Computer Communications and Networks

For further volumes:
www.springer.com/series/4198

The Computer Communications and Networks series is a range of textbooks, monographs and handbooks. It sets out to provide students, researchers and non-specialists alike with a sure grounding in current knowledge, together with comprehensible access to the latest developments in computer communications and networking.

Emphasis is placed on clear and explanatory styles that support a tutorial approach, so that even the most complex of topics is presented in a lucid and intelligible manner.

K. Erciyes

Distributed Graph Algorithms for Computer Networks

Springer

K. Erciyes
Computer Engineering Department
Izmir University
Uckuyular, Izmir, Turkey

Series Editor
A.J. Sammes
Centre for Forensic Computing
Cranfield University
Shrivenham campus
Swindon, UK

ISSN 1617-7975 Computer Communications and Networks
ISBN 978-1-4471-5172-2 ISBN 978-1-4471-5173-9 (eBook)
DOI 10.1007/978-1-4471-5173-9
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2013938954

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To the memories of Necdet Doğanata and Selçuk Erciyes, and all who believe in education

Preface

Distributed systems consisting of a number of autonomous computing elements connected over a communication network that cooperate to achieve common goals have shown an unprecedented growth in the last few decades, especially in the form of the Grid, the Cloud, mobile ad hoc networks, and wireless sensor networks. Design of algorithms for these systems, namely the distributed algorithms, has become an important research area of computer science, engineering, applied mathematics, and other disciplines as they pose different and usually more difficult problems than the sequential algorithms. A graph can be used to conveniently model a distributed system, and distributed graph algorithms or graph-theoretical distributed algorithms, in the context of this book, are considered as distributed algorithms that make use of some property of the graph that models the distributed system to solve a problem in such systems.

This book is about distributed graph algorithms as applied to computer networks with focus on implementation and hopefully without much sacrifice on the theory. It grew out of the need I have witnessed while teaching distributed systems and algorithms courses in the last two decades or so. The main observation was that although there were many books on distributed algorithms, graph theory, and ad hoc networks separately, there did not seem to be any book with detailed focus on the intersection of these three major areas of research. The second observation was the difficulty the students faced when implementing distributed algorithm code although the concepts and the idea of an algorithm in an abstract manner were perceived relatively more comfortably. For example, when and how to synchronize algorithms running on different computing nodes was one of the main difficulties. In this sense, we have attempted to provide algorithms in ready-to-be-coded format in most cases, showing minor details explicitly to aid the distributed algorithm designer and implementor.

The book is divided into three parts. After reviewing the background, Part I provides a review of the fundamental and better known distributed graph algorithms. Part II describes the core concepts of distributed graph algorithms that have wide range of applications in computer networks in an abstract manner, without considering the application environment. However, in Part III, we focus ourselves on ad hoc wireless networks and show how some of the algorithms we have investigated can be modified for this environment.

The layout of each chapter is kept quite uniform for ease of reading. Each chapter starts with an introduction describing the problem shortly by showing its possible applications in computer networks. The problem is then stated formally, and examples are provided in most of the cases. We then provide a list of algorithms usually starting by a sequential one to aid understanding the problem better. The distributed algorithms shown may be well established if they exist and sometimes algorithms that have been recently published as articles are described with examples if they have profound effect on the solution of the problem.

An algorithm is first introduced conceptually, and then, its pseudocode is given and described in detail. We provide similar simple graph templates to show the steps of the implementation of the algorithm and then provide analysis of its time and message complexity. Proof of correctness is given only when this does not seem obvious or, on the contrary, a reference is given for the proof if this requires lengthy analysis. The chapter concludes by the Chapter Notes section, which usually emphasizes main points, compares the described algorithms, and also provides a contemporary bibliographic review of the topic with open research areas where applicable. This style is repeated throughout the book for all chapters. Exercises at the end of chapters are usually in the form of small programming projects in line with the main goal of the book, which is to describe how to implement distributed algorithms.

There are few aspects of the book worth mentioning. Firstly, many self-stabilizing algorithms are included, some being very recent, for most of the topics covered in Part II. There are few algorithms, again in Part II, that are new and have not been published elsewhere. Also, an updated survey of the topic covered is provided for all chapters. Finally, a simple simulator we have designed, implemented, and used while teaching distributed algorithm courses is included as the final chapter, and its source code is given in Appendix B.

The intended audience for this book are the graduate students and researchers of computer science and mathematics and engineering or any person with basic background in discrete mathematics, algorithms, and computer networks.

I would like to thank graduate students at Ege University, University of California Davis, California State University San Marcos and senior students at Izmir University who have taken the distributed algorithms courses, sometimes under slightly different names, for their valuable feedback when parts of the material covered in the book was presented during lectures. I would like to thank Aysegul Alaybeyoglu, Deniz Cokuslu, Orhan Dagdeviren, and Jukka Suomela for their review of some chapters and valuable comments. I would also like to thank Springer editors Wayne Wheeler and Simon Rees for their continuous support during the course of this project and Donatas Akmanavičius for the final editing process.

Izmir, Turkey K. Erciyes

Contents

1 Introduction 1
1.1 Distributed Systems 1
1.2 Distributed Computing Platforms 2
1.2.1 The Grid 2
1.2.2 Cloud Computing 3
1.2.3 Mobile Ad hoc Networks 3
1.2.4 Wireless Sensor Networks 3
1.3 Models 4
1.4 Software Architecture 4
1.5 Design Issues 5
1.5.1 Synchronization 5
1.5.2 Load Balancing 6
1.5.3 Fault Tolerance 6
1.6 Distributed Graph Algorithms 6
1.7 Organization of the Book 7
References 8

Part I Fundamental Algorithms

2 Graphs 11
2.1 Definition of Graphs 11
2.1.1 Special Graphs 13
2.1.2 Graph Representations 14
2.2 Walks, Paths and Cycles 14
2.2.1 Diameter, Radius, Circumference, and Girth 15
2.3 Subgraphs 16
2.4 Connectivity 17
2.4.1 Cutpoints and Bridges 18
2.5 Trees 19
2.5.1 Minimum Spanning Trees 19

2.6 Chapter Notes . . . 20
2.6.1 Exercises . . . 20
References . . . 21

3 The Computational Model . . . 23
3.1 Introduction . . . 23
3.2 Message Passing . . . 24
3.3 Finite-State Machines . . . 26
3.3.1 Moore Machine Example: Parity Checker . . . 27
3.3.2 Mealy Machine Example: Data Link Protocol Design . . . 27
3.4 Synchronization . . . 29
3.5 Communication Primitives . . . 30
3.6 Application Level Synchronization . . . 32
3.7 Performance Metrics . . . 34
3.7.1 Time Complexity . . . 34
3.7.2 Bit Complexity . . . 34
3.7.3 Space Complexity . . . 34
3.7.4 Message Complexity . . . 34
3.8 Chapter Notes . . . 36
3.8.1 Exercises . . . 36
References . . . 37

4 Spanning Tree Construction . . . 39
4.1 Introduction . . . 39
4.2 The Flooding Algorithm . . . 39
4.2.1 Analysis . . . 40
4.3 Flooding-Based Asynchronous Spanning Tree Construction . . . 41
4.3.1 Analysis . . . 42
4.4 An Asynchronous Algorithm with Termination Detection . . . 43
4.4.1 Analysis . . . 45
4.5 Tarry's Traversal Algorithm . . . 46
4.5.1 Analysis . . . 47
4.6 Convergecast and Broadcast over a Spanning Tree . . . 47
4.7 Chapter Notes . . . 49
4.7.1 Exercises . . . 51
References . . . 51

5 Graph Traversals . . . 53
5.1 Introduction . . . 53
5.2 Breadth-First-Search Algorithms . . . 54
5.2.1 Synchronous BFS Construction . . . 54
5.2.2 Asynchronous BFS Construction . . . 58
5.2.3 Analysis . . . 60
5.3 Depth-First-Search Algorithms . . . 60
5.3.1 The Classical DFS Algorithm . . . 61
5.3.2 Awerbuch's DFS Algorithm . . . 63

5.3.3 Distributed DFS with Neighbor Knowledge 64
5.4 Chapter Notes 66
5.4.1 Exercises 66
References 67

6 Minimum Spanning Trees 69
6.1 Introduction 69
6.2 Sequential MST Algorithms 69
6.3 Synchronous Distributed Prim Algorithm 71
6.3.1 Analysis 73
6.4 Synchronous GHS Algorithm 74
6.4.1 Analysis 75
6.5 Asynchronous GHS Algorithm 77
6.5.1 States of Nodes and Links 77
6.5.2 Searching MWOE 77
6.5.3 The Algorithm 78
6.6 Chapter Notes 79
6.6.1 Exercises 81
References 82

7 Routing 83
7.1 Introduction 83
7.2 Sequential Routing Algorithms 83
7.2.1 Dijkstra's Algorithm 84
7.2.2 Bellman–Ford Algorithm 85
7.2.3 All-Pairs Shortest-Paths Routing Algorithm 86
7.3 The Distributed Floyd–Warshall Algorithm 88
7.4 Toueg's Algorithm 89
7.4.1 Analysis 90
7.5 Synchronous Distributed Bellman–Ford Algorithm 92
7.5.1 Analysis 92
7.6 Chandy–Misra Algorithm 93
7.7 Routing Protocols 94
7.7.1 Link State Protocol 94
7.7.2 Distance Vector Protocol 94
7.8 Chapter Notes 95
7.8.1 Exercises 95
References 96

8 Self-Stabilization 97
8.1 Introduction 97
8.2 Models 98
8.2.1 Anonymous or Identifier-Based Networks 98
8.2.2 Deterministic, Randomized, or Probabilistic Algorithms . . 98
8.3 Dijkstra's Self-Stabilizing Mutual Exclusion Algorithm 99

8.4 BFS Tree Construction . . . 99
8.4.1 Dolev, Israeli, and Moran Algorithm . . . 99
8.4.2 Afek, Kutten, and Yung Algorithm . . . 101
8.5 Self-Stabilizing DFS . . . 101
8.6 Chapter Notes . . . 102
8.6.1 Exercises . . . 103
References . . . 104

Part II Graph Theoretical Algorithms

9 Vertex Coloring . . . 107
9.1 Introduction . . . 107
9.2 Sequential Algorithms . . . 108
9.2.1 Analysis . . . 109
9.3 Distributed Coloring Algorithms . . . 110
9.3.1 The Greedy Distributed Algorithm . . . 110
9.3.2 Random Vertex Coloring . . . 112
9.3.3 A Simple Reduction Algorithm . . . 113
9.4 Edge Coloring . . . 117
9.4.1 Analysis . . . 122
9.4.2 The Second Version . . . 123
9.5 Coloring Trees . . . 124
9.5.1 A Simple Tree Algorithm . . . 124
9.5.2 Six Coloring Algorithm . . . 125
9.5.3 Six-to-Two Coloring Algorithm . . . 126
9.6 Self-Stabilizing Vertex Coloring . . . 128
9.6.1 Coloring Planar Graphs . . . 129
9.6.2 Coloring Arbitrary Graphs . . . 130
9.7 Chapter Notes . . . 132
9.7.1 Exercises . . . 133
References . . . 134

10 Maximal Independent Sets . . . 135
10.1 Introduction . . . 135
10.2 The Sequential Algorithm . . . 136
10.3 Rank-Based Distributed MIS Algorithm . . . 137
10.3.1 Analysis . . . 140
10.4 The First Random MIS Algorithm . . . 141
10.4.1 Analysis . . . 144
10.5 The Second Random MIS Algorithm . . . 144
10.5.1 Analysis . . . 144
10.6 MIS Construction from Vertex Coloring . . . 145
10.6.1 Analysis . . . 146
10.7 Self-Stabilizing MIS Algorithms . . . 147
10.7.1 Shukla's Algorithm . . . 147
10.7.2 Ikeda's Algorithm . . . 150
10.7.3 Turau's Algorithm . . . 151

10.8 Chapter Notes 153
10.8.1 Exercises 154
References 155

11 Dominating Sets 157
11.1 Introduction 157
11.2 Sequential Algorithms 158
11.2.1 Greedy Sequential MDS Algorithm 158
11.2.2 Greedy Sequential MCDS Algorithm 159
11.2.3 Guha–Khuller Algorithms 160
11.3 Distributed Algorithms 162
11.3.1 Greedy MDS Algorithm 163
11.3.2 Greedy MCDS Algorithm 165
11.3.3 The Two-Span MDS Algorithm 166
11.4 Self-Stabilizing Domination 167
11.4.1 Dominating Set Algorithm 167
11.4.2 Minimal Dominating Set Algorithm 168
11.5 Chapter Notes 169
11.5.1 Exercises 170
References 171

12 Matching 173
12.1 Introduction 173
12.2 Unweighted Matching 174
12.2.1 A Sequential Algorithm 175
12.2.2 The Greedy Distributed Algorithm 175
12.2.3 A Three-Phase Synchronous Distributed Algorithm 178
12.2.4 Matching from Edge Coloring 180
12.3 Weighted Matching 183
12.3.1 The Greedy Sequential Algorithm 183
12.3.2 Hoepman's Algorithm 184
12.4 Self-Stabilizing Matching 185
12.4.1 Hsu and Huang Algorithm 186
12.4.2 Synchronous Matching 186
12.4.3 Weighted Matching 187
12.5 Chapter Notes 188
12.5.1 Exercises 189
References 190

13 Vertex Cover 193
13.1 Introduction 193
13.2 Unweighted Vertex Cover 195
13.2.1 Sequential Algorithms 195
13.2.2 Greedy Distributed MVC Algorithm 198
13.2.3 Connected Vertex Cover 201
13.2.4 Vertex Cover by Bipartite Matching 202

13.3 Minimal Weighted Vertex Cover 204
13.3.1 Pricing Algorithm 205
13.3.2 The Greedy Distributed MWVC Algorithm 206
13.4 Self-Stabilizing Vertex Cover 206
13.4.1 A $2 - 1/\Delta$ Approximation Algorithm 206
13.4.2 Bipartite Matching-Based Algorithm 210
13.5 Chapter Notes 211
13.5.1 Exercises 212
References 213

Part III Ad Hoc Wireless Networks

14 Introduction 217
14.1 Ad Hoc Wireless Networks 217
14.2 Mobile Ad Hoc Networks 217
14.3 Wireless Sensor Networks 220
14.4 Ad Hoc Wireless Network Models 221
14.4.1 Unit Disk Graph Model 221
14.4.2 Quasi Unit Disk Graph Model 222
14.4.3 Interference Models 223
14.5 Energy Considerations 224
14.6 Mobility Models 225
14.7 Simulation 225
14.7.1 *ns*2 225
14.7.2 TOSSIM 226
14.7.3 Other Simulators 226
14.8 Chapter Notes 227
14.8.1 Exercises 227
References 227

15 Topology Control 229
15.1 Introduction 229
15.2 Desirable Properties 230
15.2.1 Connectivity 230
15.2.2 Low Stretch Factors 231
15.2.3 Bounded Node Degree 231
15.3 Locally Defined Graphs 232
15.3.1 Nearest-Neighbor Graphs 232
15.3.2 Gabriel Graphs 234
15.3.3 Relative Neighborhood Graphs 235
15.3.4 Delaunay Triangulation 235
15.3.5 Yao Graphs 237
15.3.6 Cone-Based Topology Control 238
15.4 Clustering 238
15.4.1 Clustering in Sensor Networks 239
15.4.2 Clustering in MANETs 240

15.4.3 Performance Metrics . . . 240
15.4.4 Lowest-ID Algorithm . . . 241
15.4.5 Highest Connectivity Algorithm . . . 243
15.4.6 Lowest-Id Algorithm: Second Version . . . 244
15.4.7 *k*-Hop Clustering . . . 246
15.4.8 Spanning-Tree-Based Clustering . . . 247
15.5 Connected Dominating Sets . . . 248
15.5.1 A Sequential Algorithm using MIS . . . 249
15.5.2 Greedy Distributed Algorithms . . . 250
15.5.3 MIS-Based Distributed CDS Construction . . . 250
15.5.4 Pruning-Based Algorithm . . . 252
15.6 Chapter Notes . . . 254
15.6.1 Exercises . . . 255
References . . . 256

16 Ad Hoc Routing . . . 259
16.1 Introduction . . . 259
16.2 Characteristics of Ad Hoc Routing Protocols . . . 259
16.2.1 Proactive and Reactive Protocols . . . 260
16.3 Routing in Mobile Ad Hoc Networks . . . 261
16.3.1 Proactive Protocols . . . 261
16.3.2 Reactive Protocols . . . 264
16.3.3 Hybrid Routing Protocols . . . 268
16.4 Routing in Sensor Networks . . . 268
16.4.1 Data-Centric Protocols . . . 269
16.4.2 Hierarchical Protocols . . . 271
16.4.3 Location-Based Routing . . . 272
16.5 Chapter Notes . . . 274
16.5.1 Exercises . . . 274
References . . . 274

17 Sensor Network Applications . . . 277
17.1 Localization . . . 277
17.1.1 Range-Based Localization . . . 278
17.1.2 Range-Free Localization . . . 279
17.1.3 Localization with Range Estimate . . . 280
17.2 Target Tracking . . . 281
17.2.1 Cluster-Based Approaches . . . 282
17.2.2 Tree-Based Approaches . . . 288
17.2.3 Prediction-Based Approaches . . . 291
17.2.4 Lookahead Target Tracking . . . 291
17.3 Chapter Notes . . . 292
17.3.1 Exercises . . . 293
References . . . 293

18 ASSIST: A Simulator to Develop Distributed Algorithms 295
18.1 Introduction 295
18.2 Memory Management by Buffer Pools 295
18.3 Interprocess Communication 296
18.4 Sliding-Window Protocol Implementation 298
18.5 Spanning Tree Construction 299
18.5.1 Data Structures and Initialization 300
18.5.2 The Algorithm Thread 301
18.6 Chapter Notes 302
18.6.1 Projects 303

Appendix A Pseudocode Conventions 305
A.1 Introduction 305
A.2 Data Structures 305
A.3 Control Structures 306
A.3.1 Selection 307
A.3.2 Repetition 308
A.4 Distributed Algorithm Structure 308
References 309

Appendix B ASSIST Code 311
B.1 Buffer Pool Management 311
B.2 Interprocess Communication 313

Appendix C Applications Using ASSIST 315
C.1 Sliding-Window Protocol Code 315
C.1.1 Data Structures and Initialization 315
C.2 Spanning Tree Code 317
C.2.1 Data Structures and Initialization 317
C.2.2 Tree Construction Thread 317
C.2.3 Actions 319
C.2.4 The Main Thread 320

Index 321

Acronyms

AoA	Angle of Arrival
APSP	All Pairs Shortest Paths
ASSIST	A Simple Simulator based on Threads
BFS	Breadth First Search
CDS	Connected Dominating Set
DFS	Depth First Search
DS	Dominating Set
DT	Delaunay Triangulation
EKF	Extended Kalman Filter
FSM	Finite State Machine
GG	Gabriel Graph
IS	Independent Set
KF	Kalman Filter
k-NNG	k-Nearest Neighbor Graph
MaxIS	Maximum Independent Set
MaxM	Maximum Matching
MaxWM	Maximum Weighted Matching
MCDS	Minimal Connected Dominating Set
MCVC	Minimal Connected Vertex Cover
MCWVC	Minimal Connected Weighted Vertex Cover
MDS	Minimal Dominating Set
MinCDS	Minimum Connected Dominating Set
MinCVC	Minimum Connected Vertex Cover
MinCWVC	Minimum Connected Weighted Vertex Cover
MinDS	Minimum Dominating Set
MinVC	Minimum Vertex Cover
MinWVC	Minimum Weighted Vertex Cover
MIS	Maximal Weighted Matching
MM	Maximal Matching
MST	Minimum Spanning Tree
MVC	Minimal Vertex Cover

MWM	Maximal Weighted Matching
MWOE	Minimum Weight Outgoing Edge
MWVC	Minimal Weighted Vertex Cover
NNG	Nearest-Neighbor Graph
PF	Particle Filter
QUDG	Quasi Unit Disk Graph
RNG	Relative Neighborhood Graph
RSSI	Received Signal Strength Indicator
SSSP	Single-Source Shortest Paths
TDoA	Time Difference of Arrival
UDG	Unit Disk Graph
VC	Vertex Cover
YG	Yao Graph