

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Distributed Programming Abstractions	3
1.2.1	Inherent Distribution	4
1.2.2	Distribution as an Artifact	6
1.3	The End-to-End Argument	7
1.4	Software Components	8
1.4.1	Composition Model	8
1.4.2	Programming Interface	11
1.4.3	Modules	13
1.5	Classes of Algorithms	16
1.6	Chapter Notes	17
2	Basic Abstractions	19
2.1	Distributed Computation	20
2.1.1	Processes and Messages	20
2.1.2	Automata and Steps	20
2.1.3	Safety and Liveness	22
2.2	Abstracting Processes	24
2.2.1	Process Failures	24
2.2.2	Crashes	24
2.2.3	Omissions	26
2.2.4	Crashes with Recoveries	26
2.2.5	Eavesdropping Faults	28
2.2.6	Arbitrary Faults	29
2.3	Cryptographic Abstractions	30
2.3.1	Hash Functions	30
2.3.2	Message-Authentication Codes (MACs)	30
2.3.3	Digital Signatures	31
2.4	Abstracting Communication	32
2.4.1	Link Failures	33
2.4.2	Fair-Loss Links	34
2.4.3	Stubborn Links	35
2.4.4	Perfect Links	37
2.4.5	Logged Perfect Links	38

2.4.6	Authenticated Perfect Links	40
2.4.7	On the Link Abstractions	43
2.5	Timing Assumptions	44
2.5.1	Asynchronous System	44
2.5.2	Synchronous System	45
2.5.3	Partial Synchrony	47
2.6	Abstracting Time	48
2.6.1	Failure Detection	48
2.6.2	Perfect Failure Detection	49
2.6.3	Leader Election	51
2.6.4	Eventually Perfect Failure Detection	53
2.6.5	Eventual Leader Election	56
2.6.6	Byzantine Leader Election	60
2.7	Distributed-System Models	63
2.7.1	Combining Abstractions	63
2.7.2	Setup	64
2.7.3	Quorums	65
2.7.4	Measuring Performance	65
2.8	Exercises	67
2.9	Solutions	68
2.10	Chapter Notes	71
3	Reliable Broadcast	73
3.1	Motivation	73
3.1.1	Client–Server Computing	73
3.1.2	Multiparticipant Systems	74
3.2	Best-Effort Broadcast	75
3.2.1	Specification	75
3.2.2	Fail-Silent Algorithm: Basic Broadcast	76
3.3	Regular Reliable Broadcast	77
3.3.1	Specification	77
3.3.2	Fail-Stop Algorithm: Lazy Reliable Broadcast	78
3.3.3	Fail-Silent Algorithm: Eager Reliable Broadcast	79
3.4	Uniform Reliable Broadcast	81
3.4.1	Specification	81
3.4.2	Fail-Stop Algorithm: All-Ack Uniform Reliable Broadcast	82
3.4.3	Fail-Silent Algorithm: Majority-Ack Uniform Reliable Broadcast	84
3.5	Stubborn Broadcast	85
3.5.1	Specification	85
3.5.2	Fail-Recovery Algorithm: Basic Stubborn Broadcast	86
3.6	Logged Best-Effort Broadcast	87
3.6.1	Overview	87
3.6.2	Specification	88
3.6.3	Fail-Recovery Algorithm: Logged Basic Broadcast	89

- 3.7 Logged Uniform Reliable Broadcast 90
 - 3.7.1 Specification 90
 - 3.7.2 **Fail-Recovery Algorithm:**
 Logged Majority-Ack Uniform Reliable Broadcast 90
- 3.8 Probabilistic Broadcast 92
 - 3.8.1 The Scalability of Reliable Broadcast 92
 - 3.8.2 Epidemic Dissemination 93
 - 3.8.3 Specification 94
 - 3.8.4 Randomized Algorithm: Eager Probabilistic Broadcast 94
 - 3.8.5 Randomized Algorithm: Lazy Probabilistic Broadcast 97
- 3.9 FIFO and Causal Broadcast 100
 - 3.9.1 Overview 101
 - 3.9.2 FIFO-Order Specification 101
 - 3.9.3 Fail-Silent Algorithm: Broadcast with Sequence Number ... 101
 - 3.9.4 Causal-Order Specification 103
 - 3.9.5 Fail-Silent Algorithm: No-Waiting Causal Broadcast 104
 - 3.9.6 Fail-Stop Algorithm: Garbage-Collection of Causal Past ... 106
 - 3.9.7 Fail-Silent Algorithm: Waiting Causal Broadcast 108
- 3.10 Byzantine Consistent Broadcast 110
 - 3.10.1 Motivation 110
 - 3.10.2 Specification 111
 - 3.10.3 **Fail-Arbitrary Algorithm:**
 Authenticated Echo Broadcast 112
 - 3.10.4 Fail-Arbitrary Algorithm: Signed Echo Broadcast 114
- 3.11 Byzantine Reliable Broadcast 116
 - 3.11.1 Specification 117
 - 3.11.2 **Fail-Arbitrary Algorithm:**
 Authenticated Double-Echo Broadcast 117
- 3.12 Byzantine Broadcast Channels 120
 - 3.12.1 Specifications 120
 - 3.12.2 Fail-Arbitrary Algorithm: Byzantine Consistent Channel ... 122
 - 3.12.3 Fail-Arbitrary Algorithm: Byzantine Reliable Channel 123
- 3.13 Exercises 124
- 3.14 Solutions 126
- 3.15 Chapter Notes 134
- 4 Shared Memory 137**
 - 4.1 Introduction 138
 - 4.1.1 Shared Storage in a Distributed System 138
 - 4.1.2 Register Overview 138
 - 4.1.3 Completeness and Precedence 141
 - 4.2 $(1, N)$ Regular Register 142
 - 4.2.1 Specification 142
 - 4.2.2 **Fail-Stop Algorithm:**
 Read-One Write-All Regular Register 144

4.2.3	Fail-Silent Algorithm:	
	Majority Voting Regular Register	146
4.3	$(1, N)$ Atomic Register	149
4.3.1	Specification	149
4.3.2	Transformation:	
	From $(1, N)$ Regular to $(1, N)$ Atomic Registers	151
4.3.3	Fail-Stop Algorithm:	
	Read-Impose Write-All $(1, N)$ Atomic Register	156
4.3.4	Fail-Silent Algorithm:	
	Read-Impose Write-Majority $(1, N)$ Atomic Register	157
4.4	(N, N) Atomic Register	159
4.4.1	Multiple Writers	159
4.4.2	Specification	160
4.4.3	Transformation:	
	From $(1, N)$ Atomic to (N, N) Atomic Registers	161
4.4.4	Fail-Stop Algorithm:	
	Read-Impose Write-Consult-All (N, N) Atomic Reg.	165
4.4.5	Fail-Silent Algorithm:	
	Read-Impose Write-Consult-Majority (N, N) Atomic Reg.	167
4.5	$(1, N)$ Logged Regular Register	170
4.5.1	Precedence in the Fail-Recovery Model	170
4.5.2	Specification	170
4.5.3	Fail-Recovery Algorithm: Logged Majority Voting	172
4.6	$(1, N)$ Byzantine Safe Register	175
4.6.1	Specification	176
4.6.2	Fail-Arbitrary Algorithm: Byzantine Masking Quorum	177
4.7	$(1, N)$ Byzantine Regular Register	179
4.7.1	Specification	179
4.7.2	Fail-Arbitrary Algorithm:	
	Authenticated-Data Byzantine Quorum	180
4.7.3	Fail-Arbitrary Algorithm:	
	Double-Write Byzantine Quorum	182
4.8	$(1, N)$ Byzantine Atomic Register	188
4.8.1	Specification	189
4.8.2	Fail-Arbitrary Algorithm:	
	Byzantine Quorum with Listeners	189
4.9	Exercises	194
4.10	Solutions	195
4.11	Chapter Notes	200
5	Consensus	203
5.1	Regular Consensus	204
5.1.1	Specification	204
5.1.2	Fail-Stop Algorithm: Flooding Consensus	205
5.1.3	Fail-Stop Algorithm: Hierarchical Consensus	208

- 5.2 Uniform Consensus 211
 - 5.2.1 Specification 211
 - 5.2.2 Fail-Stop Algorithm: Flooding Uniform Consensus 212
 - 5.2.3 Fail-Stop Algorithm: Hierarchical Uniform Consensus 213
- 5.3 Uniform Consensus in the Fail-Noisy Model 216
 - 5.3.1 Overview 216
 - 5.3.2 Epoch-Change 217
 - 5.3.3 Epoch Consensus 220
 - 5.3.4 Fail-Noisy Algorithm: Leader-Driven Consensus 225
- 5.4 Logged Consensus 228
 - 5.4.1 Specification 228
 - 5.4.2 Logged Epoch-Change 229
 - 5.4.3 Logged Epoch Consensus 230
 - 5.4.4 Fail-Recovery Algorithm:
 Logged Leader-Driven Consensus 234
- 5.5 Randomized Consensus 235
 - 5.5.1 Specification 236
 - 5.5.2 Common Coin 237
 - 5.5.3 Randomized Fail-Silent Algorithm:
 Randomized Binary Consensus 238
 - 5.5.4 Randomized Fail-Silent Algorithm:
 Randomized Consensus with Large Domain 242
- 5.6 Byzantine Consensus 244
 - 5.6.1 Specifications 244
 - 5.6.2 Byzantine Epoch-Change 246
 - 5.6.3 Byzantine Epoch Consensus 248
 - 5.6.4 Fail-Noisy-Arbitrary Algorithm:
 Byzantine Leader-Driven Consensus 259
- 5.7 Byzantine Randomized Consensus 261
 - 5.7.1 Specification 261
 - 5.7.2 Randomized Fail-Arbitrary Algorithm:
 Byzantine Randomized Binary Consensus 261
- 5.8 Exercises 266
- 5.9 Solutions 268
- 5.10 Chapter Notes 277

- 6 Consensus Variants 281**
 - 6.1 Total-Order Broadcast 281
 - 6.1.1 Overview 281
 - 6.1.2 Specifications 283
 - 6.1.3 Fail-Silent Algorithm:
 Consensus-Based Total-Order Broadcast 284
 - 6.2 Byzantine Total-Order Broadcast 287
 - 6.2.1 Overview 287
 - 6.2.2 Specification 288

- 6.2.3 **Fail-Noisy-Arbitrary Algorithm:**
 - Rotating Sender Byzantine Broadcast 288
- 6.3 Terminating Reliable Broadcast 292
 - 6.3.1 Overview 292
 - 6.3.2 Specification 293
 - 6.3.3 **Fail-Stop Algorithm: Consensus-Based**
 - Uniform Terminating Reliable Broadcast 293
- 6.4 Fast Consensus 296
 - 6.4.1 Overview 296
 - 6.4.2 Specification 297
 - 6.4.3 **Fail-Silent Algorithm:**
 - From Uniform Consensus to Uniform Fast Consensus 297
- 6.5 Fast Byzantine Consensus 300
 - 6.5.1 Overview 300
 - 6.5.2 Specification 300
 - 6.5.3 **Fail-Arbitrary Algorithm:**
 - From Byzantine Consensus to Fast Byzantine Consensus ... 300
- 6.6 Nonblocking Atomic Commit 303
 - 6.6.1 Overview 303
 - 6.6.2 Specification 304
 - 6.6.3 **Fail-Stop Algorithm:**
 - Consensus-Based Nonblocking Atomic Commit 304
- 6.7 Group Membership 307
 - 6.7.1 Overview 307
 - 6.7.2 Specification 308
 - 6.7.3 **Fail-Stop Algorithm: Consensus-Based**
 - Group Membership 309
- 6.8 View-Synchronous Communication 311
 - 6.8.1 Overview 311
 - 6.8.2 Specification 312
 - 6.8.3 **Fail-Stop Algorithm:**
 - TRB-Based View-Synchronous Communication 314
 - 6.8.4 **Fail-Stop Algorithm: Consensus-Based**
 - Uniform View-Synchronous Communication 319
- 6.9 Exercises 323
- 6.10 Solutions 324
- 6.11 Chapter Notes 337
- 7 Concluding Remarks 341**
 - 7.1 Implementation in *Appia* 341
 - 7.2 Further Implementations 342
 - 7.3 Further Reading 344

Contents	xix
References	347
List of Modules	355
List of Algorithms	357
Index	361



<http://www.springer.com/978-3-642-15259-7>

Introduction to Reliable and Secure Distributed
Programming

Cachin, C.; Guerraoui, R.; Rodrigues, L.

2011, XIX, 367 p., Hardcover

ISBN: 978-3-642-15259-7