Network Topology and Fault-Tolerant Consensus

Synthesis Lectures on Distributed Computing Theory

Synthesis Lectures on Distributed Computing Theory was founded by Nancy Lynch of the Massachusetts Institute of Technology, and is now edited by Michel Raynal of the University of Rennes, France and Hong Kong Polytechnic University. The series publishes 50- to 150-page publications on topics pertaining to distributed computing theory. The scope largely follows the purview of premier information and computer science conferences, such as ACM PODC, DISC, SPAA, OPODIS, CONCUR, DialM-POMC, ICDCS, SODA, Sirocco, SSS, and related conferences. Potential topics include, but not are limited to: distributed algorithms and lower bounds, algorithm design methods, formal modeling and verification of distributed algorithms, and concurrent data structures.

Network Topology and Fault-Tolerant Consensus
Dimitris Sakavalas and Lewis Tseng
2019

Introduction to Distributed Self-Stabilizing Algorithms
Karine Altisen, Stéphane Devismes, Swan Dubois, and Franck Petit
2019

Distributed Computing Perls
Gadi Taubenfeld
2018

Decidability of Parameterized Verification
Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder
2015

Impossibility Results for Distributed Computing
Hagit Attiya and Faith Ellen
2014

Distributed Graph Coloring: Fundamentals and Recent Developments
Leonid Barenboim and Michael Elkin
2013

Distributed Computing by Oblivious Mobile Robots
Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro
2012

Quorum Systems: With Applications to Storage and Consensus
Marko Vukolić
2012

Link Reversal Algorithms
Jennifer L. Welch and Jennifer E. Walter
2011

Cooperative Task-Oriented Computing: Algorithms and Complexity
Chryssis Georgiou and Alexander A. Shvartsman
2011

New Models for Population Protocols
Othon Michail, Ioannis Chatzigiannakis, and Paul G. Spirakis
2011

The Theory of Timed I/O Automata, Second Edition
Dilsun K. Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager
2010

Principles of Transactional Memory
Rachid Guerraoui and Michał Kapałka
2010

Fault-tolerant Agreement in Synchronous Message-passing Systems
Michel Raynal
2010

Communication and Agreement Abstractions for Fault-Tolerant Asynchronous Distributed Systems
Michel Raynal
2010

The Mobile Agent Rendezvous Problem in the Ring
Evangelos Kranakis, Danny Krizanc, and Euripides Markou
2010

Network Topology and Fault-Tolerant

Consensus Dimitris Sakavalas and Lewis Tseng

ISBN: 978-3-031-00886-3 paperback
ISBN: 978-3-031-02014-8 eBook
ISBN: 978-3-031-00132-1 hardcover

DOI 10.1007/978-3-031-02014-8

A Publication in the Springer series
SYNTHESIS LECTURES ON DISTRIBUTED COMPUTING THEORY

Lecture #16
Series Editor: Michel Raynal, *University of Rennes, France and Hong Kong Polytechnic University*
Founding Editor: Nancy Lynch, *Massachusetts Institute of Technology*
Series ISSN
Print 2155-1626 Electronic 2155-1634

Network Topology and Fault-Tolerant Consensus

Dimitris Sakavalas
Boston College

Lewis Tseng
Boston College

SYNTHESIS LECTURES ON DISTRIBUTED COMPUTING THEORY #16

ABSTRACT

As the structure of contemporary communication networks grows more complex, practical networked distributed systems become prone to component failures. Fault-tolerant consensus in message-passing systems allows participants in the system to agree on a common value despite the malfunction or misbehavior of some components. It is a task of fundamental importance for distributed computing, due to its numerous applications.

We summarize studies on the topological conditions that determine the feasibility of consensus, mainly focusing on directed networks and the case of restricted topology knowledge at each participant. Recently, significant efforts have been devoted to fully characterize the underlying communication networks in which variations of fault-tolerant consensus can be achieved. Although the deduction of analogous topological conditions for undirected networks of known topology had shortly followed the introduction of the problem, their extension to the directed network case has been proven a highly non-trivial task. Moreover, global knowledge restrictions, inherent in modern large-scale networks, require more elaborate arguments concerning the locality of distributed computations. In this work, we present the techniques and ideas used to resolve these issues.

Recent studies indicate a number of parameters that affect the topological conditions under which consensus can be achieved, namely, the fault model, the degree of system synchrony (synchronous vs. asynchronous), the type of agreement (exact vs. approximate), the level of topology knowledge, and the algorithm class used (general vs. iterative). We outline the feasibility and impossibility results for various combinations of the above parameters, extensively illustrating the relation between network topology and consensus.

KEYWORDS

consensus, network topology, message-passing systems, asynchronous systems, synchronous systems, topological conditions, broadcast, reliable message transmission, local adversary, general adversary

Contents

List of Figures

List of Tables

List of Algorithms

Preface

The rapid growth of networked systems (e.g., the Internet, sensor networks, social networks, financial networks, etc.) naturally presents increased vulnerability concerns regarding their components. Considerations arising from this phenomenon are formally addressed in the study of *fault-tolerant distributed computing*. In this book, we focus mainly on the *fault-tolerant consensus* problem, which allows interacting participants of a networked system to reach an agreement despite the presence of misbehavior. Consensus is a primitive of fundamental importance for distributed computing and cryptographic protocols due to its wide range of applications. For instance, it serves as a building-block for redundant flight control systems, for the assertion of consistency among replicated databases, or for electronic voting and cryptocurrencies.

The *fault-tolerant consensus* problem has received significant attention since the seminal works of Lamport, Shostak, and Pease [50, 78] in 1980. In their setup, each participant is given an input value initially, and after a finite amount of time, each fault-free participant should produce an output value. In this book, we summarize recent results regarding both *exact* and *approximate* consensus in the context of deterministic algorithms and message-passing networks, where the participants communicate via messages. For exact consensus, all fault-free participants eventually agree on and output a common value, which depends on the initial values of all participants. For approximate consensus, the outputs of the participants must eventually be arbitrarily close to each other. Approximate consensus is of particular interest and has been the focus of many recent studies. Apart from its wide applications in the control systems area, its significance for the distributed computing community mainly comes from the fact that it can be used to overcome the impossibility or the high complexity of achieving consensus in certain models.

In general, the feasibility and efficiency of realizing distributed tasks depends on a number of parameters considered in this book and outlined in the following.

Adversary One can easily observe that if the adversary can corrupt any subset of participants and make them misbehave in any way, then the achievement of most meaningful distributed tasks becomes impossible or vague. Therefore, fault-tolerance studies assume different restrictions on the adversary which are determined by the adversary model. Specifically, the model defines the power of the adversary in terms of the misbehavior type and the family of sets that can be corrupted during an execution of the given algorithm.

In this book, we focus on the results addressing *crash* faults (fail-stop faults), where some participants may prematurely stop executing the protocol, and *Byzantine* faults, where some participants may have arbitrary misbehavior, by blocking, rerouting, or even altering a message that they should normally relay intact to specific nodes. Regarding the corruptible sets of participants, we mainly focus on the classic threshold model in which there is a fixed bound on the number of participants that may be corrupted in the system, while we also present feasibility results for the cases of a local and a general adversary which are more recently studied. The former model has applications in systems where participants can only use local information, while the latter encompasses all known adversary models by modeling a situation where arbitrary coalitions of faulty participants are possible.

Timing and System Synchrony Another parameter that significantly affects the feasibility of distributed tasks is the amount of synchrony between interacting participants. The synchrony notion includes two parameters; namely, the message delivery delay and the relative speeds with which the participants take consecutive computational steps. In order to illustrate these notions, it is common to concentrate on two extreme models: the synchronous model and the asynchronous one.

- In the synchronous case, it is assumed that there are bounds on both message delivery delays and relative speeds of the participants. Without loss of generality, one can assume that all message deliveries are instant and all participants perform computations with the exact same speed.

- In the asynchronous model, no fixed bounds on the delays and relative computation speeds are assumed. Specifically, the delivery delay of messages is finite, but no known time bound is assumed on it, and the computation speeds may arbitrarily differ.

Network *Network topology* which defines the communication capability between all pairs of participants naturally affects the degree to which certain distributed tasks can be achieved. Moreover, the initial knowledge of the topology possessed by the participants has also proven crucial in the determination of the feasibility condition. Regarding consensus feasibility, tight conditions on the network topology have shortly followed the introduction of the problem for the case of undirected networks where participants have full topology knowledge; these well-known results are outlined in the first two chapters of the book.

We give a more detailed presentation of sufficient and necessary topological conditions in the case of *directed networks* and the case of *restricted topology knowledge*. The study of directed networks is largely motivated by wireless networks, in which the different transmit radii may result in one-way communication between two participants. More importantly, it has been shown that the arguments underlying the topological conditions in undirected networks *cannot* be trivially extended to the directed case; thus, the latter case presents an extra level of difficulty which has been addressed in a series of works summarized in this book. The motivation behind the restricted topology knowledge of participants stems from large-scale networks, in which the estimation of global topological properties may be computationally prohibitive or even impossible. Moreover, the increasing use of sensor networks in mission-critical applications further motivates restricted memory models, since these networks constitute of devices with small memory and low computing power.

The main focus of this book is to summarize the topological conditions which determine the feasibility of fault-tolerant consensus. Alternatively, we present studies on the determination of the class of networks that allow participants to reach consensus, and present protocols that solve the problem in these classes of networks. The results exactly determine the structure of networks that can optimally support the usage of fault-tolerant protocols and thus can be applied in the design of such networks. Another practical benefit of these studies is that using the outlined techniques, one can exactly determine the worst fault situations that can be tolerated in existing network infrastructures.

Dimitris Sakavalas and Lewis Tseng
April 2019

Acknowledgments

The completion of this book has been supported by Boston College. We would like to thank all authors who have contributed to the results presented in this book. We are indebted to our friends, colleagues, and research collaborators with whom we have had numerous joyful and meaningful discussions on the topics addressed in this book. These include Vartika Bhandari, Chris Litsas, Aris Pagourtzis, Giorgos Panagiotakos, Lili Su, and Nitin H. Vaidya.

We are grateful to the editor, Michel Raynal, and the referees, Hsin-Hao Su and Lili Su, for their detailed comments and suggestions which significantly improved the presentation and clarity of the book. Finally, we would like to thank Michel Raynal for his kind invitation to write a book for the Synthesis Lectures on Distributed Computing Theory series he is editing for Morgan & Claypool Publishers.

Dimitris Sakavalas and Lewis Tseng
April 2019