provides an orientation to the information available to you and tips on how to tap into these resources for lifelong learning.

CHAPTER 3

Application Layer Functionality and Protocols

Objectives

Upon completion of this chapter, you will be able to answer the following questions:

- How do the functions of the three upper OSI model layers provide network services to end-user applications?
- How do the TCP/IP application layer protocols provide the services specified by the upper layers of the OSI model?
- How do people use the application layer to communicate across the information network?
- What are the functions of well-known TCP/IP applications, such as the World Wide Web and e-mail, and their related services (HTTP, DNS, DHCP, STMP/POP, and Telnet)?
- What are the file-sharing processes that use peer-to-peer applications and the Gnutella protocol?
- How do protocols ensure that services running on one kind of device can send to and receive from many different network devices?
- How can you use network analysis tools to examine and explain how common user applications work?

Key Terms

This chapter uses the following key terms. You can find the definitions in the Glossary.

data page 67

source device page 67

Domain Name System (DNS) page 68

Request for Comments (RFC) page 68

syntax page 70

session page 71

client page 72

server page 72

daemon page 73

peer page 75

scheme page 76

IP address page 77

domain name page 77

network address page 78

resource record page 78

DNS resolver page 78

nslookup page 78

query page 78

cache page 79

authoritative page 81

plug-in page 82

HTTP page 82

distributed page 82

collaborative page 82

encryption page 82

Post Office Protocol (POP) page 83

Simple Mail Transfer Protocol (SMTP) page 83

Mail User Agent (MUA) page 83

spam page 85

gateway page 85

Dynamic Host Configuration Protocol (DHCP) page 87

subnet mask page 87

broadcast page 88

Server Message Block (SMB) page 89

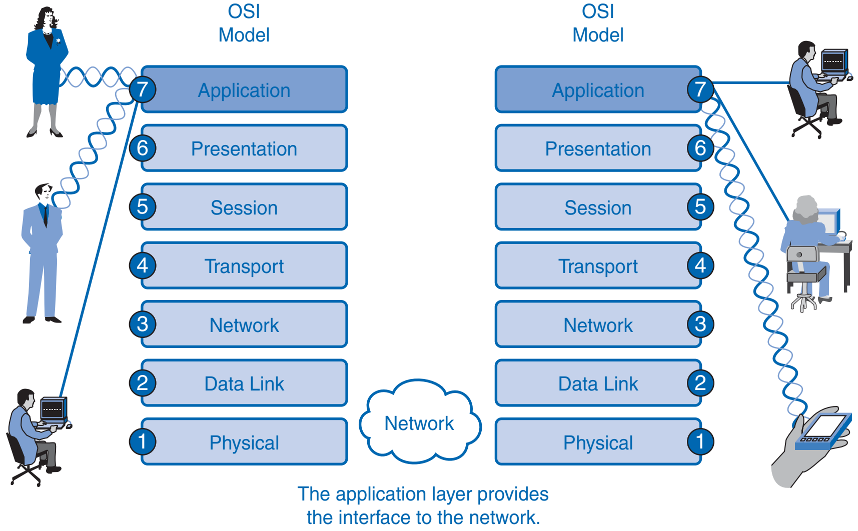
UNIX page 89

Interpret as Command (IAC) page 91

The world experiences the Internet through the use of the World Wide Web, e-mail, and file-sharing programs. These applications, as well as others, provide the human interface to the underlying network, allowing you to send and receive information with relative ease. Most of the applications are intuitive; they can be accessed and used without the need to know how they work. As you continue to study the world of networking, it becomes more important to know how an application is able to format, transmit, and interpret messages that are sent and received across the network.

Visualizing the mechanisms that enable communication across the network is made easier if you use the layered framework of the Open System Interconnection (OSI) model. Figure 3-1 depicts that framework. The OSI model is a seven-layer model, designed to help explain the flow of information from layer to layer.

Figure 3-1 Interfacing Human and Data Networks



This chapter focuses on the role of Layer 7, the application layer, and its components: applications, services, and protocols. You explore how these three elements make the robust communication across the information network possible.

Applications: The Interface Between the Networks

This section introduces two important concepts:

- **Application layer:** The application layer of the OSI model provides the first step of getting data onto the network.

- **Application software:** Applications are the software programs used by people to communicate over the network. Examples of application software, including HTTP, FTP, e-mail, and others, are used to explain the differences between these two concepts.

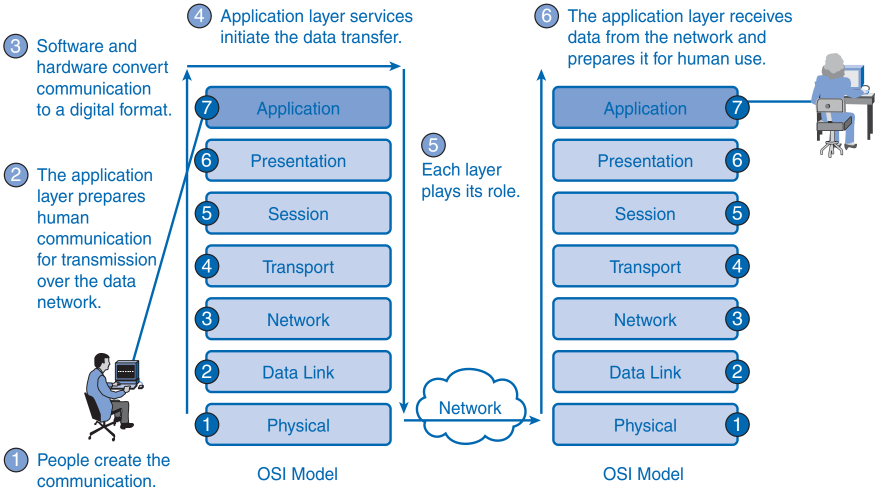
OSI and TCP/IP Model

The OSI reference model is a layered, abstract representation created as a guideline for network protocol design and instruction. The OSI model divides the networking process into seven logical layers, each of which has unique functionality and to which are assigned specific services and protocols.

In the OSI model, information is passed from one layer to the next, starting at the application layer on the transmitting host and proceeding down the hierarchy to the physical layer, then passing over the communications channel to the destination host, where the information proceeds back up the hierarchy, ending at the application layer. Figure 3-2 depicts the steps in this process. The following explains the six steps:

1. People create the communication.
2. The application layer prepares human communication for transmission over the data network.
3. Software and hardware convert communication to a digital format.
4. Application layer services initiate the data transfer.
5. Each layer plays its role. The OSI layers encapsulate data down the stack. Encapsulated data travels across the media to the destination. OSI layers at the destination unencapsulate the data up the stack.
6. The application layer receives data from the network and prepares it for human use.

Figure 3-2 OSI Encapsulation Process

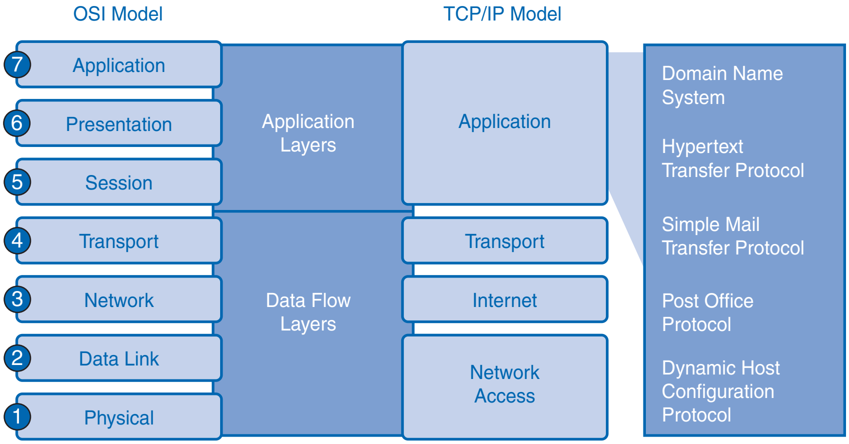

3 Software and hardware convert communication to a digital format.
2 The application layer prepares human communication for transmission over the data network.
1 People create the communication.
4 Application layer services initiate the data transfer.
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
OSI Model
5 Each layer plays its role.
Network
6 The application layer receives data from the network and prepares it for human use.
Application 7
Presentation 6
Session 5
Transport 4
Network 3
Data Link 2
Physical 1
OSI Model

The application layer, Layer 7, is the top layer of both the OSI and TCP/IP models. (Refer to the section "Protocol and Reference Models" in Chapter 2, "Communicating over the Network," for more information about the TCP/IP model.) Layer 7 provides the interface between the applications you use to communicate and the underlying network over which your messages are transmitted. Application layer protocols are used to exchange ***data*** between programs running on the source and destination hosts. There are many application layer protocols, and new protocols are always being developed. (Refer to the section "User Applications, Services, and Application Layer Protocols," later in this chapter, for examples.)

Although the TCP/IP protocol suite was developed prior to the definition of the OSI model, the functionality of the TCP/IP application layer protocols fits roughly into the framework of the top three layers of the OSI model: application, presentation, and session.

Most applications, such as web browsers or e-mail clients, incorporate functionality of the OSI Layers 5, 6, and 7. A comparison of the OSI and TCP/IP model is shown in Figure 3-3.

Figure 3-3 OSI and TCP/IP Model



Most TCP/IP application layer protocols were developed before the emergence of personal computers, GUIs, and multimedia objects. As a result, these protocols implement little of the functionality that is specified in the OSI model presentation and session layers. The next sections describe the OSI presentation and session layers in more detail.

Presentation Layer

The presentation layer has three primary functions:

- Coding and conversion of application layer data to ensure that data from the ***source device*** can be interpreted by the appropriate application on the destination device

- Compression of the data in a manner that can be decompressed by the destination device
- Encryption of the data for transmission and decryption of data upon receipt by the destination

Presentation layer implementations are not typically associated with a particular protocol stack. The standards for video and graphics are examples. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding.

Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF and JPEG are compression and coding standards for graphic images, and TIFF is a standard coding format for graphic images.

Session Layer

Functions at the session layer create and maintain dialogs between source and destination applications. The session layer handles the exchange of information to initiate dialogs and keep them active, and to restart sessions that are disrupted or idle for a long period of time.

TCP/IP Application Layer Protocols

The most widely known TCP/IP application layer protocols are those that provide the exchange of user information. These protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are the following:

- ***Domain Name System (DNS)*** is used to resolve Internet names to IP addresses.
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the web pages of the World Wide Web.
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices.
- File Transfer Protocol (FTP) is used for interactive file transfer between systems.

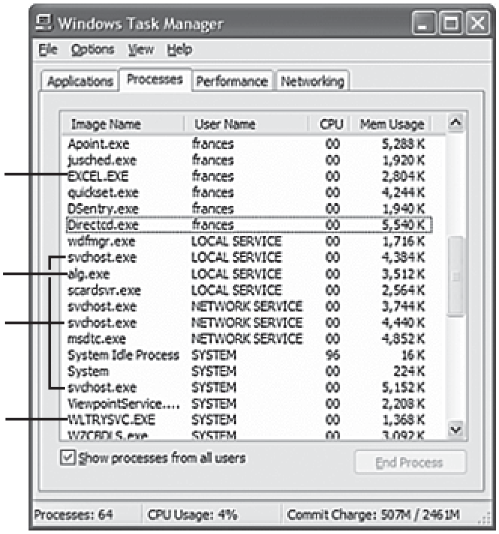
The protocols in the TCP/IP suite are generally defined by ***Requests for Comments (RFC)***. The Internet Engineering Task Force (IETF) maintains the RFCs as the standards for the TCP/IP suite.

Application Layer Software

The functions associated with the application layer protocols in both the OSI and the TCP/IP models enable the human network to interface with the underlying data network. When you open a web browser or an instant message window, an application is started, and the program is put into the device memory, where it is executed. Each executing program loaded on a device is referred to as a *process*.

Within the application layer, there are two forms of software programs or processes that provide access to the network: applications and services. This concept is shown in Figure 3-4.

Figure 3-4 Software Processes



Processes are individual software programs running concurrently.

Applications

Services

One program can be running several times, each in its own process.

System Operations

Image Name	User Name	CPU	Mem Usage
Apoint.exe	frances	00	5,288 K
jusched.exe	frances	00	1,920 K
EXCEL.EXE	frances	00	2,804 K
quickset.exe	frances	00	4,244 K
DSentry.exe	frances	00	1,940 K
Directcd.exe	frances	00	5,540 K
wdfmgr.exe	LOCAL SERVICE	00	1,716 K
svchost.exe	LOCAL SERVICE	00	4,384 K
alg.exe	LOCAL SERVICE	00	3,512 K
scardsvr.exe	LOCAL SERVICE	00	2,564 K
svchost.exe	NETWORK SERVICE	00	3,744 K
svchost.exe	NETWORK SERVICE	00	4,440 K
msdtc.exe	NETWORK SERVICE	00	4,852 K
System Idle Process	SYSTEM	96	16 K
System	SYSTEM	00	224 K
svchost.exe	SYSTEM	00	5,152 K
ViewpointService....	SYSTEM	00	2,208 K
WLTRYSVC.EXE	SYSTEM	00	1,368 K
WZCBDLS.exe	SYSTEM	00	3,092 K

☑ Show processes from all users

Processes: 64 CPU Usage: 4% Commit Charge: 507M / 2461M

Network-Aware Applications

Some end-user applications are network aware, meaning that they implement the application layer protocols and are able to communicate directly with the lower layers of the protocol stack. E-mail clients and web browsers are examples of these types of applications.

Application Layer Services

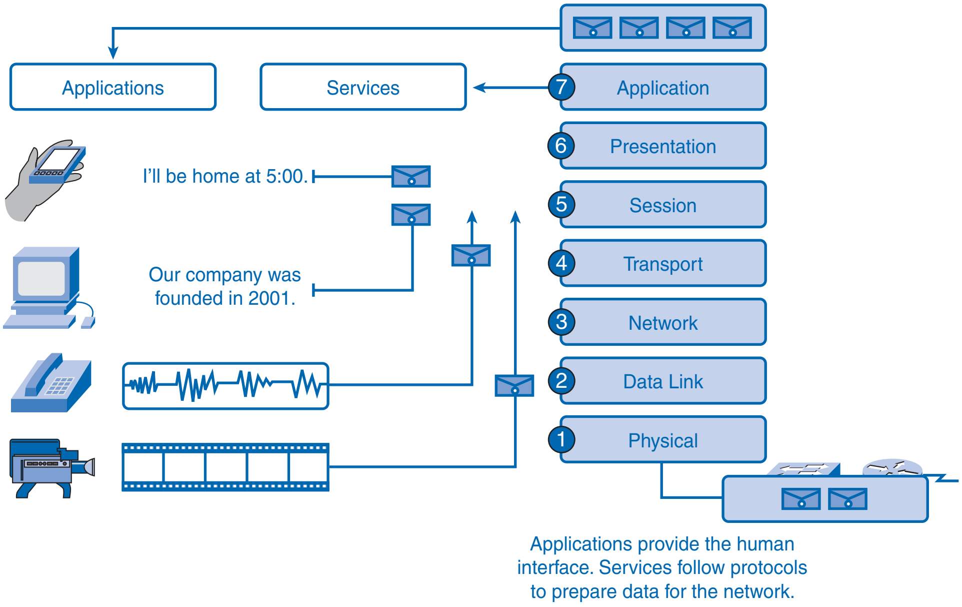
Other programs, such as file transfer or network print spooling, might need the assistance of application layer services to use network resources. Although transparent to the user, these services interface with the network and prepare the data for transfer. Different types of data—whether it is text, graphics, or video—require different network services to ensure that it is properly prepared for processing by the functions occurring at the lower layers of OSI model.

Each application or network service uses protocols that define the standards and data formats to be used. A service provides the function for doing something, and a protocol provides the rules the service uses. To understand the *function* of various network services, you need to become familiar with the underlying protocols that govern their operation.

User Applications, Services, and Application Layer Protocols

The application layer uses protocols that are implemented within applications and services. Applications provide people with a way to create messages, application layer services establish an interface to the network, and protocols provide the rules and formats that govern how data is treated, as shown in Figure 3-5. A single executable program can use all three components. For example, when discussing "Telnet," you could be referring to the Telnet application, the Telnet service, or the Telnet protocol.

Figure 3-5 Interfacing Human and Data Networks


Applications
Services
7 Application
6 Presentation
5 Session
4 Transport
3 Network
2 Data Link
1 Physical
I'll be home at 5:00.
Our company was founded in 2001.
Applications provide the human interface. Services follow protocols to prepare data for the network.

In the OSI model, applications that interact directly with people are considered to be at the top of the stack, as are the people themselves. Like all layers within the OSI model, the application layer relies on the functions of the lower layers to complete the communication process. Within the application layer, protocols specify what messages are exchanged between the source and destination hosts, the ***syntax*** of the control commands, the type and format of the data being transmitted, and the appropriate methods for error notification and recovery.

Application Layer Protocol Functions

Both the source and destination devices use application layer protocols during a communication ***session***. For the communications to be successful, the application layer protocols implemented on the source and destination host must match.

Protocols perform the following tasks:

- Establish consistent rules for exchanging data between applications and services loaded on the participating devices.
- Specify how data inside the messages is structured and the types of messages that are sent between source and destination. These messages can be requests for services, acknowledgments, data messages, status messages, or error messages.
- Define message dialogues, ensuring that a message being sent is met by the expected response and that the correct services are invoked when data transfer occurs.

Many different types of applications communicate across data networks. Therefore, application layer services must implement multiple protocols to provide the desired range of communication experiences. Each protocol has a specific purpose and contains the characteristics required to meet that purpose. The right protocol details in each layer must be followed so that the functions at one layer interface properly with the services in the lower layer.

Applications and services can also use multiple protocols in the course of a single conversation. One protocol might specify how to establish the network connection, and another might describe the process for the data transfer when the message is passed to the next lower layer.

Making Provisions for Applications and Services

When people attempt to access information on their device, whether it is a PC, laptop, PDA, cell phone, or some other device connected to a network, the data might not be physically stored on their device. If that is the case, a request to access that information must be made to the device where the data resides. The following sections cover three topics that will help you understand how the request for data can occur and how the request is filled:

- Client/server model
- Application layer services and protocols
- Peer-to-peer networking and applications

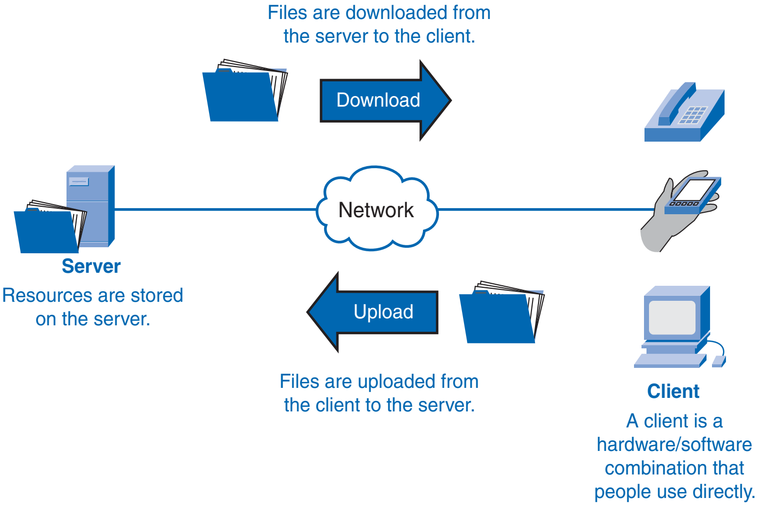
Client/Server Model

In the client/server model, the device requesting the information is called a ***client*** and the device responding to the request is called a server. Client and ***server*** processes are considered to be in the application layer. The client begins the exchange by requesting data from the server, which responds by sending one or more streams of data to the client. Application layer protocols describe the design of the requests and responses between clients and servers. In addition to the actual data transfer, this exchange can require control information, such as user authentication and the identification of a data file to be transferred.

One example of a client/server network is a corporate environment where employees use a company e-mail server to send, receive, and store e-mail. The e-mail client on an employee computer issues a request to the e-mail server for any unread mail. The server responds by sending the requested e-mail to the client.

Although data is typically described as flowing from the server to the client, some data always flows from the client to the server. Data flow can be equal in both directions or can even be greater in the direction going from the client to the server. For example, a client might transfer a file to the server for storage purposes. Data transfer from a client to a server is referred to as an *upload*, and data from a server to a client is a *download*. Figure 3-6 shows the client/server model concept.

Figure 3-6 Client/Server Model


Files are downloaded from the server to the client.
Download
Server
Resources are stored on the server.
Network
Upload
Files are uploaded from the client to the server.
Client
A client is a hardware/software combination that people use directly.

Servers

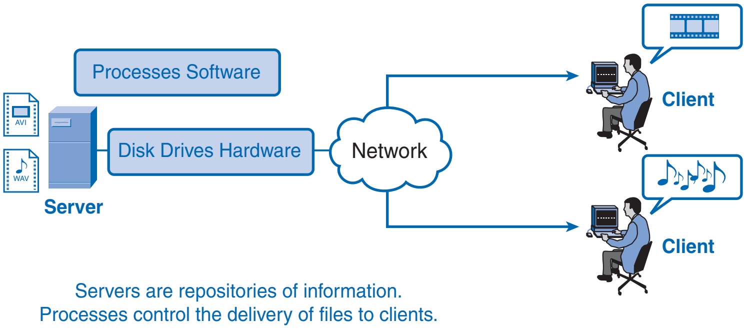
In a general networking context, any device that responds to requests from client applications is functioning as a server. A server is usually a computer that contains information to be shared with many client systems. For example, web pages, documents, databases, pictures, video, and audio files can all be stored on a server and delivered to requesting clients. In other cases, such as a network printer, the print server delivers the client print requests to the specified printer.

Different types of server applications can have different requirements for client access. Some servers can require authentication of user account information to verify whether the user has permission to access the requested data or to use a particular operation. Such servers rely on a central list of user accounts and the authorizations, or permissions (both for data access and operations), granted to each user. When using an FTP client, for example, if you request to upload data to the FTP server, you might have permission to write to your individual folder but not to read other files on the site.

In a client/server network, the server runs a service, or process, sometimes called a server ***daemon***. Like most services, daemons typically run in the background and are not under an end user's direct control. Daemons are described as "listening" for a request from a client, because they are programmed to respond whenever the server receives a request for the service provided by the daemon. When a daemon "hears" a request from a client, it exchanges appropriate messages with the client, as required by its protocol, and proceeds to send the requested data to the client in the proper format.

Figure 3-7 shows the clients requesting services from the server; specifically, one client is requesting an audio file (.wav) and the other client is requesting a video file (.avi). The server responds by sending the requested files to the clients.

Figure 3-7 Servers



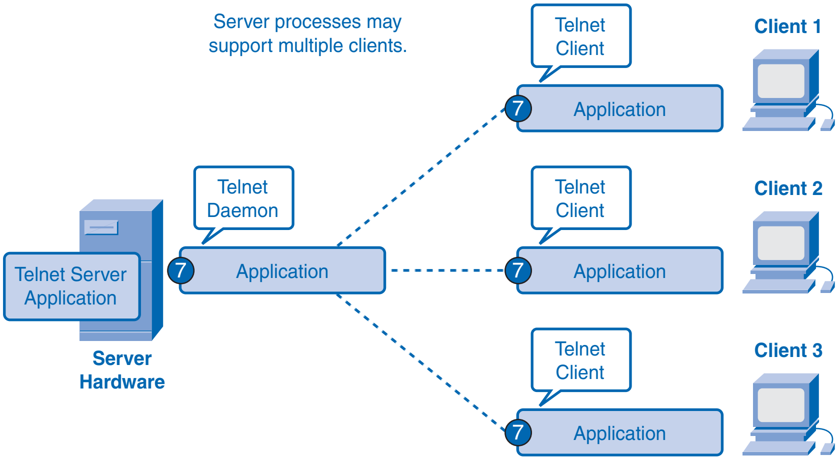
Servers are repositories of information.
Processes control the delivery of files to clients.

Application Layer Services and Protocols

A single application can employ many different supporting application layer services. Thus, what appears to the user as one request for a web page might, in fact, amount to dozens of individual requests. For each request, multiple processes can be executed. For example, the FTP requires a client to initiate a control process and a data stream process to a server.

Additionally, servers typically have multiple clients requesting information at the same time, as shown in Figure 3-8. For example, a Telnet server can have many clients requesting connections to it. These individual client requests must be handled simultaneously and separately for the network to succeed. The application layer processes and services rely on support from lower-layer functions to successfully manage the multiple conversations.

Figure 3-8 Multiple Clients' Service Requests



Packet Tracer □ Activity

Client Server Interaction (3.2.3.2)

In this activity, you will study a simple example of client/server interaction, which can serve as a model for more complex interactions later in the course. Use file e1-3232.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

Peer-to-Peer (P2P) Networking and Applications

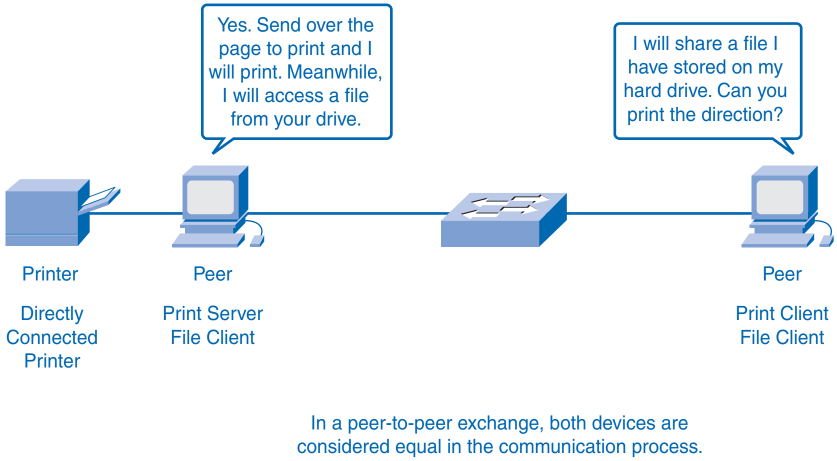
In addition to the client/server model for networking, there is a peer-to-peer (P2P) model. P2P networking involves two distinct forms: peer-to-peer network design and peer-to-peer applications. Both forms have similar features but in practice work very differently.

P2P Networks

In a peer-to-peer network, two or more computers are connected through a network and can share resources such as printers and files without having a dedicated server. Every connected end device, known as a peer, can function as either a server or a client. One computer might assume the role of server for one transaction while simultaneously serve as a client for another. The roles of client and server are set on a per-request basis, as shown in Figure

3-9. The figure shows one peer asking the other peer to provide print services, while at the same time acting as a file server that shares one of its files.

Figure 3-9 Peer-to-Peer Networking


Yes. Send over the page to print and I will print. Meanwhile, I will access a file from your drive.
I will share a file I have stored on my hard drive. Can you print the direction?
Printer
Directly Connected Printer
Peer
Print Server File Client
Peer
Print Client File Client
In a peer-to-peer exchange, both devices are considered equal in the communication process.

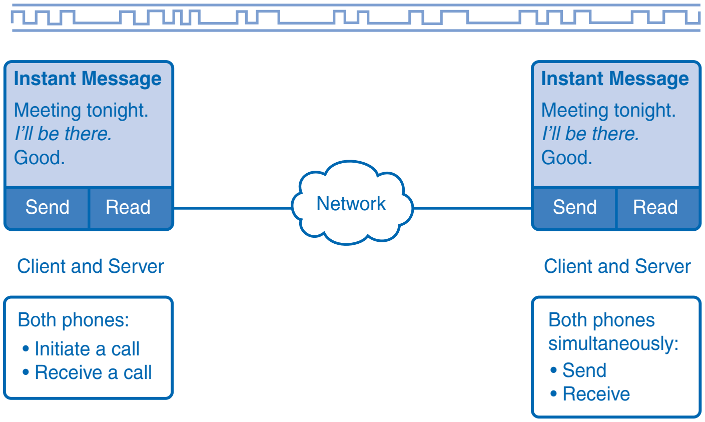
A simple home network with two connected computers sharing a printer is an example of a peer-to-peer network. Each person can set his or her computer to share files, enable networked games, or share an Internet connection. Another example of peer-to-peer network functionality is two computers connected to a large network that use software applications to share resources between one another through the network.

Unlike the client/server model, which uses dedicated servers, peer-to-peer networks decentralize the resources on a network. Instead of locating information to be shared on dedicated servers, information can be located anywhere on any connected device. Most of the current operating systems support file and print sharing without requiring additional server software. Because peer-to-peer networks usually do not use centralized user accounts, permissions, or monitors, it is difficult to enforce security and access policies in networks containing more than just a few computers. User accounts and access rights must be set individually on each ***peer*** device.

P2P Applications

A P2P application, unlike a peer-to-peer network, allows a device to act as both a client and a server within the same communication session. In this model, every client is a server and every server a client, as shown in Figure 3-10. Figure 3-10 shows two phones belonging to the same network sending an instant message. The blue lines at the top of the figure depict the digital traffic between the two phones. Both can initiate a communication and are considered equal in the communication process. However, peer-to-peer applications require that each end device provide a user interface and run a background service. When you launch a specific peer-to-peer application, it invokes the required user interface and background services. After that, the devices can communicate directly.

Figure 3-10 Peer-to-Peer Applications



A type of peer-to-peer application is the P2P hybrid system, which utilizes a centralized directory called an index server even though the files being shared are on the individual host machines. Each peer accesses the index server to get the location of a resource stored on another peer. The index server can also help connect two peers, but after they are connected, the communication takes place between the two peers without additional communication to the index server.

Peer-to-peer applications can be used on peer-to-peer networks, in client/server networks, and across the Internet.

Application Layer Protocols and Services Examples

Now that you have a better understanding of how applications provide an interface for the user and provide access to the network, you will take a look at some specific commonly used protocols.

As you will see later in this book, the transport layer uses an addressing ***scheme*** called a port number. Port numbers identify applications and application layer services that are the source and destination of data. Server programs generally use predefined port numbers that are commonly known by clients. As you examine the different TCP/IP application layer protocols and services, you will be referring to the TCP and UDP port numbers normally associated with these services. Some of these services are

- **Domain Name System (DNS):** TCP/UDP port 53
- **HTTP:** TCP port 80

- **Simple Mail Transfer Protocol (SMTP):** TCP port 25
- **Post Office Protocol (POP):** UDP port 110
- **Telnet:** TCP port 23
- **DHCP:** UDP port 67
- **FTP:** TCP ports 20 and 21

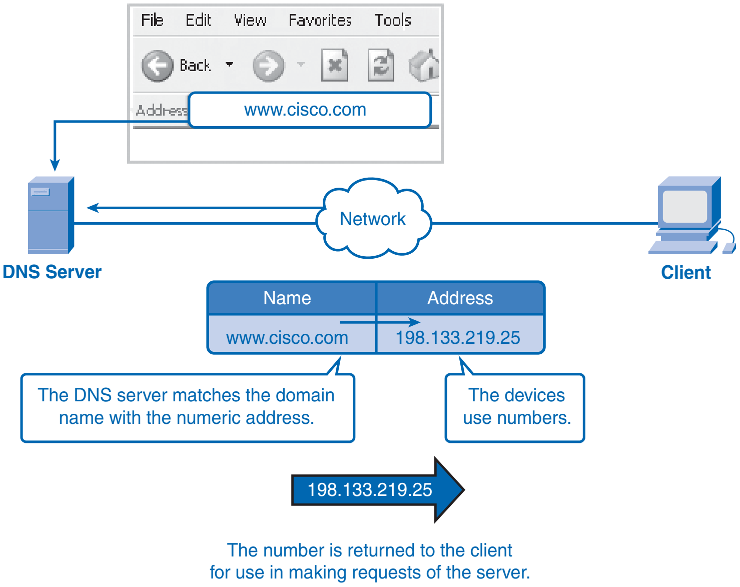
The next sections take a closer look at DNS, world wide web services, and HTTP.

DNS Services and Protocol

In data networks, devices are assigned ***IP addresses*** so that they can participate in sending and receiving messages over the network. However, most people have a hard time remembering this numeric address. Hence, domain names were created to convert the numeric address into a simple, recognizable name.

On the Internet, these domain names, such as http://www.cisco.com, are much easier for people to remember than 198.132.219.25, which, at the time of this writing, is the numeric address for this server. Also, if Cisco decides to change the numeric address, it is transparent to the user, because the ***domain name*** will remain http://www.cisco.com. The new address will simply be linked to the existing domain name and connectivity is maintained, as shown in Figure 3-11. When networks were small, it was a simple task to maintain the mapping between domain names and the addresses they represented. However, as networks began to grow and the number of devices increased, this manual system became unworkable.

Figure 3-11 Resolving DNS Addresses


File
Edit
View
Favorites
Tools
Back
Address
www.cisco.com
Network
DNS Server
Client

Name	Address
www.cisco.com	198.133.219.25

The DNS server matches the domain name with the numeric address.
The devices use numbers.
198.133.219.25
The number is returned to the client for use in making requests of the server.

DNS was created for domain name–to–address resolution for these networks. DNS uses a distributed set of servers to resolve the names associated with these numbered addresses.

How DNS Works

The DNS protocol defines an automated service that matches resource names with the required numeric ***network address***. It includes the format for queries, responses, and data formats. DNS protocol communications use a single format called a message. This message format is used for all types of client queries and server responses, error messages, and the transfer of ***resource record*** information between servers.

DNS is a client/server service; however, it differs from the other client/server services that you are examining. Whereas other services use a client that is an application (web browser, e-mail client, and so on), the DNS client runs as a service itself. The DNS client, sometimes called the ***DNS resolver***, supports name resolution for the other network applications and other services that need it.

When configuring a network device, you generally provide one or more DNS server addresses that the DNS client can use for name resolution. Usually the Internet service provider (ISP) gives you the addresses to use for the DNS servers. When a user's application requests to connect to a remote device by name, the requesting DNS client queries one of these DNS servers to resolve the name to a numeric address.

Computer operating systems also have a utility called ***nslookup*** that allows the user to manually ***query*** the name servers to resolve a given host name. You also can use this utility to troubleshoot name resolution issues and to verify the current status of the name servers.

In Example 3-1, when the **nslookup** command is issued, the default DNS server configured for your host is displayed. In this example, the DNS server is dns-sjk.cisco.com, which has an address of 171.68.226.120.

Example 3-1 nslookup Command

```
Microsoft Windows XP [Version 5.1.2600]
 Copyright 1985-2001 Microsoft Corp.

C:\> nslookup

Default Server: dns-sjk.cisco.com
Address: 171.68.226.120
>www.cisco.com
Server: dns-sj.cisco.com
Address: 171.70.168.183

Name:  www.cisco.com
Address: 198.133.219.25
```

You can then type the name of a host or domain for which you want to get the address. In the first query in Example 3-1, a query is made for www.cisco.com. The responding name server provides the address of 198.133.219.25.

Although the queries shown in Example 3-1 are only simple tests, the **nslookup** command has many options available to do extensive testing and verification of the DNS process.

Name Resolution and Caching

A DNS server provides the name resolution using the name daemon, which is often called *named* (pronounced name-dee). The DNS server acts as the phone book for the Internet: It translates human-readable computer host names, for example, http://www.cisco.com, into the IP addresses that networking equipment needs for delivering information.

The DNS server stores different types of resource records used to resolve names. These records contain the name, address, and type of record.

Some of these record types are

- **A:** An end device address
- **NS:** An authoritative name server
- **CNAME:** The canonical name (or fully qualified domain name [FQDN]) for an alias; used when multiple services have the single network address but each service has its own entry in DNS
- **MX:** Mail exchange record; maps a domain name to a list of mail exchange servers for that domain

When a client makes a query, the "named" process first looks at its own records to see whether it can resolve the name. If it is unable to resolve the name using its stored records, it contacts other servers to resolve the name.

The request can be passed along to a number of servers, which can take extra time and consume bandwidth. When a match is found and returned to the original requesting server, the server temporarily stores the numbered address that matches the name in the ***cache***.

If that same name is requested again, the first server can return the address by using the value stored in its name cache. Caching reduces both the DNS query data network traffic and the workloads of servers higher up the hierarchy. The DNS client service on Windows PCs optimizes the performance of DNS name resolution by storing previously resolved names in memory, as well. The **ipconfig/displaydns** command displays all the cached DNS entries on a Windows XP or 2000 computer system.

DNS Hierarchy

DNS uses a hierarchical system to create a name database to provide name resolution. The hierarchy looks like an inverted tree with the root at the top and branches below.

At the top of the hierarchy, the root servers maintain records about how to reach the top-level domain servers, which in turn have records that point to the secondary-level domain servers and so on.

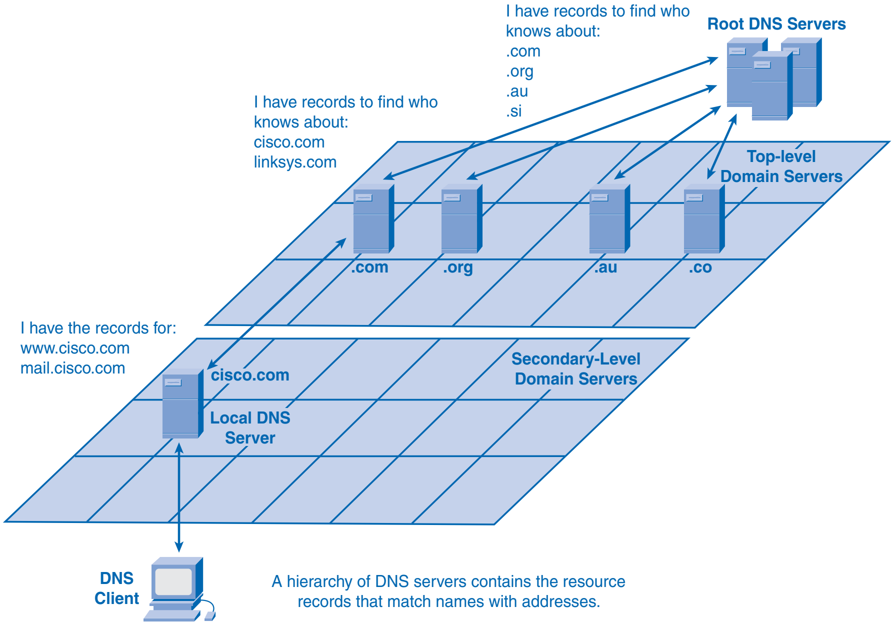
The different top-level domains represent either the type of organization or the country of origin. The following are examples of top-level domains are:

- **.au:** Australia
- **.co:** Colombia
- **.com:** A business or industry
- **.jp:** Japan
- **.org:** A nonprofit organization

After top-level domains are second-level domain names, and below them are other lower-level domains. A great example of that is the domain name http://www.cisco.netacad.net. The .net is the top-level domain, .netacad is the second-level domain, and .cisco is at the lower level.

Each domain name is a path down this inverted tree starting from the root. For example, as shown in Figure 3-12, the root DNS servers might not know exactly where the e-mail server mail.cisco.com is located, but they maintain a record for the .com domain within the top-level domain. Likewise, the servers within the .com domain might not have a record for mail.cisco.com, but they do have a record for the cisco.com secondary-level domain. The servers within the cisco.com domain have a record (an MX record to be precise) for mail.cisco.com.

Figure 3-12 DNS Server Hierarchy



DNS relies on this hierarchy of decentralized servers to store and maintain these resource records. The resource records list domain names that the server can resolve and alternative servers that can process requests. If a given server has resource records that correspond to its level in the domain hierarchy, it is said to be ***authoritative*** for those records.

For example, a name server in the cisco.netacad.net domain would not be authoritative for the mail.cisco.com record because that record is held at a higher-domain-level server, specifically the name server in the cisco.com domain.

Note

Two links to the DNS protocol RFCs are

- http://www.ietf.org/rfc/rfc1034.txt
- http://www.ietf.org/rfc/rfc1035.txt

Request for Comments (RFC) are standards documents encompassing new research, innovations, and methodologies applicable to Internet technologies. These RFCs are very technical in nature, but they can provide you with some insight to how detailed these standards really are.

WWW Service and HTTP

When a web address (or URL) is typed into a web browser, the web browser establishes a connection to the web service running on the server using HTTP. URLs and URIs (uniform resource identifiers) are the names most people associate with web addresses.

The URL http://www.cisco.com/index.html refers to a specific resource—a web page named index.html on a server identified as cisco.com.

Web browsers are the client applications computers use to connect to the World Wide Web and access resources stored on a web server. As with most server processes, the web server runs as a background service and makes different types of files available.

To access the content, web clients make connections to the server and request the desired resources. The server replies with the resources and, upon receipt, the browser interprets the data and presents it to the user.

Browsers can interpret and present many data types, such as plain text or HTML, the language in which web pages are constructed). Other types of data, however, might require another service or program, typically referred to as a ***plug-in*** or add-on. To help the browser determine what type of file it is receiving, the server specifies what kind of data the file contains.

To better understand how the web browser and web client interact, you can examine how a web page is opened in a browser. For this example, consider the URL http://www.cisco.com/web-server.htm.

First, the browser interprets the three parts of the URL:

- http: The protocol or scheme
- www.cisco.com: The server name
- web-server.htm: The specific filename requested

The browser then checks with a name server to convert http://www.cisco.com into a numeric address, which it uses to connect to the server. Using the HTTP requirements, the browser sends a GET request to the server and asks for the file web-server.htm. The server in turn sends the HTML code for this web page to the browser. Finally, the browser deciphers the HTML code and formats the page for the browser window.

HTTP, one of the protocols in the TCP/IP suite, was originally developed to publish and retrieve HTML pages and is now used for ***distributed, collaborative*** information systems. HTTP is used across the world wide web for data transfer and is one of the most used application protocols.

HTTP specifies a request/response protocol. When a client, typically a web browser, sends a request message to a server, the HTTP protocol defines the message types the client uses to request the web page and the message types the server uses to respond. The three common message types are:

- GET
- POST
- PUT

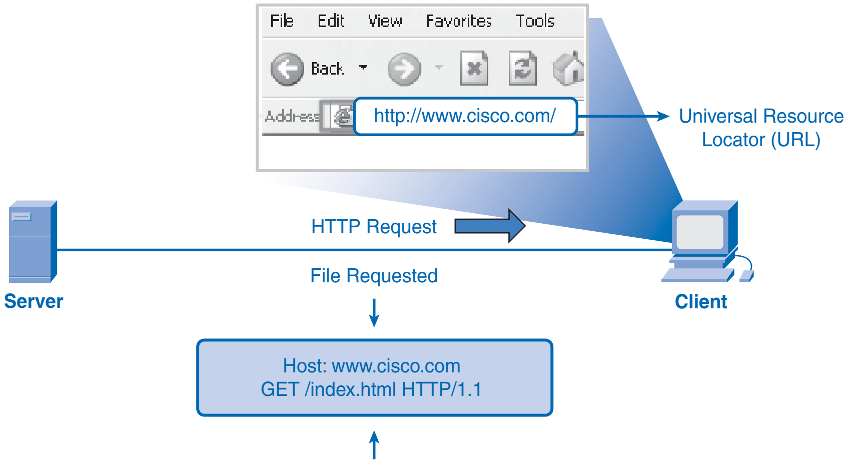
GET is a client request for data. A web browser sends the GET message to request pages from a web server. As shown in Figure 3-13, when the server receives the GET request, it responds with a status line, such as HTTP/1.1 200 OK, and a message of its own, the body of which can be the requested file, an error message, or some other information.

POST and PUT are used to send messages that upload data to the web server. For example, when the user enters data into a form embedded in a web page, POST includes the data in the message sent to the server. PUT uploads resources or content to the web server.

Although it is remarkably flexible, HTTP is not a secure protocol. The POST messages upload information to the server in plain text that can be intercepted and read. Similarly, the server responses, typically HTML pages, are unencrypted.

For secure communication across the Internet, the Secure HTTP (HTTPS) protocol is used for accessing and posting web server information. HTTPS can use authentication and ***encryption*** to secure data as it travels between the client and server. HTTPS specifies additional rules for passing data between the application layer and the transport layer.

Figure 3-13 HTTP Protocol Using GET



Host: www.cisco.com
GET /index.html HTTP/1.1

Fully Qualified Domain Name
Entering http://www.cisco.com in the address bar of a web browser generates the HTTP GET message.

Packet Tracer
□ **Activity**

Network Representations (3.3.2.3)

In this activity, you will configure DNS and HTTP services, and then study the packets that result when a web page is requested by typing a URL. Use file e1-3323.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

E-Mail Services and SMTP/POP Protocols

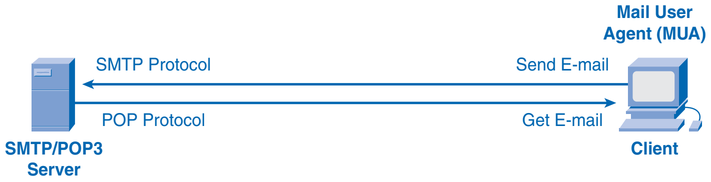
E-mail, the most popular network service, has revolutionized how people communicate through its simplicity and speed. Yet to run on a computer or other end device, e-mail requires several applications and services. Two examples of application layer protocols are ***Post Office Protocol (POP)*** and ***Simple Mail Transfer Protocol (SMTP)***. As with HTTP, these protocols define client/server processes.

POP and POP3 (Post Office Protocol, version 3) are inbound mail delivery protocols and are typical client/server protocols. They deliver e-mail from the e-mail server to the client (MUA).

SMTP, on the other hand, governs the transfer of outbound e-mail from the sending client to the e-mail server (MDA), as well as the transport of e-mail between e-mail servers (MTA). (These acronyms are defined in the next section.) SMTP enables e-mail to be transported across data networks between different types of server and client software and makes e-mail exchange over the Internet possible.

When people compose e-mail messages, they typically use an application called a ***Mail User Agent (MUA)***, or e-mail client. The MUA allows messages to be sent and places received messages into the client mailbox, both of which are distinct processes, as shown in Figure 3-14.

Figure 3-14 E-Mail Client (MUA)



To receive e-mail messages from an e-mail server, the e-mail client can use POP. Sending e-mail from either a client or a server uses message formats and command strings defined by the SMTP protocol. Usually an e-mail client provides the functionality of both protocols within one application.

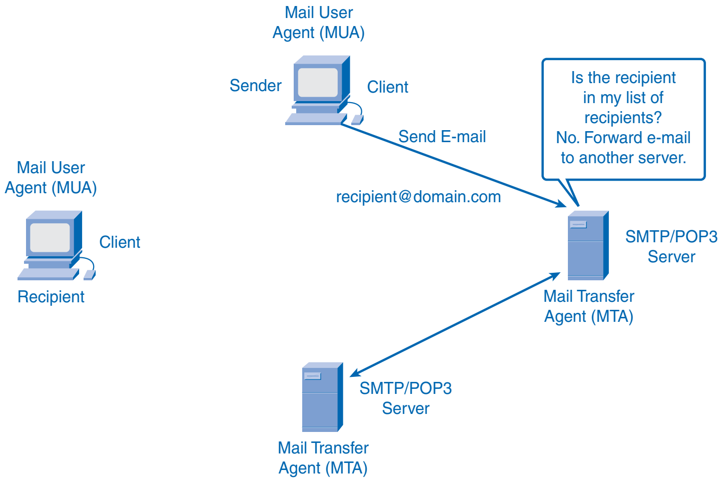
E-Mail Server Processes: MTA and MDA

The e-mail server operates two separate processes:

- Mail Transfer Agent (MTA)
- Mail Delivery Agent (MDA)

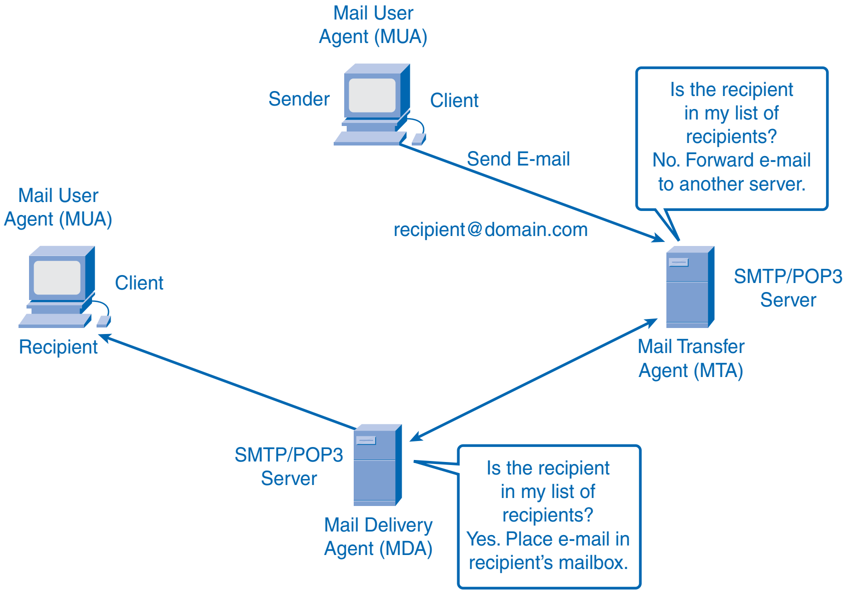
The Mail Transfer Agent (MTA) process is used to forward e-mail. As shown in Figure 3-15, the MTA receives messages from the MUA or from another MTA on another e-mail server. Based on the message header, it determines how a message has to be forwarded to reach its destination. If the mail is addressed to a user whose mailbox is on the local server, the mail is passed to the MDA. If the mail is for a user not on the local server, the MTA routes the e-mail to the MTA on the appropriate server.

Figure 3-15 E-Mail Server: MTA



In Figure 3-16, you see that the Mail Delivery Agent (MDA) accepts a piece of e-mail from a Mail Transfer Agent (MTA) and performs the delivery. The MDA receives all the inbound mail from the MTA and places it into the appropriate users' mailboxes. The MDA can also resolve final delivery issues, such as virus scanning, ***spam*** filtering, and return-receipt handling.

Figure 3-16 E-Mail Server: MDA


Mail User Agent (MUA)
Sender
Client
Send E-mail
Is the recipient in my list of recipients? No. Forward e-mail to another server.
Mail User Agent (MUA)
recipient@domain.com
Client
SMTP/POP3 Server
Recipient
Mail Transfer Agent (MTA)
SMTP/POP3 Server
Is the recipient in my list of recipients? Yes. Place e-mail in recipient's mailbox.
Mail Delivery Agent (MDA)

Most e-mail communications use the MUA, MTA, and MDA applications. However, there are other alternatives for e-mail delivery. A client can be connected to a corporate e-mail system, such as IBM Lotus Notes, Novell Groupwise, or Microsoft Exchange. These systems often have their own internal e-mail format, and their clients typically communicate with the e-mail server using a proprietary protocol.

The server sends or receives e-mail through the Internet through the product's Internet mail ***gateway***, which performs any necessary reformatting. If, for example, two people who work for the same company exchange e-mail with each other using a proprietary protocol, their messages can stay completely within the corporate e-mail system of the company.

As another alternative, computers that do not have an MUA can still connect to a mail service on a web browser to retrieve and send messages in this manner. Some computers can run their own MTA and manage interdomain e-mail themselves.

The SMTP protocol message format uses a rigid set of commands and replies. These commands support the procedures used in SMTP, such as session initiation, mail transaction, forwarding mail, verifying mailbox names, expanding mailing lists, and the opening and closing exchanges.

Some of the commands specified in the SMTP protocol are:

- **HELO:** Identifies the SMTP client process to the SMTP server process
- **EHLO:** Is a newer version of HELO, which includes services extensions
- **MAIL FROM:** Identifies the sender
- **RCPT TO:** Identifies the recipient
- **DATA:** Identifies the body of the message

FTP

FTP is another commonly used application layer protocol. FTP was developed to allow file transfers between a client and a server. An FTP client is an application that runs on a computer that is used to push and pull files from a server running the FTP daemon (FTPd).

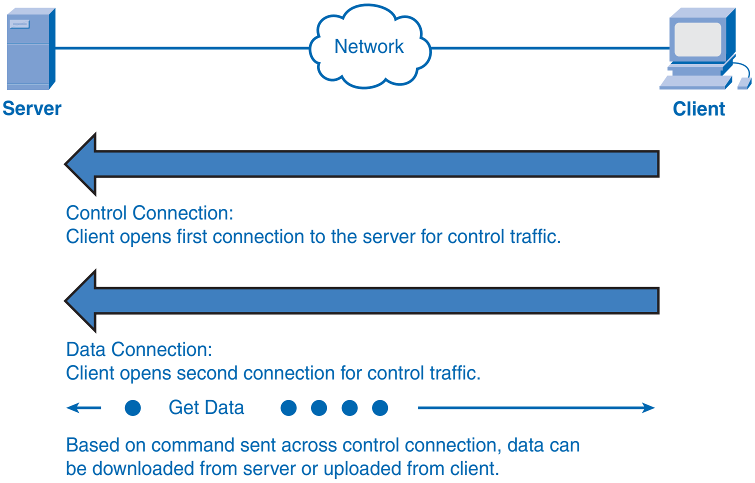
To successfully transfer files, FTP requires two connections between the client and the server: one for commands and replies, and the other for the actual file transfer.

The client establishes the first connection to the server on TCP port 21. This connection is used for control traffic, consisting of client commands and server replies.

The client establishes the second connection to the server over TCP port 20. This connection is for the actual file transfer and is created every time a file is transferred.

The file transfer can happen in either direction, as shown in Figure 3-17. The client can download (pull) a file from the server or upload (push) a file to the server.

Figure 3-17 FTP Process


Network
Server
Client
Control Connection:
Client opens first connection to the server for control traffic.
Data Connection:
Client opens second connection for control traffic.
Get Data
Based on command sent across control connection, data can
be downloaded from server or uploaded from client.

DHCP

The ***DHCP*** enables clients on a network to obtain IP addresses and other information from a DHCP server. The protocol automates the assignment of IP addresses, subnet masks, gateway, and other IP networking parameters.

DHCP allows a host to obtain an IP address dynamically when it connects to the network. The DHCP server is contacted by sending a request, and an IP address is requested. The DHCP server chooses an address from a configured range of addresses called a *pool* and assigns it to the host client for a set period.

On larger networks, local networks, or where the user population changes frequently, DHCP is preferred. New users might arrive with laptops and need a connection. Others have new workstations that need to be connected. Rather than have the network administrator assign IP addresses for each workstation, it is more efficient to have IP addresses assigned automatically using DHCP.

When a DHCP-configured device boots up or connects to the network, the client broadcasts a DHCP DISCOVER packet to identify any available DHCP servers on the network. A DHCP server replies with a DHCP OFFER, which is a lease offer message with an assigned IP address, ***subnet mask***, DNS server, and default gateway information as well as the duration of the lease.

DHCP-distributed addresses are not permanently assigned to hosts but are only leased for a period of time. If the host is powered down or taken off the network, the address is returned to the pool for reuse. This is especially helpful with mobile users who come and go on a network. Users can freely move from location to location and re-establish network connections. The host can obtain an IP address after the hardware connection is made, either through a wired or wireless LAN.

DHCP makes it possible for you to access the Internet using wireless hotspots at airports or coffee shops. As you enter the area, your laptop DHCP client contacts the local DHCP server through a wireless connection. The DHCP server assigns an IP address to your laptop.

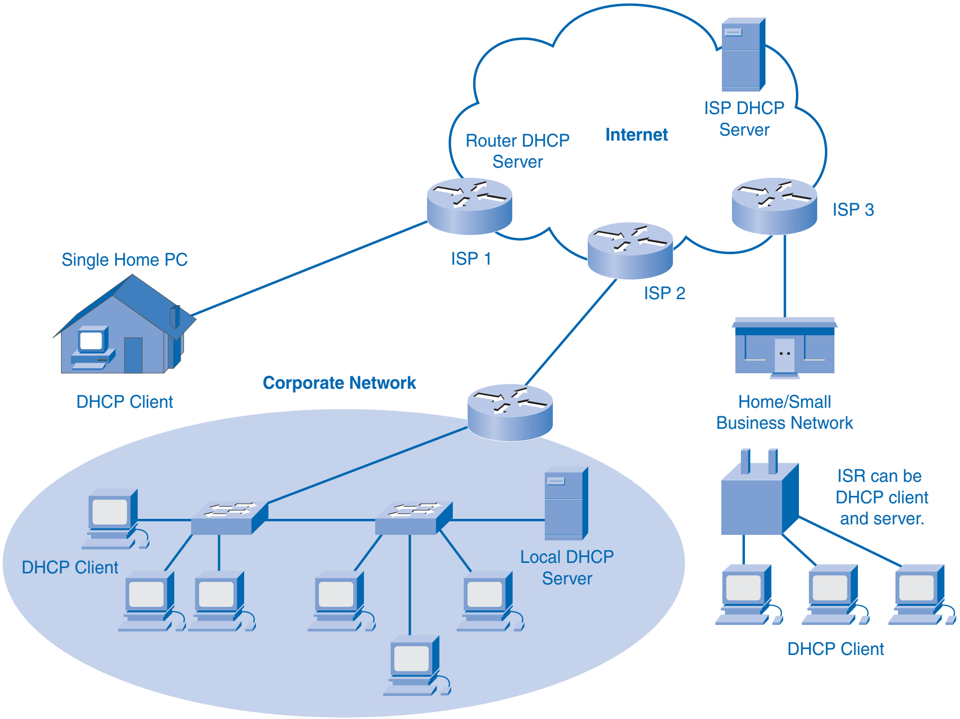
Various types of devices can be DHCP servers when running DHCP service software. The DHCP server in most medium to large networks is usually a local dedicated PC-based server.

With home networks, the DHCP server is usually located at the ISP, and a host on the home network receives its IP configuration directly from the ISP.

Many home networks and small businesses use an Integrated Services Router (ISR) device to connect to the ISP. In this case, the ISR is both a DHCP client and a server. The ISR acts as a client to receive its IP configuration from the ISP and then acts a DHCP server for internal hosts on the local network.

Figure 3-18 shows the different ways of having DHCP servers arranged.

Figure 3-18 DHCP Servers



DHCP can pose a security risk because any device connected to the network can receive an address. This risk makes physical security an important factor when determining whether to use dynamic or static (manual) addressing.

Dynamic and static addressing have their places in network designs. Many networks use both DHCP and static addressing. DHCP is used for general-purpose hosts such as end-user devices, and static, or fixed, addresses are used for network devices such as gateways, switches, servers, and printers.

The client can receive multiple DHCP OFFER packets if the local network has more than one DHCP server. The client must choose between them and ***broadcast*** a DHCP REQUEST packet that identifies the explicit server and lease offer that it is accepting. A client can choose to request an address that it had previously been allocated by the server.

Assuming that the IP address requested by the client, or offered by the server, is still valid, the chosen server would return a DHCP ACK (acknowledgment) message. The ACK message lets the client know that the lease is finalized. If the offer is no longer valid for some reason, perhaps because of a timeout or another client allocating the lease, the chosen server must respond to the client with a DHCP NAK (negative acknowledgment) message. When the client has the lease, it must be renewed prior to the lease expiration through another

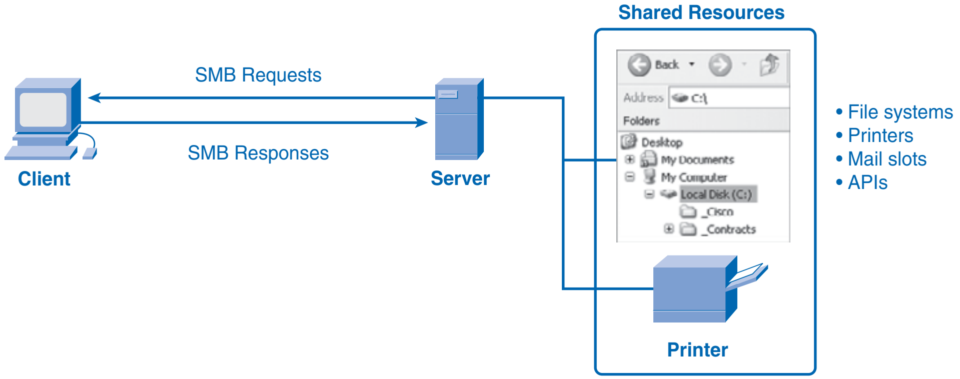
DHCP REQUEST message. The DHCP server ensures that all IP addresses are unique. (An IP address cannot be assigned to two different network devices simultaneously.)

File-Sharing Services and SMB Protocol

Server Message Block (SMB) is a client/server file-sharing protocol. IBM developed SMB in the late 1980s to describe the structure of shared network resources, such as directories, files, printers, and serial ports. It is a request/response protocol. Unlike the file sharing supported by FTP, clients establish a long-term connection to servers. After the connection is established, the user of the client can access the resources on the server as if the resource is local to the client host.

SMB file-sharing and print services have become the mainstay of Microsoft networking. With the introduction of the Windows 2000 series of software, Microsoft changed the underlying structure for using SMB. In previous versions of Microsoft products, the SMB services used a non-TCP/IP protocol to implement name resolution. Beginning with Windows 2000, all subsequent Microsoft products use DNS naming. This allows TCP/IP protocols to directly support SMB resource sharing, as shown in Figure 3-19.

Figure 3-19 File Sharing Using the SMB Protocol



The Linux and ***UNIX*** operating systems also provide a method of sharing resources with Microsoft networks using a version of SMB called SAMBA. The Apple Macintosh operating systems also support resource sharing using the SMB protocol.

The SMB protocol describes file system access and indicates how clients can make requests for files. It also describes the SMB protocol interprocess communication. All SMB messages share a common format. This format uses a fixed-sized header followed by a variable-sized parameter and data component.

SMB messages can perform the following tasks:

- Start, authenticate, and terminate sessions
- Control file and printer access
- Allow an application to send or receive messages to or from another device

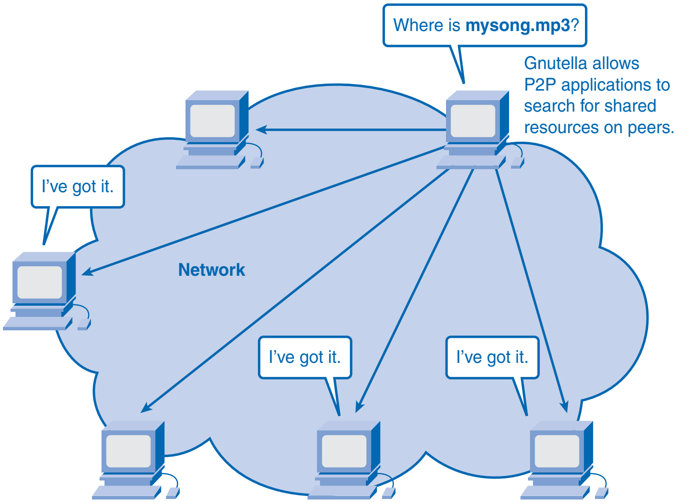
P2P Services and Gnutella Protocol

You learned about FTP and SMB as ways of obtaining files. This section describes another application protocol, Gnutella. Sharing files over the Internet has become extremely popular. With P2P applications based on the Gnutella protocol, people can make files on their hard disks available to others for downloading. Gnutella-compatible client software allows users to connect to Gnutella services over the Internet and to locate and access resources shared by other Gnutella peers.

Many client applications are available for accessing the Gnutella network, including BearShare, Gnucleus, LimeWire, Morpheus, WinMX, and XoloX. Although the Gnutella Developer Forum maintains the basic protocol, application vendors often develop extensions to make the protocol work better on their applications.

Many P2P applications do not use a central database to record all the files available on the peers. Instead, the devices on the network each tell the other what files are available when queried and use the Gnutella protocol and services to support locating resources, as shown in Figure 3-20. When a user is connected to a Gnutella service, the client applications will search for other Gnutella nodes to connect to. These nodes handle queries for resource locations and replies to those requests. They also govern control messages, which help the service discover other nodes. The actual file transfers usually rely on HTTP services.

Figure 3-20 Gnutella Protocol


Where is **mysong.mp3**?
Gnutella allows P2P applications to search for shared resources on peers.
I've got it.
Network
I've got it.
I've got it.

The Gnutella protocol defines five different packet types:

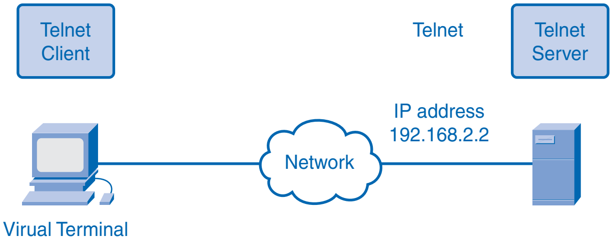
- **ping:** For device discovery
- **pong:** As a reply to a ping
- **query:** For file location
- **query hit:** As a reply to a query
- **push:** As a download request

Telnet Services and Protocol

Long before desktop computers with sophisticated graphical interfaces existed, people used text-based systems that were often just display terminals physically attached to a central computer. After networks were available, people needed a way to remotely access the computer systems in the same manner that they did with the directly attached terminals.

Telnet was developed to meet that need. It dates back to the early 1970s and is among the oldest of the application layer protocols and services in the TCP/IP suite. Telnet is a client/server protocol that provides a standard method of emulating text-based terminal devices over the data network. Both the protocol itself and the client software that implements the protocol are commonly referred to as Telnet. The Telnet service is depicted in Figure 3-21.

Figure 3-21 Telnet Service



Appropriately enough, a connection using Telnet is called a *VTY* (Virtual Terminal) *session*, or *connection*. Telnet specifies how a VTY session is established and terminated. It also provides the syntax and order of the commands used to initiate the Telnet session, and it provides control commands that can be issued during a session. Each Telnet command consists of at least 2 bytes. The first byte is a special character called the ***Interpret as Command (IAC)*** character. As its name implies, the IAC character defines the next byte as a command rather than text. Rather than using a physical device to connect to the server, Telnet uses software to create a virtual device that provides the same features of a terminal session with access to the server command-line interface (CLI).

To support Telnet client connections, the server runs a service called the Telnet daemon. A virtual terminal connection is established from an end device using a Telnet client application. Most operating systems include an application layer Telnet client. On a Microsoft Windows PC, Telnet can be run from the command prompt. Other common terminal applications that run as Telnet clients are HyperTerminal, Minicom, and TeraTerm.

When a Telnet connection is established, users can perform any authorized function on the server, just as if they were using a command-line session on the server itself. If authorized, they can start and stop processes, configure the device, and even shut down the system.

The following are some sample Telnet protocol commands:

- **Are You There (AYT):** Enables the user to request that a response, usually a prompt icon, appear on the terminal screen to indicate that the VTY session is active.
- **Erase Line (EL):** Deletes all text from the current line.
- **Interrupt Process (IP):** Suspends, interrupts, aborts, or terminates the process to which the virtual terminal is connected. For example, if a user started a program on the Telnet server through the VTY, he or she could send an IP command to stop the program.

Although the Telnet protocol supports user authentication, it does not support the transport of encrypted data. All data exchanged during a Telnet session is transported as plain text across the network. This means that the data can be intercepted and easily understood.

The Secure Shell (SSH) protocol offers an alternate and secure method for server access. SSH provides the structure for secure remote login and other secure network services. It also provides stronger authentication than Telnet and supports the transport of session data using encryption. As a best practice, network professionals should use SSH in place of Telnet, whenever possible.

Summary

The application layer is responsible for directly accessing the underlying processes that manage and deliver communication to the human network. This layer serves as the source and destination of communications across data networks. The application layer applications, protocols, and services enable users to interact with the data network in a way that is meaningful and effective.

Applications are computer programs with which the user interacts and that initiate the data transfer process at the user's request.

Services are background programs that provide the connection between the application layer and the lower layers of the networking model.

Protocols provide a structure of agreed-upon rules, much like grammar and punctuation provide "rules" in a language. These protocol rules ensure that services running on one particular device can send and receive data from a range of different network devices.

Delivery of data over the network can be requested from a server by a client. In a peer-to-peer arrangement, either device can function as a client or server, and data is delivered depending on the client/server relationship established. Messages are exchanged between the application layer services at each end device in accordance with the protocol specifications to establish and use these relationships.

Protocols like HTTP, for example, support the delivery of web pages to end devices. SMTP/POP protocols support sending and receiving e-mail. SMB enables users to share files. DNS resolves the human-legible names used to refer to network resources into numeric addresses usable by the network. Telnet provides remote, text-based access to devices. DHCP provides dynamic allocation of IP addresses and other network-enabling parameters. P2P allows two or more computers to share resources over the network.

Activities and Labs

The activities and labs available in the companion *Network Fundamentals, CCNA Exploration Labs and Study Guide* (ISBN 1-58713-203-6) provide hands-on practice with the following topics introduced in this chapter:

Activity 3-1: Data Stream Capture (3.4.1.1)

In this activity, you will use a computer that has a microphone and Microsoft Sound Recorder or Internet access so that an audio file can be downloaded.

Lab 3-1: Managing a Web Server (3.4.2.1)

In this lab, you will download, install, and configure the popular Apache web server. You will use a web browser to connect to the server and Wireshark to capture the communication. Analyzing the capture will help you understand how HTTP operates.

Lab 3-2: E-mail Services and Protocols (3.4.3.1)

In this lab, you will configure and use an e-mail client application to connect to eagle-server network services. You will then monitor the communication with Wireshark and analyze the captured packets.

Packet Tracer
□ **Companion**

Many of the hands-on labs include Packet Tracer companion activities where you can use Packet Tracer to complete a simulation of the lab. Look for this icon in the *Network Fundamentals, CCNA Exploration Labs and Study Guide* (ISBN 1-58713-203-6) for hands-on labs that have Packet Tracer companion activities.

Check Your Understanding

Complete all the review questions listed here to test your understanding of the topics and concepts in this chapter. The appendix, "Check Your Understanding and Challenge Questions Answer Key," lists the answers.

1. The application layer is ___________ of the OSI model.

   A. Layer 1

   B. Layer 3

   C. Layer 4

   D. Layer 7

2. The TCP/IP application layer consists roughly of which three OSI layers?

   A. Application, session, transport

   B. Application, presentation, session

   C. Application, transport, network

   D. Application, network, data link

3. HTTP is used to do which of the following?

   A. Resolve Internet names to IP addresses

   B. Provide remote access to servers and networking devices

   C. Transfer files that make up the web pages of the World Wide Web

   D. Transfer the mail messages and attachments

4. Post Office Protocol (POP) uses which port?

   A. TCP/UDP port 53
   B. TCP port 80
   C. TCP port 25
   D. UDP port 110

5. What is GET?

   A. A client request for data
   B. A protocol that uploads resources or content to the web server
   C. A protocol that uploads information to the server in plain text that can be intercepted and read
   D. A response from a server

6. Which is the most popular network service?

   A. HTTP
   B. FTP
   C. Telnet
   D. E-mail

7. FTP requires ____________ connection(s) between client and server to successfully transfer files.

   A. 1
   B. 2
   C. 3
   D. 4

8. DHCP enables clients on a network to do which of the following?

   A. Have unlimited telephone conversations
   B. Play back video streams
   C. Obtain IP addresses
   D. Track intermittent denial of service attacks

9. The Linux and UNIX operating systems use SAMBA, which is a version of which protocol?

   A. SMB
   B. HTTP
   C. FTP
   D. SMTP

10. Which of the following is a connection using Telnet?

    A. File Transfer Protocol (FTP) session

    B. Trivial File Transfer Protocol (TFTP) session

    C. Virtual Terminal (VTY) session

    D. Auxiliary (AUX) session

11. Is eBay a peer-to-peer or client/server application?

12. In the client/server model, the device requesting the service is referred to as the ____________.

13. HTTP is referred to as a request/response protocol. What are three typical message formats?

14. DHCP allows the automation of what?

15. What does FTP stand for, and what is it used for?

Challenge Questions and Activities

These questions require a deeper application of the concepts covered in this chapter. You can find the answers in the appendix.

1. List the six-step process for converting human communications to data.

2. Describe the two forms of application software and the purpose of each.

3. Elaborate on the meaning of the terms server and client in the context of data networks.

4. Compare and contrast client/server with peer-to-peer data transfer over networks.

5. List five general functions that application layer protocols specify.

6. Give the specific purposes of the DNS, HTTP, SMB, and SMTP/POP application layer protocols.

7. Compare and contrast the messages that application layer protocols such as DNS, HTTP, SMB, and SMTP/POP exchange between devices to enable data transfers to occur.

To Learn More

The following questions encourage you to reflect on the topics discussed in this chapter. Your instructor might ask you to research the questions and discuss your findings in class.

1. Why is it important to distinguish between a particular application layer application, the associated service, and the protocol? Discuss this in the context of network reference models.
2. What if it was possible to include all application layer services with a single all-encompassing protocol? Discuss the advantages and disadvantages of having one such protocol.
3. How would you develop a new protocol for a new application layer service? What would have to be included? Who would have to be involved in the process, and how would the information be disseminated?

This page intentionally left blank

Index