Brief Announcement: Subquadratic Multivalued Asynchronous Byzantine Agreement WHP

Shir Cohen
Technion, Haifa, Israel

Idit Keidar
Technion, Haifa, Israel

Abstract

There have been several reductions from multivalued consensus to binary consensus over the past 20 years. To the best of our knowledge, none of them solved it for Byzantine asynchronous settings. In this short paper, we close this gap. Moreover, we do so in subquadratic communication, using newly developed subquadratic binary Byzantine Agreement techniques.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Cryptographic primitives; Mathematics of computing → Probabilistic algorithms

Keywords and phrases Byzantine agreement, subquadratic communication, fault tolerance in distributed systems

Digital Object Identifier 10.4230/LIPIcs.DISC.2023.39

Related Version *Full Version*: `https://arxiv.org/abs/2308.02927` [3]

Funding *Shir Cohen*: Supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities.

1 Introduction

Byzantine Agreement (BA) is a well-studied problem where a set of correct processes have input values and aim to agree on a common decision despite the presence of malicious ones. This problem was first defined over 40 years ago [8]. However, in the past decade BA gained a renewed interest due to the emergence of blockchains as a new distributed and decentralized tool. Moreover, the scale of the systems in which this problem is solved is much larger than in the past. As a result, there is a constant effort to find new techniques that will enable the reduction of communication complexity of BA solutions.

A significant improvement in BA scalability was enabled by subquadratic solutions, circumventing Dolev and Reischuk's renown lower bound of $\Omega(n^2)$ messages [5]. This was done by King and Saia [7] in the synchronous model and later by Algorand [6] (first in the synchronous model and then with eventual synchrony). In their work, Algorand presented a validated committee sampling primitive, based on the idea of cryptographic sortition using *verifiable random functions* (VRF) [9]. This primitive allows different subsets of processes to execute different parts of the BA protocol. Each committee is used for sending exactly one protocol message and messages are sent only by committee members, thus reducing the communication cost.

In this paper, we tackle the asynchronous model, which best describes real-life settings. Importantly, subquadratic asynchronous BA was first introduced not so long ago by Cohen et. al [4] and Blum et. al [2]. The limitation of these results is that both solve a binary BA, where the inputs and outputs are in $\{0, 1\}$. We extend the binary results to multivalued BA, which is more suitable in real-world systems. Nowadays, perhaps the most extensive use of BA solution appears in blockchains to agree on the next block. These blocks carry multiple

transactions (as well as additional metadata), which is clearly not a binary value. We note that a similar extension for multivalued consensus was presented in asynchrony by Mostefaoui, Raynal, and Tronel [11] but it cannot handle Byzantine failures. Another reduction by Turpin and Coan [12] is able to handle Byzantine failures, but only in a synchronous model. Despite not solving the multivalued case in the Byzantine asynchronous model, both of them have quadratic word complexity (Following the standard complexity notions [1, 10]).

We consider a system with a static set of n processes and an adversary, in the so-called "permissioned" setting, where the ids of all processes are well-known. The adversary may adaptively corrupt up to $f = (\frac{1}{3} - \epsilon)n$ processes in the course of a run, where $\frac{1}{2 \ln n} < \epsilon < \frac{1}{3}$. In addition, we assume a trusted *public key infrastructure* (PKI) that allows us to use *verifiable random functions* (VRFs) [9]. Finally, we assume that once the adversary takes over a process, it cannot "'front run" messages that that process had already sent when it was correct, causing the correct messages to be supplanted. This assumption can be replaced if one assumes an erasure model as used in [2, 6]. That is, using a separate key to encrypt each message, and deleting the secret key immediately thereafter.

Let us first examine the "straightforward" (yet faulty) reduction from multivalued BA to binary BA, both satisfying the *strong unanimity* validity property. This property states that if all correct processes have the same input value, then this must be the decision they all output. To solve the multivalued BA, any process interprets its input as a binary string, and then all processes participate in a sequence of binary BA instances. The input for the i^{th} instance by process p is the i^{th} digit in the binary representation. It is easy to see, that if all correct processes share the same input value, they start each instance with the same binary value and by validity agree upon it. Hence, by the end of the last BA instance, they all reach the same (input) decision. Otherwise, they can agree on some arbitrary common value.

Unfortunately, the simple binary-to-multivalued reduction does not work when applied with existing asynchronous subquadratic solutions. Assume that in the multivalued version of BA, values are taken from a finite domain $\mathcal{V}$. If the size of $\mathcal{V}$ is in $O(n)$, then to represent the input value as a binary string we need $O(\log n)$ bits, and the same number of BA instances. Although this number keeps the overall complexity subquadratic, it breaks the probability arguments made in the existing solutions. Briefly, both works take advantage of logarithmic subsets of processes that drive the protocol progress. These so-called committees are elected uniformly such that with high probability (WHP) it contains "enough" correct processes, and not "too many" Byzantine ones. Since, WHP, both algorithms complete in a constant number of rounds, their safety and liveness are guaranteed WHP. However, once we apply these techniques and make the probability arguments more than a constant number of times (as we would if we were to apply the reduction), the high probability does not remain high at all.

To overcome this challenge, we take a different approach. We generalize the method in [12] to work with asynchronous committee sampling and solve for *weak unanimity* validity. Our algorithm requires only two additional committees, compared to any binary BA algorithm. Finally, we present the first multivalued BA with a word complexity of $\widetilde{O}(n)$.

Validated Committee Sampling

Using VRFs, it is possible to implement *validated committee sampling*, which is a primitive that allows processes to elect committees without communication and later prove their election. It provides every process p_i with a private function $sample_i(s, \lambda)$, which gets a string s and a threshold $1 \leq \lambda \leq n$ and returns a tuple $\langle v_i, \sigma_i \rangle$, where $v_i \in \{true, false\}$ and σ_i is a proof that $v_i = sample_i(s, \lambda)$. If $v_i = true$ we say that p_i is *sampled* to the committee

for s and λ. The primitive ensures that p_i is sampled with probability $\frac{\lambda}{n}$. In addition, there is a public (known to all) function, *committee-val*$(s, \lambda, i, \sigma_i)$, which gets a string s, a threshold λ, a process identification i and a proof σ_i, and returns *true* or *false*.

Consider a string s. For every i, $1 \leq i \leq n$, let $\langle v_i, \sigma_i \rangle$ be the return value of *sample*$_i(s, \lambda)$. The following is satisfied for every p_i:

- *committee-val*$(s, \lambda, i, \sigma_i) = v_i$.
- If p_i is correct, then it is infeasible for the adversary to compute *sample*$_i(s, \lambda)$.
- It is infeasible for the adversary to find $\langle v, \sigma \rangle$ s.t. $v \neq v_i$ and *committee-val*$(s, \lambda, i, \sigma) =$ *true*.

Due to space limitations, we present here the parameters and guarantees as presented and proven in [4] using Chernoff bounds. For simplicity, we only state the claims we are using in this paper.

Committee Sampling Properties from [4]. *Let the set of processes sampled to the committee for s and λ be $C(s, \lambda)$, where λ is set to $8 \ln n$. Let d be a parameter of the system such that $\frac{1}{\lambda} < d < \frac{\epsilon}{3} - \frac{1}{3\lambda}$. We set $W \triangleq \lceil (\frac{2}{3} + 3d)\lambda \rceil$ and $B \triangleq \lfloor (\frac{1}{3} - d)\lambda \rfloor$. With high probability the following hold:*

(S3) *At least W processes in $C(s, \lambda)$ are correct.*

(S4) *At most B processes in $C(s, \lambda)$ are Byzantine.*

(S5) *Consider $C(s, \lambda)$ for some string s and two sets $P_1, P_2 \subset C(s, \lambda)$ s.t $|P_1| = |P_2| = W$. Then, $|P_1 \cap P_2| \geq B + 1$.*

2 From Binary BA to Multivalued BA

In the Byzantine Agreement (BA) problem, a set Π of n processes attempt to reach a common decision. In addition, the decided value must be "valid" in some sense which makes the problem non-trivial. We consider two standard variants of BA for asynchrony that differ in their validity condition and the domain of inputs by processes in the system. In the binary version, all inputs are taken from the domain $\{0, 1\}$, while in the multivalued case they can be any value from any finite domain $\mathcal{V}$. For the validity condition, in order to support a larger domain we weaken the validity condition. Instead of the known *strong unanimity* property, we opt for *weak unanimity* as defined below. In this work we show how to reduce a subquadratic weak multivalued BA to a subquadratic binary strong BA, both are solved WHP. That is, a probability that tends to 1 as n goes to infinity. Formally, we take a black-box solution to:

▶ **Definition 1** (Binary Strong Byzantine Agreement WHP). *In Binary Strong Byzantine Agreement WHP, each correct process $p_i \in \Pi$ proposes a binary input value v_i and decides on an output value decision$_i$ s.t. with high probability the following properties hold:*

- *Validity (Strong Unanimity). If all correct processes propose the same value v, then any correct process that decides, decides v.*
- *Agreement. No two correct processes decide differently.*
- *Termination. Every correct process eventually decides.*

And use it to solve:

▶ **Definition 2** (Multivalued Weak Byzantine Agreement WHP). *In Multivalued Weak Byzantine Agreement WHP, each correct process $p_i \in \Pi$ proposes an input value v_i and decides on an output value decision$_i$ s.t. with high probability the following properties hold:*

Algorithm 1 Multivalued Byzantine Agreement(v_i): code for p_i.

local variables: $alert \in \{true, false\}$, initially $false$
$count \in \mathbb{N}$, initially 0
init-set, *init-values-set*, *converge-set* $\in \mathcal{P}(\Pi)$, initially 0
1: **if** $sample_i(\text{INIT}, \lambda) = true$ **then** broadcast $\langle \text{INIT}, v_i \rangle_i$
2: **upon receiving** $\langle \text{INIT}, v_j \rangle_j$ with valid v_j from validly sampled p_j **do**
3: *init-set* $\leftarrow$ *init-set* $\cup\{j\}$
4: *init-values-set* $\leftarrow$ *init-values-set* $\cup\{v_j\}$
5: **if** $sample_i(\text{CONVERGE}, \lambda) = true$ and $|\textit{init-set}| = W$ for the first time **then**
6: **if** *init-values-set* $= \{v_i\}$ **then** ▷ All received values are p_i's initial value
7: batch the W messages into QC_{v_i}
8: send $\langle \text{CONVERGE}, true, QC_{v_i} \rangle_i$ to all processes
9: **else**
10: send $\langle \text{CONVERGE}, false, \perp \rangle_i$ to all processes
11: **upon receiving** $\langle \text{CONVERGE}, is\_content, QC_v \rangle_j$ from validly sampled p_j **do**
12: *converge-set* $\leftarrow$ *converge-set* $\cup\{j\}$
13: **if** *is_content*=*true* **then**
14: *count*$+ = 1$
15: **when** $|\textit{converge-set}| = W$ for the first time
16: $alert \leftarrow count < B + 1$
17: $binary\_decision_i \leftarrow$ *Binary Byzantine Agreement(alert)*
18: **if** $binary\_decision_i = true$ **then**
19: $decision_i \leftarrow \perp$
else
20: wait for a message of the form $\langle \text{CONVERGE}, true, QC_v \rangle_j$ from validly sampled p_j if such was not already received
21: $decision_i \leftarrow v$

- *Validity (Weak Unanimity). If all processes are correct and propose the same value v, then any correct process that decides, decides v.*
- *Agreement. Same as above.*
- *Termination. Same as above.*

We employ committee sampling and a binary subquadratic strong BA to present a multivalued solution to the weak BA problem. That is, the processes' initial values are from an arbitrary domain $\mathcal{V}$. We follow the method presented in [12] and adjust it to work with an asynchronous environment and committee sampling to achieve a subquadratic solution. The algorithm, presented in Algorithm 1, consists of two communication phases followed by a binary BA execution. To reduce the communication costs of the algorithm, the two phases are being executed only by a subset of the processes that are elected uniformly in random by a committee sampling primitive.

The first step is an INIT step, in which all INIT committee members send their signed initial value to all other processes (line 1). The second is a CONVERGE step, during which all CONVERGE committee members aim to converge around one common value. To do so, CONVERGE processes are waiting to hear from sufficiently many processes in the INIT committee. Since committees are elected using randomization, it is impossible for processes to wait for all of the previous committee members, as the size of the committee is unknown. Instead, it is guaranteed that a process hears from at least W processes WHP.

For some process p in the CONVERGE committee, if all W INIT messages include the same value v, that is also p's initial value, then p is considered to be *content*. To inform all other processes, p sends a CONVERGE message claiming to be content, that also carries

a quorum certificate (QC) on the value v containing all received messages (line 8). In the complementary case, where p knows of at least two different values by line 6, it sends a CONVERGE message with a false is_content flag (line 10).

In the third part of the algorithm, processes run a binary consensus whose target is deciding whether the system as a whole is content. To do so, processes update an alert flag that is determined according to the number of content processes in the CONVERGE committee (lines 13 – 16). It is designed such that if a correct process p hears from at least one non-content correct process, it sets its alert flag to true. To do so, we utilize the parameter B which is, according to the specification, an upper bound on the number of Byzantine processes in a committee.

After processes set their boolean *alert* flag, they make a binary decision on its values at line 17. If the output is *true*, then all correct processes output $\perp$ (line 19). Notice that by the strong unanimity property of the binary BA, it is impossible that all correct processes have had *alert=false* before the execution. Namely, there were many non-content processes in the converge committee. Otherwise, if the binary decision is *true*, then it must be the case that (i) all content processes are content with respect to the same value and (ii) at least one correct process was content in the converge committee (proof appears in the full version [3]). In this case, that process carries a quorum certificate with W different signatures on a value v, that can be safely decided upon (line 21) and is guaranteed to eventually arrive at all correct processes. This is mainly thanks to the fact that two subsets of the converge committee of size W intersect by at least one correct process.

Complexity

In Algorithm 1 all correct processes that are sampled to the two committees (lines 1,5) send messages to all other processes. Each of these messages contains a value from the finite domain, a VRF proof of the sender's election to the committee, and possibly a quorum certificate of W different signatures. Therefore, each message's size is either a constant number of words or W words. Thus, the total word complexity of a multivalued weak BA WHP is $O(nWC)$ where C is the number of processes that are sampled to the committees. Since each process is sampled to a committee with probability $\frac{\lambda}{n}$, we get a word complexity of $O(nW\lambda) = O(n \log^2 n) = \widetilde{O}(n)$ in expectation.

In the full version [3], we prove the following theorem:

▶ **Theorem 3.** *Algorithm 1 implements multivalued weak Byzantine Agreement WHP with a word complexity of* $\widetilde{O}(n)$.

3 Conclusions

Real-world systems are asynchronous and prone to Byzantine failures. This paper presents an algorithm that reduces the multivalued weak BA WHP to binary strong BA. Together with the binary BA presented in [4, 2] this paper yields that first subquadratic multivalued BA.

References

1 Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. Asymptotically optimal validated asynchronous byzantine agreement. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 337–346, 2019.

2 Erica Blum, Jonathan Katz, Chen-Da Liu-Zhang, and Julian Loss. Asynchronous byzantine agreement with subquadratic communication. Cryptology ePrint Archive, Report 2020/851, 2020.

3 Shir Cohen and Idit Keidar. Subquadratic multivalued asynchronous byzantine agreement whp. *arXiv preprint arXiv:2308.02927*, 2023.

4 Shir Cohen, Idit Keidar, and Alexander Spiegelman. Not a coincidence: Sub-quadratic asynchronous byzantine agreement whp. In *34th International Symposium on Distributed Computing (DISC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

5 Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. *J. ACM*, 32(1):191–204, January 1985.

6 Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 51–68, 2017.

7 Valerie King and Jared Saia. Breaking the $O(n^2)$ bit barrier: scalable byzantine agreement with an adaptive adversary. *Journal of the ACM (JACM)*, 58(4):1–24, 2011.

8 Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

9 Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 120–130. IEEE, 1999.

10 Achour Mostéfaoui, Hamouma Moumen, and Michel Raynal. Signature-free asynchronous binary byzantine consensus with $t < n/3$, $O(n^2)$ messages, and $O(1)$ expected time. *Journal of the ACM (JACM)*, 62(4):31, 2015.

11 Achour Mostefaoui, Michel Raynal, and Frédéric Tronel. From binary consensus to multivalued consensus in asynchronous message-passing systems. *Information Processing Letters*, 73(5-6):207–212, 2000.

12 Russell Turpin and Brian A Coan. Extending binary byzantine agreement to multivalued byzantine agreement. *Information Processing Letters*, 18(2):73–76, 1984.