Technical University of Vienna

INSTITUTE OF COMPUTER ENGINEERING

Embedded Computing Systems Group

Master thesis Proposal

IoT Implementation and Evaluation of Distributed Consensus Algorithms

Student:
Christoph Echtinger-Sieghart
Matriculation Number:
00304130

Advisor:
Prof. Dr. Ulrich Schmid

November 16, 2020

Abstract

The problem of consensus is central to the field of distributed computing. Starting from a group of algorithms, described in [WSS19] and [SS20], we will implement and prove correct new algorithms solving variants of consensus in directed dynamic networks. Directed dynamic networks are synchronous distributed systems made up of an unknown number of fault-free processes under unreliable communication modeled in the form of a message adversary.

As a side product of the implementation, a testing framework to simulate communication under a message adversary will be created. After the implementation is done, we envisage to perform experimental runs of the algorithms to gain insights with regard to performance and real-world behaviour.

1 Problem Description

The problem of consensus is a central one in the field of distributed computing. Consensus is important in all scenarios, where independent processors need to agree on a common value [AW04]. Achieving distributed agreement becomes hard under unreliable communication conditions. Unreliable communication conditions and many independent processors trying to cooperate are manifold in the Internet of Things (IoT) - these systems need practical implementations of consensus algorithms able to cope with limited resources and unreliable networks.

Unreliable communication conditions can be modeled as a message adversary [AG13], which must adhere to some specification, but within this specification may try to foil reaching consensus.

The algorithm from [WSS19] describes a solution to the consensus problem in synchronous systems operating under a message adversary called $\Diamond\text{STABLE}_{\leq N,D}(D+1)$. Loosely speaking - this message adversary guarantees that, eventually, a vertex-stable root component will occur in the communication graph and that it will last sufficiently long ($D+1$ rounds). The parameter D being the dynamic network depth. Initial experiments conducted in [PS16] show a similar adversarial model to be a good match for real world conditions in synchronous wireless sensor networks.

[SS20] improved on the algorithm presented in [WSS19] in terms of robustness, by removing the need for a round counter. In the same paper, the authors introduce a novel algorithm for solving stabilizing consensus based on insights gained during discovery of the robustness improvements.

Besides the fact that only a reduction-based sketch of a correctness proof has been given for the stabilizing consensus algorithm in [SS20], the MinMax stabilizing consensus algorithm from [CBM19] reveals that this problem can be solved for stronger message adversaries than $\Diamond\text{STABLE}_{\leq N,D}(D+1)$. Exploring the space of message adversaries delimited by these two instances is hence a very interesting theoretical question.

In addition, no real-world implementation of this specific group of algorithms exists and our aim is to remedy this situation. By implementing the algorithms, we expect to gain a deeper understanding of the engineering obstacles one needs to overcome to make use of these recent discoveries in real-world applications.

Our implementation will be somewhat agnostic to the underlying time synchronisation mechanism - opening the way for experimentation with various time synchronisation methods. To the best of our knowledge, there is little to no tooling available to aid the developer in implementing such algorithms - especially no testing or simulation framework to describe communication under the assumption of a message adversary. During the process of implementing these algorithms such a framework is important to perform testing and simulation runs.

2 Expected Results

We will create a practically usable implementation of the group of algorithms described in [WSS19] and [SS20] targeted at a - yet to be decided - IoT platform capable of wireless communication. As a side product of the implementation a testing/simulation framework for communication under message adversaries will be created.

We will further give a non-reduction based correctness proof of the novel stabilizing consensus algorithm introduced in [SS20]. In addition, we plan to investigate why the stabilizing consensus algorithm from [SS20] requires unique node IDs, but the stabilizing consensus algorithm MinMax from [CBM19] does not.

After finishing the implementation, we plan to run experiments, testing the algorithms under real-world conditions. We expect insights gained during the implementation of the algorithms and during the experimental phase with regard to their behaviour in the field and with regard to real-world performance.

3 Methods

3.1 Implementation

We plan to implement the group of algorithms described in [WSS19] and [SS20]. This will lead to three variants of consensus algorithms. The first variant being the original algorithm as described in [WSS19] using round counters. The second variant being the adaption described in [SS20], that makes the round counter superfluous and the third variant is the new stabilizing consensus algorithm also described in [SS20].

An important aspect of the implementation will be the question of how to determine the message adversary's parameters (most importantly D - the dynamic network depth) that make it cover a more or less real system - the experimental results described in [PS16] can act as a guideline.

We additionally need a means to perform time synchronization of the various nodes involved. Here we have to consider, if - depending on the actual hardware - a time synchronization protocol like FTSP [MKSL04] or IEEE1588 [iee14] can be used, or, if the synchronization can be done on a lower level utilizing functionality available in the PHY/MAC layers of the actual hardware. Evaluating the correct functioning of the time synchronization protocol and the achieved precision will be another challenge.

To facilitate the development of a correct implementation, we need a way to perform tests and simulations. Various options are available for performing the simulation. It remains to be decided whether we will extend one of the numerous available simulator platforms (OMNeT++, ..., see [CBBZ17]), or - if no good match can be found - write a simulator for the task at hand. If TinyOS is chosen as the implementation platform, it remains to be seen whether the TinyOS simulator

TOSSIM can be adapted to simulate packet loss matching some message adversary's specification.

3.2 Experiments

After finishing the implementation we plan to run experiments under real-world conditions to gather performance data on these algorithms and gain insights into their real-world behaviour.

3.3 Theoretical Work

Based on existing correctness proofs for (stabilizing) consensus algorithms like [WSS19] and [CBM19], we plan to derive a non reduction-based correctness proof for the stabilizing consensus algorithm of [SS20].

4 State of the Art

Theoretical work on various forms of consensus in directed dynamic networks was done, amongst others, by [BRS12], [SWS16], [BRS$^+$18] and [CBM19]. [BRS12] introduce the notion of vertex-stable stable root components in the communication graph and provide and prove correct an algorithm that solves consensus under these assumptions. [SWS16] improve on the previous results regarding consensus algorithms for message adversaries that guarantee a vertex-stable root component. [BRS$^+$18] investigate consensus under a message adversary that allows several vertex-stable root components and in addition take a look at k-set consensus and its impossibility/solvability border. [CBM19] introduce the MinMax family of algorithms for solving the problem of stabilizing consensus (introduced in [AFJ06]). [WS19] gives an overview of recent theoretical advances in the field of consensus in directed dynamic networks.

Implementation work in the area of wireless sensor networks was done on the problem of dynamic average consensus. The dynamic average consensus problem is concerned with groups of nodes trying to track the average of some locally available signals over time ([KSC$^+$19]). [KKRF11] implemented an average consensus algorithm under TinyOS on IRIS motes from a company called Memsic. [Man13] describes the implementation of a multi-hop dynamic consensus algorithm. The implementation relied on ZigBee as the wireless transport technology. Routing of messages was implemented using the AODV (Ad-hoc On-demand Distance Vector) protocol.

[ANRE15] describes the implementation of a binary consensus algorithm - again using IRIS sensor motes running TinyOS. The authors performed simulations under TOSSIM and real world experiments with up to 139 nodes.

Regarding the simulation of consensus protocols [DHW09] provide a short note on implementing a simple consensus algorithm inside the simulator framework OMNeT++. A survey conducted by Kiess et al. [KM07] gives a comprehensive overview of the possible mismatches between simulation and real-world experiments with regard to mobile ad-hoc networks.

5 Topic Match

Operating at the intersection of theory and low level details is one of the key qualities of the "Computer Engineering" course of studies. The implementation of the algorithms requires both a firm grasp on the theoretical machinery involved as well as a deep understanding of available choices and platforms to facilitate a real-world implementation.

References

[AFJ06] Dana Angluin, Michael J Fischer, and Hong Jiang. Stabilizing consensus in mobile networks. In *International Conference on Distributed Computing in Sensor Systems*, pages 37–50. Springer, 2006.

[AG13] Yehuda Afek and Eli Gafni. Asynchrony from synchrony. In *International Conference on Distributed Computing and Networking*, pages 225–239. Springer, 2013.

[ANRE15] Noor Al-Nakhala, Ryan Riley, and Tarek Elfouly. Distributed algorithms in wireless sensor networks: An approach for applying binary consensus in a real testbed. *Computer Networks*, 79:30–38, 2015.

[AW04] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics (2nd edition)*. John Wiley Interscience, March 2004.

[BRS12] Martin Biely, Peter Robinson, and Ulrich Schmid. Agreement in directed dynamic networks. In *International Colloquium on Structural Information and Communication Complexity*, pages 73–84. Springer, 2012.

[BRS+18] Martin Biely, Peter Robinson, Ulrich Schmid, Manfred Schwarz, and Kyrill Winkler. Gracefully degrading consensus and k-set agreement in directed dynamic networks. *Theoretical Computer Science*, 726:41–77, 2018.

[CBBZ17] Maxim Chernyshev, Zubair Baig, Oladayo Bello, and Sherali Zeadally. Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647, 2017.

[CBM19] Bernadette Charron-Bost and Shlomo Moran. Minmax algorithms for stabilizing consensus. *CoRR*, abs/1906.09073, 2019.

[DHW09] Andreas Dielacher, Thomas Handl, and Christian Widtmann. Implementation of a distributed consensus algorithm with omnet++. *Simul. Notes Eur.*, 19(1):41–42, 2009.

[iee14] IEEE 1588 Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. `https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588`, 2014. [Online; accessed 10-Sept-2020].

[KKRF11] Jozef Kenyeres, Martin Kenyeres, Markus Rupp, and Peter Farkas. Wsn implementation of the average consensus algorithm. In *17th European Wireless 2011-Sustainable Wireless Technologies*, pages 1–8. VDE, 2011.

[KM07] Wolfgang Kiess and Martin Mauve. A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks*, 5(3):324–339, 2007.

[KSC+19] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[Man13] Sabato Manfredi. Design of a multi-hop dynamic consensus algorithm over wireless sensor networks. *Control Engineering Practice*, 21(4):381–394, 2013.

[MKSL04] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 39–49, 2004.

[PS16] Daniel Pfleger and Ulrich Schmid. A framework for connectivity monitoring in wireless sensor networks. In *Proceedings 10th International Conference on Sensor Technlogies and Applications (SENSORCOMM 2016)*, pages 40–48, 2016.

[SS20] Ulrich Schmid and Manfred Schwarz. Round-oblivious consensus in dynamic networks under stabilizing message adversaries. 2020.

[SWS16] Manfred Schwarz, Kyrill Winkler, and Ulrich Schmid. Fast consensus under eventually stabilizing message adversaries. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pages 1–10, 2016.

[WS19] Kyrill Winkler and Ulrich Schmid. An overview of recent results for consensus in directed dynamic networks. *Bulletin of EATCS*, 2(128), 2019.

[WSS19] Kyrill Winkler, Manfred Schwarz, and Ulrich Schmid. Consensus in rooted dynamic networks with short-lived stability. *Distributed Computing*, 32(5):443–458, 2019.