# Communication Complexity of Byzantine Broadcast

Linda Chen

June 2021

**Abstract**

Byzantine Broadcast is a fundamental problem in distributed computing, with communication complexity being an important aspect of Byzantine Broadcast protocols. In Byzantine Broadcast, a designated leader must ensure that all honest users in a distributed system reach a consensus, even in the presence of some dishonest users. Previous works have shown an $O(n^2)$ lower bound on communication complexity, as well as protocols with $O(n^2)$ communication complexity for the honest majority scenario. In this paper, we review the previous work and provide various methods and intuition towards a possible $O(n^3)$ communication complexity lower bound for dishonest majority Byzantine Broadcast.

## 1 Introduction

Byzantine Agreement and Byzantine Broadcast are fundamental problems in distributed computing [3, 4, 8] and are important in creating fault-tolerant distributed systems, which allow systems to continue operating reliably and correctly even when some of its components fail. This problem has various important applications, including in cryptocurrency and blockchain protocols [2, 6, 7].

In the Byzantine Agreement problem, $n$ users are each given an input bit and they must come to an agreement and output the same value, even in the presence of up to $f$ corrupt users. Throughout this paper, we will use $n$ to denote the total number of users in the system and $f$ to denote the maximum number of corrupt users. A variation of the Byzantine Agreement problem is Byzantine Broadcast, where a single selected leader sends a bit $b$ to all $n$ users, and the honest users must agree on this bit.

An important aspect of Byzantine Agreement, which we focus on in this paper, is its communication complexity, or the total number of bits that needs to be exchanged between all users throughout the protocol. Note that given a Byzantine Agreement protocol, it is possible to construct a Byzantine Broadcast protocol with the same communication complexity, by simply adding an initial round in which the leader sends its proposal to all users. Therefore, in this paper we will mainly use Byzantine Broadcast when considering lower bounds on communication complexity to make our lower bound stronger.

Previous works have shown that any Byzantine Agreement or Broadcast protocol requires at least $O(n^2)$ communication complexity under deterministic protocols [5] as well as randomized protocols [1]. Under the honest majority scenario, when $f < n/2$, a previous paper by Momose and Ren [9] also showed that achieving this $O(n^2)$ communication complexity is possible using methods such as threshold signatures or expander graphs. However, these methods only apply to the honest majority scenario of Byzantine Broadcast.

In this paper, we attempt to show an $O(n^3)$ communication complexity lower bound for the dishonest majority scenario, where $f > n/2$. Although we were unable to formally prove this lower bound, we provide various methods and intuition behind the communication complexity of dishonest majority protocols and discuss future work to prove a potential $O(n^3)$ lower bound.

First, in section 2, we provide the problem definition and our high level intuition. In section 3, we review previous work proving the $O(n^2)$ lower bound on communication complexity for Byzantine Broadcast. In section 4, we summarize a previous paper that showed how this $O(n^2)$ communication complexity can be achieved in honest majority protocols. Finally, in section 5, we introduce our methods to prove a potential $O(n^3)$ lower bound for the dishonest majority scenario.

# 2 Preliminaries

## 2.1 Problem Definition

In the Byzantine Broadcast problem, suppose there are $n$ users in the system. One of the users is the designated sender whose identity is known to all other users. At the start of the protocol, the designated sender will receive a bit $b$. All users then interact to learn the sender's bit $b$. At the end of the protocol, each user $i$ outputs a bit $b_i$. We say that the protocol is correct and achieves Byzantine Broadcast if it meets the following conditions:

- Consistency: for any honest users $i$ and $j$, $b_i = b_j$.
- Validity: if the leader is honest, then for any honest user $i$, $b_i = b$.
- Liveness: all honest users will eventually terminate.

## 2.2 Intuition

A Byzantine Broadcast protocol essentially consists of the leader sending its proposal to all users, and the users then exchanging votes on the leader's proposal to all other users, in order to detect if the leader was dishonest and sent equivocating proposals to different users. This part of the protocol, in which all users send a vote to all other users, creates an $O(n^2)$ communication cost.

In addition to exchanging votes on the leader's proposal, all users should also relay the $O(n)$ votes they receive to all other users, meaning that $O(n)$ users send $O(n)$ votes to $O(n)$ users. This results in $O(n^3)$ communication cost. However, under honest majority, it is possible to reduce this communication to $O(n^2)$ by using threshold signatures to combine the $O(n)$ votes into a single signature, or using expander graphs to allow each user to only send to $O(1)$ users. However, these methods do not work for the dishonest majority scenario, and the goal of our research was to show an $O(n^3)$ lower bound in the dishonest majority scenario.

# 3 $O(n^2)$ Communication Complexity Lower Bound

To begin, we explain the previous work by Dolev and Reischuk [5], which proved that in any deterministic Byzantine Broadcast protocol, users must send at least $(1 + f/2)^2$, or $O(n^2)$ messages.

**Theorem 3.1** (from [5]). *If Byzantine Agreement is achieved in a protocol where the number of corrupt users $f < n - 1$, honest users must send at least $(1 + f/2)^2$ messages*

*Proof.* Consider two similar scenarios. In the first scenario, all users are split into two subsets $A$ and $B$, where $|B| = 1 + f/2$. The adversary corrupts all users in $B$, while all users in $A$ remain honest. Users in $B$ are corrupted so that they will not send messages to each other, and will ignore the first $f/2$ messages they receive from users in $A$. Otherwise, they behave honestly to users in $A$. Now, suppose the leader is honest and sends its input bit 0. In this scenario, all honest users in $A$ will still agree and commit on 0.

However, if at most $(f/2)^2$ messages are sent in the protocol, this means that there is at least one user $p \in B$ such that $p$ receives at most $f/2$ messages from users in $A$. Therefore, in a second scenario, the adversary can instead corrupt all users in $B$ except for one user $p$, and all users in $A$ that send messages to $p$. Since $p$ receives messages from at most $f/2$ users, the adversary will be corrupting at most $f$ users. In this second scenario, the honest users in $A$ still receive the same messages as in scenario one, and therefore will still commit on 0. However, the honest user $p$ will receive no messages and thus may not commit on 0, violating consistency.

This contradiction proves that all $1 + f/2$ users in $B$ must each receive at least $1 + f/2$ messages from users in $A$, so the total communication complexity must be at least $(1 + f/2)^2$, or $O(n^2)$. □

This lower bound has also been extended to randomized protocols by a previous work [1], which showed that a randomized Byzantine Broadcast protocol would require at least expected $O(n^2)$ communication complexity under a strongly adaptive adversary, meaning that in round $r$, the adversary can observe the messages sent by an honest user $h$ and adaptively corrupt $h$ in round $r$, including removing the message sent by $h$ in round $r$.

**Theorem 3.2** (from [1]). *If a protocol solves Byzantine Broadcast with probability $\frac{3}{4} + \epsilon$ under a strongly adaptive adversary, then honest nodes will need to send at least expected $(\epsilon f)^2$ messages.*

*Proof.* A more detailed proof can be found in [1]. However, to summarize, the proof for the randomized protocol is similar to the previous proof for the deterministic protocol. As done previously, users are split into two sets $A$ and $B$, where $|B| = f/2$.

Then, consider a first scenario in which the adversary corrupts $B$ such that they do not send messages to each other and ignore the first $f/2$ messages received from $A$. Users in $A$ still act honestly and will output the leader's proposed bit with at least probability $\frac{3}{4} + \epsilon$.

In the second scenario, the adversary corrupt all users in $B$ except for one user $p$, and the first $f/2$ users that try to send messages to $p$. As shown in [1], assuming that honest nodes send less than $(\epsilon f)^2$ messages in expectation, the probability that honest users send less than $\frac{\epsilon}{2}f^2$ messages to $B$, and that a random node $p$ picked from $B$ will receive at most $f/2$ messages, is $1 - 4\epsilon$. In other words, there is a $1 - 4\epsilon$ chance that honest user $p$ will receive no messages. Assume that if $p$ receives no messages, it will output 0 with at most 1/2 probability. (Otherwise, if $p$ outputs 1 with at most 1/2 probability, then we can use a symmetric argument). Then, even if the leader proposes 0, $p$ will not output 0 with probability at least $\frac{1}{2}(1 - 4\epsilon)$.

Also in the second scenario, the honest users in $A$ can not distinguish between the two scenarios and will output 0 with probability at least $\frac{3}{4} + \epsilon$. Now, let $Y_1$ be the event that $p$ does not output 0, and let $Y_2$ be the event that all honest users in $A$ output 0. The probability that users in $A$ output 0 and $p$ outputs 1 will then be

$$\begin{aligned} \Pr[Y_1 \cap Y_2] &= \Pr[Y_1] + \Pr[Y_2] - \Pr[Y_1 \cup Y_1] \\ &> \frac{1}{2}(1 - 4\epsilon) + (\frac{3}{4} + \epsilon) - 1 \\ &= \frac{1}{4} - \epsilon. \end{aligned}$$

Therefore, the protocol violates consistency and does not solve Byzantine Broadcast with $\frac{3}{4} + \epsilon$ probability. This contradiction shows that if honest users send less $(\epsilon f)^2$ messages, or $O(n^2)$, in expectation, Byzantine Broadcast is impossible. □

# 4 $O(n^2)$ Communication Complexity Protocols for Honest Majority

Previous works [9] have constructed Byzantine Broadcast protocols for the honest majority scenario that achieve $O(n^2)$ communication complexity, using two main methods: threshold signatures and expander graphs.

In the Byzantine Broadcast protocols, we want all $O(n)$ users to send messages to all $O(n)$ other users, with each message containing the votes from $O(n)$ users. Therefore, this would result in $O(n^3)$ communication complexity. However, using either threshold signatures or expander graphs, this can be reduced to $O(n^2)$ for the honest majority scenario.

## 4.1 Threshold Signatures

In the first method shown in [9], the communication complexity can be reduced using threshold signatures. Threshold signatures combine $O(n)$ messages into a single $O(1)$ sized message. Therefore, we would have $O(n)$ users sending $O(1)$ sized messages to $O(n)$ users, resulting in a final $O(n^2)$ communication complexity. The specific protocol used and how the threshold signatures work is described in more detail in [9].

However, this method of using threshold signatures only works for the honest majority scenario, where we can require that each threshold signature contains at least $n - f$ signatures, meaning it contains at least one honest user's signature. On the other hand, for the dishonest majority scenario, the threshold signatures would no longer work since the the corrupt users would simply be able to forge threshold signatures.

## 4.2 Expander Graphs

The second method described in the previous work [9] demonstrated how to reduce communication complexity through expander graphs.

**Definition 4.1** (Expander Graph from [9])**.** *Let $\alpha$ and $\beta$ be constants satisfying $0 < \alpha < \beta < 1$. An $(n, \alpha, \beta)$-expander is a graph of $n$ vertices such that, for any set $S$ of $\alpha n$ vertices, the number of neighbors of $S$ is more than $\beta n$.*

Their protocol requires that the number of corrupt users $f \leq (1/2-\epsilon)n$, and utilizes an $(n, 2\epsilon, 1-2\epsilon)$-expander graph with constant degree. As shown in [9], for any positive constant $\epsilon$ and for all $n$, it is possible to construct an expander graph with constant degree. This means that each user only exchanges messages with a constant number of other users. In other words, $O(n)$ users would be sending $O(n)$ sized messages to only $O(1)$ users, for a total $O(n^2)$ communication complexity.

Even though each user is only sending to a constant number of other users, the protocol still ensures that enough users will receive the votes. Specifically, when relaying the votes, at least $2\epsilon n = n - 2f$ users will propagate the votes to more than $(1-2\epsilon)n = 2f$ users, of which at least $f+1$ must be honest. Therefore, these $f+1$ honest users must vote for the same value, guaranteeing the existence of a unique certificate for this value. This process is described in more detail in [9].

However, similar to the use of threshold signatures, this method again requires an honest majority, specifically that $f \leq (1/2-\epsilon)n$. There have not been any $O(n^2)$ communication complexity protocols that work for the dishonest majority scenario.

# 5 Potential $O(n^3)$ Communication Complexity Lower Bound for Dishonest Majority

While $O(n^2)$ communication complexity protocols exist for the honest majority scenario, there remains a gap between the lower bound and current results for dishonest majority protocols. Methods used in the honest majority scenario to reduce communication complexity, such as threshold signature, do not work in the dishonest majority scenario. Therefore, we believe it is necessary in dishonest majority Byzantine Broadcast for $O(n)$ users to send $O(n)$ sized messages to $O(n)$ users.

Our goal in this section is to explore a possible $O(n^3)$ lower bound for deterministic dishonest majority Byzantine Broadcast. We introduce two methods which provide important intuition and progress towards a potential proof. However, we were unable to formally prove this lower bound.

## 5.1 Method 1

In our first method, we modeled the communication in the Byzantine Broadcast protocols using relay graphs, in which a user $A$'s relay graph $G_A$, would represent the users that receive $A$'s vote. Specifically, we denote the set of users $S(i, A)$ to be the users that have received $A$'s vote from user $i$. Figure 1 below shows an example of what $A$'s relay graph may look like.

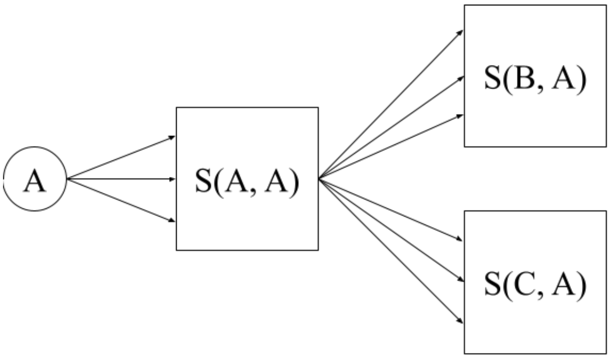

Figure 1: $A$'s relay graph $G_A$

Note that the sum of the number of edges in all users' relay graphs is equal to the total number of messages exchanged, or the communication complexity. In this method, our goal was to show that if we assumed the communication complexity was less than $O(n^3)$, then constructing a correct Byzantine Broadcast protocol would be impossible.

Therefore, we first assume the communication complexity is less than $O(n^3)$. Specifically, suppose the communication complexity were $O(n^{3-\epsilon})$, where $\epsilon$ is some small constant. This means that there exists at least one user $A \in n$ who's relay graph $G_A$ has less than $n^{3-\epsilon}/n = n^{2-\epsilon}$ edges. The average degree in $G_A$, or the number of users each user exchanges messages with on average, would then be $n^{2-\epsilon}/n = n^{1-\epsilon}$.

By corrupting all users that relay $A$'s vote to a user, the adversary can isolate $f/n^{1-\epsilon} = n^{\epsilon}$ users, who will not receive $A$'s vote. Therefore, if the leader was dishonest and sent 0 to all users except an equivocating proposal on 1 to user $A$, these $n^{\epsilon}$ users would not detect the leader's equivocation and would still output 0. However, in this case $A$ would still receive the votes from the $n^{\epsilon}$ users and would not output 1, so the protocol would not be incorrect.

In order to complete the impossibility proof, we would need to show that, under $O(n^{3-\epsilon})$ communication complexity, there would be at least one user $B$ among the $n^{\epsilon}$ users that don't receive $A$'s vote, such that $A$ also does not receive $B$'s vote. In other words, there must be two users $A, B$ who do not receive each others' vote. However, we were unable to complete the proof using this method. Although this method does not prove the $O(n^3)$ lower bound, it still provides important intuition surrounding how the adversary would be able to interfere with how messages are relayed when communication complexity is less than $O(n^3)$.

## 5.2 Method 2

In our second method, we attempted to prove the $O(n^3)$ communication complexity lower bound by setting up the users in a way such that that it would be necessary for users to exchange $O(n^3)$ messages in order for all honest users to output the leader's proposed bit. We first show a structure in which a single user $A$ is exchanging messages with a set of users $B$, where $|B| = 2h$, and a set of users $C$, where $|C| = h - 1$. However, $B$ and $C$ do not exchange messages with each other. This setup is shown in Figure 2 below.

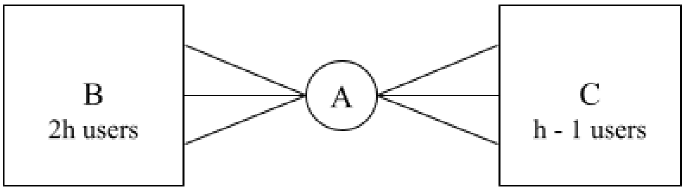

Figure 2: $A$'s relay graph $G_A$

In this case, suppose the leader were to send its proposal to only users in $A$ and $C$. Then, we want to determine the minimum number of messages $A$ must send to users in $B$ in order for users in $B$ to receive all votes. Intuitively, this means that users in $A$ must send $O(n)$ votes from the users in $C$ to $O(n)$ users in $B$, resulting in total $O(n^2)$ communication complexity.

We then modify the setup to the arrangement shown in Figure 3 below, in which we replace user $A$ with a set of $h$ users, $M$.

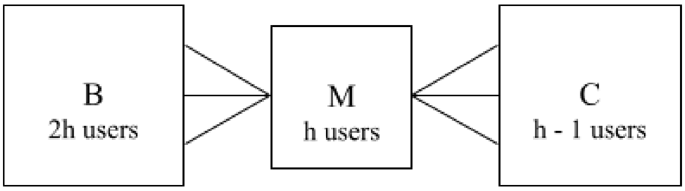

Figure 3: $A$'s relay graph $G_A$

In the case shown in Figure 3, users in $M$ would not be able to tell how many users are in $M$. Each user in $M$ therefore must assume that they are the only user $A$ in $M$, and by the previous logic, would require $O(n^2)$ communication complexity. Since all $O(n)$ users in $M$ operate with $O(n^2)$ communication complexity, this would result in total $O(n^3)$ communication complexity.

However, although this method intuitively makes sense, we have not yet been able to fully prove that $A$ actually requires $O(n^2)$ communication complexity for all users in $B$ to receive all votes. Specifically, we must prove that it is impossible for $A$ to potentially send to only a constant number of users in $B$, and for users in $B$ to then relay messages with each other. Additionally, we must also prove that when replacing $A$ with $M$, it is actually impossible for users in $M$ to determine how many users are in $M$. Future work could involve expanding on this method more to construct a more formal proof.

# 6 Conclusion

This paper reviewed previous results surrounding the communication complexity of Byzantine Broadcast, including the existing $O(n^2)$ lower bound and possible $O(n^2)$ protocols for the honest majority scenario. We then attempted to show an $O(n^3)$ communication complexity lower bound for the dishonest majority scenario. Although we have not yet fully proved this lower bound, we explored various methods that provide important intuition towards a possible proof. Future work can be done to construct a formal proof of this $O(n^3)$ lower bound.

# 7 Acknowledgments

I would like to thank my mentor, Jun Wan, for introducing me to the idea for this project and guiding me throughout the research process. I would also like to thank the MIT PRIMES program for the research opportunity.

# References

[1] Ittai Abraham et al. "Communication Complexity of Byzantine Agreement, Revisited". In: *PODC*. 2019.

[2] Ittai Abraham et al. "Solidus: An Incentive-compatible Cryptocurrency Based on Permissionless Byzantine Consensus". In: *CoRR* abs/1612.02916 (2016).

[3] Atul Adya et al. "FARSITE: Federated, available, and reliable storage for an incompletely trusted environment". In: *ACM SIGOPS Operating Systems Review* 36.SI (2002), pp. 1–14.

[4] Miguel Castro and Barbara Liskov. "Practical Byzantine fault tolerance". In: *OSDI*. Vol. 99. 1999, pp. 173–186.

[5] Danny Dolev and Rüdiger Reischuk. "Bounds on Information Exchange for Byzantine Agreement". In: *J. ACM* 32.1 (Jan. 1985), pp. 191–204. ISSN: 0004-5411. DOI: 10.1145/2455.214112. URL: http://doi.acm.org/10.1145/2455.214112.

[6] Ryan Farell. "An analysis of the cryptocurrency industry". In: (2015).

[7] Yossi Gilad et al. "Algorand: Scaling byzantine agreements for cryptocurrencies". In: *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM. 2017, pp. 51–68.

[8] Leslie Lamport. "The part-time parliament". In: *ACM Transactions on Computer Systems (TOCS)* 16.2 (1998), pp. 133–169.

[9] Ren Momose and Ling Ren. "Optimal Communication Complexity of Authenticated Byzantine Agreement". In: (2021).