Brief Announcement: Multi-Threshold Asynchronous Reliable Broadcast and Consensus

Martin Hirt
Department of Computer Science, ETH Zurich, Switzerland
hirt@inf.ethz.ch

Ard Kastrati
Dept. of Information Technology and Electrical Engineering, ETH Zurich, Switzerland
akastrati@ethz.ch

Chen-Da Liu-Zhang
Department of Computer Science, ETH Zurich, Switzerland
lichen@inf.ethz.ch

Abstract

Classical protocols for reliable broadcast and consensus provide security guarantees as long as the number of corrupted parties f is bounded by a single given threshold t. If $f > t$, these protocols are completely deemed insecure. We consider the relaxed notion of *multi-threshold* reliable broadcast and consensus where validity, consistency and termination are guaranteed as long as $f \leq t_v$, $f \leq t_c$ and $f \leq t_t$ respectively. For consensus, we consider both variants of $(1 - \epsilon)$-consensus and *almost-surely terminating* consensus, where termination is guaranteed with probability $(1 - \epsilon)$ and 1, respectively. We give a very complete characterization for these primitives in the asynchronous setting and with no signatures:

- Multi-threshold reliable broadcast is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$.
- Multi-threshold almost-surely consensus is possible if $\max\{t_c, t_v\} + 2t_t < n$, $2t_v + t_t < n$ and $t_t < n/3$. Assuming a global coin, it is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$ and $2t_v + t_t < n$.
- Multi-threshold $(1-\epsilon)$-consensus is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$ and $2t_v + t_t < n$.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Design and analysis of algorithms; Security and privacy → Cryptography

Keywords and phrases broadcast, byzantine agreement, multi-threshold

Digital Object Identifier 10.4230/LIPIcs.DISC.2020.48

Category Brief Announcement

Related Version A full version of the paper is available at `https://eprint.iacr.org/2020/958`.

1 Extended Abstract

Consensus and reliable broadcast are fundamental building blocks in fault-tolerant distributed computing. Consensus allows a set of parties, each holding an input, to agree on a common value v', where, if all honest parties hold the same input v, $v' = v$. Reliable broadcast allows a designated party, called the sender, to consistently distribute a value v among a set of recipients such that all honest recipients output v in case the sender is honest. If the sender is dishonest, either all honest recipients output the same value or none of them terminates. Both primitives are used typically in the design of more complex applications, including multi-party computation, verifiable secret-sharing or voting, just to name a few.

The first consensus protocol was introduced in the seminal work of Lamport et al. [5] for the model where parties have access to a complete network of point-to-point authenticated

© Martin Hirt and Ard Kastrati and Chen-Da Liu-Zhang;
licensed under Creative Commons License CC-BY

channels, and where at most $t < n/3$ parties are corrupted. Reliable broadcast was first introduced by Bracha [2] as a useful primitive to construct building blocks in asynchronous environments. Since then, both primitives has been extensively studied in many different settings [3, 4, 2, 1, 6].

Most known fault-tolerant distributed protocols provide security guarantees in an *all-or-nothing* fashion: if up to t parties are corrupted, all security guarantees remain. However, if more than t parties are corrupted, the protocols do not provide any security guarantees. Multi-threshold protocols (also known as hybrid security) provide different security guarantees depending on the amount of corruption, thereby allowing a graceful degradation of security.

In this work, we consider consensus and reliable broadcast protocols with separate thresholds t_v, t_c and t_t for *validity*, *consistency* and *termination*, respectively. For consensus, we consider both variants of $(1-\epsilon)$-consensus and *almost-surely terminating* consensus, where termination is guaranteed with probability $(1-\epsilon)$ and 1, respectively. Our protocols work without the use of signatures and in the purely asynchronous model without the need to make any timing assumptions. Our contributions give a very complete picture of feasibility and impossibility results: 1) Multi-threshold reliable broadcast is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$; 2) Multi-threshold almost-surely consensus is possible if $\max\{t_c, t_v\} + 2t_t < n$, $2t_v + t_t < n$ and $t_t < n/3$. Assuming a global coin, we further show that the condition $t_t < n/3$ can be dropped; 3) Multi-threshold $(1-\epsilon)$-consensus is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$ and $2t_v + t_t < n$.

Multi-Threshold Reliable Broadcast. Reliable broadcast is a fundamental primitive in distributed computing which allows a sender to consistently distribute a message towards a set of recipients. We consider a setting with $n+1$ parties, one sender S and n recipients $\mathcal{R} = \{R_1, ..., R_n\}$. Let us denote the number of corrupted recipients (not including the sender) at the end of the protocol execution by f.

▶ **Definition 1 (Reliable Broadcast).** *Let $\mathcal{M}$ be a finite message space and f be the number of corrupted recipients at the end of the execution. A protocol π where initially the sender S has an input $m \in \mathcal{M}$ and every recipient R_i upon termination outputs $m_i \in \mathcal{M}$, is a reliable broadcast protocol, with respect to thresholds t_c, t_v, and t_t, if it satisfies the following:*

- **Consistency.** *If $f \leq t_c$, then every honest recipient that terminates outputs the same message. That is, $\exists m' \in \mathcal{M} : \forall$ honest R_i that terminate $m_i = m'$.*
- **Validity.** *If $f \leq t_v$ and the sender is honest, then every honest recipient R_i that terminates outputs the sender's message. That is, $\forall$ honest R_i that terminate $m_i = m$.*
- **Termination.**
  1. *An honest sender always terminates.*
  2. *If $f \leq t_t$ and an honest recipient terminates, then every honest recipient eventually terminates.*
  3. *If $f \leq t_t$ and the sender is honest, then eventually every honest recipient terminates.*

▶ **Theorem 2.** *Multi-threshold reliable broadcast protocol is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$.*

Multi-Threshold Consensus. Stated in simple terms, consensus allows a set of parties to agree on a common value. More formally, the protocol starts with every party having an input and ends with every party having a consistent output. Moreover, if every honest party starts with the same input, they keep it. Due to the FLP impossibility proof, non-terminating executions are inevitable for every consensus protocol. Hence, we require the parties to terminate only with probability 1, termed in the literature as almost-surely terminating consensus.

▶ **Definition 3** (**Almost-Surely Terminating Consensus**). *Let $\mathcal{M}$ be a finite message space and f be the number of corrupted parties at the end of the execution. A protocol π where initially each party has an input $x_i \in \mathcal{M}$ and finally every party P_i upon termination has an output $y_i \in \mathcal{M}$, is a consensus protocol, with respect to thresholds t_c, t_v, t_t, if it satisfies the following:*

- **Consistency**. *If $f \leq t_c$, then the output of every honest party is the same value. That is, $\exists y \in \mathcal{M} : \forall$ honest P_i that output $y_i = y$.*
- **Validity**. *If $f \leq t_v$ and every honest party has the same input value $x \in \mathcal{M}$, then the output of every honest party P_i is x. That is, $\forall$ honest P_i that output $y_i = x$.*
- **Termination**. *If $f \leq t_t$, then with probability 1 eventually every honest party outputs and terminates.*

▶ **Theorem 4.** *Multi-threshold almost-surely consensus is possible if $\max\{t_c, t_v\} + 2t_t < n$, $2t_v + t_t < n$ and $t_t < n/3$.*

In the full paper, we show that the bounds $\max\{t_c, t_v\} + 2t_t < n$, $2t_v + t_t < n$ are required. We leave the feasibility of almost-surely multi-threshold consensus with $t_t \geq n/3$ as an open question. However, we provide a construction that overcomes the $n/3$ bound for the case where parties have access to a global coin.

In contrast to almost-surely terminating consensus, we show that it is possible to overcome the $n/3$ bound also if we further relax the termination guarantee as follows.

▶ **Definition 5** (**$(1-\epsilon)$-Consensus**). *The consistency and validity property of $(1-\epsilon)$-consensus are the same as in Definition 3. We only change the termination property.*

- **Termination**. *If $f \leq t_t$, then with probability $1-\epsilon$ eventually every honest party outputs and terminates.*

▶ **Theorem 6.** *Multi-threshold $(1-\epsilon)$-consensus is possible if and only if $\max\{t_c, t_v\} + 2t_t < n$ and $2t_v + t_t < n$.*

References

1 Michael Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols (extended abstract). In *Proceedings of the Second Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 27–30. ACM, 1983. doi:10.1145/800221.806707.

2 Gabriel Bracha. Asynchronous Byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987. URL: http://dx.doi.org/10.1016/0890-5401(87)90054-X, doi:10.1016/0890-5401(87)90054-X.

3 Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

4 Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM Journal on Computing*, 26(4):873–933, 1997.

5 Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

6 Achour Mostéfaoui, Moumen Hamouma, and Michel Raynal. Signature-free asynchronous Byzantine consensus with $t < n/3$ and $O(n^2)$ messages. In *ACM Symposium on Principles of Distributed Computing*, pages 2–9. ACM, 2014. doi:10.1145/2611462.2611468.