

Douglas E. Comer

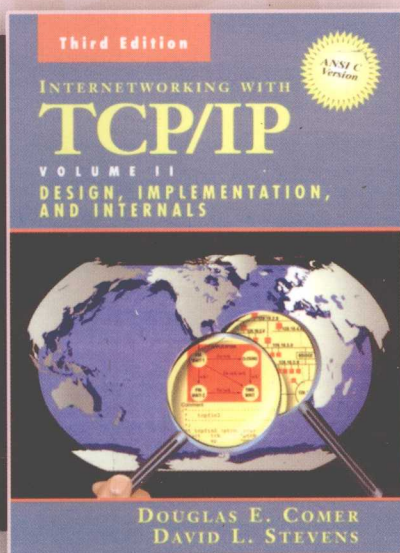
PEARSON

# 用TCP/IP 进行网际互连

## ——设计、实现与内核

### (ANSI C版) (第三版)

第二卷



Internetworking

Volume II With TCP/IP

ANSI C Version: Design  
Implementation  
and Internals, Third Edition

英文版

[美]

Douglas E. Comer  
David L. Stevens

著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国外计算机科学教材系列

# 用 TCP/IP 进行网际互连

第二卷——设计、实现与内核

( ANSI C 版 )( 第三版 )

( 英文版 )

Internetworking With TCP/IP

Volume II: ANSI C Version: Design, Implementation, and Internals

Third Edition

[ 美 ] Douglas E. Comer 著  
David L. Stevens

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书是关于计算机网络的经典教材。全书共三卷。第二卷在第一卷介绍了TCP/IP基本概念的基础上,进一步详细讨论了TCP/IP的实现细节。这一卷的突出特点是非常注重实际。作者利用程序代码实现了TCP/IP的每个技术细节,并且所有代码在书中均可找到,有助于读者深入了解并掌握TCP/IP的细节内容。书中附录分别给出了过程调用交叉参考表、程序代码中用到的C数据结构交叉参考表以及Xinu函数和常量。

Original edition, entitled INTERNETWORKING WITH TCP/IP Volume II: ANSI C VERSION: DESIGN, IMPLEMENTATION, AND INTERNALS, 3E, 9780139738432 by DOUGLAS E. COMER, DAVID L. STEVENS, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 1999 by Prentice Hall, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2009.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in the mainland of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书由Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

此版本仅限在中国大陆出版发行。

本书贴有Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2009-3938

### 图书在版编目(CIP)数据

用TCP/IP进行网际互连.第2卷,设计与内核:ANSI C版:第3版=Internetworking with TCP/IP. Vol. 2, ANSI C Version: Design, Implementation, and Internals, 3e: 英文/(美)科默(Comer, D. E.), (美)史蒂文(Stevens, D. L.)著.-北京:电子工业出版社,2009.8

(国外计算机科学教材系列)

ISBN 978-7-121-09188-9

I. 用… II. ①科… ②史… III. 计算机网络-通信协议-教材-英文 IV. TN915.04

中国版本图书馆CIP数据核字(2009)第109628号

策划编辑:谭海平

责任编辑:许菊芳

印 刷:北京市顺义兴华印刷厂

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×980 1/16 印张:42.75 字数:958千字

印 次:2009年8月第1次印刷

定 价:75.00元

凡所购买电子工业出版社的图书有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系及邮购电话:(010)88254888。

质量投诉请发邮件至zlt@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010)88258888。

## 出版说明

21 世纪初的 5 至 10 年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入 WTO 后的今天,培养一支适应国际化竞争的一流 IT 人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如 Pearson Education 培生教育出版集团、麦格劳—希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

## 教材出版委员会

主 任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委 员	王 珊	中国人民大学信息学院教授 中国计算机学会副理事长，数据库专业委员会主任
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授、博士生导师 清华大学深圳研究生院信息学部主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹 鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础课程教学指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

# Foreword

I am grateful to Doug Comer for the opportunity to share my thoughts as the third edition of Doug's book goes to press. The Internet continues its now-legendary growth, nearly doubling in size annually over the last decade. From an estimated 100,000 computers on the Internet in July 1989, the number of host reached an estimated 30,000,000 in January 1998. At the time the second edition when to press, it was estimated that 26,000 networks were interconnected to form the global Internet. While this data is no longer accurately available, it is estimated that between 200,000 and 350,000 network form the global Internet of 1998. And that does not count the number of private intranets that use internet technology but are not necessarily connected to the public network.

Apart from sheer scale, the Internet has changed dramatically in terms of the applications it is expected to support. Internet telephony is now expected to become an integral and growing part of the system, taking its place beside the burgeoning World Wide Web. The latter has sparked a revolution in the business, residential, academic and government sectors. Estimates of the number of "pages" in the WWW range up to 320 million, with many more appearing daily. Some schools tell me that more than a quarter of their admissions applications are arriving via email or web forms. Dell computer reported that it was selling \$6M/day worth of personal computers through its web site. Amazon.com reported recent quarterly revenues of \$66M making it the fastest growing company in history and the first to reach a sales run rate in excess of \$250M in less than a year.

At least two thousand radio stations are putting their audio up on the Internet and many web sites supply audio and even low-quality video on demand. The latter is very likely to improve in quality as access speeds improve into the megabit range and backbone capacity increases to match.

Internet enabled appliances are beginning to appear with the arrival of WebTV in 1996 and Internet-enabled cellular telephones in 1997 (Nokia). More appliances are bound to be similarly enabled - water heaters that can be controlled not only by home computers but also by power companies seeking to moderate peak load demands in a pinpointed fashion. Although intelligent agents have not become a mainstream item as yet, the development of the XML variant of HTML (and SGML) suggests that we are

on the edge of another explosive growth curve for transaction processing in the Internet. Standard "document" representations in SML with commonly agreed upon interpretations form a kind of transportable object which can become the basis for all kinds of business and financial transactions, database transactions and other exchanges requiring standard representations and interpretations.

Returning to telephony for a moment, both traditional suppliers and entrepreneurial start-ups are bringing Internet-enabled telephony products to market. "Soft PBX" systems that perform the functions of conventional private branch exchanges, but do so using LANs, Internet, and microprocessor-based telephones are changing the economics and the functionality of telecommunication. IP-enabled fax machines are on the horizon. And, of course, gateways to connect the older analog world to the new Internet world are a part of the equation.

Capacity requirements for Internet service have grown as quickly or even more quickly than the absolute size of the network. It is becoming common for backbone networks to operate at speeds up to 622 megabits/second. New generation routers, using hardware IP switching, are expected to be able to handle Internet packets at line speeds exceeding ten gigabits per second (OC192). Already examples of gigabit ethernet provide convincing evidence that these forwarding rates are feasible. The challenge will be to keep up with single mode fibers carrying hundreds of light colors and terabits per second of traffic.

Security concerns, always in the background, are making it into the foreground as all sectors of the economy begin to rely increasingly on the Internet for day to day operation. Firewalls, end-to-end encryption, key management, certificate systems and authentication systems are all becoming critical elements of successful Internet operation.

Looking into the more distant future, work is already starting on protocols and architectural structures for an interplanetary Internet - likely to take shape as a networks of internets. The Domain Name System, if it survives, will need to incorporate a hierarchical structure that takes into account various planetary systems. The "inter-Internet" protocols will have to deal with very high delay, high noise environments - and the conventional TCP concepts replaced with more one-way types of procedures. The Internauts of 2020 will almost surely look back on the last decade of the Twentieth Century as a time of charming naivete. The speculations in this foreword will seem to them timid and quaint as they deal with the reality of another two decades of the Internet revolution. "Why, I remember, back in ninety-eight, when people thought wearable computers were far-fetched and maybe a little crazy."

Vinton Cerf  
Camelot  
Northern Virginia  
April, 1998

# Preface

*Internetworking With TCP/IP Volume 2* was created to supply details about TCP/IP protocols that are not covered in Volume 1. Volume 2 places the TCP/IP stack under a magnifying glass, and examines the details of individual protocols. It discusses their implementation, and focuses on the internals of protocol software. The third edition contains updates and improvements throughout. The code has been revised to use the ANSI-standard C subset of C++, including function prototypes and argument declarations. In addition, we have corrected several minor problems, and made improvements to the efficiency. We updated SNMP to the version 2 standard, including replacing the address translation table and adding the UDP listener table. Finally, we added Appendix 2 that contains a cross-reference of major structure declarations and variables used in the code, and expanded Appendix 1 that contains a cross-reference of procedure declarations and calls.

The example code was compiled for an Intel architecture platform using the widely-available Gnu C++ compiler. It was then downloaded and tested on a Pentium™ system. All the code in the text is available online at:

`ftp://ftp.cs.purdue.edu/pub/comer/TCPIP-vol2.dist.tar.Z`

Although this text is copyrighted, the code is available, and has been used in many commercial products. The only restriction on use is that the source may not be published in a book.

We encourage readers to obtain a copy of the code and to use computer tools to examine, modify, compile, and test it. Indeed, although the indices in Appendix 1 and 2 provide a useful way to locate items, we find the UNIX *grep* program invaluable when examining large volumes of code.

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 the 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 or OSPF, specifies how a router obtains routes for an IP



routing table and advertises its routes to other routers. Routing protocols also specify when routes must be removed. But the interaction between routing and other protocols may not be apparent from the RFCs. The question arises, “how does route timeout affect routes in the table that were installed by ICMP?” One must also consider the question, “should routing updates override routes that the manager has installed manually?”

To illustrate the interaction among protocols and to ensure 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, SNMPv2, 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 completely 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.

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. Shawn Ostermann made significant contributions to this version of the text. Shawn earlier integrated the TCP/IP code into Xinu version 8 and ported it from the original Sun 3 platform to a DECstation 3100. In this edition, he extensively revised SNMP, originally written by Vic Norman, to meet the version 2 standard and helped debug the code. John Lin proofread the text for technical accuracy, correcting several problems.

Various members of the Internetworking Research Group at Purdue contributed to earlier versions of the code. 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 also participated in some of the software development.

My wife, Christine, edited the revision and improved both wording and accuracy. Finally, we thank the Department of Computer Sciences at Purdue for their support.

Douglas E. Comer

David L. Stevens

May, 1998

# Contents

<b>Foreword</b>	15
<b>Preface</b>	17
<b>Chapter 1 Introduction And Overview</b>	<b>1</b>
1.1 TCP/IP Protocols	1
1.2 The Need To Understand Details	1
1.3 Complexity Of Interactions Among Protocols	2
1.4 The Approach In This Text	2
1.5 The Importance Of Studying Code	3
1.6 The Xinu Operating System	3
1.7 Organization Of The Remainder Of The Book	4
1.8 Summary	4
<b>Chapter 2 The Structure Of TCP/IP Software In An Operating System</b>	<b>7</b>
2.1 Introduction	7
2.2 The Process Concept	8
2.3 Process Priority	9
2.4 Process Synchronization	9
2.5 Interprocess Communication	12
2.6 Device Drivers, Input, And Output	14
2.7 Network Input and Interrupts	14
2.8 Passing Packets To Higher Level Protocols	16
2.9 Passing Datagrams From IP To Transport Protocols	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	<i>UDP Output</i>	21
2.14	<i>Summary</i>	21

## **Chapter 3 Network Interface Layer**

**27**

3.1	<i>Introduction</i>	27
3.2	<i>The Network Interface Abstraction</i>	28
3.3	<i>Ethernet Definitions</i>	30
3.4	<i>Logical State Of An Interface</i>	34
3.5	<i>Local Host Interface</i>	35
3.6	<i>Buffer Management</i>	36
3.7	<i>Demultiplexing Incoming Packets</i>	38
3.8	<i>Summary</i>	40

## **Chapter 4 Address Discovery And Binding (ARP)**

**41**

4.1	<i>Introduction</i>	41
4.2	<i>Conceptual Organization Of ARP Software</i>	42
4.3	<i>Example ARP Design</i>	42
4.4	<i>Data Structures For The ARP Cache</i>	43
4.5	<i>ARP Output Processing</i>	46
4.6	<i>ARP Input Processing</i>	51
4.7	<i>ARP Cache Management</i>	56
4.8	<i>ARP Initialization</i>	60
4.9	<i>ARP Configuration Parameters</i>	61
4.10	<i>Summary</i>	61

## **Chapter 5 IP: Global Software Organization**

**63**

5.1	<i>Introduction</i>	63
5.2	<i>The Central Switch</i>	63
5.3	<i>IP Software Design</i>	64
5.4	<i>IP Software Organization And Datagram Flow</i>	65
5.5	<i>Byte-Ordering In The IP Header</i>	78
5.6	<i>Sending A Datagram To IP</i>	80
5.7	<i>Table Maintenance</i>	83
5.8	<i>Summary</i>	84

## **Chapter 6 IP: Routing Table And Routing Algorithm**

**87**

- 6.1 *Introduction* 87
- 6.2 *Route Maintenance And Lookup* 87
- 6.3 *Routing Table Organization* 88
- 6.4 *Routing Table Data Structures* 89
- 6.5 *Origin Of Routes And Persistence* 91
- 6.6 *Routing A Datagram* 91
- 6.7 *Periodic Routing Table Maintenance* 98
- 6.8 *IP Options Processing* 106
- 6.9 *Summary* 107

## **Chapter 7 IP: Fragmentation And Reassembly**

**109**

- 7.1 *Introduction* 109
- 7.2 *Fragmenting Datagrams* 109
- 7.3 *Implementation Of Fragmentation* 110
- 7.4 *Datagram Reassembly* 115
- 7.5 *Maintenance Of Fragment Lists* 124
- 7.6 *Initialization* 126
- 7.7 *Summary* 126

## **Chapter 8 IP: Error Processing (ICMP)**

**129**

- 8.1 *Introduction* 129
- 8.2 *ICMP Message Formats* 129
- 8.3 *Implementation Of ICMP Messages* 129
- 8.4 *Handling Incoming ICMP Messages* 132
- 8.5 *Handling An ICMP Redirect Message* 134
- 8.6 *Setting A Subnet Mask* 135
- 8.7 *Choosing A Source Address For An ICMP Packet* 137
- 8.8 *Generating ICMP Error Messages* 138
- 8.9 *Avoiding Errors About Errors* 141
- 8.10 *Allocating A Buffer For ICMP* 142
- 8.11 *The Data Portion Of An ICMP Message* 144
- 8.12 *Generating An ICMP Redirect Message* 146
- 8.13 *Summary* 147

## **Chapter 9 IP: Multicast Processing (IGMP)**

**149**

- 9.1 *Introduction* 149
- 9.2 *Maintaining Multicast Group Membership Information* 149
- 9.3 *A Host Group Table* 150
- 9.4 *Searching For A Host Group* 152
- 9.5 *Adding A Host Group Entry To The Table* 153
- 9.6 *Configuring The Network Interface For A Multicast Address* 155
- 9.7 *Translation Between IP and Hardware Multicast Addresses* 157
- 9.8 *Removing A Multicast Address From The Host Group Table* 159
- 9.9 *Joining A Host Group* 160
- 9.10 *Maintaining Contact With A Multicast Router* 161
- 9.11 *Implementing IGMP Membership Reports* 163
- 9.12 *Computing A Random Delay* 165
- 9.13 *A Process To Send IGMP Reports* 166
- 9.14 *Handling Incoming IGMP Messages* 167
- 9.15 *Leaving A Host Group* 169
- 9.16 *Initialization Of IGMP Data Structures* 170
- 9.17 *Summary* 171

## **Chapter 10 UDP: User Datagrams**

**173**

- 10.1 *Introduction* 173
- 10.2 *UDP Ports And Demultiplexing* 173
- 10.3 *UDP Input Processing* 177
- 10.4 *UDP Output Processing* 187
- 10.5 *Summary* 190

## **Chapter 11 TCP: Data Structures And Input Processing**

**193**

- 11.1 *Introduction* 193
- 11.2 *Overview Of TCP Software* 194
- 11.3 *Transmission Control Blocks* 194
- 11.4 *TCP Segment Format* 199
- 11.5 *Sequence Space Comparison* 200
- 11.6 *TCP Finite State Machine* 202
- 11.7 *Example State Transition* 204
- 11.8 *Declaration Of The Finite State Machine* 204
- 11.9 *TCB Allocation And Initialization* 206
- 11.10 *Implementation Of The Finite State Machine* 208
- 11.11 *Handling An Input Segment* 209

## Chapter 12 TCP: Finite State Machine Implementation

221

12.1	Introduction	221
12.2	CLOSED State Processing	221
12.3	Graceful Shutdown	222
12.4	Timed Delay After Closing	222
12.5	TIME-WAIT State Processing	223
12.6	CLOSING State Processing	225
12.7	FIN-WAIT-2 State Processing	226
12.8	FIN-WAIT-1 State Processing	227
12.9	CLOSE-WAIT State Processing	229
12.10	LAST-ACK State Processing	231
12.11	ESTABLISHED State Processing	232
12.12	Processing Urgent Data In A Segment	233
12.13	Processing Other Data In A Segment	235
12.14	Keeping Track Of Received Octets	237
12.15	Aborting A TCP Connection	240
12.16	Establishing A TCP Connection	241
12.17	Initializing A TCB	241
12.18	SYN-SENT State Processing	243
12.19	SYN-RECEIVED State Processing	244
12.20	LISTEN State Processing	247
12.21	Initializing Window Variables For A New TCB	248
12.22	Summary	250

## Chapter 13 TCP: Output Processing

251

13.1	Introduction	251
13.2	Controlling TCP Output Complexity	251
13.3	The Four TCP Output States	252
13.4	TCP Output As A Process	252
13.5	TCP Output Messages	253
13.6	Encoding Output States And TCB Numbers	254
13.7	Implementation Of The TCP Output Process	254
13.8	Mutual Exclusion	255
13.9	Implementation Of The IDLE State	256
13.10	Implementation Of The PERSIST State	256
13.11	Implementation Of The TRANSMIT State	257
13.12	Implementation Of The RETRANSMIT State	259
13.13	Sending A Segment	259

13.14	<i>Computing The TCP Data Length</i>	263
13.15	<i>Computing Sequence Counts</i>	264
13.16	<i>Other TCP Procedures</i>	265
13.17	<i>Summary</i>	271

## **Chapter 14 TCP: Timer Management 273**

14.1	<i>Introduction</i>	273
14.2	<i>A General Data Structure For Timed Events</i>	273
14.3	<i>A Data Structure For TCP Events</i>	274
14.4	<i>Timers, Events, And Messages</i>	275
14.5	<i>The TCP Timer Process</i>	276
14.6	<i>Deleting A TCP Timer Event</i>	278
14.7	<i>Deleting All Events For A TCB</i>	280
14.8	<i>Determining The Time Remaining For An Event</i>	281
14.9	<i>Inserting A TCP Timer Event</i>	282
14.10	<i>Starting TCP Output Without Delay</i>	283
14.11	<i>Summary</i>	285

## **Chapter 15 TCP: Flow Control And Adaptive Retransmission 287**

15.1	<i>Introduction</i>	287
15.2	<i>The Difficulties With Adaptive Retransmission</i>	288
15.3	<i>Tuning Adaptive Retransmission</i>	288
15.4	<i>Retransmission Timer And Backoff</i>	288
15.5	<i>Window-Based Flow Control</i>	291
15.6	<i>Maximum Segment Size Computation</i>	295
15.7	<i>Congestion Avoidance And Control</i>	299
15.8	<i>Slow-Start And Congestion Avoidance</i>	300
15.9	<i>Round Trip Estimation And Timeout</i>	303
15.10	<i>A Miscellaneous Note</i>	309
15.11	<i>Summary</i>	310

## **Chapter 16 TCP: Urgent Data Processing And The Push Function 313**

16.1	<i>Introduction</i>	313
16.2	<i>Out-Of-Band Signaling</i>	313
16.3	<i>Urgent Data</i>	314
16.4	<i>Interpreting The Standard</i>	314
16.5	<i>Configuration For Berkeley Urgent Pointer Interpretation</i>	317
16.6	<i>Informing An Application</i>	317



16.7	<i>Reading Data From TCP</i>	318
16.8	<i>Sending Urgent Data</i>	320
16.9	<i>TCP Push Function</i>	321
16.10	<i>Interpreting Push With Out-Of-Order Delivery</i>	322
16.11	<i>Implementation Of Push On Input</i>	323
16.12	<i>Summary</i>	324

## **Chapter 17 Socket-Level Interface**

**327**

17.1	<i>Introduction</i>	327
17.2	<i>Interfacing Through A Device</i>	327
17.3	<i>TCP Connections As Devices</i>	329
17.4	<i>An Example TCP Client Program</i>	330
17.5	<i>An Example TCP Server Program</i>	331
17.6	<i>Implementation Of The TCP Master Device</i>	333
17.7	<i>Implementation Of A TCP Slave Device</i>	341
17.8	<i>Initialization Of A Slave Device</i>	355
17.9	<i>Summary</i>	356

## **Chapter 18 RIP: Active Route Propagation And Passive Acquisition**

**359**

18.1	<i>Introduction</i>	359
18.2	<i>Active And Passive Mode Participants</i>	360
18.3	<i>Basic RIP Algorithm And Cost Metric</i>	360
18.4	<i>Instabilities And Solutions</i>	361
18.5	<i>Message Types</i>	364
18.6	<i>Protocol Characterization</i>	365
18.7	<i>Implementation Of RIP</i>	366
18.8	<i>The Principle RIP Process</i>	369
18.9	<i>Responding To An Incoming Request</i>	375
18.10	<i>Generating Update Messages</i>	377
18.11	<i>Initializing Copies Of An Update Message</i>	379
18.12	<i>Generating Periodic RIP Output</i>	384
18.13	<i>Limitations Of RIP</i>	385
18.14	<i>Summary</i>	385

## **Chapter 19 OSPF: Route Propagation With An SPF Algorithm**

**387**

19.1	<i>Introduction</i>	387
19.2	<i>OSPF Configuration And Options</i>	388
19.3	<i>OSPF's Graph-Theoretic Model</i>	388