Preface

Introduction

This book describes the TCP/IP protocol suite, but from a different perspective than other texts on TCP/IP. Instead of just describing the protocols and what they do, we'll use a popular diagnostic tool to watch the protocols in action. Seeing how the protocols operate in varying circumstances provides a greater understanding of how they work and why certain design decisions were made. It also provides a look into the implementation of the protocols, without having to wade through thousands of lines of source code.

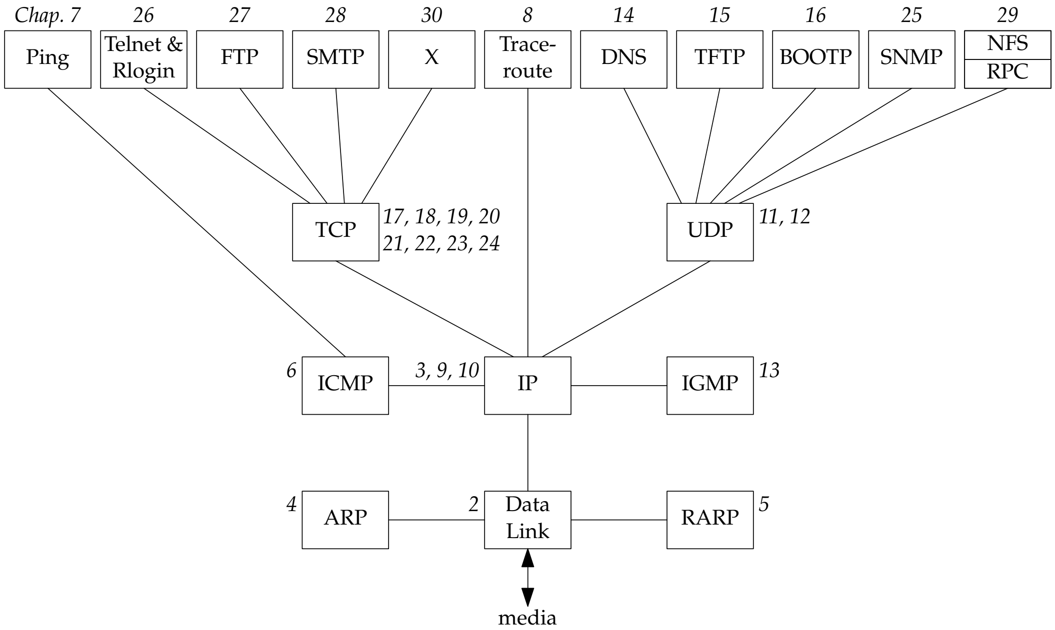
When networking protocols were being developed in the 1960s through the 1980s, expensive, dedicated hardware was required to see the packets going "across the wire." Extreme familiarity with the protocols was also required to comprehend the packets displayed by the hardware. Functionality of the hardware analyzers was limited to that built in by the hardware designers.

Today this has changed dramatically with the ability of the ubiquitous workstation to monitor a local area network [Mogul 1990]. Just attach a workstation to your network, run some publicly available software (described in Appendix A), and watch what goes by on the wire. While many people consider this a tool to be used for *diagnosing* network problems, it is also a powerful tool for *understanding* how the network protocols operate, which is the goal of this book.

This book is intended for anyone wishing to understand how the TCP/IP protocols operate: programmers writing network applications, system administrators responsible for maintaining computer systems and networks utilizing TCP/IP, and users who deal with TCP/IP applications on a daily basis.

Organization of the Book

The following figure shows the various protocols and applications that are covered. The italic number by each box indicates the chapter in which that protocol or application is described.



(Numerous fine points are missing from this figure that will be discussed in the appropriate chapter. For example, both the DNS and RPC use TCP, which we don't show.)

We take a bottom-up approach to the TCP/IP protocol suite. After providing a basic introduction to TCP/IP in Chapter 1, we will start at the link layer in Chapter 2 and work our way up the protocol stack. This provides the required background for later chapters for readers who aren't familiar with TCP/IP or networking in general.

This book also uses a functional approach instead of following a strict bottom-to-top order. For example, Chapter 3 describes the IP layer and the IP header. But there are numerous fields in the IP header that are best described in the context of an application that uses or is affected by a particular field. Fragmentation, for example, is best understood in terms of UDP (Chapter 11), the protocol often affected by it. The time-to-live field is fully described when we look at the Traceroute program in Chapter 8, because this field is the basis for the operation of the program. Similarly, many features of ICMP are described in the later chapters, in terms of how a particular ICMP message is used by a protocol or an application.

We also don't want to save all the good stuff until the end, so we describe TCP/IP applications as soon as we have the foundation to understand them. Ping and Traceroute are described after IP and ICMP have been discussed. The applications built on UDP (multicasting, the DNS, TFTP, and BOOTP) are described after UDP has been examined. The TCP applications, however, along with network management, must be saved until the end, after we've thoroughly described TCP. This text focuses on how these applications use the TCP/IP protocols. We do not provide all the details on running these applications.

Readers

This book is self-contained and assumes no specific knowledge of networking or TCP/IP. Numerous references are provided for readers interested in additional details on specific topics.

This book can be used in many ways. It can be used as a self-study reference and covered from start to finish by someone interested in all the details on the TCP/IP protocol suite. Readers with some TCP/IP background might want to skip ahead and start with Chapter 7, and then focus on the specific chapters in which they're interested. Exercises are provided at the end of the chapters, and most solutions are in Appendix D. This is to maximize the usefulness of the text as a self-study reference.

When used as part of a one- or two-semester course in computer networking, the focus should be on IP (Chapters 3 and 9), UDP (Chapter 11), and TCP (Chapters 17−24), along with some of the application chapters.

Many forward and backward references are provided throughout the text, along with a thorough index, to allow individual chapters to be studied by themselves. A list of all the acronyms used throughout the text, along with the compound term for the acronym, appears on the inside back covers.

If you have access to a network you are encouraged to obtain the software used in this book (Appendix F) and experiment on your own. Hands-on experimentation with the protocols will provide the greatest knowledge (and make it more fun).

Systems Used for Testing

Every example in the book was run on an actual network and the resulting output saved in a file for inclusion in the text. Figure 1.11 (p. 18) shows a diagram of the different hosts, routers, and networks that are used. (This figure is also duplicated on the inside front cover for easy reference while reading the book.) This collection of networks is simple enough that the topology doesn't confuse the examples, and with four systems acting as routers, we can see the error messages generated by routers.

Most of the systems have a name that indicates the type of software being used: `bsdi`, `svr4`, `sun`, `solaris`, `aix`, `slip`, and so on. In this way we can identify the type of software that we're dealing with by looking at the system name in the printed output.

A wide range of different operating systems and TCP/IP implementations are used:

- BSD/386 Version 1.0 from Berkeley Software Design, Inc., on the hosts named `bsdi` and `slip`. This system is derived from the BSD Networking Software, Release 2.0. (We show the lineage of the various BSD releases in Figure 1.10 on p. 17.)

- Unix System V/386 Release 4.0 Version 2.0 from U.H. Corporation, on the host named `svr4`. This is vanilla SVR4 and contains the standard implementation of TCP/IP from Lachman Associates used with most versions of SVR4.
- SunOS 4.1.3 from Sun Microsystems, on the host named `sun`. The SunOS 4.1.x systems are probably the most widely used TCP/IP implementations. The TCP/IP code is derived from 4.2BSD and 4.3BSD.
- Solaris 2.2 from Sun Microsystems, on the host named `solaris`. The Solaris 2.x systems have a different implementation of TCP/IP from the earlier SunOS 4.1.x systems, and from SVR4. (This operating system is really SunOS 5.2, but is commonly called Solaris 2.2.)
- AIX 3.2.2 from IBM on the host named `aix`. The TCP/IP implementation is based on the 4.3BSD Reno release.
- 4.4BSD from the Computer Systems Research Group at the University of California at Berkeley, on the host `vangogh.cs.berkeley.edu`. This system has the latest release of TCP/IP from Berkeley. (This system isn't shown in the figure on the inside front cover, but is reachable across the Internet.)

Although these are all Unix systems, TCP/IP is operating system independent, and is available on almost every popular non-Unix system. Most of this text also applies to these non-Unix implementations, although some programs (such as Traceroute) may not be provided on all systems.

Typographical Conventions

When we display interactive input and output we'll show our typed input in a **`bold font`**, and the computer output `like this`. *Comments are added in italics.*

```
bsdi % telnet svr4 discard          connect to the discard server
Trying 140.252.13.34...             this line and next output by Telnet client
Connected to svr4.
```

Also, we always include the name of the system as part of the shell prompt (`bsdi` in this example) to show on which host the command was run.

> Throughout the text we'll use indented, parenthetical notes such as this to describe historical points or implementation details.

We sometimes refer to the complete description of a command in the Unix manual as in `ifconfig`(8). This notation, the name of the command followed by a number in parentheses, is the normal way of referring to Unix commands. The number in parentheses is the section number in the Unix manual of the "manual page" for the command, where additional information can be located. Unfortunately not all Unix systems organize their manuals the same, with regard to the section numbers used for various groupings of commands. We'll use the BSD-style section numbers (which is the same for BSD-derived systems such as SunOS 4.1.3), but your manuals may be organized differently.

Acknowledgments

Although the author's name is the only one to appear on the cover, the combined effort of many people is required to produce a quality text book. First and foremost is the author's family, who put up with the long and weird hours that go into writing a book. Thank you once again, Sally, Bill, Ellen, and David.

The consulting editor, Brian Kernighan, is undoubtedly the best in the business. He was the first one to read various drafts of the manuscript and mark it up with his infinite supply of red pens. His attention to detail, his continual prodding for readable prose, and his thorough reviews of the manuscript are an immense resource to a writer.

Technical reviewers provide a different point of view and keep the author honest by catching technical mistakes. Their comments, suggestions, and (most importantly) criticisms add greatly to the final product. My thanks to Steve Bellovin, Jon Crowcroft, Pete Haverlock, and Doug Schmidt for comments on the entire manuscript. Equally valuable comments were provided on portions of the manuscript by Dave Borman, Tony DeSimone, Bob Gilligan, Jeff Gitlin, John Gulbenkian, Tom Herbert, Mukesh Kacker, Barry Margolin, Paul Mockapetris, Burr Nelson, Steve Rago, James Risner, Chris Walquist, Phil Winterbottom, and Gary Wright. A special thanks to Dave Borman for his thorough review of all the TCP chapters, and to Bob Gilligan who should be listed as a coauthor for Appendix E.

An author cannot work in isolation, so I would like to thank the following persons for lots of small favors, especially by answering my numerous e-mail questions: Joe Godsil, Jim Hogue, Mike Karels, Paul Lucchina, Craig Partridge, Thomas Skibo, and Jerry Toporek.

This book is the result of my being asked lots of questions on TCP/IP for which I could find no quick, immediate answer. It was then that I realized that the easiest way to obtain the answers was to run small tests, forcing certain conditions to occur, and just watch what happens. I thank Pete Haverlock for asking the probing questions and Van Jacobson for providing so much of the publicly available software that is used in this book to answer the questions.

A book on networking needs a real network to work with along with access to the Internet. My thanks to the National Optical Astronomy Observatories (NOAO), especially Sidney Wolff, Richard Wolff, and Steve Grandi, for providing access to their networks and hosts. A special thanks to Steve Grandi for answering lots of questions and providing accounts on various hosts. My thanks also to Keith Bostic and Kirk McKusick at the U.C. Berkeley CSRG for access to the latest 4.4BSD system.

Finally, it is the publisher that pulls everything together and does whatever is required to deliver the final product to the readers. This all revolves around the editor, and John Wait is simply the best there is. Working with John and the rest of the professionals at Addison-Wesley is a pleasure. Their professionalism and attention to detail show in the end result.

Camera-ready copy of the book was produced by the author, a Troff die-hard, using the Groff package written by James Clark. I welcome electronic mail from any readers with comments, suggestions, or bug fixes.

Tucson, Arizona W. Richard Stevens
October 1993 `rstevens@noao.edu`