Brief Announcement: Byzantine Agreement, Broadcast and State Machine Replication with Optimal Good-Case Latency

Ittai Abraham
VMware Research, Herzliya, Israel
iabraham@vmware.com

Kartik Nayak
Duke University, Durham, NC, USA
kartik@cs.duke.edu

Ling Ren
University of Illinois at Urbana-Champaign, Champaign, IL, USA
renling@illinois.edu

Zhuolun Xiang
University of Illinois at Urbana-Champaign, Champaign, IL, USA
xiangzl@illinois.edu

Abstract

This paper investigates the problem *good-case latency* of Byzantine agreement, broadcast and state machine replication in the synchronous authenticated setting. The good-case latency measure captures the time it takes to reach agreement when all non-faulty parties have the same input (or in BB/SMR when the sender/leader is non-faulty) and all messages arrive instantaneously. Previous result implies a lower bound showing that any Byzantine agreement or broadcast protocol tolerating more than $n/3$ faults must have a good-case latency of at least Δ. Our first result is a matching tight upper bound for a family of protocols we call 1Δ. We propose a protocol 1Δ-BA that solves Byzantine agreement in the synchronous and authenticated setting with optimal good-case latency of Δ and optimal resilience $f < n/2$. We then extend our protocol and present 1Δ-BB and 1Δ-SMR for Byzantine fault tolerant broadcast and state machine replication, respectively, in the same setting and with the same optimal good-case latency of Δ and $f < n/2$ fault tolerance.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Security and privacy → Distributed systems security

Keywords and phrases Byzantine broadcast, synchrony, latency, state machine replication

Digital Object Identifier 10.4230/LIPIcs.DISC.2020.47

Related Version A full version of the paper is available at `https://arxiv.org/abs/2003.13155`.

Funding This work is supported in part by a VMware early career faculty grant.

1 Introduction

Byzantine agreement (BA) and Byzantine broadcast (BB) are fundamental problems in distributed computing, and one of the most important practical applications of BA and BB is to implement Byzantine fault tolerant (BFT) state machine replication (SMR).

In this paper, we argue that a new model for BA and BB is needed due to the following practical considerations. First, lock-step execution where replicas start and end each round at the same time, should not be assumed. Most theoretical works assume lock-step execution, as a result, the latency of BA/BB protocols is typically measured by the round complexity. Such a lock-step assumption simplifies the protocol design but it is considered impractical because it not only is hard to enforce but also leads to poor performance. Secondly, BFT

SMR protocols typically care about the *good-case*, in which a stable honest leader stays in charge and drives consensus on many decisions. However, classical BA/BB protocols tend to optimize their *worst-case* latency, which is $f+1$ rounds for tolerating f faults [2]. Since f is typically assumed to be linear in n, any BA/BB protocol will inevitably have a poor worst-case latency as n increases. Furthermore, for a more accurate characterization of latency, we adopt the separation between the conservative bound Δ and the actual (unknown) bound δ [3]. Finally, instead of measuring only the traditional *worst-case* latency to terminate, practical SMR protocols are intended to run forever; replicas *commit* or *decide* on an ever-growing sequence of values. Motivated by the above considerations, we propose the *good-case latency* to commit as the main metric, which is defined as follows.

▶ **Definition 1** (good-case latency for Byzantine fault tolerant state machine replication). *The good-case latency is the maximal latency (over all adversarial strategies) until all honest replicas commit given:*

1. *(good leader) An honest leader is in charge.*
2. *(good network) All messages arrive instantaneously, i.e., $\delta = 0$.*
3. *(good compute) All local computation is instantaneous.*

Of course, in practice, communication and computation are never instantaneous, but the good case definition captures the intuition that they are much smaller than the pessimistic upper bound Δ. From the good-case latency definition of BFT SMR above, it is natural to define the *good-case latency for BB* by replacing the good leader property with *good sender*: the designated sender is honest. Similarly, we define the *good-case latency for BA* by replacing the good leader property with *good input*: all honest replicas have the same input. The good input property also ties back to the good-case of BFT SMR because if the leader is honest, then all honest replicas receive the same input from the leader.

The main goal of our paper is to develop BA and BB protocols with optimal good-case latency and apply them to BFT SMR protocols under the synchronous and authenticated setting. Due to space constraints, in this brief announcement we only present a synchronous Byzantine agreement protocol 1Δ-BA with optimal good-case latency of Δ (Theorem 3 and 4) and optimal resilience of $f < n/2$.

2 Byzantine Agreement with Optimal Good-case Latency

We consider n replicas in a reliable, authenticated all-to-all network, where up to f replicas can be malicious and behave in a Byzantine fashion, and rest of the replicas are honest. Without loss of generality, we assume $n = 2f+1$. We assume standard digital signatures and public-key infrastructure (PKI). We use $\langle x \rangle_p$ to denote a signed message x by replica p.

▶ **Definition 2** (Byzantine Agreement). *A Byzantine agreement protocol provides the following three guarantees.*

- *Agreement. If two honest replicas commit value b and b' respectively, then $b = b'$.*
- *Termination. All honest replicas eventually commit and terminate.*
- *Validity. If all honest replicas have the same input value, then all honest replicas commit on the value.*

We first show a lower bound of Δ on the good-case latency, adapted from [1].

▶ **Theorem 3.** *Any Byzantine agreement or broadcast protocol that is resilient to $f \geq n/3$ faults must have a good-case latency at least Δ.*

Initially, every replica i has an input b_i, starts Step 1 at the same time $t = 0$, and sets $b_{lck} = \perp$.

1. **Propose.** Sign and send the input value b_i to all others. Once receiving $f + 1$ distinct signed messages of the same value b, form a proposal B with these messages as $B = \{\langle b \rangle_j\}_{f+1}$.
2. **Forward.** Upon forming or receiving a valid new proposal B containing $f + 1$ distinct signed messages of the same value b, forward B to all other replicas, set $\texttt{vote-timer}_B$ for proposal B to Δ and start counting down.
3. **Vote.** When $\texttt{vote-timer}_B$ for proposal B containing value b reaches 0, if the replica does not receive another valid proposal B' containing $f + 1$ distinct signed messages of a different value $b' \neq b$, it broadcasts a vote in the form of $\langle \texttt{vote}, B \rangle_i$.
4. **Commit.** Upon collecting $f+1$ distinct signed votes $\langle \texttt{vote}, B \rangle$ of a valid proposal B containing value b at time t, (i) if $t \leq 3\Delta$, it broadcasts these $f + 1$ votes, sets $b_{lck} = b$, and commits b, (ii) if $t > 3\Delta$, it sets $b_{lck} = b$.
5. **Byzantine agreement.** At time 4Δ, invoke an instance of Byzantine agreement with b_{lck} as the input. If not committed, commit on the output of the Byzantine agreement. Terminate.

Figure 1 1Δ-BA Protocol under the Synchronous Model.

We now briefly describe the 1Δ-BA presented in Figure 1. Initially, all replicas have an input, and set their locked value b_{lck} to be some default value $\perp$. When protocol starts, all replicas first sign and broadcast its input value, and try to form a proposal containing $f + 1$ signed messages of an identical value (Step 1). When any replica i has a proposal of value b, it forwards the proposal to detect conflict (Step 2). After the replica forwards the proposal, it locally starts a timer called `vote-timer` to wait for Δ time period. If the timer expires and the replica does not receive any conflicting proposal containing a different value $b' \neq b$, it broadcasts a `vote` message for the proposal (Step 3). Once the replica gathers $f+1$ distinct `vote` messages for the proposal containing value b within time 3Δ, it broadcasts these `vote` messages, sets locked value $b_{lck} = b$ and commits the value b. If the $f + 1$ distinct `vote` messages for value b are received later than 3Δ, the replica only sets locked value $b_{lck} = b$ without committing the value (Step 4). At time 4Δ, the replica initiates an instance of Byzantine agreement with its locked value as the input, and any replica that has not committed yet will commit on the output of the agreement (Step 5).

▶ **Theorem 4.** *1Δ-BA protocol solves Byzantine agreement in the synchronous authenticated setting with optimal good-case latency of Δ and optimal resilience of $f < n/2$.*

3 Conclusion

We propose using the non-lock-step models and the good-case latency metric for Byzantine agreement and broadcast as they better capture what matters in practical BFT SMR. In the full version of this paper, we propose the first Byzantine agreement, broadcast, and state machine replication protocols with optimal good-case latency of Δ.

References

1 Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. Sync hotstuff: Simple and practical synchronous state machine replication. *IEEE Symposium on Security and Privacy (SP)*, 2020.

2 Michael J Fischer and Nancy A Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183–186, 1982.

3 A Ierzberg and S Kutten. Efficient detection of message forwarding faults. In *Proceeding of the 8th ACM Symposium on Principles of Distributed Computing*, pages 339–353, 1989.