INTERNETWORKING WITH TCP/IP

PRINCIPLES, PROTOCOLS, AND ARCHITECTURE

Prentice-Hall International Editions

DOUGLAS COMER

Internetworking With TCP/IP
Principles, Protocols, and Architecture

DOUGLAS E. COMER

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

Prentice-Hall International, Inc.

0-13-470188-7

A Division of Simon & Schuster
Englewood Cliffs, NJ 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall, Inc., *Englewood Cliffs, New Jersey*

Foreword

In this book Professor Douglas Comer has provided a long sought overview and introduction to TCP/IP. There have been many requests for *the* article, report, or book to read to get started on understanding the TCP/IP protocols. At last, this book satisfies those requests. Writing an introduction to TCP/IP for the uninitiated is a very difficult task. While combining the explanation of the general principles of computer communication with the specific examples from the TCP/IP protocol suite, Doug Comer has provided a very readable book.

While this book is specifically about the TCP/IP protocol suite, it is a good book for learning about computer communications protocols in general. The principles of architecture, layering, multiplexing, encapsulation, addressing and address mapping, routing, and naming are quite similar in any protocol suite, though, of course, different in detail (See Chapters 3, 10, 17, and 18).

Computer communication protocols do not do anything themselves. Like operating systems, they are in the service of application processes. Processes are the active elements that request communication and are the ultimate senders and receivers of the data transmitted. The various layers of protocols are like the various layers in a computer operating system, especially the file system. Understanding protocol architecture is like understanding operating system architecture. In this book Doug Comer has taken the ''bottom up'' approach – starting with the physical networks and moving up in levels of abstraction to the applications.

Since application processes are the active elements using the communication supported by the protocols, TCP/IP is an ''interprocess communication'' (IPC) mechanism. While there are several experiments in progress with operating system style message passing and procedure call types of IPC based on IP, the focus in this book is on more traditional applications that use the UDP datagram or TCP logical connection forms of IPC (See Chapters 11, 12, 17, 18, and 19).

One of the key ideas inherent in TCP/IP and in the title of this book is ''internetworking.'' The power of a communication system is directly related to the number of entities in that system. The telephone network is very useful because (nearly) all the telephones are in (as it appears to the users) one network. Computer communication systems and networks are currently separated and fragmented. The goal of interconnection and internetworking, to have a single powerful computer communication network, is fundamental to the design of TCP/IP. Essential to internetworking is addressing (See Chapters 4, 5, and 6), and a universal protocol – the Internet Protocol (See Chapters 7, 8, and 9).

To have an internetwork the individual networks must be connected. The connecting devices are called gateways. Further, these gateways must have some procedures for forwarding data from one network to the next. The data is in the form of IP datagrams and the destination is specified by an IP address, but the gateway must make a routing decision based on the IP address and what it knows about the connectivity of the networks making up the Internet. The procedures for distributing the current connectivity information to the gateways are called routing algorithms, and these are currently the subject of much study and development (See Chapters 13, 14, 15, and 16).

Like all communication systems, the TCP/IP protocol suite is an unfinished system. It is evolving to meet changing requirements and new opportunities. Thus, this book is, in a sense, a snapshot of TCP/IP circa 1987. And, as Doug Comer points out, there are many loose ends (See Chapter 20).

Most chapters end with a few pointers to material ''for further study''. Many of these refer to memos of the RFC series of notes. This series of notes is the result of a policy of making the working ideas and the protocol specifications developed by the TCP/IP research and development community widely available. This availability of the basic and detailed information about these protocols, and the availability of the early implementations of them, has had much to do with their current widespread use. This commitment to public documentation at this level of detail is unusual for a research effort, and has had significant benefits for the development of computer communication (See Appendix 3).

This book brings together information about the various parts of the TCP/IP architecture and protocols and makes it accessible. Its publication is a very significant milestone in the evolution of computer communications.

Jon Postel,
Internet Protocol Designer and
Deputy Internet Architect

December, 1987

Preface

In the last century, railroads revolutionized the world by providing a transportation network that moved raw materials and manufactured products. They made an industrialized society possible. Digital communication networks have started a new revolution by providing the technology that transports the data needed by a society in which information plays a key role. Networking already permeates industry, education, and government. It has already begun to change the way we view the world by shrinking geographic distances and forming new communities of people who interact frequently. More important, network growth is explosive. The revolution is well underway.

To understand networking and the selection of topics discussed in this book, it is important to realize that network research and development occurred in three stages. Before the 1960s, the main question was, "How can we transmit bits across a communication medium efficiently and reliably?" The results include the development of information theory, the sampling theorem, and other ideas commonly referred to as signal processing. Beginning around the mid 1960s, emphasis shifted to packet switching and the question became, "How can we transmit packets across a communication medium efficiently and reliably?" The results include the development of packet switching technologies, local area networks, and statistical analysis of network response to load. From approximately the mid 1970s to the present, emphasis has centered on network architecture and the question, "How can we provide communication services across a series of interconnected networks?" The results include the development of internetwork technologies, protocol layering models, datagram and stream transport services, and the client-server interaction paradigm.

Most textbooks and network courses concentrate on the first two stages of network research, presenting the well-known theories of data communications and queueing analysis. Although such information is important to engineers who design network technologies and hardware products, most network architects purchase commercially available network hardware. Instead of detailed knowledge about how bits or packets flow across communication media, they need to know how to interconnect such hardware and how to use the resulting system.

This text concentrates on the third stage of networking. It examines the architecture of interconnected networks and explains the principles and protocols that make such interconnected architectures function as a single unified communication system. More important, it shows how an interconnected architecture can be used for distributed computation.

The entire text focuses on the concept of internetworking in general and the TCP/IP internet technology in particular. Internetworking is a powerful abstraction that allows us to deal with the complexity of multiple underlying communication technologies. It hides the details of network hardware and provides a high level communication environment. As the book shows, the ultimate goal of internetworking is maximal interoperability, that is, maximizing the ability of programs on diverse computer and network systems to communicate reliably and efficiently.

The text reviews both the architecture of network interconnections as well as internet communication services and the protocols needed to provide those services. By the end of the book, the reader will understand how it is possible to interconnect multiple physical networks into a coordinated system, how internet protocols operate in that environment, and how application programs use the resulting system. As a specific example, the reader learns the details of the DARPA (TCP/IP) Internet, including the architecture of the gateway system and the application protocols it supports. In addition, the book discusses some of the limitations of the internet approach.

Writing about internetworking is both exciting and challenging. It is challenging because, as in any rapidly changing research area, nothing is stable. It is exciting because the TCP/IP Internet is an active, rapidly expanding entity. Researchers working on it generate new ideas constantly and the possibilities seem endless. Looking back over TCP/IP and the DARPA Internet makes it clear that much has been accomplished. Knowing that the research has taken a little over a decade makes one realize how intense the effort has been.

Designed as both a college text and as a professional reference, the book is written at an advanced undergraduate or graduate level. For professionals, the book provides a comprehensive introduction to the TCP/IP technology and the architecture of the Internet. Although it is not intended to replace protocol standards, the book is a good starting point for learning about internetworking because it provides a uniform overview that emphasizes principles. Moreover, it gives the reader perspective that can be extremely difficult to obtain from individual protocol documents.

When used in the classroom, the text provides more than sufficient material for a single semester network course at either the undergraduate or graduate level. Such a course can be extended to a 2-semester sequence if accompanied by programming projects and readings from the literature. For undergraduate courses, it can be taken at face value. Students should be expected to grasp the basic concepts described in the text, and they should be able to describe or use them. At the graduate level, students should be expected to use the material here as a basis for further exploration of current research. They should understand it well enough to answer exercises or solve problems that require them to explore subtleties and consequences. Many of the exercises suggest such subtleties; solving them often requires students to read protocol standards and apply creative energy to comprehend consequences.

At all levels, hands-on experience sharpens the concepts and helps students gain intuition. Thus, I encourage instructors to invent projects that force students to use internetwork services and protocols. Although such experimentation is safest when the instructional laboratory network is isolated from production computing facilities, we have found that students exhibit the most enthusiasm, and benefit the most, when they have access to the ''real'' TCP/IP Internet.

The book is organized into four main parts. Chapters 1 and 2 form an introduction that provides an overview and discusses existing technologies. In particular, Chapter 2 reviews physical network hardware. The intention is to provide basic intuition about what is possible, not to spend inordinate time on hardware details. Chapters 3-12 describe the TCP/IP Internet from the viewpoint of a single host, showing the basic services available and the protocols a host uses to access them. They cover the basics of Internet addressing and routing as well as the notion of protocol layering. Chapters 13-16 describe the architecture of the Internet when viewed globally. They explore the core gateway system and the protocols gateways use to exchange routing information. Finally, Chapters 17-19 discuss application level services available in the Internet. They present the client-server model of interaction and give several examples of how one can organize client and server software. The last section discusses electronic mail and the domain name system, two topics that are extremely popular.

The chapters have been organized "bottom up." They begin with an overview of hardware and continue to build new functionality on top of it. This view will appeal to anyone who has developed Internet software because it follows the same pattern one uses in implementation. The concept of layering does not appear until Chapter 10. The discussion of layering emphasizes the distinction between conceptual layers of functionality and the reality of layered protocol software in which multiple objects appear at each layer.

Although it is difficult to omit any chapter completely, the instructor will find that students are often satisfied to know that something is possible without knowing the details. For example, one can skim through Chapters 5, 6, and 9 by covering only the functionality and not the details of the protocols. In addition, several chapters (especially 16) contain engineering techniques. While such techniques are crucial to efficient implementations, they can be skipped to save time.

A modest background is required to understand the material. The reader is expected to have programmed in a high level language and to be familiar with basic data structures like stacks, queues, and trees. Readers need basic intuition about the organization of computer software into an operating system that supports concurrent programming and application programs that users invoke to perform computation. Readers do not need sophisticated mathematics, nor do they need to know information theory or theorems from data communications; the book describes the physical network as a black box around which an internetwork can be built. It states design principles in English and discusses motivations and consequences.

Many people have contributed to this book. I thank Steve Chapin, Jim Griffioen, Chris Kent, Tim Korb, Dan Lynch, Thomas Narten, Shawn Ostermann, John Steele, and Raj Yavatkar, who all read drafts and made valuable comments. Craig Partridge supplied numerous suggestions, including a few exercises, and corrected several technical errors. He and Van Jacobson supplied the graph of Internet round trip delays in Chapter 12. Barry Shein graciously allowed me to use his example UNIX client and server code in Appendix 1. Charlotte Tubis provided valuable editing. Special thanks go to my wife, Chris, who has read the text more times than I can count and made extensive suggestions.

Contents

1

Introduction and Overview

1.1 The Need For An Internet

Data communication has become a fundamental part of computing. World-wide networks gather data about such diverse subjects as atmospheric conditions, crop production, and airline traffic. Groups establish electronic mailing lists so they can share information of common interest. Hobbyists exchange programs for their home computers. In the scientific world, data networks are essential because they allow scientists to send programs and data to remote supercomputers for processing, to retrieve the results, and to exchange scientific information with colleagues.

Unfortunately, most networks are independent entities, established to serve the needs of a single group. The users choose a hardware technology appropriate to their communication problems. More important, it is impossible to build a universal network from a single hardware technology, because no single network suffices for all uses. Some users need a high-speed network to connect machines, but such networks cannot be expanded to span large distances. Others settle for a slower speed network that connects machines thousands of miles apart.

Recently, however, a new technology has emerged that makes it possible to interconnect many disparate physical networks and make them function as a coordinated unit. The new technology, called an *internetwork*, or *internet*, accommodates multiple, diverse underlying hardware technologies by adding both physical connections and a new set of conventions. The internet technology hides the details of network hardware and permits computers to communicate independent of their physical network connections.

To appreciate internet technology, think of how it affects research. Imagine for a minute the effects of interconnecting all the computers used by scientists. Any scientist would be able to exchange data resulting from an experiment with any other scientist. It would be possible to establish national data centers to collect data from natural phenome-