Byzantine Agreement in the Full-Information Model in O(log n) Rounds

Michael Ben-Or*
The Hebrew University
Jerusalem, Israel
benor@cs.huji.ac.il

Elan Pavlov
MIT
Cambridge, MA, USA
elan@mit.edu

Vinod Vaikuntanathan†
CSAIL, MIT
Cambridge, MA, USA
vinodv@mit.edu

Abstract

We present a randomized Byzantine Agreement (BA) protocol with an expected running time of $O(\log n)$ rounds, in a synchronous *full-information* network of n players. For any constant $\epsilon > 0$, the constructed protocol tolerates t non-adaptive Byzantine faults, as long as $n \geq (4+\epsilon)t$. In the full-information model, no restrictions are placed on the computational power of the faulty players or the information available to them. In particular, the faulty players may be infinitely powerful, and they can observe all communication among the honest players.

This constitutes significant progress over the best known randomized BA protocol in the same setting which has a round-complexity of $\Theta(\frac{t}{\log n})$ rounds [9], and answers an open problem posed by Chor and Dwork [10].

*Incumbent of the Jean and Helena Alfassa Chair in Computer Science. Research supported in part by an Israel Defense Ministry Research Authority grant and by the Israel Science Foundation.

†Supported in part by an NSF grant CNS-0430450

1 Introduction

Byzantine Agreement (BA) is arguably the central problem in distributed computation tolerating faulty behavior. Informally, the problem is to maintain a common view of the world in the presence of faulty processes that strive to prevent the good processes from reaching agreement. The faults can range from simple "crashing of processes" to processes with malicious intent co-ordinating to mislead the good ones into disagreement. Byzantine Agreement is equivalent to the problem of reliable broadcast among a set of processes.

The problem of Byzantine Agreement, in its most basic form, is as defined below.

Definition 1 (Byzantine Agreement)**.** *Let $\mathcal{P}$ be a protocol among n players, in which each player P_i starts with an input bit b_i, and P_i outputs a bit c_i at the end of the protocol. $\mathcal{P}$ is a Byzantine Agreement protocol, if the following conditions hold:*

- Agreement: *For any two non-faulty players*[1] *P_i and P_j, $c_i = c_j$.*
- Validity: *If $b_i = b_j = b$ for all non-faulty players P_i and P_j, then $c_i = b$ for all non-faulty players P_i.*
- Termination: *Protocol $\mathcal{P}$ terminates with probability* 1*.*

The problem was introduced by Pease, Shostak and Lamport [20] in 1980. One can consider different network models, models of interprocessor communication, and fault models. In this paper, we focus on the case of a synchronous network, with point-to-point authenticated channels between every pair of processors. A synchronous network is an idealized model, where a common clock governs the delivery of messages. Messages are sent at the end of a round and are delivered at the beginning of the next round. The fault is modeled by a coalition of t players, who are corrupted by an adversary before the protocol begins (i.e, the adversary is *non-adaptive*). The adversary is *computationally unbounded* and has full information about the state of all the players, and the communication between any pair of them. The adversary decides the messages that the t corrupted players send in a round, depending on the messages sent by the good players in all the previous rounds, including the current round (i.e, the adversary is *rushing*). This is referred to as the *full-information model* in the literature [4, 16].

A point worth re-emphasizing is that, we do not assume secrecy of communication between any two players. We cannot use cryptography to ensure the secrecy either, since the adversary is computationally unbounded. In particular, the adversary can listen to *any* communication in the network, but a good player can only hear the messages sent to it.

Our protocol is probabilistic, and therefore, we assume that every player has a source of perfect randomness (say, independent and unbiased coin flips).

1.1 Previous Work

Byzantine Agreement

Since its introduction in the work of Pease et al.[20], the problem of Byzantine Agreement has been a source of enormous attention. Pease et al. proved (in [20] itself) that no deterministic algorithm can achieve Byzantine Agreement among n players in the presence of t faults if $n \leq 3t$

[1] The meaning of which players are non-faulty depends on the adversary.

(This bound was later extended to the case of randomized algorithms by Karlin and Yao [19]). They also constructed a (deterministic) algorithm that solves BA for any $n > 3t$, in a synchronous full-information network. Once the feasibility of BA was shown, further attempts concentrated on reducing the complexity of achieving agreement. The standard complexity measures of interest are the number of rounds, and the total communication and computational complexity of the protocol, the former being the most interesting of them. The protocol of [20] had a round complexity of $t+1$ rounds, which was shown to be optimal for *deterministic* protocols by Fischer and Lynch [14]. However, the communication complexity of the protocol was exponential in n. Following a series of works [5, 6], Garay and Moses [15] constructed a BA protocol that runs for $t+1$ rounds, with a polynomial communication.

Randomized Byzantine Agreement

Faced with the lower bound on the round complexity for deterministic protocols, the natural direction of research was to find ways to overcome this limitation, the first choice being to resort to randomization.[2] This direction was pursued early on, starting with the work of Ben-Or, Rabin and Bracha [3, 21, 8] who put forth the idea of a common coin as the correct notion of randomization to achieve Byzantine Agreement. A common coin is a "sufficiently random" coin seen by "sufficiently many processors", as defined by Dwork, Shmoys and Stockmeyer [11]. In particular, [21] and [11] showed how to achieve Byzantine Agreement in $O(1)$ extra rounds, given progressively weaker notions of a common coin. Thus, the bulk of the attention was concentrated on constructing protocols that generate a common-coin in a network.

Under the assumption that the point-to-point channels connecting pairs of processors are *private*, or that the processors are computationally bounded and cryptography exists, Feldman and Micali [13] constructed a protocol to generate a common coin in $O(1)$ rounds and with polynomial communication. This, in turn, gave Byzantine Agreement protocols that run in $O(1)$ rounds (by Rabin's result [21]). On the other hand, without the assumption of private channels or a computationally bounded adversary (i.e, in the full-information model), the best known protocol that achieved Byzantine Agreement had a round complexity of $\Theta(\frac{t}{\log n})$ rounds [9].

The Full-Information Model

The full-information model was introduced by Ben-Or and Linial [4] to study collective coin-flipping, which is the problem of generating a common bounded-bias bit in a network of n players with t faults. This problem was studied in a series of works that aimed to improve the fault-tolerance and round-complexity, resulting in the protocols of Russell and Zuckerman [22] and Feige [12], that construct $\log^* n + O(1)$ round protocols that tolerate, for any $\epsilon > 0$, $t < \frac{n}{2+\epsilon}$ faults. Goldreich et al. [16] consider the problem of multiparty computation in the full-information model. Note that these coin-flipping protocols *assume* the existence of a *broadcast channel*, and therefore, cannot be used *as such* to construct a broadcast protocol![3] Nevertheless, we use the ideas from the coin-flipping protocol of Feige [12] in an essential way in our Byzantine Agreement protocol.

[2]Note that the Agreement, Validity and Termination conditions in the definition of Byzantine Agreement are required to hold *with probability* 1 over the coin-tosses of the processors. The complexity measure of interest is the *expected* running time of the protocol.

[3]Recall that the problem of reliable broadcast among n processors is *equivalent* to the problem of Byzantine Agreement.

Our Results

Main Theorem 1. *For any constant $\epsilon > 0$, there exists a (explicit) protocol BA_ϵ that reaches Byzantine Agreement in a synchronous full-information network tolerating $t < (\frac{1}{4} - \epsilon)n$ non-adaptive Byzantine faults, and runs for expected $O(\frac{\log n}{\epsilon^2})$ rounds.*

We remark that the restriction that the adversary be nonadaptive is *essential* – Ben-Or and Bar-Joseph [2] show that any BA protocol that tolerates $t = \Theta(n)$ adaptive fail-stop faults runs for $\tilde{\Omega}(\sqrt{n})$ rounds.

Perspective

As early as the mid-1980s, researchers observed that the bulk of the work in randomized Byzantine Agreement actually introduced two twists in the original setting of Byzantine Agreement: (a) Allowing randomization (and a probabilistic analysis, consequently), and (b) Using cryptographic primitives and computational assumptions (either explicitly, or implicitly by using a physical assumption such as the existence of point-to-point "private channels"). A natural question raised then was whether both these elements are essential. That is, can Byzantine Agreement be sped up significantly by relying *only on randomization*, without using cryptographic techniques ? Theorem 1 provides a positive answer to this question.

A Synopsis of Our Solution

Following the time-tested notion of reducing randomized Byzantine Agreement to the problem of flipping a global bounded-bias coin, we concentrate our attention to designing such a protocol. As pointed out earlier, standard techniques (due to Rabin [21] and Dwork et al. [11]) show how to get a BA protocol with a constant-round overhead, given a common-coin protocol.

The "immediately obvious" approach is to use the collective coin-flipping protocols of Feige [12] or Russell and Zuckerman [22]. However, we cannot use these protocols as such, because they work under the assumption that there exists a broadcast channel, which we cannot assume.

Nevertheless, our protocol borrows ideas from the collective coin-flipping protocol of Feige [12]. Feige's protocol for collective coin-flipping works as follows: All the players are alive in the beginning of the protocol. In the first round, the players throw a ball each at random into one of $O(\frac{n}{\log n})$ bins. The players who threw their balls into the *lightest bin* survive for the next round. The protocol is then recursively invoked on the $O(\log n)$ players in the lightest bin. The crucial idea is that, assuming that the good players throw their balls randomly, their balls are almost uniformly distributed among the bins. Thus, the lightest bin contains approximately the right fraction of good players. Therefore, this protocol can be viewed as a way of electing a small subset (a "committee") of the n players, that contains a "large enough" fraction of good players. After $\log^* n$ recursive invocations of this process, a leader is elected. We let the leader flip a coin, and broadcast it.

Note that each step of this protocol assumes that the players broadcast their choices of the bins to all the players. Since we do not have a broadcast channel, we have to implement it, and that requires Byzantine Agreement. It looks like we are back to the same problem. The trick to avoid this circularity is to use a certain weak version of broadcast (called graded broadcast) to implement the first stage of Feige's protocol. We then proceed to show that this reduces BA among n players to BA among $O(\log n)$ players. Thus, assuming that we can implement graded broadcast in $O(1)$ rounds, we get an $O(\log n)$ round BA protocol.

2 The Toolkit

Notations. Letters such as $\mathcal{P}, \mathcal{S}$ denote protocols. We usually denote subprotocols of a protocol $\mathcal{P}$ by subscripts, such as $\mathcal{P}_i$, and subprotocols of $\mathcal{P}_i$ by superscripts, such as $\mathcal{P}_i^j$. Players are denoted by the letter P, possibly with subscripts. The set $\{1, 2, \ldots, n\}$ is also denoted $[n]$.

Graded Broadcast

The first tool we need is an appropriate notion of a weak version of broadcast along with an implementation of such a notion with low round-complexity. The correct weakening of broadcast we need is the notion of graded broadcast [13], defined below. Informally, a graded broadcast protocol is a protocol with a designated player called "dealer" (the one who broadcasts) such that:

- If the dealer is good, all the players get the same message.
- Even if the dealer is bad, if some good player accepts the message, all the good players get the same message (but they may or may not accept it).

Definition 2 (Graded Broadcast [13]). *A protocol $\mathcal{P}$ is said to be achieve graded broadcast if, at the beginning of the protocol, a designated player D (called* the dealer*) holds a value v, and at the end of the protocol, every player P_i outputs a pair* $(\mathsf{value}_i, \mathsf{confidence}_i)$ *such that the following properties hold: ($\forall i$, $\mathsf{confidence}_i \in \{0, 1, 2\}$)*

1. *If D is honest, then $\mathsf{value}_i = v$ and $\mathsf{confidence}_i = 2$ for every honest player P_i.*
2. *For any two honest players P_i and P_j, $|\mathsf{confidence}_i - \mathsf{confidence}_j| \leq 1$.*
3. *(Consistency) For any two honest players P_i and P_j, if $\mathsf{confidence}_i > 0$ and $\mathsf{confidence}_j > 0$, then $\mathsf{value}_i = \mathsf{value}_j$.*

An $O(1)$-rounds deterministic protocol with these guarantees appears, for instance, in Feldman and Micali [13] as a "gradecast" protocol. The gradecast protocol from [13] is described, for completeness, in Table 1.

Lemma 1 ([13]). *The protocol* Gradecast *is a graded broadcast protocol.*

Proof. The proof follows from the following series of observations:

- Suppose D is honest. Then, after Step 1, all the honest players get the same v_i. In Step 3, every honest player will set $\mu = v$, since all the $n - t$ honest players send m to him in Step 2. This implies that all honest players output $(v, 2)$ in step 4.
- Suppose some honest player outputs $(\mu, 2)$. This means for him, $\mathsf{num}_\mu \geq 2t + 1$. Since at most t of these come from bad players, $\mathsf{num}_\mu \geq t + 1$ for all other honest players. This shows that for any two honest players P_i and P_j, $|\mathsf{confidence}_i - \mathsf{confidence}_j| \leq 1$.
- Suppose an honest player outputs $(\mu, *)$ after step 4, where $* \in \{1, 2\}$. This means $\mathsf{num}_\mu \geq t + 1$ for him, and therefore, he received μ from at least one honest player after step 3. The honest player that sent μ to him did it because he got $\geq n - t$ v_i^j's equal to μ in step 3. At least $n - 2t \geq t + 1$ of these came from honest players. Thus, for any other μ', the number of honest players that transmit μ' in step 2 is at most t. It follows that no honest player sets $\mu = m'$ in step 3. This proves that no other honest player outputs $(\mu', *)$ for a $\mu' \notin \{\mu, \perp\}$.

$\square$

Protocol Gradecast
Input to the Dealer D: A value v. Output of player P_i: A pair $(\mathsf{value}_i, \mathsf{confidence}_i)$. 1. **(The dealer D)** The dealer D distributes v to all the players. 2. **(Every player P_i)** Let v_i denote the message received from D in Step 1. Send v_i to all the players. 3. **(Every player P_j)** Let v_i^j denote the message received from player P_i in Step 2. If there is a value μ such that $\geq n-t$ of the v_i^j's are equal to μ, then send μ to all the players. Else, send $\perp$. 4. **(Every player P_i)** Let num_μ denote the number of players that sent μ to P_i in Step 3. • If $\mathsf{num}_\mu \geq 2t+1$ for some μ, output $(\mu, 2)$. • If $2t \geq \mathsf{num}_\mu \geq t+1$ for some μ, output $(\mu, 1)$. • If $\mathsf{num}_\mu \leq t$ for all μ, output $(\perp, 0)$.

Table 1: The Graded Broadcast Protocol of Feldman and Micali

Reaching Byzantine Agreement Given a Common Coin

We briefly describe how to get Byzantine Agreement given a common coin protocol, for the sake of completeness. A common-coin protocol is defined as follows.

Definition 3 (Common Coin)**.** *A protocol $\mathcal{P}$ is said to be a common-coin protocol if, at the end of the protocol, every player P_i outputs* a bit b_i *and there exist constants $\delta, \epsilon > 0$ such that:* $\Pr[\exists b\ \forall i\ b_i = b] \geq \delta$*, and* $1 - \epsilon \geq \Pr[b = 0 \mid \exists b\ \forall i\ b_i = b] \geq \epsilon$*.*

The reduction from Byzantine Agreement to Common-coin (which is essentially from Rabin [21] and Dwork et al. [11]) is best described using the graded broadcast protocol (given in Table 1). Each player P_i has an input b_i (which we assume to be a bit, for simplicity). The protocol proceeds as follows.

(A) Start executing the gradecast protocol (of Table 1), from Step 2, with player P_i setting $v_i = b_i$.

(B) If the output of the gradecast is $(\mu, 2)$, for some μ, then decide on μ and terminate. If the output of the gradecast is $(\mu, 1)$, then set $v_i := \mu$ and go back to Step (A). If the output is $(\perp, 0)$, then set $m_i := r_i$, where r_i is the output of a common-coin protocol, and go back to Step (A).

The idea of the protocol is that, if a good player P_i terminates in a round, he must have had a confidence of 2 for some μ. By the property of gradecast, all other good players receive the same value μ with confidence at least 1 and therefore, all of them will decide by the next round. Suppose some subset of players have a value μ with confidence 1 and the others have $\perp$, at step (B). Then, r_i will be equal to μ, with constant probability. If this happens, all the good players will start the next iteration of gradecast with the same input, and therefore, will agree by the end of the iteration. This gives a constant expected-time reduction from the problem of Byzantine Agreement to the problem of generating a common coin.

Tail Bounds

We record the following version of the Chernoff bound for use in the analysis of our protocol.

Lemma 2 (Chernoff)**.** *Let $X_1, X_2, \ldots, X_n$ be random variables that take values in $\{0,1\}$, such that $\mathbb{E}(X_i) = p_i$. Let $\mu = \mathbb{E}(\sum_i X_i) = \sum_i p_i$. Then,* $\Pr[|\sum_i X_i - \mu| > \epsilon\mu] \leq 2e^{-\frac{1}{2}\epsilon^2\mu}$.

3 The Byzantine Agreement Protocol

We construct a sequence of protocols BA_ϵ, parametrized by an $\epsilon > 0$. BA_ϵ is expected to be a Byzantine Agreement protocol that tolerates any $t < (\frac{1}{4} - \epsilon)n$ faulty processors. BA_ϵ consists of three stages, executed sequentially. Define the constant a to be $\frac{7}{\epsilon^2}$.

Stage 1

The first stage of our protocol is the same as the first step of Feige's protocol [12] – each player chooses one of the $\frac{n}{a \log n}$ committees at random, and announces this choice to all the players. The difficulty with this implementation is that, a bad player need not give all the good players a consistent view of which committee he chose. To partially remedy this problem, we ask the players to announce their choices via a gradecast protocol. This, among other things, ensures that a bad player cannot convince two different good players that he chose different committees. Note, however, that it is still possible that a good player thinks player P_i is in a committee, whereas another good player thinks that P_i is not in the committee. In particular, Stage 1 of the protocol works as follows.

Stage 1: Form $\frac{n}{a \log n}$ Committees

1. (Each player P_i) Choose a random number B_i from $[1 \ldots \frac{n}{a \log n}]$, and gradecast B_i to all the players. The gradecast results in every other player P_j receiving a pair $(\mathsf{value}_j^i, \mathsf{confidence}_j^i)$.

2. (Each player P_j) Construct a local view of the composition of the committees. Define player P_j's view of committee C_k, denoted as $\mathsf{view}_j(C_k)$,[4] as the set of all players P_i whose gradecast resulted in $\mathsf{value}_j^i = C_k$ and $\mathsf{confidence}_j^i = 2$. That is,

$$\mathsf{view}_j(C_k) = \{P_i \mid \mathsf{value}_j^i = C_k \text{ and } \mathsf{confidence}_j^i = 2\}.$$

   Further, define player P_j's extended view of committee C_k, denoted as $\overline{\mathsf{view}}_j(C_k)$, to be the set of all players P_i whose gradecast resulted in $\mathsf{value}_j^i = C_k$ and $\mathsf{confidence}_j^i > 0$. That is,

$$\overline{\mathsf{view}}_j(C_k) = \{P_i \mid \mathsf{value}_j^i = C_k \text{ and } \mathsf{confidence}_j^i > 0\}.$$

[4] Strictly speaking, this should be $\mathsf{view}_{P_j}(C_k)$. We abbreviate it to $\mathsf{view}_j(C_k)$ for notational convenience.

Whenever a player $P \in \mathsf{view}_j(C_k)$, we say that player P_j *accepts* player P into committee C_k. Analogously, when a player $P \in \overline{\mathsf{view}}_j(C_k)$, we say that P_j *adopts* P into committee C_k. *Adopting* a player into a committee can be thought of as a safety mechanism. If a good player accepts a player P into a committee, then *all good players* will at least adopt P into the same committee. Lemma 3 below records this observation.

Lemma 3. *If a player $P \in \mathsf{view}_j(C)$ for some committee C and some honest player P_j, then $P \in \overline{\mathsf{view}}_k(C)$ for* all *honest players P_k.*

Proof. This follows from the consistency property of graded broadcast which says that for any two honest players P_i and P_j, $|\mathsf{confidence}_i - \mathsf{confidence}_j| \leq 1$. The fact that $P \in \mathsf{view}_j(C)$ for a honest player P_j implies (by the definition of views) that P's gradecast was accepted by P_j with confidence 2. Therefore, all other honest players P_k will accept P's gradecast with confidence at least 1, which means that $P \in \overline{\mathsf{view}}_k(C)$ for all honest P_k. $\square$

A few more general facts about the views are worth noting. First of all, note that $\overline{\mathsf{view}}_j(C_k) \supseteq \mathsf{view}_j(C_k)$ for all honest players P_j and committees C_k – this follows trivially from the definition of a view (and an extended view). Secondly, it is quite possible that for two different honest players P_j and $P_{j'}$, and committee C_k, $\mathsf{view}_j(C_k)$ and $\mathsf{view}_{j'}(C_k)$ are different. This is because the faulty players may present conflicting values to two different honest players (An analogous statement is true for $\overline{\mathsf{view}}$'s of two honest players).

Since the good players choose their committee uniformly at random, the expected number of good players that chose each committee is $(\frac{3}{4} + \epsilon)a \log n$. A simple application of Chernoff bound followed by a union bound shows that, with high probability, *every* committee has at least $\frac{3}{4}a \log n$ good players.

Lemma 4. *Let $t = (\frac{1}{4} - \epsilon)n$, $a = \frac{7}{\epsilon^2}$, and the number of committees be $\frac{n}{a \log n}$. With probability at least $1 - n^{-1}$, all committees have more than $\frac{3}{4}a \log n$* good players.

Proof. The expected number of good players in any committee is $(\frac{3}{4} + \epsilon)a \log n$. The probability that some committee (say, C_i) has at most $\frac{3}{4}a \log n$ good players is, by a Chernoff bound, at most $2e^{-\frac{2\epsilon^2}{3+4\epsilon}a \log n} < n^{-2}$, by our choice of a. The probability that *some* committee has less than $\frac{3}{4}a \log n$ good players is, by a union bound, at most $\frac{1}{n}$. $\square$

Since with high probability, every committee has $\frac{3}{4}a \log n$ good players, any committee of size k has at most $k - \frac{3}{4}a \log n$ bad players (with high probability).

Corollary 5. *With probability at least $1 - n^{-1}$, in any committee of size k, there are at most $k - \frac{3}{4}a \log n$ bad players.*

Since the number of good players is bounded from below, with high probability, any *small* committee has a large *fraction* of good players. If a committee is small, it can be much easier to reach agreement in the small committee and notify all of the players. The only problem is that there can be many contenders for the small committee in question. This means we need two agreements: first, the players agree on a single small committee (this is Stage 2), and secondly, the players in the chosen committee toss a bounded-bias coin among themselves, and notify all the players of the outcome (this is Stage 3).

In what follows, *we will assume that all committees have at least $\frac{3}{4}a \log n$ good players.* If this does not hold, we have no guarantee on the outcome of the coin-flip. But since this happens with a probability of $O(\frac{1}{n})$ (by Lemma 4), the extra bias that this event adds to the coin-flip is negligible.

Stage 2

We now wish to agree on a single small committee. If all the good players could agree on the composition of each committee, then the good players could choose a committee with the smallest number of players. Unfortunately, we do not know how to do this without a Byzantine Agreement protocol. In order to circumvent this, we allow (the players in) a committee to decide that the committee is too large and therefore should be *disqualified*.

This would have worked, if all committees were of size $O(\log n)$. But, a committee could be of size as much as $\Theta(n)$ because a large number of faulty players would decide to choose a particular committee in Stage 1. Thus, running a BA protocol on this committee would be prohibitively expensive. We solve this problem by letting *each small subset of a committee* run a BA among themselves to decide whether the committee is too large. More concretely, for every committee C_i, we look at all subsets of players S_j of size $\frac{3}{4}a \log n$ (There are $\binom{n}{\frac{3}{4}a\log n}$ such sets S_j). Each such set agrees (using a deterministic BA protocol) on the composition of a set $\overline{\mathsf{view}}_{S_j}(C_i)$, where a player P is considered to be in $\overline{\mathsf{view}}_{S_j}(C_i)$ if *all* of the players in S_j adopt P (i.e, $P \in \bigcap_{P_k \in S_j} \overline{\mathsf{view}}_k(C_i)$). If this set is larger than $a \log n$, then each member of S_j publicizes a disqualification of C_i. The players in S_j also compute $\bigcap_{P_k \in S_j} \mathsf{view}_k(C_i)$ to be what S_j thinks about the composition of C_i, and publicize it.

We note that it is quite possible for different S_j's to reach different conclusions about disqualifying C_i. The advantage we have here, contrary to Byzantine Agreement, is that there is a *preferred outcome* (i.e, the committee is too large and therefore is disqualified).

Once this is done, every other player P has to decide how to interpret the set of all messages it gets from the different S_j's. A good player P_k decides to disqualify a committee C_i if P_k receives at least $\frac{1}{2}a \log n$ valid disqualifications from a set S_j, and P_k has initially *accepted* all the players in S_j as belonging to C_i (i.e, $S_j \subseteq \mathsf{view}_k(C_i)$). Finally, if the committee is not disqualified, P_k chooses the largest advertised composition of C_i as the eventual composition of C_i. This is done to ensure that a large fraction of honest players survive in the chosen committee. More specifically, Stage 2 works as follows.

Stage 2 : Eliminate large committees, and agree on the composition of one of the remaining (small) committees.

1. Run the protocols $\mathcal{P}_1, \ldots, \mathcal{P}_{\frac{n}{a \log n}}$ in parallel. $\mathcal{P}_i$ is the protocol that decides whether to disqualify committee C_i.

2. Protocol $\mathcal{P}_i$ consists of subprotocols executed concurrently by each of the $\binom{n}{\frac{3}{4}a\log n}$ sets $S_j \subseteq \{P_1, \ldots, P_n\}$, where each S_j is of size $\frac{3}{4}a \log n$. Denote the subprotocol of the protocol $\mathcal{P}_i$ executed by set S_j as $\mathcal{P}_i^j$.

3. We now describe the subprotocol $\mathcal{P}_i^j$.

   (a) (To disqualify or not) Run a deterministic BA protocol in S_j to compute

$$\overline{\mathsf{view}}_{S_j}(C_i) \stackrel{def}{=} \bigcap_{P_k \in S_j} \overline{\mathsf{view}}_k(C_i)$$

Each player P in S_j computes a local variable $\mathsf{disq}_{P,S_j}(C_i)$ as follows.

$$\mathsf{disq}_{P,S_j}(C_i) = \begin{cases} 1 & \text{if } |\overline{\mathsf{view}}_{S_j}(C_i)| > a \log n. \\ 0 & \text{if } |\overline{\mathsf{view}}_{S_j}(C_i)| \leq a \log n. \end{cases}$$

(b) (If not disqualified, what is the composition of the committee ?) Run a deterministic BA protocol in S_j at the end of which each honest player P in S_j computes

$$\mathsf{composition}_{P,S_j}(C_i) \stackrel{def}{=} \bigcap_{P_k \in S_j} \mathsf{view}_k(C_i)$$

(c) P sends the tuple

$$(S_j, C_i, \mathsf{disq}_{P,S_j}(C_i), \mathsf{composition}_{P,S_j}(C_i))$$

to all other players.

4. (Each player P_k) If P_k receives messages of the form $(S_j, C_i, 1, *)$ from some set of players S'_j such that $|S'_j| \geq \frac{1}{2} a \log n$ and $S'_j \subseteq S_j \subseteq \mathsf{view}_k(C_i)$, then set $\mathsf{disq}_{P_k}(C_i) = 1$.

5. (Each player P_k) For every committee C_i and each set $S_j \subseteq \mathsf{view}_k(C_i)$, P_k does the following: If player P_k receives messages of the form $(S_j, C_i, 0, \mathsf{D})$ from some set of players $S'_j \subseteq S_j$ of size at least $\frac{1}{2} a \log n$, then set $\mathsf{composition}_{P_k,S_j}(C_i) = \mathsf{D}$, else set $\mathsf{composition}_{P_k,S_j}(C_i) = \perp$.

   Given this, player P_k defines the final composition of the committee C_i (denoted $\mathsf{final\_comp}_{P_k}(C_i)$) as the largest $\mathsf{composition}_{P_k,S_j}(C_i)$ among all sets S_j (where $\mathsf{composition}_{P_k,S_j}(C_i)$ was computed as above).

6. The players choose the lexicographically smallest committee that was not disqualified, as the chosen committee.

Define a "good set" as a set S_j (of size $\frac{3}{4} a \log n$) all whose members are good players. By Lemma 4, a "good set" exists for every committee [5].

We first need to show that at the end of this stage, all the players have a consistent view of which committees have been disqualified and which ones remain. Intuitively, the reason for this is as follows. An honest player will accept "disqualify committee C" messages only from players that it has *accepted* to C (in Stage 2, Step (e)). By Lemma 3, all such players are *adopted* to committee C by every other honest player. Suppose many of these players (who disqualified committee C) are bad. This means that every honest player adopts a lot of players into committee C, and therefore, "good set" will disqualify C and let this fact be known to the world. On the other hand, if many of the players that disqualified committee C are good, then they will themselves tell the world the right decision. This is the *agreement lemma* (Lemma 6).

We also need to show not all committees are eliminated and the composition agreed on for the chosen committee has more than $\frac{2}{3}$ fraction of good players. For the proof, see Section 3.1.

[5] There may be many good sets for a committee, if the number of good players in the committee C_i is large. In such a case, we designate an arbitrary such set as the good set for C_i.

Stage 3

Now that all the players have agreed on a small committee with a large fraction of good players, we run a leader election protocol (such as the ones of Feige [12] or Russell and Zuckerman [22]) within the chosen committee. The leader-election protocols assume a broadcast primitive, which we implement using a randomized Byzantine Agreement protocol that runs in expected $O(\frac{\log n}{\log\log n})$ rounds (such as the one of Chor and Coan [9]). Finally, we ask the chosen leader to flip a coin and send it to all the players.

3.1 Proof of the Main Theorem

Lemma 6 **(Agreement Lemma).** *At the end of Stage* 2*, the following holds:*

- *If some good player* P_i *sets* $\mathsf{disq}_{P_i}(C) = 1$ *for some committee* C*, then every other good player* P_j *sets* $\mathsf{disq}_{P_j}(C) = 1$.
- *Furthermore, for every committee* C *that is not disqualified,* $\mathsf{final\_comp}_{P_k}(C) = \mathsf{final\_comp}_{P_l}(C)$ *for any two honest players* P_k, P_l.

Proof. We divide the proof of the first assertion into two cases:

- Case 1: P_i set $\mathsf{disq}_{P_i}(C) = 1$ due to receiving messages from some set $S'_j \subseteq S_j$ such that S_j consists of more than $\frac{1}{4}a\log n$ bad players. (In this case, the Byzantine Agreement in S_j may not succeed)
  In this case, since P_i accepts S_j's disqualification only if $S_j \subseteq \mathsf{view}_i(C)$, it follows that for every good player $P_{i'}$, $S_j \subseteq \overline{\mathsf{view}}_{i'}(C)$. Thus, $\overline{\mathsf{view}}_{i'}(C)$ consists of more than $\frac{1}{4}a\log n$ bad players. There are at least $\frac{3}{4}a\log n$ honest players in C and all of them are in $\overline{\mathsf{view}}_{i'}(C)$.

  This means that $|\bigcap_{P_k \in \mathsf{goodset}(C)} \overline{\mathsf{view}}_k(C)| > \frac{3}{4}a\log n + \frac{1}{4}a\log n = a\log n$. Thus the good set will decide to disqualify C, every honest player will be notified by the good set, and therefore, every honest player will disqualify C.

- Case 2: P_i set $\mathsf{disq}_{P_i}(C) = 1$ due to receiving messages from some set $S'_j \subseteq S_j$ such that S_j consists of at most $\frac{1}{4}a\log n$ bad players.
  Then, since at most $\frac{1}{3}$ fraction of S_j is corrupt, the Byzantine agreement protocol in S_j would succeed and all honest players $P_k \in S_j$ have the same value for $\mathsf{disq}_{P_k,S_j}(C)$. Since $|S'_j| \geq \frac{1}{2}a\log n$, a fortiori, there is at least one honest player in S'_j, who supported the disqualification of C. Because of agreement, all good players in S_j support the disqualification of C, and this decision will be sent to all the honest players.

To prove the second assertion, we observe that for any committee C that is *not disqualified*, if a set $S_j \subseteq \mathsf{view}_{P_k}(C)$ for some honest player P_k, then S_j has *more than two-thirds fraction* of honest players (Otherwise, C would have been disqualified by the good set using an argument similar to Case 1 above). Therefore the BA in S_j will succeed, and S_j will present a uniform view of $\mathsf{composition}_{P_k,S_j}$ to all honest players P_k which, a fortiori, means that all honest players P_k compute the same value for $\mathsf{final\_comp}_{P_k}(C)$. □

Let $\mathcal{H}$ denote the set of all honest players. Define $\mathbb{C}_i \stackrel{def}{=} \bigcup_{k\in\mathcal{H}} \overline{\mathsf{view}}_k(C_i)$. Intuitively, $\mathbb{C}_i$ is the set of all players that are *adopted* by *some* honest player into the committee C_i. Our goal is to prove that there exists a committee C_i such that $|\mathbb{C}_i|$ is small. Lemma 7 and Corollary 8 show that there indeed exists such a committee. Lemma 9 (the "survivor lemma") shows that such a committee cannot be disqualified.

Lemma 7. *At the end of Stage* 1*, the following holds: For any two committees* $C_i \neq C_j$*,* $\mathbb{C}_i \cap \mathbb{C}_j = \phi$*.*

Proof. If a player $P \in \overline{\mathsf{view}}_k(C_i)$ for some honest player P_k and committee C_i, then $P \notin \overline{\mathsf{view}}_{k'}(C_{i'})$ for any honest player $P_{k'}$ and committee $C_{i'}, i' \neq i$ (due to the consistency property of gradecast). The statement of the lemma follows immediately from this observation. ☐

Corollary 8. *There exists at least one committee* C_i *such that* $|\mathbb{C}_i| \leq a \log n$*.*

Proof. Lemma 7 says that the $\mathbb{C}_i$ for the different i's are disjoint. Moreover, $|\bigcup_{i \in [\frac{n}{a \log n}]} \mathbb{C}_i| \leq n$. It immediately follows that for at least one i, $|\mathbb{C}_i| \leq a \log n$. ☐

Lemma 9 (Survivor Lemma). *At the end of Stage* 2*, there is at least one committee that is not disqualified.*

Proof. Consider the committee C_i, such that $|\mathbb{C}_i| \leq a \log n$, guaranteed by corollary 8. We claim that C_i cannot be disqualified. Suppose not. Then, there exists a set $S_j \subseteq \mathsf{view}_k(C_i)$ for some honest player P_k such that S_j disqualifies C_i.

We first note that any such set S_j has at least $\frac{1}{2} a \log n$ good players. Indeed, if S_j had more than $\frac{1}{4} a \log n$ bad players, and a honest P_k accepts S_j's disqualification of C_i, the good set will disqualify C_i too (the argument is the same as that in Case 1 of Lemma 6). And this would mean that $\mathbb{C}_i$ is too large, contrary to assumption. Thus, S_j has more than $\frac{2}{3}$ fraction of good players, and therefore, Byzantine Agreement in S_j succeeds.

The good players in S_j compute $\bigcap_{P_l \in S_j} \overline{\mathsf{view}}_l(C_i)$ to decide if the committee is too big. But, note that

$$\bigcap_{P_l \in S_j} \overline{\mathsf{view}}_l(C_i) \subseteq \bigcup_{P_l \in \mathcal{H}} \overline{\mathsf{view}}_l(C_i) \stackrel{def}{=} \mathbb{C}_i$$

By the choice of C_i, $|\mathbb{C}_i| \leq a \log n$, and therefore the good players in S_j do not disqualify C_i, contrary to assumption. ☐

Define $\mathbb{D}_i \stackrel{def}{=} \bigcup_{k \in \mathcal{H}} \mathsf{view}_k(C_i)$. Intuitively, $\mathbb{D}_i$ is the set of all players that are *accepted* by *some* honest player into the committee C_i (Note the difference between the definitions of $\mathbb{D}_i$ and $\mathbb{C}_i$). Lemma 10 is a kind of partial converse to Lemma 9, in the following sense: Lemma 9 says that, if $|\mathbb{C}_i|$ is small, C_i survives. Lemma 10, on the other hand, says that if C_i survives, then $|\mathbb{D}_i|$ is small.

Lemma 10 (Partial Converse of Survivor Lemma). *If committee* C_i *is not disqualified at the end of Stage 2, then* $|\mathbb{D}_i| \leq a \log n$*.*

Proof. C_i was not disqualified implies, in particular, that the good set did not disqualify C_i. Suppose, for contradiction, that $|\mathbb{D}_i| > a \log n$. Any player P in $\mathbb{D}_i$ is in $\mathsf{view}_k(C_i)$ for all players P_k in the good set (by Lemma 3). Thus, $\bigcap_{(P_k \in \mathsf{good\ set})} \mathsf{view}_k(C_i) \supseteq \mathbb{D}$, and thus of size more than $a \log n$. Thus, the good set will disqualify C_i, contrary to assumption. ☐

Finally, we show that the composition agreed for the chosen committee consists of more than $\frac{2}{3}$ fraction of good players (Note the degradation from $\frac{3}{4}$).

Lemma 11 (**"The Chosen One is Good" Lemma**). *At the end of Stage 2, the final composition that the players agree for the chosen committee C_i ($\mathsf{final\_comp}_{P_k}(C_i)$) contains at most a $\frac{1}{3}$ fraction of bad players.*

Proof. As in Lemma 9, any set S_j whose decision matters has less than $\frac{1}{4}a \log n$ bad players.

The composition for C_i advertised by the good set contains all the $\frac{3}{4}a \log n$ good players in C_i. Suppose this is not the final composition chosen. This means, there exists another set S_j that advertised a larger composition for C_i. i.e, a composition of size $s > \frac{3}{4}a \log n$.

Now, we upper bound the number of bad players in any such advertised (and accepted) composition. The players in S_j agree on $\bigcap_{P_k \in S_j} \mathsf{view}_k(C_i)$ as the composition. For a good player $P_k \in S_j$, $\mathsf{view}_k(C_i) \subseteq \bigcup_{P_k \in \mathcal{H}} \mathsf{view}_k(C_i)$, which is smaller than $a \log n$ (by Lemma 10). Thus, it contains less than $\frac{1}{4}a \log n$ bad players. Since we take the intersection of all such views, the composition computed by S_j has at most $\frac{1}{4}a \log n$. But, by the previous paragraph, it has size $s > \frac{3}{4}a \log n$. Thus it has at most $\frac{1}{3}$ fraction of bad players. □

Lemma 12 (**Round Complexity**). $\mathsf{BAP}_\epsilon(n)$ *achieves Byzantine agreement among n players tolerating $t < (\frac{1}{4} - \epsilon)n$ faults, and runs in expected time $O(\frac{\log n}{\epsilon^2})$.*

Proof. Stage 1 takes $O(1)$ rounds. Stage 2 consists of a number of parallel executions all of which terminate by $a \log n + 1$ rounds. Stage 3 uses an $O(\log^* n)$-round leader election protocol in which each step consists of all the $\log n$ players broadcasting a string of length $\log^{O(1)} n$ to every other player. Such a broadcast is simulated with the Chor-Coan protocol [9], which is a Byzantine Agreement protocol to agree on *a single bit*. Thus, we are executing $\log^{O(1)} n$ instances of the Chor-Coan protocol, in parallel. We use the following fact about the Chor-Coan protocol: when executed among n players out of which at most t are faulty, the probability that the protocol does not terminate in $\frac{2t}{\log n} + 2\sqrt{n}$ rounds is at most $e^{-\sqrt{n}}$. Thus, the probability that one invocation of the Chor-Coan BA (among the $a \log n$ players) does not terminate in $\frac{a \log n}{\log(a \log n)} + 2\sqrt{\log n}$ rounds is at most $e^{-\sqrt{\log n}} \ll \frac{1}{\log^{O(1)} n}$. Thus, with probability at least $1 - \frac{1}{\log^{O(1)} n}$, all the parallel invocations terminate in $O(\frac{a \log n}{\log \log n})$ rounds. Thus, Stage 3 elects a good leader with a constant probability, and takes $\frac{\log n \log^* n}{\epsilon^2 \log \log n} = O(\frac{\log n}{\epsilon^2})$ rounds. Thus, BA_ϵ runs in expected $O(\frac{\log n}{\epsilon^2})$ rounds. □

Lets put together these lemmas to prove the main theorem. By Lemmas 9 and 11, at least one committee is not disqualified, and the final composition agreed on for the chosen committee, has more than $\frac{2}{3}$ fraction of good players (which is necessary for the BA in Stage 3). Thus, in $O(\frac{\log n}{\epsilon^2})$ rounds, we get a coin with bounded bias. By [11] (sketched in Appendix B), this gives BA with $O(\frac{\log n}{\epsilon^2})$ expected rounds.

When The Number of Faults is Small

When the number of faults $t = O(\frac{n}{\log^2 n})$, it is possible to construct an $O(1)$-round BA protocol. The protocol is based on the following result of Ajtai and Linial [1], who show the existence of a 1-round collective coin-flipping protocol (in other words, a Boolean function) whose output cannot be influenced by any coalition of less than $\frac{n}{\log^2 n}$ players. For a string $\mathbf{x} \in \{0,1\}^n$ and $B \subseteq [n]$, let $\mathbf{x}_B$ denote the $|B|$-bit string formed by projecting $\mathbf{x}$ onto indices in B. For any $B \subseteq [n]$, we can thus write $\mathbf{x}$ as a pair $(\mathbf{x}_B, \mathbf{x}_{[n]\setminus B})$.

Theorem 13 (Ajtai-Linial [1]). *There are constants $c, \epsilon > 0$ and a family of Boolean functions $\{f_n\}_{n=1}^{\infty}$ where $f_n : \{0,1\}^n \to \{0,1\}$ such that, for any set of variables $B \subseteq [n]$ of size at most $\frac{cn}{\log^2 n}$, $\epsilon \leq \Pr_{\mathbf{x}_B \in_U \{0,1\}^{|B|}}[\exists \mathbf{x}_{[n]\setminus B} \in \{0,1\}^{n-|B|}$ such that $f(\mathbf{x}_B, \mathbf{x}_{[n]\setminus B}) = 0] \leq 1 - \epsilon$.*

This theorem directly gives a way to get a common coin with bounded bias, when $t = O(\frac{n}{\log^2 n})$, and thus an $O(1)$ expected rounds BA.

4 Conclusion and Future Work

In this paper, we construct a Byzantine Agreement protocol that tolerates $t < (\frac{1}{4} - \epsilon)n$ faults, and runs for $O(\log n)$ rounds. Many interesting questions remain.

1. **Achieving Polynomial Communication:** Our protocol, implemented in a straight-forward way, has a quasi-polynomial communication complexity (the total number of bits sent by the good players is $n^{O(\log n)}$). A modified version of our protocol can be run with an optimized way of sending messages so that the total communication is polynomial in n. We omit the details.

2. **Tolerating More Faults:** Our protocol tolerates $t < (\frac{1}{4} - \epsilon)n$ faulty players (for any constant $\epsilon > 0$). It seems reasonable to conjecture that one can get $t < (\frac{1}{3} - \epsilon)n$.

3. **Better Round Complexity:** The Kahn-Kalai-Linial result [18] shows that if our source of randomness for the coin flipping protocol is just one unbiased coin flip from each player, then there is always a group of size $O(\frac{n}{\log n})$ that has high probability of setting the value of the coin. Extending the KKL lower bound to general sources of randomness is a long standing open problem and for this reason, we cannot even rule out the possibility of an $O(1)$ rounds solution to the BA problem in our setting. Also, a "recursive approach"(a.l.a [22]) fails to reduce the round complexity in our case.

4. $O(\log n)$ **rounds for stronger models :** Recall that our protocol works in a synchronous network, against a non-adaptive Byzantine adversary. The restriction that the adversary be non-adaptive is essential. This difficulty is inherent, since Bar-Joseph and Ben-Or [2] show that if the adversary is adaptive, $\tilde{\Omega}(\sqrt{n})$ rounds are necessary to achieve Byzantine Agreement in a synchronous network. In the case of an asynchronous network, achieving even a polynomial-rounds BA protocol is open. We note that the best known asynchronous BA protocols [3, 7] have exponential expected round-complexity.

In subsequent work [17], we resolve the first and second questions, by constructing a BA protocol that runs in $O(\frac{\log n}{\epsilon^2})$ rounds, tolerates any $t < (\frac{1}{3} - \epsilon)n$ faults and has a communication complexity of $\tilde{O}(n^2)$.

Acknowledgments. The authors wish to thank Danny Dolev, Shafi Goldwasser and Dahlia Malkhi for thoughtful discussions and useful directions. We also thank an anonymous reviewer for an insightful perspective on the result.

References

[1] Miklós Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[2] Ziv BarJoseph and Michael BenOr. A tight lower bound for randomized synchronous consensus. In *PODC*, pages 193–199, 1998.

[3] Michael BenOr. Another advantage of free choice: Completely asynchronous agreement protocols (extended abstract). In *PODC*, pages 27–30, 1983.

[4] Michael BenOr and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *FOCS*, pages 408–416, 1985.

[5] Piotr Berman and Juan A. Garay. Asymptotically optimal distributed consensus (extended abstract). In *ICALP*, pages 80–94, 1989.

[6] Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Optimal early stopping in distributed consensus (extended abstract). In *WDAG*, pages 221–237, 1992.

[7] Gabriel Bracha. An asynchronous [(n-1)/3]-resilient consensus protocol. In *PODC*, pages 154–162, 1984.

[8] Gabriel Bracha. An $O(\log n)$ expected rounds randomized byzantine generals protocol. *J. ACM*, 34(4):910–920, 1987.

[9] Benny Chor and Brian A. Coan. A simple and efficient randomized byzantine agreement algorithm. *IEEE Trans. Software Eng.*, 11(6):531–539, 1985.

[10] Benny Chor and Cynthia Dwork. Randomization in byzantine agreement. *Advances in Computing Research*, 5:443–497, 1989.

[11] Cynthia Dwork, David B. Shmoys, and Larry J. Stockmeyer. Flipping persuasively in constant time. *SIAM J. Comput.*, 19(3):472–499, 1990.

[12] Uriel Feige. Noncryptographic selection protocols. In *FOCS*, pages 142–153, 1999.

[13] Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM J. Comput.*, 26(4):873–933, 1997.

[14] Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Inf. Process. Lett.*, 14(4):183–186, 1982.

[15] Juan A. Garay and Yoram Moses. Fully polynomial byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. *SIAM J. Comput.*, 27(1):247–290, 1998.

[16] Oded Goldreich, Shafi Goldwasser, and Nathan Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput.*, 27(2):506–544, 1998.

[17] Shafi Goldwasser, Elan Pavlov, and Vinod Vaikuntanathan. Better byzantine agreement protocols in the full-information model. *Manuscript*, 2006.

[18] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions. In *Proc. 29-th Annual Symposium on Foundations of Computer Science*, pages 68–80, 1988.

[19] Anna Karlin and Andrew Chi-Chih Yao. Probabilistic lower bounds for byzantine agreement. *Manuscript*, 1986.

[20] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM.*, 27:228–234, 1980.

[21] Michael O. Rabin. Randomized byzantine generals. *FOCS*, pages 403–409, 1983.

[22] Alexander Russell and David Zuckerman. Perfect information leader election in $\log^* n + O(1)$ rounds. *J. Comput. Syst. Sci.*, 63(4):612–626, 2001.