

Contents

Abstract (English/Français)	v
Acknowledgements	xi
List of figures	xvi
List of tables	xviii
1 Introduction	1
1.1 Thesis Context	1
1.2 Thesis Motivation	4
1.3 Thesis Goal	4
1.4 Thesis Contribution	4
1.5 Outline	6
2 Preliminaries	7
2.1 Consensus Problem	7
2.1.1 Arbitrary Process faults	7
2.1.2 Transmission Faults	8
2.2 Total-order Broadcast	8
2.3 State Machine Replication	9
2.4 Partially Synchronous System Model	9
2.5 The Basic Round Model	10
3 Unifying Byzantine Consensus	
Algorithms with WIC	11
3.1 Introduction	11
3.2 Weak interactive consistency: an informal introduction	13
3.2.1 On the use of signatures	13
3.2.2 Safe updates requires neither signatures nor a coordinator	13
3.2.3 Coordinator for liveness	14
3.3 Definition of WIC	15
3.4 Implementing WIC	16
3.4.1 Simulation with signatures	17
3.4.2 Simulation without signatures	18
	xiii

3.5	Achieving Consensus with WIC	21
3.5.1	On the use of WIC	21
3.5.2	MA algorithm	21
3.5.3	CL algorithm	24
3.6	Related work	32
3.7	Conclusion	33
4	Tolerating Permanent and Transient Value Faults	35
4.1	Introduction	35
4.2	Model	38
4.2.1	Heard-Of Sets and Consistent Rounds	39
4.2.2	HO Machines	40
4.2.3	Simulation of communication predicates	40
4.2.4	Consensus	41
4.3	Communication predicates	41
4.3.1	Predicates that capture static and dynamic value faults	42
4.3.2	Predicates that restrict asynchrony of communication and dynamism of faults	42
4.3.3	Permanent versus Transient Faults	43
4.3.4	Weak Interactive Consistency	44
4.4	Simulating weak interactive consistency $\mathcal{P}_{\diamond cons}$ from eventually safe kernels $\mathcal{P}_{\diamond SK}$	44
4.4.1	Generic predicate	51
4.5	Solving consensus with eventual consistency	52
4.5.1	The <i>BLV</i> algorithm	52
4.6	Deriving the overall resilience of <i>BLV</i>	59
4.7	Communication predicates and corresponding systems	61
4.8	Conclusion	63
5	Generic Consensus Algorithm for Benign and Byzantine Faults	65
5.1	Introduction	65
5.2	Model	67
5.3	Deriving a generic consensus algorithm	68
5.3.1	Very simple consensus algorithm	68
5.3.2	Generic Algorithm: Draft 1	69
5.3.3	Generic Algorithm: Draft 2	71
5.3.4	Generic Algorithm: final version	73
5.3.5	Correctness of the Generic Algorithm	75
5.3.6	Optimizations	77
5.4	Instantiations of Parameters and Classification of Algorithms	78
5.4.1	Instantiations of $FLV(\vec{\mu}_p^r)$	78
5.4.2	Instantiations of $Validator(p, \phi)$	91
5.5	Instantiations of Algorithm 5.4	92
5.5.1	Class 1 algorithms	92

5.5.2	Class 2 algorithms	93
5.5.3	Class 3 algorithms	96
5.5.4	Where is the leader in the generic algorithm ?	97
5.6	Conclusion	98
6	On the Reduction of Total-Order Broadcast to Consensus	99
6.1	Introduction	99
6.2	Definitions	101
6.2.1	Reliable Unique Broadcast	101
6.2.2	Consensus	101
6.3	Total-order broadcast reduction and the validity property of consensus	102
6.3.1	Total-order broadcast is harder than weak unanimity consensus	103
6.3.2	Strong validity consensus is harder than total-order broadcast	104
6.3.3	Consensus problems equivalent to total-order broadcast	105
6.4	Reducing total-order broadcast to consensus with range validity	106
6.4.1	Reduction algorithm	106
6.4.2	Proof of Algorithm 6.1	108
6.4.3	Why <i>reliable unique broadcast</i> ?	110
6.4.4	Time complexity of Algorithm 6.1	111
6.5	Solving range validity consensus	112
6.5.1	Reduction of range validity consensus to binary consensus	112
6.5.2	Solving range validity consensus in the partially synchronous system model	114
6.6	Reducing total-order broadcast to binary consensus	114
6.7	Related Work	115
6.8	Conclusion	117
7	Bounded Delay in Byzantine-Tolerant State Machine Replication	119
7.1	Introduction	119
7.2	Related work	123
7.3	Definitions	125
7.3.1	Model	125
7.4	Abortable Timely Announced Broadcast	125
7.4.1	Solving ATAB	126
7.5	Solving Total-Order Broadcast with ATAB	128
7.5.1	Basic idea	128
7.5.2	Process p skipping its own instances	129
7.5.3	Process p skipping instances of other processes	129
7.5.4	Correctness proof	129
7.6	BFT-Mencius	131
7.6.1	Generalities	132
7.6.2	Dealing with faulty servers	132
7.6.3	Dealing with slow servers	132
7.6.4	Ensuring Bounded Delay	134

Contents

7.6.5	Improving the Delay Bound	134
7.6.6	Blacklisting cannot impair liveness	135
7.7	Evaluation	135
7.7.1	Experimental setup and methodology	137
7.7.2	Failure-free executions	138
7.7.3	Executions with performance attacks by faulty servers	139
7.8	Conclusion	141
8	Concluding Remarks	143
A	CL-ATAB	147
A.1	Normal case	147
A.2	Changing view	148
A.3	Proof of correctness	149
	Bibliography	165