

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

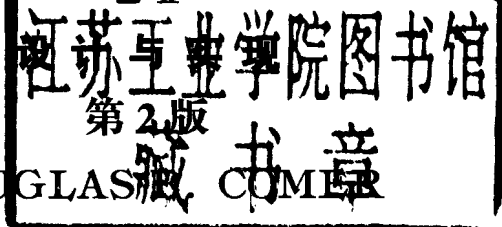
Vol II :

Design, Implementation, and Internals

Second Edition

TCP/IP 网络互连技术

卷 II



DOUGLAS R. 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

© 1994 by Prentice Hall, Inc.

Original English Language Edition published by Prentice Hall, Inc., a Simon & Schuster Company.

All Rights Reserved.

For sale in Mainland China only.

本书影印版由西蒙与舒斯特国际出版公司授权清华大学出版社在中国境内(不包括中国香港特别行政区、澳门地区和台湾地区)独家出版发行。

未经出版者书面许可,不得用任何方式复制或抄袭本书的任何部分。

本书封面贴有 Prentice Hall 激光防伪标签,无标签者不得销售。

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 Douglas 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

Foreword	xv
-----------------	-----------

Preface	xvii
----------------	-------------

Chapter 1 Introduction And Overview	1
--	----------

1.1	<i>TCP/IP Protocols</i>	1
1.2	<i>The Need To Understand Details</i>	1
1.3	<i>Complexity Of Interactions Among Protocols</i>	2
1.4	<i>The Approach In This Text</i>	2
1.5	<i>The Importance Of Studying Code</i>	3
1.6	<i>The Xinu Operating System</i>	3
1.7	<i>Organization Of The Remainder Of The Book</i>	4
1.8	<i>Summary</i>	4

Chapter 2 The Structure Of TCP/IP Software In An Operating System	7
--	----------

2.1	<i>Introduction</i>	7
2.2	<i>The Process Concept</i>	8
2.3	<i>Process Priority</i>	9
2.4	<i>Communicating Processes</i>	9
2.5	<i>Interprocess Communication</i>	12
2.6	<i>Device Drivers, Input, And Output</i>	14
2.7	<i>Network Input and Interrupts</i>	14
2.8	<i>Passing Packets To Higher Level Protocols</i>	16
2.9	<i>Passing Datagrams From IP To Transport Protocols</i>	16
2.10	<i>Delivery To Application Programs</i>	18
2.11	<i>Information Flow On Output</i>	19
2.12	<i>From TCP Through IP To Network Output</i>	20

- 2.13 *UDP Output* 21
- 2.14 *Summary* 21

Chapter 3 Network Interface Layer

27

- 3.1 *Introduction* 27
- 3.2 *The Network Interface Abstraction* 28
- 3.3 *Logical State Of An Interface* 31
- 3.4 *Local Host Interface* 31
- 3.5 *Buffer Management* 32
- 3.6 *Demultiplexing Incoming Packets* 35
- 3.7 *Summary* 36

Chapter 4 Address Discovery And Binding (ARP)

39

- 4.1 *Introduction* 39
- 4.2 *Conceptual Organization Of ARP Software* 40
- 4.3 *Example ARP Design* 40
- 4.4 *Data Structures For The ARP Cache* 41
- 4.5 *ARP Output Processing* 44
- 4.6 *ARP Input Processing* 49
- 4.7 *ARP Cache Management* 53
- 4.8 *ARP Initialization* 58
- 4.9 *ARP Configuration Parameters* 59
- 4.10 *Summary* 59

Chapter 5 IP: Global Software Organization

61

- 5.1 *Introduction* 61
- 5.2 *The Central Switch* 61
- 5.3 *IP Software Design* 62
- 5.4 *IP Software Organization And Datagram Flow* 63
- 5.5 *Byte-Ordering In The IP Header* 76
- 5.6 *Sending A Datagram To IP* 77
- 5.7 *Table Maintenance* 80
- 5.8 *Summary* 82

Chapter 6 IP: Routing Table And Routing Algorithm	85
6.1 Introduction	85
6.2 Route Maintenance And Lookup	85
6.3 Routing Table Organization	86
6.4 Routing Table Data Structures	87
6.5 Origin Of Routes And Persistence	89
6.6 Routing A Datagram	89
6.7 Periodic Route Table Maintenance	96
6.8 IP Options Processing	104
6.9 Summary	105
 Chapter 7 IP: Fragmentation And Reassembly	 107
7.1 Introduction	107
7.2 Fragmenting Datagrams	107
7.3 Implementation Of Fragmentation	108
7.4 Datagram Reassembly	113
7.5 Maintenance Of Fragment Lists	122
7.6 Initialization	124
7.7 Summary	124
 Chapter 8 IP: Error Processing (ICMP)	 127
8.1 Introduction	127
8.2 ICMP Message Formats	127
8.3 Implementation Of ICMP Messages	127
8.4 Handling Incoming ICMP Messages	130
8.5 Handling An ICMP Redirect Message	132
8.6 Setting A Subnet Mask	133
8.7 Choosing A Source Address For An ICMP Packet	135
8.8 Generating ICMP Error Messages	136
8.9 Avoiding Errors About Errors	139
8.10 Allocating A Buffer For ICMP	140
8.11 The Data Portion Of An ICMP Message	142
8.12 Generating An ICMP Redirect Message	144
8.13 Summary	145

Chapter 9 IP: Multicast Processing (IGMP) 147

9.1	<i>Introduction</i>	147
9.2	<i>Maintaining Multicast Group Membership Information</i>	147
9.3	<i>A Host Group Table</i>	148
9.4	<i>Searching For A Host Group</i>	150
9.5	<i>Adding A Host Group Entry To The Table</i>	151
9.6	<i>Configuring The Network Interface For A Multicast Address</i>	152
9.7	<i>Translation Between IP and Hardware Multicast Addresses</i>	154
9.8	<i>Removing A Multicast Address From The Host Group Table</i>	156
9.9	<i>Joining A Host Group</i>	157
9.10	<i>Maintaining Contact With A Multicast Router</i>	158
9.11	<i>Implementing IGMP Membership Reports</i>	160
9.12	<i>Computing A Random Delay</i>	161
9.13	<i>A Process To Send IGMP Reports</i>	163
9.14	<i>Handling Incoming IGMP Messages</i>	164
9.15	<i>Leaving A Host Group</i>	165
9.16	<i>Initialization Of IGMP Data Structures</i>	167
9.17	<i>Summary</i>	168

Chapter 10 UDP: User Datagrams 171

10.1	<i>Introduction</i>	171
10.2	<i>UDP Ports And Demultiplexing</i>	171
10.3	<i>UDP</i>	175
10.4	<i>UDP Output Processing</i>	185
10.5	<i>Summary</i>	188

Chapter 11 TCP: Data Structures And Input Processing 191

11.1	<i>Introduction</i>	191
11.2	<i>Overview Of TCP Software</i>	192
11.3	<i>Transmission Control Blocks</i>	192
11.4	<i>TCP Segment Format</i>	196
11.5	<i>Sequence Space Comparison</i>	198
11.6	<i>TCP Finite State Machine</i>	199
11.7	<i>Example State Transition</i>	200
11.8	<i>Declaration Of The Finite State Machine</i>	200
11.9	<i>TCB Allocation And Initialization</i>	202
11.10	<i>Implementation Of The Finite State Machine</i>	204
11.11	<i>Handling An Input Segment</i>	205

11.12	Summary	214
-------	---------	-----

Chapter 12 TCP: Finite State Machine Implementation

217

12.1	Introduction	217
12.2	CLOSED State Processing	217
12.3	Graceful Shutdown	218
12.4	Timed Delay After Closing	218
12.5	TIME-WAIT State Processing	219
12.6	CLOSING State Processing	221
12.7	FIN-WAIT-2 State Processing	222
12.8	FIN-WAIT-1 State Processing	223
12.9	CLOSE-WAIT State Processing	225
12.10	LAST-ACK State Processing	227
12.11	ESTABLISHED State Processing	228
12.12	Processing Urgent Data In A Segment	229
12.13	Processing Other Data In A Segment	231
12.14	Keeping Track Of Received Octets	233
12.15	Aborting A TCP Connection	236
12.16	Establishing A TCP Connection	237
12.17	Initializing A TCB	237
12.18	SYN-SENT State Processing	239
12.19	SYN-RECEIVED State Processing	240
12.20	LISTEN State Processing	243
12.21	Initializing Window Variables For A New TCB	244
12.22	Summary	246

Chapter 13 TCP: Output Processing

247

13.1	Introduction	247
13.2	Controlling TCP Output Complexity	247
13.3	The Four TCP Output States	248
13.4	TCP Output As A Process	248
13.5	TCP Output Messages	249
13.6	Encoding Output States And TCB Numbers	250
13.7	Implementation Of The TCP Output Process	250
13.8	Mutual Exclusion	251
13.9	Implementation Of The IDLE State	252
13.10	Implementation Of The PERSIST State	252
13.11	Implementation Of The TRANSMIT State	253
13.12	Implementation Of The RETRANSMIT State	255
13.13	Sending A Segment	255

13.14	<i>Computing The TCP Data Length</i>	259
13.15	<i>Computing Sequence Counts</i>	260
13.16	<i>Other TCP Procedures</i>	261
13.17	<i>Summary</i>	267

Chapter 14 TCP: Timer Management **269**

14.1	<i>Introduction</i>	269
14.2	<i>A General Data Structure For Timed Events</i>	269
14.3	<i>A Data Structure For TCP Events</i>	270
14.4	<i>Timers, Events, And Messages</i>	271
14.5	<i>The TCP Timer Process</i>	272
14.6	<i>Deleting A TCP Timer Event</i>	274
14.7	<i>Deleting All Events For A TCB</i>	275
14.8	<i>Determining The Time Remaining For An Event</i>	276
14.9	<i>Inserting A TCP Timer Event</i>	277
14.10	<i>Starting TCP Output Without Delay</i>	279
14.11	<i>Summary</i>	280

Chapter 15 TCP: Flow Control And Adaptive Retransmission **283**

15.1	<i>Introduction</i>	283
15.2	<i>The Difficulties With Adaptive Retransmission</i>	284
15.3	<i>Tuning Adaptive Retransmission</i>	284
15.4	<i>Retransmission Timer And Backoff</i>	284
15.5	<i>Window-Based Flow Control</i>	287
15.6	<i>Maximum Segment Size Computation</i>	291
15.7	<i>Congestion Avoidance And Control</i>	295
15.8	<i>Slow-Start And Congestion Avoidance</i>	296
15.9	<i>Round Trip Estimation And Timeout</i>	299
15.10	<i>Miscellaneous Notes And Techniques</i>	305
15.11	<i>Summary</i>	306

Chapter 16 TCP: Urgent Data Processing And The Push Function **309**

16.1	<i>Introduction</i>	309
16.2	<i>Out-Of-Band Signaling</i>	309
16.3	<i>Urgent Data</i>	310
16.4	<i>Interpreting The Standard</i>	310
16.5	<i>Configuration For Berkeley Urgent Pointer Interpretation</i>	313
16.6	<i>Informing An Application</i>	313

Contents

16.7	<i>Reading Data From TCP</i>	314
16.8	<i>Sending Urgent Data</i>	316
16.9	<i>TCP Push Function</i>	317
16.10	<i>Interpreting Push With Out-Of-Order Delivery</i>	318
16.11	<i>Implementation Of Push On Input</i>	319
16.12	<i>Summary</i>	320

Chapter 17 Socket-Level Interface

323

17.1	<i>Introduction</i>	323
17.2	<i>Interfacing Through A Device</i>	323
17.3	<i>TCP Connections As Devices</i>	325
17.4	<i>An Example TCP Client Program</i>	326
17.5	<i>An Example TCP Server Program</i>	327
17.6	<i>Implementation Of The TCP Master Device</i>	329
17.7	<i>Implementation Of A TCP Slave Device</i>	337
17.8	<i>Initialization Of A Slave Device</i>	351
17.9	<i>Summary</i>	352

Chapter 18 RIP: Active Route Propagation And Passive Acquisition

355

18.1	<i>Introduction</i>	355
18.2	<i>Active And Passive Mode Participants</i>	356
18.3	<i>Basic RIP Algorithm And Cost Metric</i>	356
18.4	<i>Instabilities And Solutions</i>	357
18.5	<i>Message Types</i>	361
18.6	<i>Protocol Characterization</i>	361
18.7	<i>Implementation Of RIP</i>	362
18.8	<i>The Principle RIP Process</i>	365
18.9	<i>Responding To An Incoming Request</i>	370
18.10	<i>Generating Update Messages</i>	372
18.11	<i>Initializing Copies Of An Update Message</i>	373
18.12	<i>Generating Periodic RIP Output</i>	378
18.13	<i>Limitations Of RIP</i>	379
18.14	<i>Summary</i>	379

Chapter 19 OSPF: Route Propagation With An SPF Algorithm

381

19.1	<i>Introduction</i>	381
19.2	<i>OSPF Configuration And Options</i>	382
19.3	<i>OSPF's Graph-Theoretic Model</i>	382

19.4	<i>OSPF Declarations</i>	386
19.5	<i>Adjacency And Link State Propagation</i>	391
19.6	<i>Discovering Neighboring Gateways With Hello</i>	392
19.7	<i>Sending Hello Packets</i>	394
19.8	<i>Designated Router Concept</i>	399
19.9	<i>Electing A Designated Router</i>	400
19.10	<i>Reforming Adjacencies After A Change</i>	404
19.11	<i>Handling Arriving Hello Packets</i>	406
19.12	<i>Adding A Gateway To The Neighbor List</i>	408
19.13	<i>Neighbor State Transitions</i>	410
19.14	<i>OSPF Timer Events And Retransmissions</i>	412
19.15	<i>Determining Whether Adjacency Is Permitted</i>	414
19.16	<i>Handling OSPF input</i>	415
19.17	<i>Declarations And Procedures For Link State Processing</i>	418
19.18	<i>Generating Database Description Packets</i>	421
19.19	<i>Creating A Template</i>	422
19.20	<i>Transmitting A Database Description Packet</i>	424
19.21	<i>Handling An Arriving Database Description Packet</i>	426
19.22	<i>Handling Link State Request Packets</i>	432
19.23	<i>Building A Link State Summary</i>	434
19.24	<i>OSPF Utility Procedures</i>	435
19.25	<i>Summary</i>	439

Chapter 20 SNMP: MIB Variables, Representations, And Bindings 441

20.1	<i>Introduction</i>	441
20.2	<i>Server Organization And Name Mapping</i>	442
20.3	<i>MIB Variables</i>	443
20.4	<i>MIB Variable Names</i>	444
20.5	<i>Lexicographic Ordering Among Names</i>	445
20.6	<i>Prefix Removal</i>	445
20.7	<i>Operations Applied To MIB Variables</i>	446
20.8	<i>Names For Tables</i>	446
20.9	<i>Conceptual Threading Of The Name Hierarchy</i>	447
20.10	<i>Data Structure For MIB Variables</i>	448
20.11	<i>A Data Structure For Fast Lookup</i>	450
20.12	<i>Implementation Of The Hash Table</i>	452
20.13	<i>Specification Of MIB Bindings</i>	452
20.14	<i>Internal Variables Used In Bindings</i>	457
20.15	<i>Hash Table Lookup</i>	458
20.16	<i>SNMP Structures And Constants</i>	461
20.17	<i>ASN.1 Representation Manipulation</i>	464
20.18	<i>Summary</i>	474