大学计算机教育丛书（影印版）网络互连技术系

Second Edition

BSD Socket Version with ANSI C

INTERNETWORKING WITH TCP/IP VOLUME III

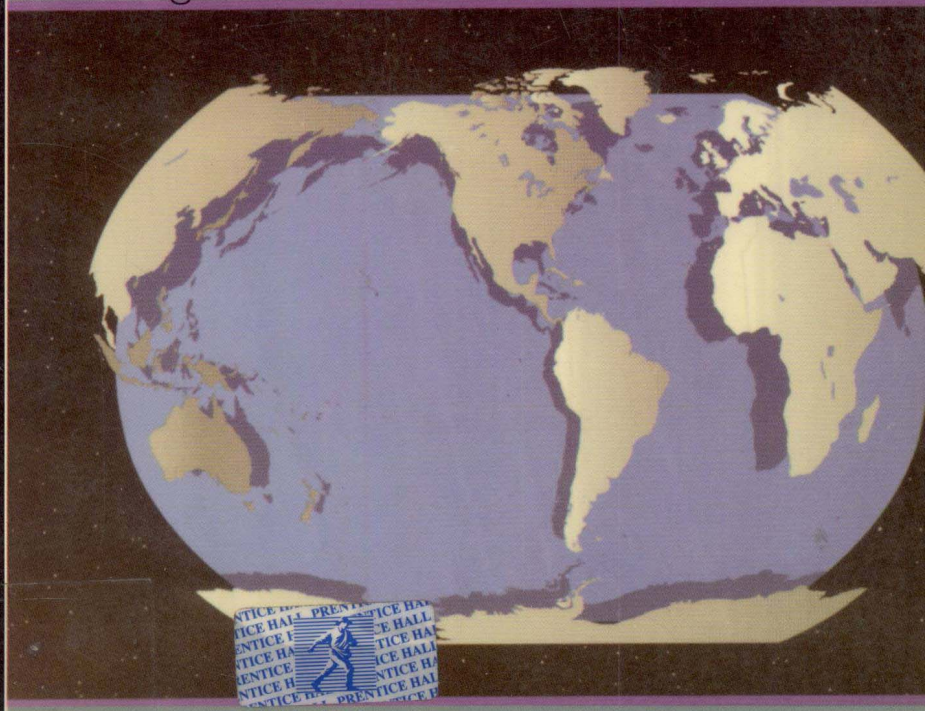
CLIENT-SERVER PROGRAMMING AND APPLICATIONS

TCP/IP 网络互连技术

卷III：客户服务器编程和应用

BSD套接字版　第2版

Douglas E. Comer · David L. Stevens



清华大学出版社 · PRENTICE HALL
http://www.tup.tsinghua.edu.cn

Internetworking With TCP/IP

Vol Ⅲ:

Client-Server Programming And Applications BSD Socket Version

Second Edition

TCP/IP 网络互连技术

卷Ⅲ

客户/服务器编程和应用BSD套接字版

第2版

DOUGLAS E. COMER

Prentice-Hall International, Inc.

(京) 新登字 158 号

Internetworking with TCP/IP Vol Ⅲ: Client-Server programming and applications, BSD Socket Version, 2nd ed./Douglas E. Comer, David L. Stevens

北京市版权局著作权合同登记号: 01-98-0959

图书在版编目(CIP)数据

TCP/IP 网络互连技术 卷Ⅲ: BSD 套接字版: 第 2 版: 英文/(美)科默(Comer, D. E.), 史蒂文斯(Stevens, D. L.). - 影印版. - 北京: 清华大学出版社, 1998.7
(大学计算机教育丛书)
ISBN 7-302-02948-2

Ⅰ.T… Ⅱ.①科… ②史… Ⅲ.计算机网络-连接技术-英文 Ⅳ.TP393

中国版本图书馆 CIP 数据核字(98)第 09324 号

出版者: 清华大学出版社(北京清华大学校内, 邮编 100084)
http://www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京科技发行所
开 本: 850×1168 1/32 印张: 17.125
版 次: 1998 年 9 月第 1 版 1998 年 12 月第 2 次印刷
书 号: ISBN 7-302-02948-2/TP・1559
印 数: 3001～8000
定 价: 32.00 元

出 版 前 言

清华大学出版社与 Prentice Hall 出版公司合作推出的“大学计算机教育丛书(影印版)”和“ATM 与 B-ISDN 技术丛书(影印版)”受到了广大读者的欢迎。很多读者通过电话、信函、电子函件给我们的工作以积极的评价,并提出了不少中肯的建议。其中,很多读者希望我们能够出版一些网络方面较深层次的书籍,这也就成为我们出版这套“网络互连技术系列”的最初动机。

众所周知,网络协议是网络与通信技术的关键组成部分。而今,因特网技术、移动通信技术的飞速发展,为网络协议注入了新内容。本套丛书以 Duglas Comer 教授的网络协议的经典名著 TCP/IP 网络互连技术系列为主干,并补充以论述新协议如 IPV6 和移动 IP 等国外最新专著,力求为从事网络互连技术研究与开发的人员以及大专院校师生提供充分的技术支持。

衷心希望所有阅读这套丛书的读者能从中受益。

清华大学出版社
Prentice Hall 公司

1998.9

Foreword

It is indeed a pleasure to introduce the reader to the revised third volume of Dr. Douglas E. Comer's remarkable series: Internetworking with TCP/IP. This series, which began so innocently back in 1987, is now the premiere source for learning about the suite of protocols that have made vendor-independent computer-communications possible – the Internet suite of protocols.

To my mind, this seminal work is our best hope against the "dumbing down of the Internet." Whilst the media and entrepreneurs fill the popular imagination with visions of "Internet mysticism," it is Dr. Comer who clearly explains the technical reality of the technology that makes the Internet possible.

Although I have learned from all three books in the series, I feel that Volume *3*, *Client-Server Programming and Applications*, which Doug has authored with David L. Stevens, is particularly relevant to the Internet today. It teaches us how to architect and build client-server applications, and – more importantly – how to understand what trade-offs are involved with each design decision.

So, I invite you to undertake a memorable journey into "how's and why's" of the theory, design, and realization of internetworking technology.

Marshall T. Rose
Theorist, Implementor, and Agent Provocateur
Del Mar, California

Preface

We are pleased to introduce a revised version of Volume 3 in the Internetworking Series. Broadly speaking, Volume *1* examines the question, "What is a TCP/IP internet?" Volume *2* examines the question, "How does TCP/IP software work?" It presents more details and explores greater depth than the first volume. This volume examines the question, "How does application software use TCP/IP?" It focuses on the client-server paradigm, and examines algorithms for both the client and server components of a distributed program. It shows an implementation that illustrates each design, and discusses techniques including application-level gateways and tunneling. In addition, it reviews several standard application protocols, and uses them to illustrate the algorithms and implementation techniques.

The revision follows the latest standards. For example, code in examples has been rewritten to use ANSI C, and the chapter on NFS discusses changes in version *3*. In addition, new sections have been added to explain concepts behind programs like *slirp* that provide Internet access across a dialup telephone connection. The discussion of ways client-server systems fail has been expanded: an entire new chapter focuses on deadlock and livelock. The chapter examines causes of the problems and techniques for preventing them. Finally, minor typos and ambiguities in wording have been corrected throughout the text.

The code is available on-line. To access a copy via the Web, look for Volume *3* in the list of networking books at location:

http://www.cs.purdue.edu/homes/comer/books.html

To access the code via FTP, use location:

ftp://ftp.cs.purdue.edu/pub/Xinu/TCPIP-vol3.bsd.dist.tar.Z

The organization of the text remains the same as the previous version. Beginning chapters introduce the client-server paradigm and the socket interface that application programs use to access TCP/IP protocol software. They also describe concurrent processes and the operating system functions used to create them. Chapters that follow the introductory material discuss client and server designs.

The text explains that the myriad of possible designs are not random. Instead, they follow a pattern that can be understood by considering the choice of concurrency and transport. For example, one chapter discusses a nonconcurrent server design that uses connection-oriented transport (e.g., TCP), while another discusses a similar design that uses connectionless transport (e.g., UDP).

We describe how each design fits into the space of possible implementations, but do not try to develop an abstract "theory" of client-server interactions. Instead, we emphasize practical design principles and techniques that are important to programmers. Each technique has advantages in some circumstances, and each has been used in working software. We believe that understanding the conceptual ties among the designs will help the reader appreciate the strengths and weaknesses of each approach, and will make it easier to choose among them.

The text contains example programs that show how each design operates in practice. Most of the examples implement standard TCP/IP application protocols. In each case, we tried to select an application protocol that would convey a single design idea without being too complex to understand. Thus, while few of the example programs are exciting, they each illustrate one important concept. This version of Volume *3* uses the BSD UNIX socket mechanism in all programming examples; a companion edition contains the same examples using AT&T's TLI protocol interface.

Later chapters discuss the remote procedure call concept and describe how it can be used to construct distributed programs. They relate the remote procedure call technique to the client-server model, and show how software can be used to generate client and server programs from a remote procedure call description. The chapters on TELNET show how small details dominate a production program and how complex the code can become for even a simple, character-oriented protocol.

Much of the text concentrates on concurrent processing. Many of the concepts described may seem familiar to students who have written concurrent programs because they apply to all concurrent programs, not only network applications. Students who have not written concurrent programs may find the concepts difficult.

The text is suitable for a single semester introductory networking course at the senior or graduate level. Because the text concentrates on how to use an internet rather than on how it works, students need little background in networking to understand the material. No particular concept is too difficult for lower level courses as long as the instructor proceeds at a suitable pace. A basic course in operating systems concepts or experience with concurrent programming may provide the best background.

Students will not appreciate the material until they use it first hand. Thus, any course should have programming exercises that force the students to apply the ideas to practical programs. Undergraduates can learn the basics by repeating the designs on other application protocols. Graduate students should build more complex distributed programs that emphasize some of the subtle techniques (e.g., the concurrency management techniques in Chapter *15* and the interconnection techniques in Chapter *17*).

Many people deserve credit for their help. Members of the Internet Research Group at Purdue contributed technical information and suggestions to the original text. Christine Comer edited the revision and improved both wording and consistency.

Douglas E. Comer
David L. Stevens

Contents