1. Record Nr. UNINA9910830875303321

Autore	Santoro N (Nicola), <1951->
Titolo	Design and analysis of distributed algorithms [[electronic resource] /] / Nicola Santoro
Pubbl/distr/stampa	Hoboken, N.J., : Wiley-Interscience, c2007
ISBN	1-280-72154-5 9786610721542 0-470-07264-4 0-470-07263-6
Descrizione fisica	1 online resource (610 p.)
Collana	Wiley series on parallel and distributed computing
Disciplina	005.1
Soggetti	Electronic data processing - Distributed processing Computer algorithms
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and indexes.
Nota di contenuto	DESIGN AND ANALYSIS OF DISTRIBUTED ALGORITHMS; CONTENTS; Preface; 1 Distributed Computing Environments; 1.1 Entities; 1.2 Communication; 1.3 Axioms and Restrictions; 1.3.1 Axioms; 1.3.2 Restrictions; 1.4 Cost and Complexity; 1.4.1 Amount of Communication Activities; 1.4.2 Time; 1.5 An Example: Broadcasting; 1.6 States and Events; 1.6.1 Time and Events; 1.6.2 States and Configurations; 1.7 Problems and Solutions (*); 1.8 Knowledge; 1.8.1 Levels of Knowledge; 1.8.2 Types of Knowledge; 1.9 Technical Considerations; 1.9.1 Messages; 1.9.2 Protocol; 1.9.3 Communication Mechanism 1.10 Summary of Definitions1.11 Bibliographical Notes; 1.12 Exercises, Problems, and Answers; 1.12.1 Exercises and Problems; 1.12.2 Answers to Exercises; 2 Basic Problems And Protocols; 2.1 Broadcast; 2.1.1 The Problem; 2.1.2 Cost of Broadcasting; 2.1.3 Broadcasting in Special Networks; 2.2 Wake-Up; 2.2.1 Generic Wake-Up; 2.2.2 Wake-Up in Special Networks; 2.3 Traversal; 2.3.1 Depth-First Traversal; 2.3.2 Hacking (*); 2.3.3 Traversal in Special Networks; 2.3.4 Considerations on Traversal; 2.4 Practical Implications: Use a Subnet; 2.5 Constructing a Spanning Tree 2.5.1 SPT Construction with a Single Initiator: Shout2.5.2 Other SPT

Constructions with Single Initiator; 2.5.3 Considerations on the Constructed Tree; 2.5.4 Application: Better Traversal; 2.5.5 Spanning-Tree Construction with Multiple Initiators; 2.5.6 Impossibility Result; 2.5.7 SPT with Initial Distinct Values; 2.6 Computations in Trees; 2.6.1 Saturation: A Basic Technique; 2.6.2 Minimum Finding; 2.6.3 Distributed Function Evaluation; 2.6.4 Finding Eccentricities; 2.6.5 Center Finding; 2.6.6 Other Computations; 2.6.7 Computing in Rooted Trees; 2.7 Summary; 2.7.1 Summary of Problems
2.7.2 Summary of Techniques2.8 Bibliographical Notes; 2.9 Exercises, Problems, and Answers; 2.9.1 Exercises; 2.9.2 Problems; 2.9.3 Answers to Exercises; 3 Election; 3.1 Introduction; 3.1.1 Impossibility Result; 3.1.2 Additional Restrictions; 3.1.3 Solution Strategies; 3.2 Election in Trees; 3.3 Election in Rings; 3.3.1 All the Way; 3.3.2 As Far As It Can; 3.3.3 Controlled Distance; 3.3.4 Electoral Stages; 3.3.5 Stages with Feedback; 3.3.6 Alternating Steps; 3.3.7 Unidirectional Protocols; 3.3.8 Limits to Improvements (*); 3.3.9 Summary and Lessons; 3.4 Election in Mesh Networks; 3.4.1 Meshes
3.4.2 Tori3.5 Election in Cube Networks; 3.5.1 Oriented Hypercubes; 3.5.2 Unoriented Hypercubes; 3.6 Election in Complete Networks; 3.6.1 Stages and Territory; 3.6.2 Surprising Limitation; 3.6.3 Harvesting the Communication Power; 3.7 Election in Chordal Rings (*); 3.7.1 Chordal Rings; 3.7.2 Lower Bounds; 3.8 Universal Election Protocols; 3.8.1 Mega-Merger; 3.8.2 Analysis of Mega-Merger; 3.8.3 YO-YO; 3.8.4 Lower Bounds and Equivalences; 3.9 Bibliographical Notes; 3.10 Exercises, Problems, and Answers; 3.10.1 Exercises; 3.10.2 Problems; 3.10.3 Answers to Exercises
4 Message Routing and Shortest Paths

Sommario/riassunto

This text is based on a simple and fully reactive computational model that allows for intuitive comprehension and logical designs. The principles and techniques presented can be applied to any distributed computing environment (e.g., distributed systems, communication networks, data networks, grid networks, internet, etc.). The text provides a wealth of unique material for learning how to design algorithms and protocols perform tasks efficiently in a distributed computing environment.