# Brief Announcement: Practical Synchronous Byzantine Consensus*

**Ittai Abraham[1], Srinivas Devadas[2], Kartik Nayak[3], and Ling Ren[2]**

1 **VMware Research** – iabraham@vmware.com
2 **MIT** – {devadas, renling}@mit.edu
3 **UMD** – kartik@cs.umd.edu

**Abstract**

This paper presents new protocols for Byzantine state machine replication and Byzantine agreement in the synchronous and authenticated setting. The PBFT state machine replication protocol tolerates $f$ Byzantine faults in an asynchronous setting using $n = 3f + 1$ replicas. We improve the Byzantine fault tolerance to $n = 2f + 1$ by utilizing the synchrony assumption. Our protocol also solves synchronous authenticated Byzantine agreement in fewer expected rounds than the best existing solution (Katz and Koo, 2006).

**1998 ACM Subject Classification** C.2.4

**Keywords and phrases** consensus, agreement, Byzantine fault tolerance, replication, synchrony

**Digital Object Identifier** 10.4230/LIPIcs.DISC.2017.41

## 1 Introduction

Byzantine consensus is a fundamental problem in distributed computing and cryptography. Broadly speaking, Byzantine consensus considers the problem of reaching agreement among a group of $n$ parties, among which up to $f$ can have Byzantine faults and deviate from the protocol arbitrarily. There exist a few variant formulations for the Byzantine consensus problem. Two theoretical formulations are Byzantine broadcast and Byzantine agreement [4]. In Byzantine broadcast (BB), there is a designated *sender* who tries to broadcast a value; In Byzantine agreement (BA), every party holds an input value. To rule out trivial solutions, both problems have additional validity requirements. BA and BB have been studied under various combinations of timing (synchrony, asynchrony or partial synchrony) and cryptographic assumptions (whether or not to assume digital signatures). It is now well understood that these assumptions drastically affect the bounds on fault tolerance. In particular, BA requires $f < n/3$ under partial synchrony or asynchrony even with digital signatures, but can be solved with $f < n/2$ under synchrony with digital signatures.

A more practice-oriented problem formulation is Byzantine fault tolerant (BFT) state machine replication (SMR) [1]. In this formulation, the goal is to design a replicated service that provides the same interface as a single server, despite some replicas experiencing Byzantine faults. In particular, honest replicas agree on a sequence of values and their *order*, while the validity of the values is left outside the protocol. PBFT is an asynchronous SMR protocol that tolerates $f < n/3$ Byzantine faults [1]. As the first BFT protocol designed for practical efficiency, PBFT has since inspired numerous follow-up works.

* The full version of this paper is available at https://arxiv.org/abs/1704.02397. We use the word "consensus" as a collective term for all variants; other papers have different conventions.


Perhaps somewhat surprisingly, we do not yet have a practical solution for Byzantine consensus in the seemingly easier synchronous and authenticated (i.e., with digital signatures) setting. To the best of our knowledge, the most efficient BA protocol with the optimal $f < n/2$ fault tolerance in this setting is due to Katz and Koo [2], which requires in expectation 24 rounds of communication (not counting the random leader election subroutine). The only SMR protocol we know of in this setting is XFT [5]. Relying on an active group of $f+1$ honest replicas to make progress, XFT is designed to optimize efficiency for small $n$ and $f$ (e.g., $f = 1$). Its performance degrades as $n$ and $f$ increase, especially when $f = \lfloor \frac{n-1}{2} \rfloor$. In that case, among the $\binom{n}{f+1}$ $f+1$-sized groups in total, only one is all-honest. The simplest variant of XFT, as presented in [5], requires an exponential number of view changes to find that group. The best XFT variant we can think of still requires $\Theta(n^2)$ view changes.

This paper presents efficient Byzantine consensus protocols for the synchronous and authenticated setting tolerating $f < n/2$ faults. Our main focus is BFT SMR, for which our protocol requires amortized 4 rounds per slot independent of $n$ and $f$. (We say each value in the sequence fills one *slot*.) Meanwhile, our protocol can also solve multi-valued BA and BB for $f < n/2$ in expected 10 rounds assuming a random leader oracle. (The higher round complexity is due to the fact that BA/BB considers a single slot and cannot be amortized.)

## 1.1 Overview of the Our Protocols

Interestingly, our core protocol draws inspiration from the Paxos protocol [3], which is neither synchronous nor Byzantine fault tolerant. Since our main focus is SMR, we will describe the core protocol with "replicas" instead of "parties". The core of our protocol resembles the synod algorithm in Paxos, but is adapted to the synchronous and Byzantine setting. In a nutshell, it runs in iterations with a unique leader in each iteration (how to elect leaders is left to higher level protocols). Each new leader picks up the states left by previous leaders and drives agreement in its iteration. A Byzantine leader can prevent progress but cannot violate safety. As soon as an honest leader emerges, then all honest replicas reach agreement and terminate at the end of that iteration.

While synchrony is supposed to make the problem easier, it turns out to be non-trivial to adapt the synod algorithm to the synchronous and Byzantine setting while achieving the optimal $f < n/2$ fault tolerance. The major challenge is to ensure *quorum* intersection [3] at one *honest* replica. The core idea of Paxos is to form a quorum of size $f + 1$ before a commit. With $n = 2f + 1$, two quorums always intersect at one replica, which is honest in Paxos. In order to tolerate $f$ Byzantine faults, PBFT uses quorums of size $2f + 1$ out of $n = 3f + 1$, so that two quorums intersect at $f + 1$ replicas, among which one is guaranteed to be honest. At first glance, our goal of one honest intersection seems implausible with the $n = 2f + 1$ constraint. Following PBFT, we need two quorums to intersect at $f + 1$ replicas which seems to require quorums of size $1.5f + 1$. On the other hand, a quorum size larger than $f + 1$ (the number of honest replicas) seems to require participation from Byzantine replicas and thus loses liveness. Our solution is to utilize the synchrony assumption to form a *post-commit quorum* of size $2f + 1$. A post-commit quorum does not affect liveness and intersects with any *pre-commit quorum* (of size $f + 1$) at $f + 1$ replicas. This satisfies the requirement of one honest replica in intersection.

To implement the above quorum intersection idea, each iteration in our core protocol consists of 4 rounds. The first three rounds are conceptually similar to Paxos: (1) the leader learns the states of the system, (2) the leader proposes a safe value, and (3) every replica sends a commit request to every other replica. If a replica receives $f+1$ commit requests for the same value, it commits on that value and *notifies* all other replicas about the commit

using a 4th round. Upon receiving a notification, other replicas *accept* the committed value and will vouch for that value to future leaders. To tolerate Byzantine faults, we need to add equivocation checks and other proofs of honest behaviors at various steps. We can then apply the core synod protocol to SMR as well as BA/BB.

For SMR, a simple strategy is to rotate the leader role among the replicas after each iteration. Because each honest leader is able to fill at least one slot, the protocol spends amortized 2 iterations (8 rounds) per slot with $f < n/2$ faults. We then improve the protocol to allow a stable leader and only replace the leader if it is not making progress. The improved protocol fills one slot in every iteration (4 rounds). While our view change protocol resembles that of PBFT at a high level, the increased fault threshold $f < n/2$ again creates new challenges. In particular, two views in PBFT cannot make progress concurrently: $f + 1$ honest replicas need to enter the new view to make progress there, leaving not enough replicas for a quorum in the old view. In contrast, with a quorum size of $f + 1$ and $n = 2f + 1$ in our protocol, if a single honest replica is left behind in the old view, the $f$ Byzantine replicas can exploit it to form a quorum and violate safety. Thus, our view change protocol needs to ensure that two honest replicas are never in different views. In the end, our protocol achieves the result in Theorem 1.

▶ **Theorem 1.** *There exists a synchronous leader-based SMR protocol with optimal Byzantine fault tolerance $n = 2f + 1$. If a leader is non-faulty, each decision takes 4 rounds. A view change (replacing a leader) takes 4 rounds.*

To solve BB, we let the designated sender be the leader for the first iteration. After the first iteration, we rotate the leader role among all $n$ parties. It is straightforward to see that this solution achieves both agreement and validity. If the designated sender is honest, every honest party agrees on its value and terminates. Otherwise, the first honest leader that appears down the line will ensure agreement and termination for all honest parties. Assuming we have a random leader oracle, there is a $(f + 1)/(2f + 1) > 1/2$ probability that each leader after the first iteration is honest, so the protocol terminates in expected 2 iterations after the first iteration. To solve BA, we can use the classical transformation from Lamport et al. [4]. These give rise to the results in Theorem 2.

▶ **Theorem 2.** *Assuming a random leader election oracle, there exist synchronous BA and BB protocols for $f < n/2$ that terminate in expected 10 rounds.*

We remark that the $f < n/2$ Byzantine fault tolerance in our protocols is optimal for synchronous authenticated BA and SMR, but not for BB. Our quorum-based approach cannot solve BB in the dishonest majority case ($f \geq n/2$).

## References


1 Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *OSDI*, pages 173–186. USENIX Association, 1999.

2 Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. *J. Comput. Syst. Sci.*, 75(2):91–112, 2009.

3 Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, 1998.

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

5 Shengyun Liu, Christian Cachin, Vivien Quéma, and Marko Vukolic. XFT: practical fault tolerance beyond crashes. In *OSDI*, pages 485–500. USENIX Association, 2016.