大学计算机教育丛书（影印版）

网络互连技术系

Second Edition

Internetworking with TCP/IP

VOLUME II

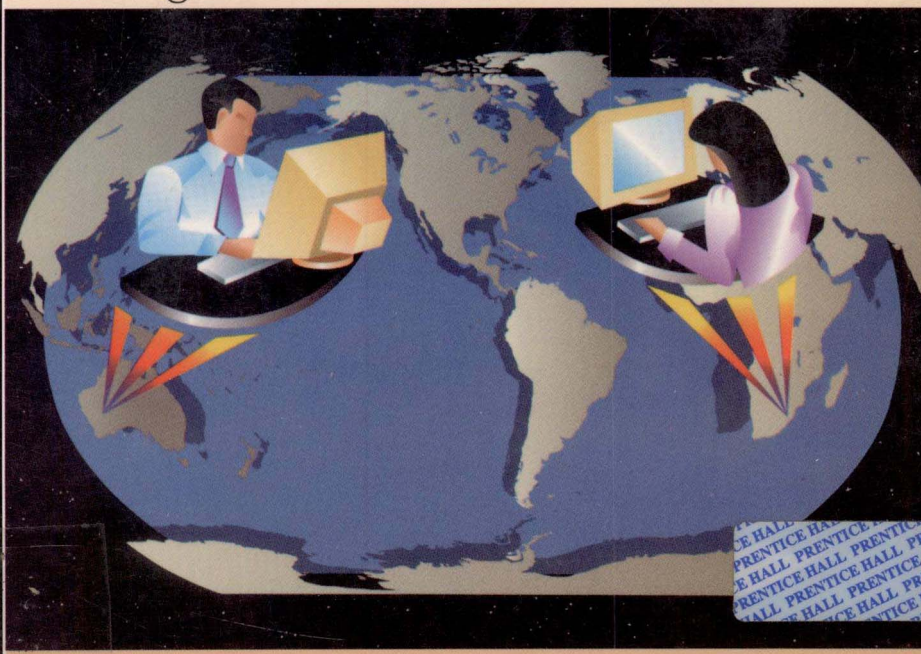
Design, Implementation, and Internals

TCP/IP 网络互连技术

卷 II

设计与实现

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 Ⅱ:

Design, Implementation, and Internals

Second Edition

TCP/IP 网络互连技术

卷Ⅱ

设计、实现与内部

第2版

DOUGLAS E. COMER

and

DAVID L. STEVENS

Prentice-Hall International, Inc.

(京)新登字 158 号

Internetworking with TCP/IP Vol Ⅱ: design, implementation, and internals 2nd ed./Douglas E. Comer, David L. Stevens

北京市版权局著作权合同登记号:01-98-0958

图书在版编目(CIP)数据

TCP/IP 网络互连技术 卷Ⅱ:第 2 版:英文科默(Comer, D. E.),史蒂文斯(Stevens, D. L.)著. - 影印版. - 北京:清华大学出版社,1998.7
(大学计算机教育丛书)
ISBN 7-302-02947-4

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 09297 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:新华书店总店北京发行所
开 本:850×1168 1/32 印张:19.75
版 次:1998 年 8 月第 1 版 1999 年 2 月第 2 次印刷
书 号:ISBN 7-302-02947-4/TP·1558
印 数:5001~10000
定 价:30.00 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

About the Authors

Dr. Douglas Comer is a full professor of Computer Science at Purdue University, where he teaches courses on operating systems and computer networks. He has written numerous research papers and textbooks, and currently heads several networking research projects. He has been involved in TCP/IP and internetworking since the late 1970s, and is an internationally recognized authority. He designed and implemented X25NET and Cypress networks, and the Xinu operating system. He is director of the Internetworking Research Group at Purdue, editor-in-chief of the *Journal of Internetworking*, editor of *Software - Practice and Experience*, and a former member of the Internet Architecture Board.

David Stevens is a programmer for the Purdue University Computing Center, where he has done UNIX and Networking development since 1986. He is currently a member of the Purdue Data Networking Group, he is co-author of several computer networking textbooks, and is a member of the Internetworking Research Group at Purdue. He holds a Master's Degree in Computer Science from Purdue University.

Foreword

It is an honor to take this space to share some thoughts with you as the second edition of this very popular text is published. In the time between the first and second edition, the Internet has grown exponentially, as has the level of interest and awareness in its existence. Readers might be interested to know that twice as many private networks have been fielded as have been connected to the global Internet. There are, as of this writing, 26,000 networks linked to the global Internet and on the order of twice that many that use the technology of TCP/IP, but are private.

Among the original design principles behind the TCP/IP protocols and their predecessors, the ARPANET host-host protocol, was the notion that most communication would take place between processes in separate computers somewhere in the Internet. Moreover, many processes might be housed in each host, each of them in contact with one or more other processes elsewhere in the Net. We can see solid evidence that this model is increasingly important. One need only read the trade press and, in fact, some popular press, to find that intelligent agents have become a major new component in the Internet environment. Control of these processes and support for their interaction with each other and with distributed databases around the Internet leads to some very high estimates of data traffic as the network population increases.

The TCP/IP protocols face some very important architectural challenges as the Internet continues its phenomenal expansion. Handling 100% growth in the number of networks annually puts a strain on the routing system. In the longer term, the address space itself will be stressed. Already, the choice class B address space is under pressure and the use of Classless Inter-Domain Routing (CIDR) techniques is only a temporary palliative. A new version of the IP protocol is needed to support much larger address spaces and improved handling of the general scaling problem.

At the same time, new and very ambitious applications are already making demands on the capacity of the Internet to support packet voice and video in ever-increasing quantities. Many hope that the new cellular switching technologies such as Asynchronous Transfer Mode (ATM) will make it possible to serve substantial amounts of these kinds of applications.

Security, too, is a major issue, especially as the network expands and is more applied to business use. Finding a uniform design to support the Internet and still deal with a variety of technologies world-wide, some of which are subject to export control, is a challenge of major proportions.

In such a fertile, rapidly-changing and innovative environment, it is essential to understand and internalize the basic function of the Internet protocols, especially to prepare for the exciting work ahead, guiding the evolution of these popular standards towards their use in the 21st century. I hope you find that this volume contributes to your depth of understanding and appreciation of the Internet and its technologies.

Vinton Cerf
President
Internet Society
Annandale, VA

Preface

We published the first edition of *Internetworking With TCP/IP Volume 2* in response to readers who asked for more details about TCP/IP protocols than Volume *1* contains. Volume *2* places TCP/IP under a magnifying glass, and examines the details of individual protocols. It discusses their implementation, and focuses on the internals of protocol software. The second edition updates several protocols and adds two new chapters that discuss the IGMP protocol used for IP multicasting and the OSPF routing protocol. Our implementation of TCP urgent data has been changed to illustrate the data-mark interpretation, and the text discusses the consequences.

The official specifications for individual protocols, as well as discussions of their implementation and use, appear in Request For Comments documents (RFCs). Although some RFCs can be difficult for beginners to understand, they remain the authoritative source of detailed information; no author can hope to reproduce all that information in a textbook. While the RFCs cover individual protocols, however, they sometimes leave unanswered questions about the interactions among protocols. For example, a routing protocol such as RIP specifies how a gateway installs routes in an IP routing table, and how the gateway advertises routes in its table to other gateways. RIP also specifies that routes must be timed out and removed. But the interaction between RIP and other protocols may not be apparent from the RFC. The question arises, "how does route timeout affect routes in the table that were not installed by RIP?" One must also consider the question, "should RIP updates override routes that the manager installs manually?"

To help explain the interaction among protocols and to insure that our solutions fit together, we designed and built a working system that serves as a central example throughout the text. The system provides most of the protocols in the TCP/IP suite, including: TCP, IP, ICMP, IGMP, UDP, ARP, RIP, SNMP, and a significant part of OSPF. In addition, it has an example client and server for the finger service. Because the text contains the code for each protocol, the reader can study the implementation and understand its internal structure. Most important, because the example system integrates the protocol software into a working whole, the reader can clearly understand the interaction among protocols.

The example code attempts to conform to the protocol standards and to include current ideas. For example, our TCP code includes silly window avoidance and the Jacobson-Karels slow-start and congestion avoidance optimizations, features sometimes missing from commercial implementations. However, we are realistic enough to realize that the commercial world does not always follow the published standards, and have

tried to adapt the system for use in a practical environment. For example, the code includes a configuration parameter that allows it to use either the Internet standard or BSD UNIX implementation of TCP's urgent data pointer.

We do not claim that the code presented here is bug-free, or even that it is better than other implementations. Indeed, after many years of using it, we continue to find ways to improve the software, and hope that readers will look for them as well. To help, the publisher has agreed to make machine-readable copies of all the code available, so readers can use computer tools to examine, modify, and test it. The archive is available via anonymous FTP from file *pub/comer/v2.dist.tar.Z* on computer *ftp.cs.purdue.edu*.

The text can be used in an upper-division course on networking or in a graduate course. Undergraduate courses should focus on the earlier chapters, omitting the chapters on *OSPF*, *SNMP* and *RIP*. Graduate students will find the most interesting and challenging concepts in the chapters on TCP. Adaptive retransmission and the related heuristics for high performance are especially important and deserve careful attention. Throughout the text, exercises suggest alternative implementations and generalizations; they rarely call for rote repetition of the information presented. Thus, students may need to venture beyond the text to solve many of the exercises.

As in any effort this size, many people share the credit; we thank them. David Stevens, one of the authors, implemented most of the software, including a complete version of TCP. Victor Norman built the SNMP software, and revised it several times. Shawn Ostermann integrated the TCP/IP code into Xinu version 8, and ported it from the original Sun 3 platform to a DECstation 3100. Andy Muckelbauer and Steve Chapin built a UNIX compatibility library, and, along with Shawn Ostermann and Scott Mark, used the TCP code to run an X window server. Their testing exercised TCP extensively, and pointed out several performance problems. Scott M. Ballew participated in some of the software development, and provided an extensive review of all the text and code. Various other members of the Internetworking Research Group at Purdue contributed to earlier versions of the code. Christine Comer reviewed the manuscript and made many suggestions. Finally, we thank the Department of Computer Sciences and the Computing Center at Purdue University for their support.

Douglas E. Comer
David L. Stevens

Contents