

Distributed Systems

An Algorithmic Approach

CHAPMAN & HALL/CRC COMPUTER and INFORMATION SCIENCE SERIES

Series Editor: Sartaj Sahni

PUBLISHED TITLES

HANDBOOK OF SCHEDULING: ALGORITHMS, MODELS, AND PERFORMANCE ANALYSIS

Joseph Y.-T. Leung

THE PRACTICAL HANDBOOK OF INTERNET COMPUTING

Munindar P. Singh

HANDBOOK OF DATA STRUCTURES AND APPLICATIONS

Dinesh P. Mehta and Sartaj Sahni

DISTRIBUTED SENSOR NETWORKS

S. Sitharama Iyengar and Richard R. Brooks

SPECULATIVE EXECUTION IN HIGH PERFORMANCE COMPUTER ARCHITECTURES

David Kaeli and Pen-Chung Yew

SCALABLE AND SECURE INTERNET SERVICES AND ARCHITECTURE

Cheng-Zhong Xu

HANDBOOK OF BIOINSPIRED ALGORITHMS AND APPLICATIONS

Stephan Olariu and Albert Y. Zomaya

HANDBOOK OF ALGORITHMS FOR WIRELESS NETWORKING AND MOBILE COMPUTING

Azzedine Boukerche

HANDBOOK OF COMPUTATIONAL MOLECULAR BIOLOGY

Srinivas Aluru

FUNDAMENTALS OF NATURAL COMPUTING: BASIC CONCEPTS, ALGORITHMS, AND APPLICATIONS

Leandro Nunes de Castro

ADVERSARIAL REASONING: COMPUTATIONAL APPROACHES TO READING THE OPPONENT'S MIND

Alexander Kott and William M. McEneaney

DISTRIBUTED SYSTEMS: AN ALGORITHMIC APPROACH

Sukumar Ghosh

Distributed Systems

An Algorithmic Approach

Sukumar Ghosh

University of Iowa

Iowa City, U.S.A.



Chapman & Hall/CRC

Taylor & Francis Group

Boca Raton London New York

Chapman & Hall/CRC is an imprint of the
Taylor & Francis Group, an informa business

Cover design by Soumya Ghosh.

Chapman & Hall/CRC
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2007 by Taylor & Francis Group, LLC
Chapman & Hall/CRC is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 1-58488-564-5 (Hardcover)
International Standard Book Number-13: 978-1-58488-564-1 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC) 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Ghosh, Sukumar.

Distributed systems : an algorithmic approach / Sukumar Ghosh.
p. cm. -- (Computer and information science series ; no. 13)

Includes bibliographical references and index.

ISBN-13: 978-1-58488-564-1

ISBN-10: 1-58488-564-5

1. Electronic data processing--Distributed processing. 2. Computer algorithms. I. Title. II. Series.

QA76.9.D5G487 2006

004'.36--dc22

2006017600

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Preface

Distributed systems have witnessed phenomenal growth in the past few years. The declining cost of hardware, the advancements in communication technology, the explosive growth of the Internet, and our ever-increasing dependence on networks for a wide range of applications ranging from social communication to financial transactions have contributed to this growth. The breakthroughs in nanotechnology, and wireless communication have opened up new frontiers of applications like sensor networks and wearable computers. We have witnessed the rise and fall of Napster, but we have also seen the rise of peer-to-peer networks from the ashes of Napster. Most applications in distributed computing center around a set of common subproblems. A proper understanding of these subproblems requires a background of the underlying theory and algorithmic issues. This book is a presentation of the foundational topics of distributed systems and their relationships to real-life applications.

The distributed systems community is polarized into two camps. Some practitioners shun the theory as impractical or irrelevant. Some theoreticians pay little attention to the relevance of the theory, and are unable to relate them to real-life results. This book aims at bringing the two communities closer to each other, by striking a better balance between theory and practice.

The book has 21 chapters that can be broadly divided into 5 parts: **Part A (Chapters 1–2)** deals with background materials that include various interprocess communication techniques, and middleware services. **Part B (Chapter 3–6)** presents foundational topics, which address system models, correctness criteria, and proof techniques. **Part C (Chapters 7–11)** presents several important paradigms in distributed systems — topics include logical clocks, distributed snapshots, deadlock detection, termination detection, election, and a few graph algorithms relevant to distributed systems design. **Part D (Chapters 12–17)** addresses failures and fault-tolerance techniques in various applications — it covers consensus, transactions, group communication, replicated data management, and self-stabilization. Topics like group-communication or consensus are certainly not techniques of fault-tolerance, but their implementations become challenging when process crashes are factored in. Finally, **Part E (Chapters 18–21)** addresses issues in the real world: these include distributed discrete-event simulation and security, sensor networks, and peer-to-peer networks. Each chapter has a list of exercises that will challenge the readers. A small number of these are programming exercises. Some exercises will encourage the readers to learn about outside materials.

The book is intended for use in a one-semester course at the senior undergraduate or the first-year graduate level. About 75% of the materials can be covered in one semester. Accordingly, the chapters can be picked and packaged in several different ways. Based on inputs from people who used the material, a theory oriented offering is possible using Chapters 1, 3–17, and 19. For a more practical flavor, use Chapters 1–2, parts of Chapters 3–5, 6, parts of Chapters 7, 9, 11–16, and Chapters 18–21, supplemented by a semester-long project chosen from the areas on replicated data management, wireless networks, group communication, discrete-event simulation, mobile agents, sensor networks, or P2P networks. Readers with background in networking can skip the first two chapters.

In several chapters, readers will find topics that do not have an immediate relevance in the practical world. For example, one may wonder, who cares about designing mutual exclusion algorithms now, when application designers have well-developed tools for mutual exclusion? But remember that these tools did not come from nowhere! Some of these are intellectually challenging, but only of historic interest and good sources of enrichment, while others tell readers about what goes on “under

the hood.” If the coverage of topics is constrained by immediate practical relevance, then creativity takes a back seat. Those who do not agree to this view can conveniently skip such topics.

Here is a disclaimer: this book is **not** about programming distributed systems. [Chapter 2](#) is only a high-level description that we expect everyone to know, but is **not** an introduction to programming. If programming is the goal, then I encourage readers to look for other materials. There are several good books available.

Several years ago, a well-respected computer scientist advised me about the importance of maintaining a low length-to-content ratio in technical writing. I took his advice to heart while writing this book.

It is a pleasure to acknowledge the help and support of my friends and colleagues from all over the world in completing this project. Steve Bruell helped with improving the initial write-up. Ted Herman has been a constant source of wisdom. Discussions with Sriram Pemmaraju on several topics have been stimulating. Various parts of this book have been used in several offerings of the courses of 22C:166, 22C:194, and 22C:294 at the Computer Science department of the University of Iowa — special thanks to the students of these courses for their constructive criticisms and suggestions. Amlan Bhattacharya, Kajari Ghosh Dastidar, and Shridhar Dighe helped with several examples and exercises. Encouraging feedbacks from readers around the world on the earlier drafts of the material provided the motivation for the project. Thanks to Anand Padmanabhan, Shrisha Rao, Alan Kaminsky, Clifford Neuman, Carl Hauser, Michael Paultisch, Chandan Mazumdar, Arobinda Gupta, and several anonymous reviewers for numerous feedbacks that helped improve the contents of this book. Paul Crockett’s early encouragement and Bob Stern’s patience have played a key role in completing the manuscript.

Despite best efforts on my part, there will be errors. Conscientious readers are requested to report these to ghosh@cs.uiowa.edu. I thank them in advance.

Sukumar Ghosh
Iowa City

Table of Contents

Part A	
Background Materials	1
Chapter 1	
Introduction	3
1.1 What Is a Distributed System?	3
1.2 Why Distributed Systems?	3
1.3 Examples of Distributed Systems	4
1.4 Important Issues in Distributed Systems	6
1.5 Common Subproblems	8
1.6 Implementing a Distributed System	9
1.7 Parallel vs. Distributed Systems	10
1.8 Bibliographic Notes	10
Chapter 2	
Interprocess Communication: An Overview	13
2.1 Introduction	13
2.1.1 Processes and Threads	13
2.1.2 Client–Server Model	13
2.1.3 Middleware	14
2.2 Network Protocols	14
2.2.1 The Ethernet	15
2.2.2 Wireless Networks	15
2.2.3 The OSI Model	17
2.2.4 Internet Protocol	19
2.2.5 Transport Layer Protocols	20
2.2.6 Interprocess Communication Using Sockets	21
2.3 Naming	22
2.3.1 Domain Name Service	23
2.3.2 Naming Service for Mobile Clients	24
2.4 Remote Procedure Call	25
2.4.1 Implementing RPC	25
2.4.2 SUN RPC	27
2.5 Remote Method Invocation	27
2.6 Web Services	28
2.7 Messages	29
2.7.1 Transient and Persistent Messages	29
2.7.2 Streams	29
2.8 Event Notification	29
2.9 CORBA	30
2.10 Mobile Agents	31

2.11	Basic Group Communication Services	32
2.12	Concluding Remarks.....	32
2.13	Bibliographic Notes.....	32
2.14	Exercises	33

Part B
Foundational Topics **35**

Chapter 3		
Models of Communication.....		37
3.1 The Need for a Model		37
3.2 A Message-Passing Model for Interprocess Communication.....		37
3.2.1 Process Actions		37
3.2.2 Channels		38
3.2.3 Synchronous vs. Asynchronous Systems.....		39
3.3 Shared Variables		41
3.3.1 Linda		42
3.4 Modeling Mobile Agents		43
3.5 Relationship among Models		44
3.5.1 Strong and Weak Models		44
3.5.2 Implementing a FIFO Channel Using a Non-FIFO Channel		45
3.5.3 Implementing Message Passing on Shared Memory		46
3.5.4 Implementing Shared Memory Using Message Passing.....		46
3.5.5 An Impossibility Result with Channels.....		47
3.6 Classification Based on Special Properties		48
3.6.1 Reactive vs. Transformational Systems		48
3.6.2 Named vs. Anonymous Systems.....		48
3.7 Complexity Measures.....		48
3.8 Concluding Remarks.....		51
3.9 Bibliographic Notes.....		51

Chapter 4		
Representing Distributed Algorithms: Syntax and Semantics.....		55
4.1 Introduction		55
4.2 Guarded Actions		55
4.3 Nondeterminism		57
4.4 Atomic Operations		58
4.5 Fairness		60
4.5.1 Unconditionally Fair Scheduler.....		61
4.5.2 Weakly Fair Scheduler		61
4.5.3 Strongly Fair Scheduler		62
4.6 Central vs. Distributed Schedulers		62
4.7 Concluding Remarks.....		64
4.8 Bibliographic Notes.....		65

Chapter 5		
Program Correctness		69
5.1 Introduction		69
5.2 Correctness Criteria.....		70
5.2.1 Safety Properties.....		70
5.2.2 Liveness Properties		71

5.3	Correctness Proofs	74
5.4	Predicate Logic	74
5.4.1	A Review of Propositional Logic	74
5.4.2	Brief Overview of Predicate Logic	75
5.5	Assertional Reasoning: Proving Safety Properties	76
5.6	Proving Liveness Properties Using Well-Founded Sets	77
5.7	Programming Logic	79
5.8	Predicate Transformers	82
5.9	Concluding Remarks	84
5.10	Bibliographic Notes	84

Chapter 6

Time in a Distributed System	89
6.1 Introduction	89
6.1.1 The Physical Time	89
6.1.2 Sequential and Concurrent Events	90
6.2 Logical Clocks	90
6.3 Vector Clocks	93
6.4 Physical Clock Synchronization	94
6.4.1 Preliminary Definitions	94
6.4.2 Clock Reading Error.....	95
6.4.3 Algorithms for Internal Synchronization	96
6.4.4 Algorithms for External Synchronization	97
6.5 Concluding Remarks.....	99
6.6 Bibliographic Notes.....	100

Part C

Important Paradigms	103
----------------------------------	------------

Chapter 7

Mutual Exclusion	105
7.1 Introduction	105
7.2 Solutions Using Message Passing	105
7.2.1 Lamport's Solution	106
7.2.2 Ricart–Agrawala's Solution	108
7.2.3 Maekawa's Solution	109
7.3 Token Passing Algorithms	113
7.3.1 Suzuki–Kasami Algorithm	113
7.3.2 Raymond's Algorithm	114
7.4 Solutions on the Shared-Memory Model	114
7.4.1 Peterson's Algorithm	115
7.5 Mutual Exclusion Using Special Instructions	117
7.5.1 Solution Using Test-and-Set	117
7.5.2 Solution Using Load-Linked and Store-Conditional	118
7.6 The Group Mutual Exclusion Problem	118
7.6.1 A Centralized Solution	119
7.6.2 Decentralized Solution on the Shared-Memory Model	119
7.7 Concluding Remarks	120
7.8 Bibliographic Notes	121

Chapter 8

Distributed Snapshot	127
8.1 Introduction	127
8.2 Properties of Consistent Snapshots	128
8.3 The Chandy–Lamport Algorithm	129
8.3.1 Two Examples	131
8.4 The Lai–Yang Algorithm	133
8.5 Concluding Remarks.....	134
8.6 Bibliographic Notes.....	134

Chapter 9

Global State Collection	137
9.1 Introduction	137
9.2 An Elementary Algorithm for Broadcasting.....	137
9.3 Termination Detection Algorithm.....	139
9.3.1 The Dijkstra–Scholten Algorithm	140
9.3.2 Termination Detection on a Unidirectional Ring	143
9.4 Distributed Deadlock Detection	144
9.4.1 Detection of Resource Deadlock.....	145
9.4.2 Detection of Communication Deadlock	147
9.5 Concluding Remarks.....	148
9.6 Bibliographic Notes.....	149

Chapter 10

Graph Algorithms	151
10.1 Introduction	151
10.2 Routing Algorithms.....	151
10.2.1 Computation of Shortest Path.....	152
10.2.2 Distance Vector Routing	154
10.2.3 Link-State Routing	155
10.2.4 Interval Routing	156
10.3 Graph Traversal	159
10.3.1 Spanning Tree Construction	159
10.3.2 Tarjan’s Graph Traversal Algorithm.....	161
10.3.3 Minimum Spanning Tree.....	162
10.4 Graph Coloring.....	166
10.4.1 A Simple Coloring Algorithm	167
10.4.2 Planar Graph Coloring	168
10.5 Concluding Remarks.....	169
10.6 Bibliographic Notes.....	170

Chapter 11

Coordination Algorithms.....	173
11.1 Introduction	173
11.2 Leader Election	173
11.2.1 The Bully Algorithm	174
11.2.2 Maxima Finding on a Ring.....	175
11.2.2.1 Chang–Roberts Algorithm	175
11.2.2.2 Franklin’s Algorithm.....	176
11.2.2.3 Peterson’s Algorithm.....	177

11.2.3	Election in Arbitrary Networks	179
11.2.4	Election in Anonymous Networks	179
11.3	Synchronizers	180
11.3.1	The ABD Synchronizer	180
11.3.2	Awerbuch's Synchronizer	181
11.3.2.1	The α -Synchronizer	181
11.3.2.2	The β -Synchronizer	183
11.3.2.3	The γ -Synchronizer	183
11.3.2.4	Performance of Synchronizers	185
11.4	Concluding Remarks	185
11.5	Bibliographic Notes	186
11.6	Exercises	186

Part D

Faults and Fault-Tolerant Systems 189

Chapter 12

Fault-Tolerant Systems	191
12.1 Introduction	191
12.2 Classification of Faults	191
12.3 Specification of Faults	194
12.4 Fault-Tolerant Systems	196
12.4.1 Masking Tolerance	196
12.4.2 NonMasking Tolerance	196
12.4.3 Fail-Safe Tolerance	196
12.4.4 Graceful Degradation	197
12.4.5 Detection of Failures	197
12.5 Tolerating Crash Failures	198
12.5.1 Triple Modular Redundancy	198
12.6 Tolerating Omission Failures	199
12.6.1 The Sliding Window Protocol	200
12.6.2 The Alternating Bit Protocol	202
12.6.3 How TCP Works	203
12.7 Concluding Remarks	203
12.8 Bibliographic Notes	204
12.9 Exercises	205

Chapter 13

Distributed Consensus	209
13.1 Introduction	209
13.2 Consensus in Asynchronous Systems	210
13.3 Consensus on Synchronous Systems: Byzantine Generals Problem	213
13.3.1 The Solution with No Traitor	214
13.3.2 Solution with Traitors: Interactive Consistency Requirements	214
13.3.3 Consensus with Oral Messages	214
13.3.3.1 An Impossibility Result	215
13.3.3.2 The OM(m) Algorithm	216
13.3.4 Consensus Using Signed Messages	219
13.4 Failure Detectors	220
13.4.1 Solving Consensus Using Failure Detectors	222

13.4.1.1	Consensus with P	223
13.4.1.2	Consensus Using S	223
13.5	Concluding Remarks	224
13.6	Bibliographic Notes	225
13.7	Exercises	225

Chapter 14

Distributed Transactions	227
14.1 Introduction	227
14.2 Classification of Transactions	228
14.3 Implementing Transactions	229
14.4 Concurrency Control and Serializability	229
14.4.1 Testing for Serializability	230
14.4.2 Two-Phase Locking	231
14.4.3 Timestamp Ordering	231
14.5 Atomic Commit Protocols	232
14.5.1 One-Phase Commit	233
14.5.2 Two-Phase Commit	233
14.5.3 Non-Blocking Atomic Commit	234
14.6 Recovery from Failures	235
14.6.1 Stable Storage	235
14.6.2 Checkpointing and Rollback Recovery	236
14.6.3 Message Logging	237
14.7 Concluding Remarks	238
14.8 Bibliographic Notes	239
14.9 Exercises	239

Chapter 15

Group Communication	243
15.1 Introduction	243
15.2 Atomic Multicast	243
15.3 IP Multicast	244
15.4 Application Layer Multicast	246
15.5 Ordered Multicasts	247
15.5.1 Implementing Total Order Multicast	248
15.5.2 Implementing Causal Order Multicast	249
15.6 Reliable Ordered Multicast	250
15.6.1 The Requirements of Reliable Multicast	250
15.6.2 Scalable Reliable Multicast	251
15.7 Open Groups	252
15.7.1 View-Synchronous Group Communication	253
15.8 An Overview of Transis	254
15.9 Concluding Remarks	255
15.10 Bibliographic Notes	255
15.11 Exercises	256

Chapter 16

Replicated Data Management	263
16.1 Introduction	263
16.1.1 Reliability vs. Availability	263
16.2 Architecture of Replicated Data Management	264

16.2.1	Passive vs. Active Replication	264
16.2.2	Fault-Tolerant State Machines	266
16.3	Data-Centric Consistency Models	267
16.3.1	Strict Consistency	267
16.3.2	Linearizability	268
16.3.3	Sequential Consistency	269
16.3.4	Causal Consistency	270
16.4	Client-Centric Consistency Protocols	270
16.4.1	Eventual Consistency	271
16.4.2	Consistency Models for Mobile Clients	271
16.5	Implementation of Data-Centric Consistency Models	272
16.5.1	Quorum-Based Protocols	272
16.6	Replica Placement	273
16.7	Case Studies	274
16.7.1	Replication Management in Coda	274
16.7.2	Replication Management in Bayou	275
16.7.3	Gossip Architecture	276
16.8	Concluding Remarks	277
16.9	Bibliographic Notes	278
16.10	Exercises	278

Chapter 17

Self-Stabilizing Systems	281
17.1 Introduction	281
17.2 Theoretical Foundations	282
17.3 Stabilizing Mutual Exclusion.....	283
17.3.1 Mutual Exclusion on a Unidirectional Ring	283
17.3.2 Mutual Exclusion on a Bidirectional Array	285
17.4 Stabilizing Graph Coloring	287
17.5 Stabilizing Spanning Tree Protocol	290
17.6 Distributed Reset	291
17.7 Stabilizing Clock Synchronization	294
17.8 Concluding Remarks	295
17.9 Bibliographic Notes.....	296
17.10 Exercises	296

Part E

Real World Issues	301
-------------------------	-----

Chapter 18

Distributed Discrete-Event Simulation	303
18.1 Introduction	303
18.1.1 Event-Driven Simulation	303
18.2 Distributed Simulation	305
18.2.1 The Challenges	305
18.2.2 Correctness Issues	307
18.3 Conservative Simulation	308
18.4 Optimistic Simulation and Time Warp	308
18.4.1 Global Virtual Time	309
18.5 Concluding Remarks	310

18.6	Bibliographic Notes	310
18.7	Exercises	310

Chapter 19

Security in Distributed Systems	313
19.1 Introduction	313
19.2 Security Mechanisms	314
19.3 Common Security Attacks	314
19.4 Encryption	315
19.5 Secret-Key Cryptosystem	317
19.5.1 Confusion and Diffusion	317
19.5.2 DES	318
19.5.3 3DES	319
19.5.4 AES	319
19.5.5 One-Time Pad	319
19.5.6 Stream Ciphers	320
19.5.7 Steganography	321
19.6 Public-Key Cryptosystems	321
19.6.1 The Rivest–Shamir–Adleman (RSA) Method	322
19.6.2 ElGamal Cryptosystem	323
19.7 Digital Signatures	324
19.7.1 Signatures in Secret-Key Cryptosystems	324
19.7.2 Signatures in Public-Key Cryptosystems	325
19.8 Hashing Algorithms	325
19.8.1 Birthday Attack	325
19.9 Elliptic Curve Cryptography	326
19.10 Authentication Server	327
19.10.1 Authentication Server for Secret-Key Cryptosystems	327
19.10.2 Authentication Server for Public-Key Cryptosystems	328
19.11 Digital Certificates	328
19.12 Case Studies	329
19.12.1 Kerberos	329
19.12.2 Pretty Good Privacy (PGP)	330
19.12.3 Secure Socket Layer (SSL)	331
19.13 Virtual Private Networks (VPN) and Firewalls	332
19.13.1 VPN	332
19.13.2 Firewall	333
19.14 Sharing a Secret	333
19.15 Concluding Remarks	334
19.16 Bibliographic Notes	334
19.17 Exercises	334

Chapter 20

Sensor Networks	339
20.1 The Vision	339
20.2 Architecture of a Sensor Node	339
20.2.1 MICA Mote	340
20.2.2 ZigBee Enabled Sensor Nodes	340
20.2.3 TinyOS Operating System	342
20.3 The Challenges in Wireless Sensor Networks	344
20.3.1 Energy Conservation	345

20.3.2	Fault-Tolerance	346
20.3.3	Routing	346
20.3.4	Time Synchronization	346
20.3.5	Location Management	346
20.3.6	Middleware Design	346
20.3.7	Security	346
20.4	Routing Algorithms	347
20.4.1	Directed Diffusion	347
20.4.2	Cluster-Based Routing	348
20.4.2.1	LEACH	348
20.4.3	PEGASIS	348
20.4.4	Meta-Data Based Routing: SPIN	349
20.5	Time Synchronization Using Reference Broadcast	349
20.5.1	Reference Broadcast	350
20.6	Localization Algorithms	351
20.6.1	RSSI Based Ranging	352
20.6.2	Ranging Using Time Difference of Arrival	352
20.6.3	Anchor-Based Ranging	352
20.7	Security in Sensor Networks	353
20.7.1	SPIN for Data Security	353
20.7.1.1	An Overview of SNEP	354
20.7.1.2	An Overview of μ TESLA	354
20.7.2	Attacks on Routing	355
20.8	Sample Application: Pursuer–Evader Games	355
20.9	Concluding Remarks	357
20.10	Bibliographic Notes	358
20.11	Exercises	358

Chapter 21

Peer-to-Peer Networks	363
21.1 Introduction	363
21.2 The First-Generation P2P Systems	363
21.2.1 Napster	364
21.2.2 Gnutella	364
21.3 The Second-Generation P2P Systems	365
21.3.1 KaZaA	366
21.3.2 Chord	366
21.3.3 Content Addressable Network (CAN)	368
21.3.4 Pastry	370
21.4 Koorde and De Bruijn Graph	371
21.5 The Small-World Phenomenon	372
21.6 Skip Graph	374
21.7 Replication Management	376
21.8 Free Riders and Bittorrent	377
21.9 Censorship Resistance, Anonymity, and Ethical Issues	377
21.10 Concluding Remarks	378
21.11 Bibliographic Notes	378

Bibliography	383
--------------------	-----

Dedication

*This book is dedicated to the memory of my parents.
They would have been the happiest persons to see this in print.*

*I acknowledge the support of my wife Sumita and my son Soumya for
their patience. I sincerely appreciate the encouragement of my former
students Chandan Mazumdar and Arobinda Gupta in completing this
project.*