

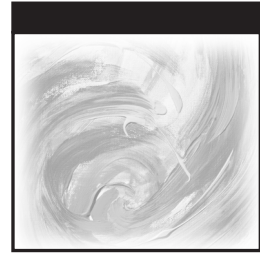
# **TCP/IP Analysis and Troubleshooting Toolkit**

Kevin Burns



Wiley Publishing, Inc.





# **TCP/IP Analysis and Troubleshooting Toolkit**

Kevin Burns



Wiley Publishing, Inc.

Executive Publisher: Robert Ipsen  
Vice President and Publisher: Joe Wikert  
Editor: Carol A. Long  
Developmental Editor: Kevin Kent  
Editorial Manager: Kathryn Malm  
Production Editor: Pamela M. Hanley  
Text Design & Composition: Wiley Composition Services

This book is printed on acid-free paper. ☺

Copyright © 2003 by Kevin Burns. All rights reserved.

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8700. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-mail: permcoordinator@wiley.com.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

**Trademarks:** Wiley, the Wiley Publishing logo and related trade dress are trademarks or registered trademarks of Wiley Publishing, Inc., in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Screenshot(s) Copyright © 2002 Wildpackets, Inc. All rights reserved.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

*Library of Congress Cataloging-in-Publication Data: is available from the publisher*

ISBN: 0-471-42975-9

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

*To my parents, who always believed in me*





# Contents

<b>Acknowledgments</b>	<b>xi</b>
<b>About the Author</b>	<b>xiii</b>
<b>Introduction</b>	<b>xv</b>
<b>Part I Foundations of Network Analysis</b>	<b>1</b>
<b>Chapter 1 Introduction to Protocol Analysis</b>	<b>3</b>
A Brief History of Network Communications	3
OSI to the Rescue	5
Defining the Layers	6
Layer 1: Physical Layer	6
Layer 2: Data Link Layer	7
Layer 3: Network Layer	7
Layer 4: Transport Layer	7
Layer 5: Session Layer	7
Layer 6: Presentation Layer	8
Layer 7: Application Layer	8
Protocol Analysis of the Layers	8
Layer 1: The Physical Layer	8
Layer 2: The Data Link Layer	10
Layer 3: Network Layer	18
Layer 4: Transport Layer	21
Layer 5: Session Layer	23
Layer 6: Presentation Layer	23
Layer 7: Application Layer	24
Putting It All Together	24
History of TCP/IP	26
Summary	28
<b>Chapter 2 Analysis Tools and Techniques</b>	<b>29</b>
Reviewing Network Management Tools	30
Categorizing Network Management Tools by Function	30
Fault Management Systems	31
Performance Management and Simulation	31

Protocol Analyzers	32
Application-Specific Tools	33
Classifying Tools by How They Perform Functions	33
Protocol Analyzers—Problem-Solving Tools	35
Why Protocol Analysis?	36
Protocol Analyzer Functions	37
Data Capture	37
Network Monitoring	42
Data Display	42
Notification	44
Logging	45
Packet Generator	45
Configuring and Using Your Analyzer	45
Capture Configuration	45
Filtering	48
Expert Analysis	52
Measuring Performance	56
Analysis Tips	61
Placing Your Analyzers	61
Using Proper Filters	62
Troubleshooting from the Bottom Up	63
Knowing Your Protocols	63
Comparing Working Traces	63
Analyzing after Each Change	65
Summary	65
<b>Part II     The Core Protocols</b>	<b>67</b>
<b>Chapter 3     Inside the Internet Protocol</b>	<b>69</b>
Reviewing Layer 2 Communications	70
Multiplexing	70
Error Control	71
Addressing	71
Case Study: NetBEUI Communications	72
Name Resolution	73
Reliable Connection Setup	74
NetBIOS Session Setup	75
Application Process	75
Limitations of Layer 2 Communication Networks	76
Network Layer Protocols	77
Internet Protocol Addressing	79
IP Addressing	81
Reserved Addressing	85
Classful Addressing	85
Classless Addressing	88
IP Communications	92
Address Resolution Protocol (ARP)	93
ARP Packet Format	94
Case Study: Troubleshooting IP Communications	
with ARP and PING	97
ARP Types	100
ARP in IP Communication	101
Case Study: Incomplete ARP	101



IP Routing	104
The Routing Table	104
Route Types	108
Router Routing Tables	110
The Forwarding Process	112
Case Study: Local Routing	114
IP Packet Format	117
Version	117
Header Length	117
Type of Service	117
Datagram Length	119
Fragment ID	119
Fragmentation Flags	119
Fragment Offset	119
Time to Live	120
Protocol	120
Header Checksum	121
Source IP Address	121
Destination IP Address	121
Options	121
Data	121
Case Study: TTL Expiring	122
Case Study: Local Routing Revisited	124
A Word about IP Version 6	126
The IPv6 Header	128
IPv6 Address Format	129
Other Changes to IPv6	130
Summary	130
<b>Chapter 4    Internet Control Message Protocol</b>	<b>131</b>
Reliability in Networks	132
Connection-Oriented versus Connectionless Networks	132
Feedback	133
Exploring the Internet Control Messaging Protocol	134
ICMP Header	134
ICMP Types and Codes	135
ICMP Message Detail	137
Destination Unreachable (Type 3)	137
Diagnostic Messages	144
Redirect Codes (type 5)	146
Time Exceeded (Type 11)	151
Informational Messages	151
Network Diagnostics with ICMP	152
Summary	154
<b>Chapter 5    User Datagram Protocol</b>	<b>155</b>
Revisiting the Transport Layer	156
UDP Header	157
Source Port	157
Destination Port	157
UDP Length	158
UDP Checksum	158
Data	159
UDP Communication Process	160

Case Studies in UDP Communications	164
Name Resolution Services	165
Routing Information Protocol	166
Simple Network Management Protocol	169
UDP and Firewalls	169
Case Study: Failed PCAnywhere Session	170
Case Study: NFS Failures	171
Traceroute Caveats	173
Summary	174
<b>Chapter 6 Transmission Control Protocol</b>	<b>175</b>
Introduction to TCP	175
Requirements for a Reliable Transport Protocol	176
Fast Sender and Slow Receiver	177
Packet Loss	177
Data Duplication	178
Priority Data	178
Out-of-Order Data	179
The TCP Header	179
Source Port	180
Destination Port	180
Sequence Number	181
Acknowledgment Number	181
Header Offset	181
Reserved Bits	181
Connection Flags	182
Window Size	182
TCP Checksum	182
Urgent Pointer	182
Options	183
Data	183
TCP Implementation	183
Multiplexing	183
Data Sequencing and Acknowledgment	183
Flow Control	183
TCP Connection Management	184
TCP Open	185
Initial Sequence Number (ISN)	185
TCP Connection States	189
TCP Options	189
TCP Close	192
Half-Close	193
TCP Reset	194
Case Study: Missing Drive Mappings	194
Case Study: No Telnet	196
Case Study: Dropped Sessions	197
TCP Data Flow Management	200
Data Sequencing and Acknowledgment	200
TCP Retransmissions	202
Retransmission Time-Out	202
Case Study: Bad RTO	203
Delayed Acknowledgments	204
Case Study: Slow Surfing	206

	The Push Flag	207
	TCP Sliding Windows	209
	Slow Start and Congestion Avoidance	212
	Nagle Algorithm	213
	Data Protection	215
	Case Study: TCP Checksum Errors	215
	TCP Expert Symptoms	217
	TCP Application Analysis	218
	TCP and Throughput	218
	Segment Size	218
	Latency	219
	Window Size	220
	Case Study: Slow Web Server	221
	Case Study: Bad Windowing	222
	Case Study: Inefficient Applications	224
	High-Performance Extensions to TCP	225
	Selective Acknowledgments	225
	Window Scale Option	227
	Timestamp Option	229
	Summary	229
<b>Part III</b>	<b>Related TCP/IP Protocols</b>	<b>231</b>
<b>Chapter 7</b>	<b>Upper-Layer Protocols</b>	<b>233</b>
	Introduction to Upper-Layer Protocols	233
	Analyzing Upper-Layer Protocols	235
	Chapter Goals	238
	Domain Name System (DNS)	240
	DNS Database	242
	DNS Message Format	244
	Using NSLookup	247
	Name Servers	249
	ROOT Name Servers	250
	Name Server Caching	254
	Resource Records	254
	Analyzing DNS	260
	IPCONFIG	260
	CyberKit	261
	DNS Expert	262
	Common DNS Configuration Mistakes	264
	File Transfer Protocol (FTP)	265
	FTP Commands and Responses	265
	Case Study: Active Transfer Failure	269
	Case Study: Passive Transfer Failure	272
	Case Study: FTP Failures through Firewall	273
	Case Study: Revisiting FTP Transfer Failures	276
	Hypertext Transport Protocol (HTTP)	278
	HTTP Requests	278
	HTTP Responses	281
	HTTP Headers and Messages	284
	Host Header	285
	Redirection	285
	Cookies	285
	Cache Control Headers	288

HTTP Proxies	289
Measuring Proxy Latency	290
Analyzing Advanced Web Architectures	291
Case Study: Web Site Failure	293
Simple Mail Transport Protocol	294
Summary	298
<b>Chapter 8    Microsoft-Related Protocols</b>	<b>299</b>
Dynamic Host Configuration Protocol	299
DHCP Header	300
DHCP Process	302
DHCP Messages	306
DHCP Options	308
DHCP Leases	311
NetBIOS over TCP/IP	312
NetBIOS Names	312
NetBIOS Services	315
Datagram Service	316
Session Service	317
Name Service	319
NetBIOS Operations	320
Name Service Operations	321
NetBIOS Datagram Operations	326
NetBIOS Session Operations	326
Server Message Block	329
SMB Header	330
SMB Commands	332
SMB Responses	336
SMB Operations Analysis	338
Initial Connection	339
File Transfer	344
File Locking	351
Interprocess Communication	352
Named Pipes	353
Mailslots	353
DCE/RPC	354
Microsoft Applications	359
NetLogon	359
Browser Protocol	362
Summary	368
<b>Appendix A    What's on the Web Site</b>	<b>369</b>
System Requirements	369
What's on the Web Site	370
Standards and RFCs	370
Author-Created Materials	371
Applications	371
Using the Flash Video Examples	371
Troubleshooting	372
<b>Appendix B    BSMB Status Codes</b>	<b>373</b>
<b>Index</b>	<b>399</b>



# Acknowledgments

This book never would have been a reality without the following people: Emily Roche, who helped me open the door to writing and took me to my first book proposal seminar; Toni Lopopolo, who taught the seminar and put me in contact with my great agent Jawahara Saidullah. I want to thank Tony Fortunato for patiently reviewing my book for technical accuracy. Thanks also goes out to everyone at Wiley Publishing who worked so hard on this book, including my great development editor Kevin Kent, who held me to task on making sure readers would be able to easily understand the complex case studies and examples in the book. Last but not least, I want to thank my parents, who have given me everything and asked for nothing in return. This book is for you.





## About the Author

**Kevin Burns** is the founder of Tracemasters, Inc., of Philadelphia, Pennsylvania, a consulting organization specializing in network analysis and training. Kevin's 10 years of experience consist of the design, implementation, and analysis of various multiprotocol, multivendor networks. This book comprises the techniques he has used in diagnosing complex network and application problems, which he also teaches to students at various seminars and corporate settings. Kevin can be reached at [kburns@tracemasters.com](mailto:kburns@tracemasters.com).







# Introduction

## Why I Wrote This Book

---

Network engineers face difficult challenges on a daily basis. Servers can crash, WAN links can become saturated, and for unknown reasons, an application's performance can come to a crawl, pitting network engineers against application developers in a complicated blame game, usually without facts. Without the proper tools and training, when something breaks, network engineers often have to ask why: Why can't users obtain DHCP addresses, why can't users log into the server, and—the ever so bothersome question—why is the network slow? During all of this commotion, upper management is usually also asking why—Why haven't these problems been resolved? Most large network infrastructures have a mix of troubleshooting tools at their disposal, but more often than not the wrong tools are selected for the wrong job. How can you best use the tools at your disposal and the knowledge of your networks to assist you in quickly and decisively solving problems on your network infrastructures? The answer to that question is the subject of this book.

I wrote this book for the people on the front lines, the network field engineers. I have a great respect for field engineers. They are the doers, the people that make things work; they are also the first people whose pagers start beeping when things don't work. In my over 10 years of experience supporting desktops, servers, and large complex network infrastructures, I've come to the conclusion that the best field engineers are the ones who can solve the really tough problems.

People who are good problem solvers are usually tenacious and curious. These two qualities drive these people to stay up all night to try to solve a problem. They know the answer is there somewhere, waiting to be uncovered, and

they are tenacious enough to dig until they find it. The truly curious will most likely have read many good books on the TCP/IP protocol, including W. Richard Stevens' *TCP/IP Illustrated* (Addison-Wesley January 1994) and Douglas Comer's *Internetworking with TCP/IP* (Prentice Hall January 2000). To date these books are the flagship manuscripts on understanding TCP/IP, but they focus intensely on theory and lack in practical examples. (That said, I still recommend every analyst have a copy of them on their bookshelves.) I have attempted to bridge the gap left by these two books by taking the most important concepts on the protocols and applying them to the most common problems a network analyst sees on TCP/IP networks. For the more curious, interested in the intricate details and inner workings of the protocol, I have provided an appendix further detailing the website.

The goal behind the *TCP/IP Analysis and Troubleshooting Toolkit* is to give the reader the information needed to successfully maintain the protocol in real-world networks. Since TCP/IP is the most common protocol in use today, this made the decision to concentrate an entire book on the subject of its analysis and troubleshooting methods easy. Rather than write a book about the many intricate and often-mundane details of the protocol, I attempt to empower you with the knowledge to understand and diagnose problems related to the TCP/IP protocol.

You will quickly notice that many of the examples in the book are either Cisco or Microsoft specific. Since those are the two most prevalent vendors in use today, I have chosen to use examples pertaining to their systems. The examples are by no means exclusive to either Cisco or Microsoft. In almost all cases, you can take the examples and apply them to any vendor's hardware or software. Specific examples that apply to a certain vendor are noted. Along this line, you might also notice several analysis tools mentioned or used in the examples. The type of tool is not typically important, just as long as it provides the functionality needed or described.

An understanding of the technology is what's important and that is what this book concentrates on.

## **Who Should Read This Book**

---

Although this book does provide an introduction to network analysis techniques and the TCP/IP protocol, it is not for beginners. A basic understanding of the OSI model is important, as well as a decent level of experience managing server operating systems running TCP/IP.

More advanced readers already familiar with the protocol will benefit greatly from the case studies presented in each chapter. This book will help you become a better network analyst. If you are a network administrator eager

to learn more about understanding communications between clients and servers, this is a good place to start. If you are already familiar with configuring routers and switches, this book will teach you the technology behind the configuration commands; it will help you learn to think “outside the box.”

This book is about technology and how to best use tools at your disposal to keep your networks running smoothly.

## How This Book Is Organized

---

The book is organized into three parts:

- **Part I: Foundations of Network Analysis** answers such questions as “Why protocol analysis?” and “What tools do I use?” It explains the process of capturing and manipulating trace files. It also provides a refresher of the OSI model and the basic concepts of network communication that are needed to benefit from the material presented in the later chapters.
- **Part II: The Core Protocols** builds the foundation for understanding the protocols that TCP/IP is built upon. It is these protocols that provide the support for all other application-layer protocols.
- **Part III: Related TCP/IP Protocols** extends the search for understanding by revealing the inner workings of standard and vendor-independent protocol implementations. Applications such as DNS (Domain Name System), HTTP (Hypertext Transport Protocol), and FTP (File Transport Protocol) are thoroughly analyzed, and a deep investigation is conducted into Microsoft’s TCP/IP implementation, including the ever-so-mysterious Server Message Block protocol.

In each chapter, the material is complemented with numerous case studies and examples from real, live networks. These examples and case studies are given to illustrate how the knowledge and techniques discussed can be put to use.

## Tools

---

This book uses several different analysis tools to illustrate the troubleshooting examples. While the tools are not necessary to understand the examples, you do need them to view the trace files included on the companion Web site. The Web site includes instructions for downloading the freeware version of the Ethereal protocol analyzer, which can be used to view the traces.

## **The Companion Web Site**

---

The companion Web site to this book (which can be found by pointing your browser to [www.wiley.com/compbooks/burns](http://www.wiley.com/compbooks/burns)) contains protocol standards such as RFCs (Requests for Comment), IETF (Internet Engineering Task Force) standards, and other resources concerning the protocols discussed in the book. It also contains online videos of most of the books example materials and trace files from the actually case studies, which you can load and examine for yourself. Finally, it includes several freeware and shareware utilities that are a must in the network analyst's toolkit. For more specific information as to what is on the Web site, see Appendix A.

**PART**



**One**

# **Foundations of Network Analysis**

---



# Introduction to Protocol Analysis

What is protocol analysis? A *protocol* is defined as a standard procedure for regulating data transmission between computers. Protocol analysis is the process of examining those procedures. The way we go about this analysis is with special tools called *protocol analyzers*. Protocol analyzers decode the stream of bits flowing across a network and show you those bits in the structured format of the protocol. Using protocol analysis techniques to understand the procedures occurring on your network is the focus of this book. In my 10 years of analyzing and implementing networks, I have learned that in order to understand how a vendor's hardware platform, such as a router or switch, functions you need to understand how the protocols that the hardware implements operate. Routers, switches, hubs, gateways, and so on are simply nothing without the protocols. Protocols make networks happen. Routers and other devices implement those protocols. Understand the protocol, and you can largely understand what happens inside the box.

## A Brief History of Network Communications

---

For years, complex processing needs have been the driving factors behind the development of computer systems. Early on, these needs were met by the development of supercomputers. Supercomputers were designed to service a single

application at a very high speed, thus saving valuable time in performing manual calculations.

Supercomputers, with their focus on servicing a single application, couldn't fully meet the business need for a computing system supporting multiple users. Applications designed for use by many people required multiple input/output systems for which supercomputers were not designed. These systems were known as time-sharing systems because each user was given a small slice of time from the overall processing system. The earliest of these systems were known as mainframes. Although not as fast as supercomputers, mainframes could service the business needs of many users running multiple applications simultaneously. This feature made them far more effective at servicing multiple business needs.

The advent of mainframes thus led to the birth of centralized computing. With its debut, centralized computing could provide all aspects of a networked communications system within a tightly controlled cohesive system. Such systems as IBM's S/390 provided the communication paths, applications, and storage systems within a large centralized processing system. Client workstations were nothing more than text screens that let users interact with the applications running on the centralized processing units.

Distributed computing followed on the heels of centralized computing. Distributed computing is characterized by the division of business processes on separate computer systems. In the late 80's and early 90's the dumb terminal screens used in centralized computing architectures started to be replaced by computer workstations that had their own processing power and memory and, more importantly, the ability to run applications separate from the mainframe. Early distributed systems were nothing more than extensions of a single-vendor solution (bought from a single vendor) over modem or dedicated leased lines. Because the vendor controlled all aspects of the system, it was easy for that vendor to develop the communication functions that were needed to make their centralized systems distributed. These types of systems are known as "closed" systems because they only interoperate with other systems from the same manufacturer. Apple Computer and Novell were among the first companies to deliver distributed (although still proprietary) networking systems.

Distributed processing was complicated. It required addressing, error control, and synchronized coordination between systems. Unfortunately, the communication architectures designed to meet those requirements were not compatible across vendors' boundaries. Many closed proprietary systems were developed, most notably IBM's System Network Architecture (SNA) and Digital Equipment Corporation's DECNet. Down the road, other companies such as Novell and Apple followed suit. In order to open up these "closed systems," a



framework was needed which would allow interoperability between various vendors' systems.

## OSI to the Rescue

---

OSI (Open System Interconnection), developed by the International Organization for Standardization (ISO), was the solution designed to promote interoperability between vendors. It defines an architecture for communications that support distributed processing. The OSI model describes the functions that allow systems to communicate successfully over a network. Using what is called a layered approach, communications functions are broken down into seven distinct layers. The seven layers, beginning with the bottom layer of the OSI model, are as follows:

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

Each layer provides a service to the layers above it, but also depends on services from the layers below it. The model also provides a layer of abstraction because upper layers do not need to know the details of how the lower layers operate; they simply must possess the ability to use the lower layers' services. The model was created so that in a perfect world any network layer protocol, such as IP (Internet Protocol), IPX (Internet Packet Exchange), or X.25, could operate regardless of the physical media it runs over. This concept applies to all of the layers, and in later chapters you can see how some application protocols function identically over different network protocols (and sometimes even different vendors—Server Message Block (SMB) is a perfect example of this as it is used by Microsoft, IBM, and Banyan's server operating systems). Most communication protocols map very nicely to the OSI model.

**NOTE** OSI actually consists of not only the model but also a suite of complex protocols. Although the protocols are rarely used today, their original purpose was to provide a single protocol suite that all vendors could adopt into their systems, allowing for interoperability. The model survived, but unfortunately, the protocols did not.

The OSI Model	
	Example Protocols
Application	SMB, HTTP, FTP, SMTP, NCP, TELNET
Presentation	JPG, GIF, MPEG, ASN.1, SMB Negotiation
Session	NetBIOS, TCP 3-way handshake
Transport	TCP, SPX
Network	IP, IPX, DDP
Data Link	Ethernet, Token Ring, FDDI, Frame Relay, HDLC
Physical	X.21, RS-232, DS1, DS3

**Figure 1-1** The OSI model.

## Defining the Layers

Because almost all protocols are based on the OSI model, it is important to completely understand how the model operates, and to understand the protocols, you must first understand the framework. The following sections explain the seven layers in more detail, and Figure 1-1 gives examples of protocols that reside at each layer.

### *Layer 1: Physical Layer*

The simplest definition of the physical layer is that it deals with how binary data is translated into signals and transmitted across the communications medium. (I talk more about media in the “Detailed Layer Analysis” section later in this chapter.) The physical layer also comprises the functions and procedures that are responsible for the transmission of bits. Examples would be procedures such as RS-232 handshaking or zero substitution functions on B8ZS T1 circuits. The physical layer concerns itself only with sending a stream of bits between two devices over a network.

## ***Layer 2: Data Link Layer***

Layer 2, the data link layer, handles the functions and procedures necessary for coordinating frames between devices. At the data link layer, zeros and ones are logically grouped into frames with a defined beginning and end. Unlike the physical layer, the data link layer contains a measure of intelligence. Ethernet, a common Layer 2 protocol, contains detection algorithms for controlling collision detection, corrupted frames, and address recognition. Higher layers depend on the data link layer not only to provide an error-free path but also to detect errors that may occur. Corrupted data should never be passed to upper layers.

## ***Layer 3: Network Layer***

Layer 3 is the end-to-end communications provider. Whereas the data link layer's responsibility ends at the next Layer 2 device, the network layer is responsible for routing data from the source to the destination over multiple Layer 2 paths. Applications utilizing a Layer 3 protocol do not need to know the details of the underlying Layer 2 network. Layer 3 networks, such as those using the Internet Protocol, will span many different Layer 2 technologies such as Ethernet, Token Ring, Frame Relay, and Asynchronous Transfer Mode (ATM). Some examples of Layer 3 protocols are IP, IPX, and AppleTalk Datagram Delivery Protocol (DDP). Although the network layer is responsible for the addressing and routing of data from source to destination, it is not responsible for guaranteeing its delivery.

## ***Layer 4: Transport Layer***

Networks are not reliable. On Ethernet networks, collisions can occur resulting in data loss, switches can drop packets due to congestion, and networks themselves can lose data due to overloaded links (the Internet itself experiences anomalies such as these on a daily basis). Protocols that operate in the transport layer may retransmit lost data, perform flow control between end systems, and many times add an extra layer of error protection to application data. While the network layer delivers data between two endpoints, the transport layer can guarantee that it gets to its destination.

## ***Layer 5: Session Layer***

The session layer provides the ability to further control communications between end systems by providing another layer of abstraction between transport protocols and the application. If an application layer protocol possesses this functionality, a session layer protocol may not be needed. NetBIOS, as you will see later in this chapter, is a perfect example of a session layer protocol. Sometimes the session layer does not reveal itself as a protocol, but rather as a

procedure performed to allow a protocol to continue its functions. Even though a protocol will exist at a certain layer, a procedure of that protocol can sometimes perform functions that normally reside in another layer. I will note instances in later chapters where this anomaly takes place.

### ***Layer 6: Presentation Layer***

The presentation layer is another layer that sometimes does not manifest itself in obvious ways. The presentation layer handles making sure that data formats used by application layer protocols are compatible between end systems. Some examples of Layer 6 would be ASCII, JPG, and ASN.1. Just as I indicated was the case with Layer 5, some protocol functions performed in other layers fit nicely into the description of the presentation layer.

### ***Layer 7: Application Layer***

Many people confuse Layer 7 with the applications used on servers or workstations. Application layer protocols are not user applications but instead the protocols that allow those applications to operate over a network. A user browsing the Internet with Internet Explorer utilizes an application layer protocol called HTTP. Microsoft Word users saving files to a network server make use of the Server Message Block (SMB) protocol. To a user, a network drive simply appears as G:\, but in the background there are powerful application layer protocols that allow G:\ to represent a location on a remote server. Other examples of application layer protocols are FTP and Telnet.

## **Protocol Analysis of the Layers**

The following sections comprise a protocol analysis approach to the OSI model. They explain what each layer does and, more importantly, why. How each layer performs its function is left up to the protocol designers. I discuss how TCP/IP performs its functions in Chapters 3 through 6. More advanced readers may notice some vague or overly generic descriptions of packet descriptions in the following sections. I have written the descriptions this way to provide a generic blueprint for describing the layer's functionality; the details follow later in the book.

### ***Layer 1: The Physical Layer***

As I indicated earlier in the chapter, the physical layer concerns itself with how communications signals are transmitted across a medium. Appropriately, a medium is defined as a path where communication signals can be carried. A path is anything from copper, water, or air to even barbed wire if you can get the signals to successfully transmit over it. Media carry communication

signals. In wireless networks, signals travel over air as RF (radio frequency) radio waves. On 10BaseT Ethernet networks, they are carried as electrical voltage. In Fiber Distributed Data Interface (FDDI) networks, glass is used as the medium; the signals travel as pulses of light over glass fiber-optic cables. Many reasons exist as to why specific types of media are used in different technologies. Theoretically, you should be able to use whatever medium you want to carry the signals; unfortunately, the way those signals are represented places limitations on the types of media you can use.

### **Analog Signaling**

Communications signals are transmitted in two ways. The first method, analog, is used to transmit signals that have values that vary over time. Sound is a perfect example of an analog signal. Sound is measured as an analog signal in cycles per second or hertz. The range of the human voice varies from about 100 Hz to 1,500 Hz. When early telephone networks were developed, it was difficult to create good-quality long-distance communications using analog signals because when these analog signals were amplified there was no way to distinguish the noise from the voice signal. As the analog voice signal was amplified, so was the noise. Converting analog voice signals to digital signals was one way to solve this problem.

### **Digital Signaling**

Unlike analog signals, digital signals have only discrete values, either a one or a zero. Early digital telephone engineers figured out a way to modulate an analog signal onto a digital carrier using something called pulse code modulation, or PCM. PCM lets the instantaneous frequency of an analog signal be represented by a binary number. Instead of an amplifier having to guess at which signal to amplify, now it just had to repeat either a zero or a one. Using this method greatly improved the quality of long-distance communications. When computer data needed to be transmitted across network links, the decision to use digital signaling was easy. Since computers already represented data using zeros and ones, these zeros and ones could very easily be transmitted across networks digitally.

How these ones and zeros are represented is what digital signaling is all about. On 10BaseT Ethernet networks, data is represented by electrical voltage; a one is represented by a transition from  $-2.05$  V to 0 V and a zero is represented by a transition from 0 V to  $-2.05$  V. Over fiber-optic networks, a one might be represented by a pulse of light and a zero by the absence of light. The process isn't quite that simple, but the concept is basically the same. Different digital-signaling methods create ones and zeros on the media. Now, with the ability to have only two kinds of signals to recognize, it is much easier for amplifiers to pick out the digital ones and zeros from the background noise. With this ability to tell signals apart from noise, it became much easier to build networks capable of carrying computerized binary data over long distances.

**NOTE** There are many types of digital signaling. One of the factors that drives the type of digital signaling used in a specific technology is its efficiency and method of bit representation. For example 10-Mb Ethernet uses what is called Manchester encoding (a type of digital signaling), but for 100-Mb Fast Ethernet, Manchester was inefficient if not impossible to use because the cabling available at the time (the late 1980s) couldn't support its high bandwidth. Instead, Fast Ethernet uses what is called Non Return to Zero Inverted (NRZI) encoding and in certain configurations Multi-Level Three (MLT-3). Other data link technologies use different digital signaling methods. Token Ring uses Differential Manchester and T1 circuits use AMI or B8ZS encoding.

## ***Layer 2: The Data Link Layer***

So how do a bunch of ones and zeros become IP packets that traverse the network? For the network interface card (NIC) to put bits on the wire, it first must have a method of accessing the media. This method is called the media access method. All data link protocols designed for use in shared networks have one. One function of the media access method is letting the destination station recognize which bit is the first bit of the Media Access Control (MAC) frame. Once the first bit of the frame is found, the NIC can start grouping the ones and zeros into a Data Link Control (DLC) frame. Just as there are different methods of digital signaling, there are different types of DLC frames. In Ethernet, the IP protocol is carried by Ethernet II frames. On Token Ring, IP is carried by Token\_Ring\_SNAP frames.

**NOTE** Since the objective of this book is to learn how best to analyze TCP/IP networks, I won't detail the many frame types that exist. For more information on the various frame types, refer to *Data Link Protocols* by Ulysses Black (Prentice Hall Professional 1993).

It is important, however, to understand the basic details of Layer 2 framing. Each DLC frame has five basic parts:

- Media access portion
- Addressing
- Service access points
- Upper layer data
- Frame protection

These five basic parts are illustrated in Figure 1-2 and discussed in detail in the sections that follow.