Networks and Distributed Computation

Concepts, tools and algorithms

Michel Raynal

IRISA, Department of Computer Science
University of Rennes

Translated by
Meg Sanders

Technische Hochschule Darmstadt
FACHBEREICH INFORMATIK
BIBLIOTHEK
Inventar-Nr.: 7828
Sachgebiete: ....
Standort: ....

NORTH OXFORD ACADEMIC
A Division of Kogan Page

Contents

Foreword xi

Preface xiii

PART 1: Introductory examples 1

1. First example: a data transfer protocol 3

1.1 The problem 3
1.1.1. Specification of constraints 3
1.1.2. Problem invariant 4
1.2 A centralized implementation 5
1.3 A distributed implementation 7
1.3.1. Assumptions and structure 7
1.3.2. Events: location and observation 7
1.3.3. Example of an implementation 10
1.4 The unreliable environment case 12
1.4.1. Assumptions concerning channels 12
1.4.2. Fundamentals of the solution 13
1.4.2.1. Sequence numbers 13
1.4.2.2. Basic protocol 13
1.4.2.3. Calibration of the guard delay 14
1.4.2.4. Windows and credits 14
1.4.3. Structure of the solution 17
1.4.3.1. Choice of implementation 17
1.4.3.2. Structure 17
1.4.4. An implementation 18
1.4.4.1. The site of production 18
1.4.4.2. The site of consumption 19
1.4.4.3. Flow control 21
1.5. Bounding the ranges of variation 22
1.5.1. Counters 22
1.5.1.1. Further counters 22
1.5.1.2. Modulo implementation 22

1.5.1.3. Implementation with ancillary variables 24
1.5.1.4. Comparison of the two implementations 25
1.5.2. Sequence numbers 26
1.5.2.1. The problem of duplication 26
1.5.2.2. Definition of the range of variation 26
1.5.2.3. Timing constraints 27
1.6. References 29

2. Second example: mutual control of logical clocks 31

2.1. The problem 31
2.2. The two-clock case 32
2.2.1. System structure 32
2.2.2. A solution 33
2.2.2.1. Principle 33
2.2.2.2. The Carvalho–Roucairol algorithm 34
2.2.3. A variant 35
2.2.3.1. The assumption of non-desequencing 35
2.2.3.2. Range of values carried by the messages 35
2.2.3.3. The algorithm 36
2.2.3.4. Importance 37
2.2.4. Desymmetrizing the problem 37
2.3. The *n*-clock 37
2.3.1. Principle of generalization 37
2.3.2. First solution 38
2.3.3. Second solution 40
2.4. Arbitrary network 41
2.4.1. Algorithms in an arbitrary network 41
2.4.2. Channel breakdowns 42
2.4.2.1. Effects of breakdowns 42
2.4.2.2. Resistance to breakdowns 44
2.5. References 45

PART 2: Two fundamental elements: elementary algorithms and global state 47

3. Basic algorithms and protocols 49

3.1. Introduction 49
3.2. Assumptions 50
3.2.1. Assumptions concerning the network 50
3.2.2. Assumptions concerning initial knowledge 50
3.2.3. Importance of these assumptions 50
3.3. Traversal of a network 50
3.3.1. Depth-first traversal 51
3.3.1.1. The problem 51
3.3.1.2. Principle of the solution 51

3.3.1.3. The algorithm 51
3.3.1.4. Evaluation 52
3.3.1.5. Uses 53
3.3.2. Parallel traversal 53
3.3.2.1. The problem and the principle of its solution 53
3.3.2.2. The algorithm 54
3.3.2.3. Comments 56
3.3.2.4. Evaluation 56
3.3.2.5. Uses 56
3.3.3. Generalizations 57
3.4. Learning the network 58
3.4.1. Learning problems 58
3.4.2. Learning the identities 59
3.4.2.1. Identities of neighbours 59
3.4.2.2. Identities of all the sites 60
3.4.3. Learning the network 61
3.5. Finding the shortest routes 63
3.5.1. Context of the problem 63
3.5.2. The algorithm 63
3.5.2.1. Context of the sites 63
3.5.2.2. Principle of the algorithm 64
3.5.2.3. The algorithm 65
3.5.2.4. Termination 66
3.6. Resilience to modifications of the network 66
3.6.1. Types of modification 66
3.6.2. Breakdowns and channel reinsertions 67
3.6.2.1. Modification messages 67
3.6.2.2. The initial context 67
3.6.2.3. Site behaviour 68
3.6.2.4. The algorithm 69
3.6.2.5. Properties of the algorithm 70
3.7. A general principle 71
3.8. References 71

4. Determination of a global state 73

4.1. The global state problem 73
4.1.1. Introduction 73
4.1.2. Context and definitions 74
4.1.2.1. Assumptions 74
4.1.2.2. Operational view 74
4.1.2.3. Observational view 75
4.1.3. The global state problem 75
4.1.3.1. General characteristics 75
4.1.3.2. Difficulties with the algorithm 76
4.2. Derivation of an algorithm 77

4.2.1. A centralized solution 77
4.2.1.1. Principle 77
4.2.1.2. The algorithm 77
4.2.1.3. A control message: the marker 78
4.2.2. Local logical clocks: use of a virtual global time 78
4.2.3. An initial algorithm 79
4.3. The Chandy and Lamport algorithm 79
4.3.1. Elimination of logical clocks 79
4.3.2. The algorithm 80
4.3.3. Termination 81
4.3.4. The Chandy and Lamport formulation 81
4.3.5. Generalization 82
4.3.6. Characteristics of the global states obtained 82
4.3.6.1. Global state and event set 82
4.3.6.2. Properties of the state obtained 84
4.4. Two uses of the global states 84
4.4.1. Detection of stable properties 84
4.4.1.1. General detection algorithm 85
4.4.1.2. Interpretation of the result 85
4.4.1.3. Specific algorithms 85
4.4.2. Recovery after breakdown 86
4.4.2.1. The restart principle 86
4.4.2.2. A recovery algorithm 86
4.4.2.3. Eliminating the clocks 87
4.5. An example 88
4.5.1. A very simple system 88
4.5.2. Graph of the global states 89
4.5.3. A global state calculation 89
4.5.4. Space-time view 90
4.6. References 91

PART 3: Tools, methods and algorithms 93

5: Distributing a global synchronization constraint 95

5.1. Introduction 95
5.1.1. Synchronization problems 95
5.1.2. Types of state variable 97
5.1.3. Distribution of a constraint 97
5.2. Site-specific counters 98
5.2.1. Types of global constraint 98
5.2.1.1. The producer–consumer problem reconsidered 98
5.2.1.2. A canonical form 98
5.2.2. Lower-bound values and delayed updating 99
5.2.3. Upper-bound values and anticipated updating 100
5.2.3.1. Increase of the upper-bound values 100

5.2.3.2. Reduction of the upper-bound values 100
5.2.4. An illustration 101
5.2.5. Message loss 103
5.3. Variables with arbitrary behaviour 103
5.3.1. Assumptions and types of global constraints 103
5.3.2. Reliable environment case 103
5.3.3. Principle of the general solution 104
5.3.3.1. Resilience to an unreliable environment 104
5.3.3.2. Ensuring compliance with constraints 104
5.3.4. A protocol guaranteeing invariance 105
5.3.4.1. An implementation of the principles 105
5.3.4.2. Control algorithms 108
5.3.5. Towards a solution of the general case 109
5.4. Non-site-specific variables 111
5.4.1. Principle of solution 111
5.4.2. Splitting up variables 111
5.4.3. Partitioning and isarithmic control 112
5.4.4. Duplication and exclusion 113
5.4.5. A specific case 113
5.5. References 115

6. Elements and algorithms for a toolbox 116

6.1. Introduction 116
6.1.1. The idea of the toolbox 116
6.1.2. The elements to be studied 116
6.2. Summary of tools introduced and used 117
6.2.1. Sequence numbers and acknowledgements 117
6.2.2. Counters and logical clocks 118
6.2.3. Markers and tokens 118
6.3. Temporal aspect: bounded delay 119
6.3.1. Introduction 119
6.3.2. Avoiding false deadlock detection 119
6.3.2.1. Deadlock and false deadlock 119
6.3.2.2. A consistent global state 121
6.3.3. Mutual exclusion and order of satisfaction 122
6.3.3.1. A mutual exclusion algorithm 122
6.3.3.2. Order of satisfaction of requests 124
6.3.4. A termination computation 125
6.3.4.1. The problem 125
6.3.4.2. Principle of the solution 125
6.3.4.3. The algorithm 126
6.3.4.4. Characteristics of the algorithm 127
6.3.5. Distributing a finite-state machine 128
6.3.5.1. The problem 128
6.3.5.2. Principles of the algorithm 128

6.4. Structural aspect: specific topologies 129
6.4.1. Introduction 129
6.4.2. The ring 129
6.4.2.1. General remarks 129
6.4.2.2. Mutual exclusion 130
6.4.2.3. Termination 132
6.4.2.4. The producer–consumer problem 136
6.4.3. The tree structure 139
6.4.3.1. General remarks 139
6.4.3.2. A ranking algorithm 140
6.4.3.3. A sorting algorithm 141
6.4.4. The complete graph 143
6.4.4.1. General remarks 143
6.4.4.2. Distributing a state machine 143
6.5. Knowledge of neighbours 147
6.5.1. Introduction 147
6.5.2. An election algorithm 148
6.6. References 152

Bibliography 155

Index 163