Good-Case and Bad-Case Latency of Unauthenticated Byzantine Broadcast: A Complete Categorization

Ittai Abraham
VMware Research, Herzliya, Israel

Ling Ren
University of Illinois at Urbana-Champaign, IL, USA

Zhuolun Xiang
University of Illinois at Urbana-Champaign, IL, USA

Abstract

This paper studies the *good-case latency* of *unauthenticated* Byzantine fault-tolerant broadcast, which measures the time it takes for all non-faulty parties to commit given a non-faulty broadcaster. For both asynchrony and synchrony, we show that $n \geq 4f$ is the tight resilience threshold that separates good-case 2 rounds and 3 rounds. For asynchronous Byzantine reliable broadcast (BRB), we also investigate the *bad-case latency* for all non-faulty parties to commit when the broadcaster is faulty but some non-faulty party commits. We provide matching upper and lower bounds on the resilience threshold of bad-case latency for BRB protocols with optimal good-case latency of 2 rounds. In particular, we show 2 impossibility results and propose 4 asynchronous BRB protocols.

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

Keywords and phrases Byzantine broadcast, asynchrony, synchrony, latency, good-case, optimal

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2021.5

Acknowledgements The authors would like to thank Kartik Nayak for helpful discussions related to the paper.

1 Introduction

Byzantine fault-tolerant broadcast is a fundamental primitive in distributed systems, where a designated broadcaster sends its value to all parties, such that all non-faulty parties commit on the same value despite arbitrary deviation from Byzantine parties. Moreover, if the broadcaster is non-faulty, then all honest parties are required to commit the same value as the broadcaster's input. Byzantine broadcast (BB) requires all non-faulty parties to eventually commit, while Byzantine reliable broadcast (BRB) relaxes the condition to only require termination when the broadcaster is honest or if some non-faulty party terminates. When the network is asynchronous, meaning the message delays are unbounded, it is well-known that BB is unsolvable with even a single fault. On the other hand, BRB is solvable under asynchrony as long as there are $n \geq 3f + 1$ parties.

Recent work of Abraham et al. [4] investigates the notion of good-case latency of Byzantine fault-tolerant broadcast, which is the time for all honest parties to commit given that the broadcaster is honest. Theoretically, the good-case latency is a natural and interesting metric that has not been formally studied by the literature until recently; Practically, for applications like leader-based Byzantine fault-tolerant state machine replication (BFT SMR), the good-case latency study answers the fundamental question of how fast can leader-based

25th International Conference on Principles of Distributed Systems (OPODIS 2021).
Editors: Quentin Bramas, Vincent Gramoli, and Alessia Milani; Article No. 5; pp. 5:1–5:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Upper and lower bounds for good-case latency of *unauthenticated* Byzantine fault-tolerant broadcast.

Problem	Timing Model	Resilience	Lower Bound	Upper Bound
BRB	Asynchrony	$n \geq 4f$	2 rounds [4]	**2 rounds** (Thm 9)
		$3f+1 \leq n \leq 4f-1$	**3 rounds** (Thm 10)	3 rounds [5]
BB	Synchrony	$n \geq 4f$	2δ [4, 9]	$\mathbf{2\delta}$ (Thm 18)
		$3f+1 \leq n \leq 4f-1$	$\mathbf{3\delta}$ (Thm 17)	3δ (Thm 19) [5]

Table 2 Comparison of our results and previous results of asynchronous *unauthenticated* Byzantine reliable broadcast.

Result	Resilience	Good-case	Bad-case	Comm. cost	Reference
Bracha	$n \geq 3f+1$	3 rounds	4 rounds	$O(n^2)$	[5]
Imbs and Raynal	$n \geq 5f+1$	2 rounds	3 rounds	$O(n^2)$	[12]
Impossibility of $(2,2)$	$f \geq 2$	2 rounds	2 rounds	–	Thm 11
F1-BRB	$n \geq 4f, f=1$	2 rounds	2 rounds	$O(n^2)$	Thm 12
Impossibility of $(2,3)$	$n \leq 5f-2, f \geq 3$	2 rounds	3 rounds	–	Thm 13
F2-BRB	$n \geq 4f, f=2$	2 rounds	3 rounds	$O(n^3)$	Thm 15
$(2,4)$-BRB	$n \geq 4f$	2 rounds	4 rounds	$O(n^2)$	Thm 9
$(2,3)$-BRB	$n \geq 5f-1$	2 rounds	3 rounds	$O(n^2)$	Thm 16

BFT SMR commit decisions during the steady state when the leader is non-faulty. Moreover, for asynchronous Byzantine reliable broadcast, good-case latency is particularly important since BRB may not terminate under a Byzantine leader.

The work of Abraham et al. [4] reveals a surprisingly rich structure in the good-case latency tight bounds for *authenticated* Byzantine broadcast, where digital signatures are used and the adversary is assumed to be computationally bounded. In this work, we study the good-case latency and bad-case latency of *unauthenticated* Byzantine fault-tolerant broadcast. Our results are summarized in Table 1 and 2.

Complete categorization for good-case latency under asynchrony and synchrony. Under asynchrony when the message delays are unbounded, we show that $n \geq 4f$ is the tight resilience threshold that separates good-case latency of 2 rounds and 3 rounds. For $n \geq 4f$, [4] shows a 2-round lower bound, and we present a protocol with good-case latency of 2 rounds. For $3f+1 \leq n \leq 4f-1$, Bracha's reliable broadcast [5] has good-case latency of 3 rounds, and we prove a matching 3-round lower bound.

▶ **Theorem 1** (Informal; tight bounds on good-case latency in asynchrony). *For unauthenticated Byzantine reliable broadcast with f Byzantine parties under asynchrony, in the good-case:*

1. *2 rounds are necessary and sufficient if $n \geq 4f$ (Section 3.1), and*
2. *3 rounds are necessary and sufficient if $3f+1 \leq n < 4f$ (Section 3.2).*

The above asynchronous good-case latency bounds also imply similar results for good-case latency of BB and BRB under synchrony as well. Let δ denote the actual message delay bound during the execution (see Section 5 for details). For $n \geq 3f+1$, [4] shows a 2δ lower bound (also implied by the early-stopping results [9]), and we present a synchronous BB protocol with good-case latency of 2δ under $n \geq 4f$, inspired by our 2-round asynchronous

BRB protocol. For $3f + 1 \le n \le 4f - 1$, we show a synchronous BB protocol that has good-case latency of 3δ inspired by Bracha's reliable broadcast [5], and the aforementioned 3δ lower bound also applies to synchrony.

▶ **Theorem 2** (Informal; tight bounds on good-case latency in synchrony)**.** *For unauthenticated Byzantine broadcast and Byzantine reliable broadcast with f Byzantine parties under synchrony (message delay bounded by δ), in the good-case:*

1. *2δ are necessary and sufficient if $n \ge 4f$ (Section 5), and*
2. *3δ are necessary and sufficient if $3f + 1 \le n < 4f$ (Section 5).*

Complete categorization for bad-case latency of asynchronous Byzantine reliable broadcast. In addition to the good-case commit path, asynchronous BRB protocols usually have a second commit path to ensure all honest parties eventually commit, when the Byzantine broadcaster and Byzantine parties deliberately make only a few honest parties commit in the good-case commit path. We use *bad-case latency* to denote the latency of such second commit path, and say a BRB protocol is (R_g, R_b)-round if it has good-case latency of R_g rounds and bad-case latency of R_b rounds. For instance, Bracha's reliable broadcast [5] is $(3, 4)$-round.

We provide a complete categorization of the threshold resilience for BRB with good-case latency of 2. We show two lower bound results on the resilience threshold: for $(2, 2)$ and for $(2, 3)$. We also show 4 protocols with matching resilience bounds: these protocols have the optimal good-case latency of 2 rounds, but with different trade-offs in resilience and bad-case latency, matching the lower bound results. As summarized in Table 2, prior upper bound results include Bracha's $(3, 4)$-round BRB for $n \ge 3f + 1$, and the $(2, 3)$-round BRB for $n \ge 5f + 1$ by Imbs and Raynal [12].

- First, we show it is impossible to achieve $(2, 2)$-round BRB, except for the special case of $f = 1$ where we propose a protocol F1-BRB that has $(2, 2)$-round and optimal resilience $n \ge 4f$.
- Next, we show another impossibility result stating that no BRB protocol can achieve $(2, 3)$-round under $n \le 5f - 2$ for $f \ge 3$. That is, for $f \ge 3$, no BRB protocol can have optimality in all three metrics: good-case latency, bad-case latency and resilience. For the special case of $f = 2$, we propose a protocol F2-BRB that has $(2, 3)$-round and optimal resilience $n \ge 4f$. For the general case of $f \ge 3$, we have two protocols – a protocol named $(2, 4)$-BRB under $n \ge 4f$ that has $(2, 4)$-round, and a protocol named $(2, 3)$-BRB which improves the resilience of Imbs and Raynal [12] from $n \ge 5f + 1$ to $n \ge 5f - 1$ while keeping the protocol $(2, 3)$-round. Both $(2, 4)$-BRB and $(2, 3)$-BRB have tight resilience and latencies due to the impossibility result.

2 Preliminaries

Model of execution. We define a protocol for a set of n parties, among which at most f are Byzantine faulty and can behave arbitrarily and has unbounded computational power. If a party remains non-faulty for the entire protocol execution, we call the party honest. During an execution E of a protocol, parties perform sequences of events, including *send, receive/deliver, local computation.*

In this paper, we investigate results for deterministic unauthenticated protocols. If the protocol is deterministic, for any two executions, if an honest party has the same initial state and receives the same set of messages at the same corresponding time points (by its local clock), the honest party cannot distinguish two executions. We will use the standard indistinguishability argument to prove lower bounds.

We consider both synchronous and asynchronous network models. Under synchrony, any message between two honest parties will be delivered within δ time during the execution. More details about the synchrony model assumption is deferred to Section 5. Under asynchrony, the adversary can control the message delay of any message to be an arbitrary non-negative value. We assume all-to-all, reliable and authenticated communication channels, such that the adversary cannot fake, modify or drop the messages sent by honest parties.

Byzantine broadcast variants. We investigate two standard variants of Byzantine broadcast problem for synchrony and asynchrony.

▶ **Definition 3** (Byzantine Broadcast (BB)). *A Byzantine broadcast protocol must satisfy the following properties.*

- *Agreement. If two honest parties commit values v and v' respectively, then $v = v'$.*
- *Validity. If the designated broadcaster is honest, then all honest parties commit the broadcaster's value and terminate.*
- *Termination. All honest parties commit and terminate.*

▶ **Definition 4** (Byzantine Reliable Broadcast (BRB)). *A Byzantine reliable broadcast protocol must satisfy the following properties.*

- *Agreement. Same as above.*
- *Validity. Same as above.*
- *Termination. If an honest party commits a value and terminates, then all honest parties commit a value and terminate.*

We will also use *Byzantine agreement* as a primitive to simplify the construction of our BB protocols under synchrony in Section 5. The Byzantine agreement gives each party an input, and its validity requires that if all honest parties have the same input value, then all honest parties commit that value.

Good-case latency of broadcast. Depending on the network model, the measurement of latency is different. Under synchrony, we can measure the latency using the physical clock time.

▶ **Definition 5** (Good-case Latency under Synchrony [4]). *A Byzantine broadcast (or Byzantine reliable broadcast) protocol has good-case latency of T under synchrony, if all honest parties commit within time T since the broadcaster starts the protocol (over all executions and adversarial strategies), given the designated broadcaster is honest.*

Under asynchrony, the network delay is unbounded. To measure the latency of asynchronous protocols, we use the natural notion of *asynchronous rounds* from the literature [6], where a protocol runs in R asynchronous rounds if its running time is at most R times the maximum message delay between honest parties during the execution.

▶ **Definition 6** (Good-case Latency under Asynchrony [4]). *A Byzantine reliable broadcast protocol has good-case latency of R rounds under asynchrony, if all honest parties commit within asynchronous round R (over all executions and adversarial strategies), given the designated broadcaster is honest.*

When the broadcaster is dishonest, Byzantine broadcast will have worst-case latency of $f + 1$ rounds [11], and for Byzantine reliable broadcast by definition it does not guarantee termination (the broadcaster can just remain silent). Therefore, the notion of good-case latency is the natural metric to measure the latency performance of reliable broadcast. Another important latency metric for reliable broadcast is to measure how fast can all honest parties commit, once an honest party commit. We formally define it as the bad-case latency as below.

▶ **Definition 7** (Bad-case Latency under Asynchrony). *A Byzantine reliable broadcast protocol has bad-case latency of $R' = R + R_{ex}$ rounds under asynchrony, if all honest parties commit within R_{ex} asynchronous round after an honest party commits (over all executions and adversarial strategies), and the good-case latency of the protocol is R.*

We will use the notation (R_g, R_b)-round BRB to denote an authenticated Byzantine reliable broadcast protocol that has good-case latency or R_g rounds and bad-case latency of R_b rounds. For instance, the classic Bracha reliable broadcast [5] has a good-case latency of 3 rounds and a bad-case latency of 4 rounds ($R_{ex} = 1$), under $n \geq 3f + 1$ parties; it is thus a $(3, 4)$-round BRB.

3 Good-case Latency of Asynchronous Byzantine Reliable Broadcast

Under asynchrony, Byzantine reliable broadcast is solvable if and only if $n \geq 3f+1$. We show the *tight* lower and upper bound on the good-case latency of asynchronous unauthenticated BRB is 2 rounds when $n \geq 4f$, and 3 rounds when $3f + 1 \leq n \leq 4f - 1$.

3.1 2-round Unauthenticated BRB under $n \geq 4f$

We show the tightness of the bound by presenting a 2-round unauthenticated BRB protocol, which has good-case latency of 2 rounds and bad-case latency of 4 rounds with $n \geq 4f$ parties, as presented in Figure 1.

In the protocol, in the first round the broadcaster sends its proposal to all parties. Then in the second round, all parties send an `ack` for the first proposal received. Parties commit in 2 rounds when receiving $n - f - 1$ `ack` for the same value from distinct parties other than the broadcaster, which will happen when the broadcaster is honest. To ensure termination, the protocol has another 4-round commit path, to guarantee that all honest parties will commit even if the Byzantine parties deliberately make only a few honest parties commit in round 2. The 4-round commit path consists of a Bracha-style reliable broadcast, where the parties send `vote-1` and `vote-2` messages upon receiving enough messages as specified in Step 4. Finally, when receiving enough `vote-2` messages, party can also commit in round 4.

▶ **Lemma 8.** *If an honest party commits v at Step 3, then no honest party will send `vote-1` or `vote-2` for any other value $v' \neq v$.*

Proof. Since the honest party commit v at Step 3, it receives $n - f - 1$ `ack` messages for v from distinct non-broadcaster parties. If the broadcaster is honest, then no honest party will send `vote-1` or `vote-2` message for v' since there are at most f Byzantine parties. If the broadcaster is Byzantine, and suppose there are t Byzantine parties, then there are at most $t - 1$ Byzantine parties among all non-broadcaster parties, and there must be at least $(n - f - 1) - (t - 1) = n - f - t$ honest parties sending `ack` for v. Suppose an honest party receives $n - 2f$ `ack` messages for v' from distinct non-broadcaster parties, then there must be at least $(n - 2f) - (t - 1) = n - 2f - t + 1$ honest parties sending `ack` for v'. Since there are only $n - t$ honest parties, there must be at least $(n - f - t) + (n - 2f - t + 1) - (n - t) = n - 3f - t + 1 \geq n - 4f + 1 \geq 1$ honest party that sends `ack` for both v and v', contradiction. Hence no honest party can receive $n - 2f$

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v\rangle$ to all parties.
2. **Ack.** When receiving the first proposal $\langle \texttt{propose}, v\rangle$ from the broadcaster, a party sends an $\langle \texttt{ack}, v\rangle$ message to all parties.
3. **2-round Commit.** When receiving $\langle \texttt{ack}, v\rangle$ from $n-f-1$ distinct non-broadcaster parties, a party commits v, sends $\langle \texttt{vote-1}, v\rangle$ and $\langle \texttt{vote-2}, v\rangle$ to all parties, and terminates.
4. **Vote.**
   - When receiving $\langle \texttt{ack}, v\rangle$ from $n-2f$ distinct non-broadcaster parties, a party sends a $\langle \texttt{vote-1}, v\rangle$ message to all parties, if it has not already sent `vote-1` for any value.
   - When receiving $\langle \texttt{vote-1}, v\rangle$ from $n-f-1$ distinct non-broadcaster parties, a party sends a $\langle \texttt{vote-2}, v\rangle$ message to all parties, if it has not already sent `vote-2` for any value.
   - When receiving $\langle \texttt{vote-2}, v\rangle$ from $f+1$ distinct non-broadcaster parties, a party sends a $\langle \texttt{vote-2}, v\rangle$ message to all parties, if it has not already sent `vote-2` for any value.
5. **4-round Commit.** When receiving $\langle \texttt{vote-2}, v\rangle$ from $n-f-1$ distinct non-broadcaster parties, a party commits v and terminates.

Figure 1 $(2,4)$-round BRB protocol under $n \geq 4f$.

`ack` messages for v'. Moreover, since the thresholds in Step 4 are larger than the number of Byzantine parties, i.e., $n-f-1 \geq 3f-1 > f$ and $f+1 > f$, no honest party will send `vote-1` or `vote-2` for $v' \neq v$. ◀

▶ **Theorem 9.** *The protocol in Figure 1 solves Byzantine reliable broadcast under asynchrony with optimal resilience $n \geq 4f$ and optimal good-case latency of 2 rounds, and has bad-case latency of 4 rounds.*

Proof.

Validity and Good-case Latency. If the broadcaster is honest, it sends the same proposal of value v to all parties. Then all $n-f-1$ non-broadcaster honest parties will multicast the `ack` message for v. The Byzantine parties cannot make any honest party to send `vote-1`, `vote-2`, for any other value $v' \neq v$ since f is below any threshold specified in the protocol. All honest parties will eventually commit v after receiving $n-f-1$ `ack` messages at Step 3 and terminate. The good-case latency is 2 rounds, including broadcaster sending the proposal and all parties sending `ack` message.

Agreement. If the broadcaster is honest, by validity all honest parties will commit the same value. Now consider when the broadcaster is Byzantine, there are at most $f-1$ Byzantine parties among non-broadcasters.

If any two honest parties commit different values at Step 3, then there must be at least $n-f-1-(f-1) = n-2f \geq 2f$ honest parties sending `ack` for each of these different values. It is impossible by quorum intersection since there are only $3f$ honest parties. Similarly, no two honest parties can commit different values at Step 5.

Now we show that if an honest party $h1$ commits v at Step 3 and another honest party $h2$ commits v' at Step 5, then it must be $v = v'$. Suppose $h1$ commits v at Step 3, then by Lemma 8, no honest party will send `vote-1` or `vote-2` for $v' \neq v$, and thus not enough `vote-2` for any $v' \neq v$ to be committed at Step 5. Suppose $h2$ commit v' at Step 5, then $h2$ receives at least $n - f - 1 - (f - 1) = n - 2f \geq 2f$ `vote-2` messages for v' from honest parties. By the contrapositive of Lemma 8, no honest party commits $v \neq v'$ at Step 3.

Termination and Bad-case Latency. If the broadcaster is honest, by validity all honest parties will commit the same value. Now consider when the broadcaster is Byzantine, there are at most $f - 1$ Byzantine parties among non-broadcasters.

Suppose that an honest party commits v at Step 3, by Lemma 8, no honest party will send `vote-1` or `vote-2` for any $v' \neq v$. Since there are at least $n - f - 1 - (f - 1) = n - 2f$ non-broadcaster honest parties sending `ack` for v, all honest parties will receive at least $n - 2f$ `ack` for v from non-broadcasters, and thus send `vote-1` for v. Since there are $n - f$ honest non-broadcasters, all honest parties will send `vote-2` for v, and then commit after receiving $n - f - 1$ `vote-2` messages.

Suppose that an honest party commits v at Step 5, then at least $n - f - 1 - (f - 1) = n - 2f \geq f + 1$ honest non-broadcasters send `vote-2` for v. We only need to show that no honest party send `vote-2` for $v' \neq v$, then all honest parties will send `vote-2` for v and thus commit v. Suppose there is an honest party that send `vote-2` for $v' \neq v$, then there exists two sets of `vote-1` messages from $n-f-1$ distinct non-broadcasters for v and v' respectively. Suppose there are $t > 0$ Byzantine parties, then at least $n - f - 1 - (t - 1) \geq n - t - f$ honest parties send `vote-1` for v and v' respectively, which is impossible as there are $n - t < 2(n - t - f)$ honest parties. Therefore no honest party sends `vote-2` for $v' \neq v$, and all honest parties commits v.

It is clear from the protocol that after at most 2 rounds (`vote-1` and `vote-2`) since any honest party commits, all honest parties also commit. Hence the bad-case latency is 4 rounds. ◀

3.2 3-round Lower Bound for Unauthenticated BRB under $n \leq 4f - 1$

▶ **Theorem 10.** *Any unauthenticated Byzantine reliable broadcast protocol under* $3f + 1 \leq n \leq 4f - 1$ *must have a good-case latency of at least* 3 *rounds even under synchrony.*

Proof of Theorem 10. The proof is illustrated in Figure 2. We assume all parties start their protocol at the same time, which strengthens the lower bound result. Under synchrony, any message between all honest parties will be delivered within δ time, and hence each round of the protocol is of δ time. Without loss of generality, we prove the lower bound for $n = 4f - 1$. Suppose there exists a BRB protocol Π that has a good-case latency of 2 round, which means the honest parties can always commit after receiving two rounds of messages but before receiving any message from the third round, if the designated broadcaster is honest. Let party s be the broadcaster, and divide the remaining $n - 1 = 4f - 2$ parties into 4 groups A, B, C, D where $|A| = |D| = f$ and $|B| = |C| = f - 1$. For brevity, we often use A (B, C, D) to refer all the parties in A (B, C, D). Consider the following three executions of Π.

- Execution 1. The broadcaster s is honest and has input 0. Parties in D are Byzantine, they behave honestly according to the protocol except that they pretend to receive from a broadcaster whose input is 1. Since the broadcaster is honest, by validity and good-case latency, parties in A, B, C will commit 0 after receiving two rounds of messages but before receiving any message from the third round.

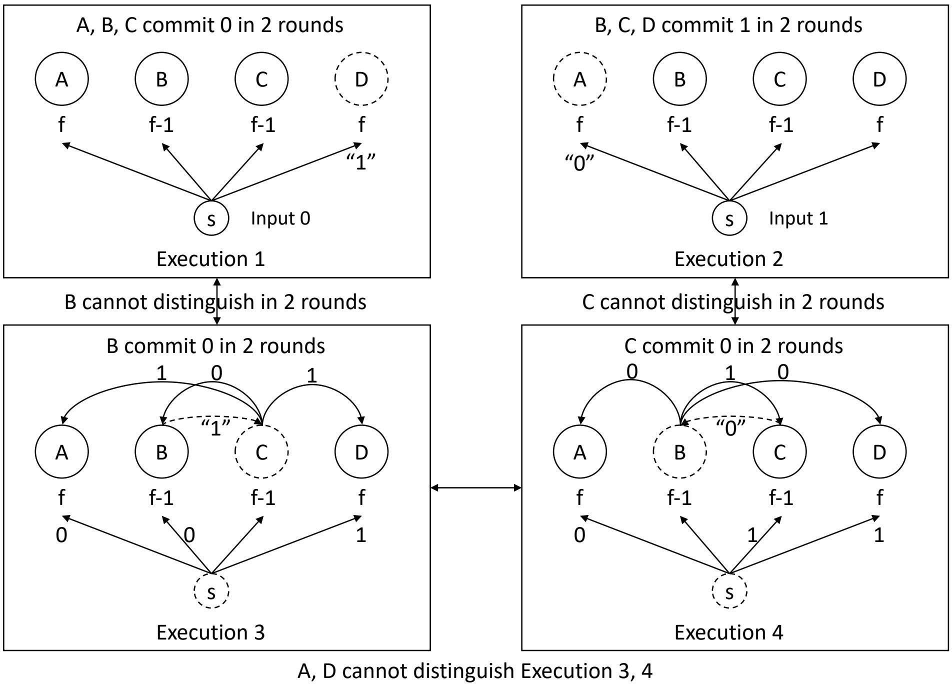
A, B, C commit 0 in 2 rounds
A
B
C
D
f
f-1
f-1
f
"1"
s
Input 0
Execution 1
B, C, D commit 1 in 2 rounds
"0"
Input 1
Execution 2
B cannot distinguish in 2 rounds
C cannot distinguish in 2 rounds
B commit 0 in 2 rounds
1
0
1
"1"
0
0
1
Execution 3
C commit 0 in 2 rounds
0
1
0
"0"
0
1
1
Execution 4
A, D cannot distinguish Execution 3, 4

Figure 2 Unauthenticated BRB Good-case Latency Lower Bound: 3 rounds under $n = 4f - 1$. Dotted circles denote Byzantine parties.

- Execution 2. This execution is a symmetric case of Execution 1. The broadcaster s is honest and has input 1. Parties in A are Byzantine, they behave honestly according to the protocol except that they pretend to receive from a broadcaster whose input is 0. Since the broadcaster is honest, by validity and good-case latency, parties in B, C, D will commit 1 after receiving two rounds of messages but before receiving any message from the third round.
- Execution 3. The broadcaster s and the parties in C are Byzantine. s behaves to A, B identically as in Execution 1 and to D identically as in Execution 2. Parties in C behave to B honestly according to the protocol except that they pretend to receive the same messages from the broadcaster as in Execution 1, and only send messages to B in the first two rounds. Parties in C behave to A, D honestly except that they pretend to receive the same messages from the broadcaster as in Execution 2, and pretend to receive messages from B as in Execution 2 only in the first two rounds.
- Execution 4. This execution is a symmetric case of Execution 3. The broadcaster s and the parties in B are Byzantine. s behaves to A identically as in Execution 1 and to C, D identically as in Execution 2. Parties in B behave to C honestly according to the protocol except that they pretend to receive the same messages from the broadcaster as in Execution 2, and only send messages to C in the first two rounds. Parties in B behave to A, D honestly except that they pretend to receive the same messages from the broadcaster as in Execution 1, and pretend to receive messages from C as in Execution 1 only in the first two rounds.

We show the following indistinguishability and contradiction.

- B cannot distinguish Execution 1 and 3 in the first two rounds, and thus will commit 0 in the end of round 2 in Execution 3. The broadcaster s behaves to B identically in both executions. The messages sent to B in the first round by any non-broadcaster party are identical in Execution 1 and 3, since the first round message only depends on the initial state and all Byzantine parties behave honestly in the first round. For the second round, in Execution 3, since parties in D pretends to B that it receives messages from the broadcaster with input 1, and parties in C pretends to B that it receives the same messages from the broadcaster as in Execution 1, the parties in B observe the same messages in the first two rounds of both executions. Hence, B cannot distinguish Execution 1 and 3 in the first two rounds. Since B commit 0 in the end of round 2 in Execution 1 due to validity and good-case latency, B also commit 0 in the end of round 2 in Execution 3.
- Similarly, C cannot distinguish Execution 2 and 4 in the first two rounds, and thus will commit 1 in the end of round 2 in Execution 4.
- A, D cannot distinguish Execution 3 and 4. Similarly, the messages sent to A, D in the first round are identical in both executions. The broadcaster s behaves to A, B identically in Execution 3 and 4 as in Execution 1, and to C, D identically in Execution 3 and 4 as in Execution 2. In Execution 3, parties in B only receive messages from C in the first two rounds, and Byzantine parties in C pretend to receive messages from a broadcaster whose input is 1. In Execution 4, Byzantine parties in B pretends only receiving two rounds of messages from C. Since the first two rounds of messages only depend on the initial state and the message received from the broadcaster in the first round, parties in B receives the same messages from C. Therefore, A, D receive the same messages from B in both Execution 3 and 4. Similarly, A, D receive the same messages from C in both Execution 3 and 4, and thus cannot distinguish these two executions.

Contradiction. Since parties in B commit 0 in Execution 3, parties in C commit 1 in Execution 4, and parties in A, D cannot distinguish Execution 3 and 4, either agreement or termination of BRB will be violated. Therefore no such protocol Π exists. ◀

4 Bad-case Latency of Asynchronous Byzantine Reliable Broadcast

In this section, we present 2 impossibility results and 4 asynchronous BRB protocols with tight trade-offs between resilience, good-case latency and bad-case latency.

Recall that the classic Bracha reliable broadcast [5] has optimal resilience of $n \geq 3f+1$, non-optimal good-case latency of 3 rounds and bad-case latency of 4 rounds (1 extra round). The 2-round BRB protocol by Imbs and Raynal [12] has non-optimal resilience of $n \geq 5f+1$, optimal good-case latency of 2 rounds and bad-case latency of 3 rounds (1 extra round). Meanwhile, our 2-round BRB protocol from Section 3 has optimal resilience $n \geq 4f$, optimal good-case latency of 2 rounds and bad-case latency of 4 round (2 extra rounds). All protocols above have optimal communication complexity of $O(n^2)$, matching the lower bound [8].

On the other hand, we can show that for any $f > 1$, no asynchronous BRB protocol can achieve both good-case latency of 2 rounds and bad-case latency of 2 rounds (Theorem 11 in Section 4.1). For the special case of $f = 1$, we show it is possible to have a $(2, 2)$-round BRB (Theorem 12).

Therefore, it is interesting to ask:

Under what conditions can BRB achieve optimality in all three metrics – optimal resilience of $n \geq 4f$, optimal good-case latency of 2 rounds and optimal bad-case latency of 3 rounds (1 extra round)?

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v\rangle$ to all parties.
2. **Ack.** When receiving the first proposal $\langle \texttt{propose}, v\rangle$ from the broadcaster, a party sends an `ack` message for v to all parties in the form of $\langle \texttt{ack}, v\rangle$.
3. **2-round Commit.** When receiving $\langle \texttt{ack}, v\rangle$ from $n-2$ distinct non-broadcaster parties, a party commits v and terminates.

Figure 3 $(2,2)$-round BRB Protocol under $n \geq 4f, f = 1$.

We show it is impossible for the general case of $f \geq 3$, by proving that no BRB protocol under $n \leq 5f-2, f \geq 3$ can achieve $(2,3)$-round (Theorem 13). For $f \geq 3$, our BRB (Figure 1) in earlier Section 3.1 has optimal good-case latency of 2 rounds and optimal resilience $n \geq 4f$, but with bad-case latency of 4 rounds. On the other hand, we give a $(2,3)$-round BRB protocol (Figure 5) with tight resilience $n \geq 5f-1$, improving the $n \geq 5f+1$ resilience of Imbs and Raynal [12]. For the special case of $f = 2$, we show it is possible to construct a $(2,3)$-round BRB (Figure 4) with optimal resilience $n \geq 4f$.

4.1 Impossibility of $(2,2)$-round BRB

For the general case of $f \geq 2$, we show any asynchronous BRB protocol cannot achieve $(2,2)$-round. The proof of Theorem 11 is deferred to Appendix A due to space limit.

▶ **Theorem 11.** *Any asynchronous unauthenticated Byzantine reliable broadcast protocol under $f \geq 2$ and has a good-case latency of 2 rounds must have a bad-case latency of at least 3 rounds.*

4.2 $(2,2)$-round BRB Protocol under $n \geq 4f, f = 1$

For the special case of $f = 1$, we can show a simple BRB protocol (Figure 3) that has optimal good-case latency and bad-case latency of 2 rounds, while having optimal resilience $n \geq 4$.

▶ **Theorem 12.** *The protocol in Figure 3 solves Byzantine reliable broadcast under asynchrony with optimal resilience $n \geq 4, f = 1$, optimal good-case latency and bad-case latency of 2 rounds.*

Proof.

Validity and Good-case Latency. If the broadcaster is honest, it sends the same proposal of value v to all parties, and all $n-2 \geq 2$ non-broadcaster honest parties will multicast the `ack` message for v. Since there is just one Byzantine party, its `ack` is below the $n-2$ threshold. Then all honest parties will commit v after receiving $n-2$ `ack` messages at Step 3 and terminate. The good-case latency is 2 rounds, including broadcaster sending the proposal and all parties sending `ack` message.

Agreement, Termination and Bad-case Latency. If the broadcaster is honest, by validity all honest parties will commit the same value. If the broadcaster is Byzantine, then all $n-1$ non-broadcaster parties are honest. If an honest party commits v at Step 3, then it receives $n-2$ `ack` messages of v from distinct non-broadcaster parties, and thus all honest parties will also receive these `ack` messages and commit v. Since all honest parties commit in the same asynchronous round, the bad-case latency is also 2 rounds. ◀

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v\rangle$ to all parties.
2. **Ack.** When receiving the first proposal $\langle \texttt{propose}, v\rangle$ from the broadcaster, a party sends a `ack` message for v to all parties in the form of $\langle \texttt{ack}, v\rangle$.
3. **2-round Commit.** When receiving $\langle \texttt{ack}, v\rangle$ from $n-f-1$ distinct non-broadcaster parties, a party commits v and terminates.
4. **Vote and Lock.**
   - When receiving $\langle \texttt{ack}, v\rangle$ from a non-broadcaster party j, a party sends $\langle \texttt{vote}, j, v\rangle$ to all parties if not yet sent $\langle \texttt{vote}, j, v\rangle$.
   - When receiving $\langle \texttt{vote}, j, v\rangle$ from $n-f-2$ distinct *non-broadcaster parties other than* j, a party locks on v for party j.
5. **3-round Commit.** When locking on the same v for $n-2f$ distinct non-broadcaster parties, a party commits v and terminates.

Figure 4 $(2,3)$-round BRB under $n \geq 4f, f = 2$.

4.3 Impossibility of $(2,3)$-round BRB

▶ **Theorem 13.** *Any asynchronous unauthenticated Byzantine reliable broadcast protocol under $n \leq 5f-2, f \geq 3$ and has a good-case latency of 2 rounds must have a bad-case latency of at least 4 rounds.*

The proof of Theorem 13 is deferred to Appendix B due to the space limit.

4.4 $(2,3)$-round BRB Protocol under $n \geq 4f, f = 2$

For the special case of $f = 2$, we propose a $(2,3)$-round BRB protocol (Figure 4) that has optimal resilience $n \geq 4f$. The main idea is that all parties send `ack` for broadcaster's proposal, and also send `vote` for other parties' `ack`. When receiving enough `vote` messages of v for the same party, a party locks on v. The protocol guarantees that all honest parties lock on the same value for each party when $f = 2$. Then, the 3-round commit step let a party commits if the party locks on the same value for a majority of the parties. Since all parties send a `vote` for all other parties, the message and communication complexity are both $O(n^3)$.

▶ **Lemma 14.** *If the broadcaster is Byzantine and an honest party locks on v for party j, then all honest parties also lock on v for party j.*

Proof. Since an honest party locks on v for party j, it receives $n-f-2$ `vote` messages from non-broadcaster parties other than j. If j is honest, then it sends the same `ack` to all parties, and thus all honest parties receive $n-f-2$ `vote` for party j from non-broadcaster honest parties other than j. If j is Byzantine, then the parties other than j and the broadcaster are all honest. Since an honest party receives $n-f-2$ `vote` messages from these honest parties, all honest parties will also receive the messages. Therefore, all honest parties also lock on v for party j. ◀

▶ **Theorem 15.** *The protocol in Figure 4 solves Byzantine reliable broadcast under asynchrony with optimal resilience $n \geq 4f, f = 2$ and optimal good-case latency of 2 rounds, and has bad-case latency of 3 rounds.*

Proof.

Validity and Good-case Latency. If the broadcaster is honest, it sends the same proposal of value v to all parties, and all $n-f-1$ honest non-broadcaster parties will multicast the ack message for v. Since there are only f Byzantine party, their ack messages is below the $n-f-1$ threshold. Then all honest parties will commit v after receiving $n-f-1$ ack messages at Step 3 and terminate. The good-case latency is 2 rounds, including broadcaster sending the proposal and all parties sending ack message.

Agreement. If the broadcaster is honest, by validity all honest parties will commit the same value. Now consider when the broadcaster is Byzantine, and suppose there are $t > 0$ Byzantine parties there are at most $t-1$ Byzantine parties among non-broadcasters.

If any two honest parties commit different values at Step 3, then there must be at least $n-f-1-(t-1) = n-f-t$ honest parties sending ack for each of these different values. It is impossible by quorum intersection since there are only $n-t$ honest parties.

Suppose any two honest parties commit different values at Step 5. Then, there must exists at least $2(n-2f)-(n-1) \geq 1$ party for which the two committed honest parties lock different values. However, this contradicts Lemma 14, which states honest parties lock on the same value for any party when the broadcaster is Byzantine. Hence, no two honest parties can commit different values at Step 5.

Now we show that if an honest party $h1$ commits v at Step 3 and another honest party $h2$ commits v' at Step 5, then it must be $v = v'$. Suppose $h1$ commits v at Step 3, then at least $n-f-1-(f-1) = n-2f$ honest non-broadcaster parties send ack for v. All honest parties will lock on v for these $n-2f$ non-broadcaster parties, which is a majority of the $n-1$ non-broadcaster parties. Therefore any honest party that commits v' at Step 5 must have $v' = v$.

Termination and Bad-case Latency. If the broadcaster is honest, by validity all honest parties will commit the same value. If the broadcaster is Byzantine, once an honest party commits v at Step 3, there are $n-2f$ non-broadcaster honest parties that send ack for v, and all honest parties will eventually lock on v for these parties after receiving the vote messages. Therefore all honest parties will commit v at Step 5 after 1 extra round. ◀

4.5 $(2,3)$-round BRB under $n \geq 5f-1$

In this section, we improve the resilience of 2-round BRB protocol in the previous work [12] from $5f+1$ to $5f-1$, while keeping the bad-case latency 3 rounds. The protocol is presented in Figure 5, and the main difference compared to Imbs and Raynal [12] is that in Step 2, parties send ack for v if receiving $n-2f$ ack from *non-broadcaster parties*, instead of from any parties as in [12]. The intuition is that when the broadcaster is Byzantine, the above set of non-broadcaster parties only contains $f-1$ Byzantine parties, and thus we can reduce the total number of parties but still ensure quorum intersection.

▶ **Theorem 16.** *The protocol in Figure 5 solves Byzantine reliable broadcast under asynchrony with resilience $n \geq 5f-1$ and optimal good-case latency of 2 rounds, and has bad-case latency of 3 rounds.*

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v\rangle$ to all parties.
2. **Ack.**
   - When receiving the first proposal $\langle \texttt{propose}, v\rangle$ from the broadcaster, a party sends a `ack` message for v to all parties in the form of $\langle \texttt{ack}, v\rangle$.
   - When receiving $\langle \texttt{ack}, v\rangle$ from $n-2f$ distinct *non-broadcaster parties*, a party sends $\langle \texttt{ack}, v\rangle$ to all parties if not yet sent $\langle \texttt{ack}, v\rangle$.
3. **Commit.** When receiving $\langle \texttt{ack}, v\rangle$ from $n-f-1$ distinct non-broadcaster parties, a party commits v and terminates.

Figure 5 $(2,3)$-round BRB under $n \geq 5f-1$.

Proof.

Validity and Good-case Latency. If the broadcaster is honest, it sends the same proposal of value v to all parties, and all $n-f-1$ honest non-broadcaster parties will multicast the `ack` message for v. Since there are only f Byzantine party, their `ack` messages is below the $n-f-1$ threshold. Then all honest parties will commit v after receiving $n-f-1$ `ack` messages at Step 3 and terminate. The good-case latency is 2 rounds, including broadcaster sending the proposal and all parties sending `ack` message.

Agreement. If the broadcaster is honest, by validity all honest parties will commit the same value. If the broadcaster is Byzantine, and suppose there are $t > 0$ Byzantine parties, then there are $t-1$ Byzantine parties among all non-broadcaster parties. Suppose that two honest parties commit different values $v \neq v'$, then by Step 3 there are at least $n-f-1-(t-1) = n-f-t$ honest parties A that send `ack` for v and at least $n-f-1-(t-1) = n-f-t$ honest parties B that send `ack` for v'. Since there are $n-t$ honest parties in total, $|A \cap B| \geq 2(n-f-t)-(n-t) = n-2f-t \geq 3f-t-1 > 0$, there must exist some honest party that sends `ack` due to the second condition of Step 2. If the above only happens to v, then there are at least $n-2f-(t-1) = n-2f-t+1$ honest parties that send `ack` for v due to receiving the `propose` from the broadcaster. This contradicts the fact that at least $n-f-t$ honest parties send `ack` for v' due to receiving `propose`, since $(n-2f-t+1)+(n-f-t) > n-t$. It the above happens to both v, v', then there are at least $n-2f-(t-1) = n-2f-t+1$ honest parties that send `ack` for v (and for v', respectively) due to receiving the `propose` from the broadcaster. This is also impossible since $2(n-2f-t+1) \geq n+f-2t+1 > n-t$. Therefore, all honest parties commit the same value.

Termination and Bad-case Latency. If the broadcaster is honest, by validity all honest parties will commit the same value. If the broadcaster is Byzantine, once an honest party commits v, there are $n-2f$ non-broadcaster honest parties that send `ack` for v. Therefore all honest parties will send `ack` for v and hence commit v after 1 extra round. ◀

5 Extension to Unauthenticated Byzantine Broadcast under Synchrony

In this section, we extend the previous results to show the good-case latency results for unauthenticated Byzantine broadcast under synchrony. It is well-known that unauthenticated Byzantine broadcast or Byzantine reliable broadcast is solvable if and only if $n \geq 3f+1$.

We adopt the synchrony model assumptions from [4], including distinguishing the latency bounds δ and Δ, and the clock assumption, briefly as follows. More details about the model assumptions can be found in [4].

Network delays. We separate the *actual bound* δ, and the *conservative bound* Δ on the network delay:

- For one execution, δ is the upper bound for message delays between any pair of honest parties, but the value of δ is *unknown* to the protocol designer or any party. Different executions may have different δ values.
- For all executions, Δ is the upper bound for message delays between any pair of honest parties, and the value of Δ is *known* to the protocol designer and all parties.

Clock synchronization. Each party is equipped with a local clock that starts counting at the beginning of the protocol execution. We assume the *clock skew* is at most σ, i.e., they start the protocol at most σ apart from each other. We assume parties have *no clock drift* for convenience. There exist clock synchronization protocols [7, 1] that guarantee a bounded clock skew of $\sigma \leq \delta$. Since the value of δ is unknown to the protocol designer or any party, our protocol will use Δ as the parameter for clock skew in the protocol. Note that the actual clock skew is still $\sigma \leq \delta$, guaranteed by the clock synchronization protocols [7, 1]. In addition, due to clock skew, the BA primitive used in our BB protocol (Figure 6) needs to tolerate up to σ clock skew. For instance, any synchronous lock-step BA can do so by using a clock synchronization algorithm [7, 1] to ensure at most Δ clock skew, and setting each round duration to be 2Δ to enforce the abstraction of lock-step rounds.

▶ **Theorem 17.** *Any unauthenticated Byzantine reliable broadcast protocol under $3f + 1 \leq n \leq 4f - 1$ must have a good-case latency of at least 3δ under synchrony.*

The proof of Theorem 17 is analogous to that of Theorem 10, and is omitted here for brevity. Next, we show a synchronous BB protocol in Figure 6 that has good-case latency of 2δ under $n \geq 4f$.

Protocol description. The protocol is presented in Figure 6, and is inspired by our $(2, 4)$-round asynchronous BRB protocol (Figure 1) from Section 3.1. The main idea is to add a Byzantine agreement at the end of the protocol to ensure termination, since BRB does not require termination when the broadcaster is Byzantine. The input of the BA is called `lock`, which is set to be some default value $\perp$ initially, and will be set when commit in Step 3 or receiving enough `vote` in Step 4. One guarantee implied by the $(2, 4)$-round BRB protocol is that, when any honest party commit v in Step 3, all honest parties will lock on v, and therefore the BA will only output v.

▶ **Theorem 18.** *The protocol in Figure 6 solves Byzantine broadcast under synchrony with optimal resilience $n \geq 4f$ and optimal good-case latency of 2δ.*

Proof.

Validity and Good-case Latency. If the broadcaster is honest, it proposes the same value v to all parties, and all honest parties will send `ack` for v. Then at Step 3, all honest parties receive $n - f - 1$ `ack` messages of v after 2δ time (which is before local time $2\Delta + \sigma$), and commits v.

Initially, every party i starts the protocol at most δ time apart with a local clock and sets $\texttt{lock} = \bot$, $\sigma = \Delta$.

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v \rangle$ to all parties.
2. **Ack.** When receiving the first proposal $\langle \texttt{propose}, v \rangle$ from the broadcaster, a party sends an `ack` message for v to all parties in the form of $\langle \texttt{ack}, v \rangle$.
3. **Commit.** When receiving $\langle \texttt{ack}, v \rangle$ from $n - f - 1$ distinct non-broadcaster parties at time t, a party sets $\texttt{lock} = v$. If $t \leq 2\Delta + \sigma$, the party commits v.
4. **Vote.**
   - When receiving $\langle \texttt{ack}, v \rangle$ from $n - 2f$ distinct non-broadcaster parties, a party sends a `vote` message for v to all parties in the form of $\langle \texttt{vote}, v \rangle$ if not yet sent `vote` for any value.
   - When receiving $\langle \texttt{vote}, v \rangle$ from $n - f - 1$ distinct non-broadcaster parties, a party sets $\texttt{lock} = v$.
5. **Byzantine agreement.** At local time $3\Delta + 2\sigma$, a party invokes an instance of Byzantine agreement with `lock` as the input. If not committed, the party commits on the output of the Byzantine agreement. Terminate.

Figure 6 2δ unauthenticated BB protocol under $n \geq 4f$.

Agreement. If all honest parties commit at Step 5, all honest parties commit on the same value due to the agreement property of the BA. Otherwise, there must be some honest party that commits at Step 3. First, no two honest parties can commit different values at Step 3 due to quorum intersection. Now suppose any honest party h that commits v at Step 3. If the broadcaster is honest, by validity, all honest parties commits v. If the broadcaster is Byzantine, then there are $f - 1$ Byzantine parties among non-broadcasters. Since h receives $n - f - 1$ `ack` messages from non-broadcasters, at least $n - f - 1 - (f - 1) = n - 2f$ of them are from honest parties. Then, all honest parties receive these $n - 2f$ `ack` messages and set $\texttt{lock} = v$ at their local time $\leq (2\Delta + \sigma) + \Delta + \sigma = 3\Delta + 2\sigma$, before invoking the Byzantine agreement primitive at Step 5, since the clock skew is σ and message delay is bounded by Δ. Also by quorum intersection, there cannot be $n - 2f$ `ack` messages for $v' \neq v$, since the set of $(n - 2f) - (f - 1) = n - 3f + 1$ honest parties who voted for v' and the set of $n - 2f$ honest parties who voted for v intersect at $\geq (n - 3f + 1) + (n - 2f) - (n - f) \geq 1$ honest parties. Therefore, at Step 5, all honest parties have the same input $\texttt{lock} = v$ to the BA. Then by the validity condition of the BA primitive, the output of the agreement is also v. Any honest party that does not commit at Step 3 will commit v at Step 5.

Termination. According to the protocol, honest parties terminate at Step 5, and they commit a value before termination. ◀

6 Related Work

Byzantine fault-tolerant broadcast, first proposed by Lamport et al. [13], have received a significant amount of attention for several decades. Under synchrony, the deterministic Dolev-Strong protocol [10] solves Byzantine broadcast in worst-case $f + 1$ rounds, matching a lower bound [11]. Under asynchrony, Byzantine broadcast is unsolvable even with a single failure. Byzantine reliable broadcast relaxes the termination property of Byzantine broadcast, and the classic Byzantine reliable broadcast by Bracha [5] has a good-case latency of 3 rounds and bad-case latency of 4 rounds with optimal resilience $n \geq 3f+1$. Later works improves the good-case latency of reliable broadcast to 2 rounds by trading off resilience [12] or using authentication (signatures) [4]. A recent line of work studies the good-case latency of authenticated BFT protocols, including [2, 3, 4].

7 Conclusion

In this paper, we investigate the good-case latency of unauthenticated Byzantine fault-tolerant broadcast, which is time for all honest parties to commit given that the broadcaster is honest. We show the tight results are 2 rounds under $n \geq 4f$ and 3 rounds under $3f+1 \leq n \leq 4f-1$ for asynchronous Byzantine reliable broadcast, which can be extended for synchronous Byzantine broadcast as well. In addition, we also study the bad-case latency for asynchronous BRB which measures how fast can all honest parties commit when the broadcaster is dishonest and some honest party commits. We show 2 impossibility results and 4 matching asynchronous BRB protocols, including $(2,4)$-BRB under $n \geq 4f$, F2-BRB of $(2,3)$-round under $n \geq 4f, f=2$, F1-BRB of $(2,2)$-round under $n \geq 4f, f=1$, and $(2,3)$-BRB under $n \geq 5f-1$.

References

1 Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Synchronous byzantine agreement with expected $O(1)$ rounds, expected $O(n^2)$ communication, and optimal resilience. In *International Conference on Financial Cryptography and Data Security (FC)*, pages 320–334. Springer, 2019.

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

3 Ittai Abraham, Kartik Nayak, Ling Ren, and Zhuolun Xiang. Brief announcement: Byzantine agreement, broadcast and state machine replication with optimal good-case latency. In *34th International Symposium on Distributed Computing (DISC)*, 2020.

4 Ittai Abraham, Kartik Nayak, Ling Ren, and Zhuolun Xiang. Good-case latency of byzantine broadcast: A complete categorization. In *Proceedings of the third annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 331–341, 2021.

5 Gabriel Bracha. Asynchronous byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987.

6 Ran Canetti and Tal Rabin. Fast asynchronous byzantine agreement with optimal resilience. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing (STOC)*, pages 42–51, 1993.

7 Danny Dolev, Joseph Y Halpern, Barbara Simons, and Ray Strong. Dynamic fault-tolerant clock synchronization. *Journal of the ACM (JACM)*, 42(1):143–185, 1995.

8 Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. *Journal of the ACM (JACM)*, 32(1):191–204, 1985.

9 Danny Dolev, Ruediger Reischuk, and H Raymond Strong. Early stopping in byzantine agreement. *Journal of the ACM (JACM)*, 37(4):720–741, 1990.

10 Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

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

12 Damien Imbs and Michel Raynal. Trading off t-resilience for efficiency in asynchronous byzantine reliable broadcast. *Parallel Processing Letters*, 26(04):1650017, 2016.

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

A Proof of Theorem 11

Proof. Suppose on the contrary there exists an asynchronous BRB protocol Π that tolerates $f = 2$ and has $(2, 2)$-round. We assume $n \geq 4f = 8$, otherwise no protocol can solve BRB with good-case latency of 2 rounds by Theorem 10. Denote the broadcaster as party 0 always, and remaining parties as party $1, ..., n - 1$. We construct the following executions.

- Execution 1. The broadcaster is honest, and has input 0. Party $n - 1$ is Byzantine and remain silent. Then all honest parties commit 0 in 2 rounds by assumption.
- Execution 2. The broadcaster is Byzantine, and behaves honestly to parties $1, ..., n - 3$ with input 0, and remains silent to other parties. Party $n - 2$ is Byzantine, and behaves identically to party $n - 3$ as in Execution 1, but remains silent to rest of the parties. Any messages from party $n - 1$ are delayed and not delivered in 2 rounds. It is easy to see that party $n - 3$ cannot distinguish Execution 2 and 1 in 2 rounds, therefore it commits 0 in 2 rounds in Execution 2 as well. By assumption, parties $1, ..., n - 4$ also commit 0 in 2 rounds in Execution 2.
- Execution x for $x = 3, ..., n - 3$. The broadcaster is Byzantine, and behaves honestly to parties $1, ..., n - x - 1$ with input 0, and remains silent to other parties. Party $n - x$ is Byzantine, and behaves identically to party $n - x - 1$ as in Execution $x - 1$, but remains silent to rest of the parties. Any messages from party $n - x + 1$ are delayed and not delivered in 2 rounds. It is easy to see that party $n - x - 1$ cannot distinguish Execution x and $x - 1$ in 2 rounds, therefore it commits 0 in 2 rounds in Execution x as well. By assumption, parties $1, ..., n - x - 2$ also commit 0 in 2 rounds in Execution 2.
- Execution $n - 2$. The broadcaster is Byzantine, and behaves honestly to party 1 with input 0, and remains silent to other parties. Party 2 is Byzantine, and behaves identically to party 1 as in Execution $n - 3$, but remains silent to rest of the parties. Any messages from party 3 are delayed and not delivered in 2 rounds. It is easy to see that party 1 cannot distinguish Execution $n - 2$ and $n - 3$ in 2 rounds, therefore it commits 0 in 2 rounds in Execution $n - 2$ as well.

Similarly, we can construct $n - 2$ symmetric executions, where the broadcaster has input 1, and in the last execution the broadcaster only behaves honestly to party $n - 1$ with input 1, and party $n - 1$ commits 1 in 2 rounds.

Contradiction. Now we consider another execution, where the broadcaster is Byzantine, it behaves to party 1 honestly with input 0, and to party $n - 1$ honestly with input 1, and remain silent to other parties. Party 2 is Byzantine, it behaves to party 1 identically as in Execution $n - 3$, and to party $n - 1$ identically as the party $n - 2$ to party $n - 1$ in the last execution of the constructed symmetric executions (due to symmetric of the non-broadcaster parties, the index does not matter). Any messages between parties $1, n - 1$ are delayed and not delivered in 2 rounds. Then, party 1 commits 0 in 2 rounds while party $n - 1$ commit 1 in 2 rounds, breaking agreement of the BRB. Therefore, such protocol Π does not exist. ◀

B Proof of Theorem 13

Proof. Suppose on the contrary that there exists an asynchronous BRB protocol Π under $n = 5f - 2, f \geq 3$ that has $(2,3)$-round. Denote the broadcaster as party 0 always, denote 2 non-broadcaster parties as p, q, and divide the remaining $5f-5$ parties into 5 groups $G_1, ..., G_5$ each of size $f - 1$ (recall $f - 1 \geq 2$). Denote $G_L = \{p\} \cup G_1 \cup G_2$ and $G_R = G_4 \cup G_5 \cup \{q\}$. We use $S[i]$ to denote the i-th party in set S, where S can be any set defined above (such as G_j for $j = 1, ..., 5$ and G_L, G_R). We construct the following executions. In all constructed executions, all messages are delivered by the recipient after Δ time by default, and we will explicitly specify the messages that are delayed by the adversary due to asynchrony.

- E_1^0. The broadcaster is honest and has input 0. Parties in $G_5 \cup \{q\}$ are Byzantine, and they behave honestly except that they pretend to receive from a broadcaster whose input is 1. Since the broadcaster is honest, by validity and good-case latency, all honest parties commit 0 after receiving two rounds of messages.
- E_1^1. This execution is a symmetric case of E_1^0. The broadcaster is honest and has input 1. Parties in $G_1 \cup \{p\}$ are Byzantine, and they behave honestly except that they pretend to receive from a broadcaster whose input is 0. Since the broadcaster is honest, by validity and good-case latency, all honest parties commit 1 after receiving two rounds of messages.
- E_2^0. The broadcaster is Byzantine, it behaves to $G_L \cup G_3$ identically as in E_1^0, and to $G_5 \cup \{q\}$ identically as in E_1^1. Parties in G_4 are Byzantine, they behave to the party $G_3[f-1]$ honestly (recall that $f - 1 \geq 2$ so $G_3[f-1] \neq G_3[1]$) but pretending to receive from the broadcaster in E_1^0, and to other parties honestly but pretending to receive from the broadcaster in E_1^1.

  ▷ Claim. The honest party $G_3[f-1]$ cannot distinguish E_2^0 and E_1^0 in 2 rounds, and thus will commit 0 in round 2. Then, by assumption, all honest parties also commit 0 in round 3 in E_2^0. The broadcaster behaves to $G_3[f-1]$ identically in both executions. The messages sent to $G_3[f-1]$ in the first round by any non-broadcaster party are identical in E_2^0 and E_1^0, since the first round message only depends on the initial state and all Byzantine parties behave honestly in the first round. For the second round, since in E_1^0 the Byzantine parties in $G_5 \cup \{q\}$ pretend to receive from a broadcaster with input 1, they send the same round-2 messages as in E_2^0. For the Byzantine parties in G_4, they behave identically to $G_3[f-1]$ by construction. All honest parties in G_L also behave identically to $G_3[f-1]$ in round 2 since they receive the same round-1 messages. Therefore party $G_3[f-1]$ cannot distinguish E_2^0 and E_1^0 in 2 rounds, and thus will commit 0 in round 2.

- E_3^0. The broadcaster is Byzantine, it behaves to G_L identically as in E_1^0, and to G_R identically as in E_1^1. Parties in G_3 are Byzantine, they behave to other parties identically as in E_2^0.

  ▷ Claim. The honest parties in $G_L \cup G_R$ cannot distinguish E_3^0 and E_2^0 in 3 rounds, and thus will commit 0 in round 3. For the round-1 message, honest parties receive the same messages in both executions since Byzantine parties including the broadcaster send the same messages. For the round-2 message, the Byzantine parties of G_4 in E_2^0 behave to $G_L \cup G_R$ as if they receive from a broadcaster with input 1, which would be identically to E_3^0. The Byzantine parties of G_3 in E_3^0 behave to other parties identically as in E_2^0 by construction. Similarly, $G_l \cup G_R$ receive the same round-3 messages in both executions, and thus cannot distinguish E_3^0 and E_2^0 in 3 rounds, and will commit 0 in round 3 in E_3^0 as well.

- E^0_{2j+2} for $j = 1, 2, ..., |G_R| = 2f - 1$. The broadcaster is Byzantine, it behaves to $G_L \cup \{G_3[1]\}$ identically as in E^0_1, and to G_R identically as in E^1_1. Parties in $G_3 \setminus \{G_3[1]\}$ are Byzantine (recall that $|G_3| = f - 1 \geq 2$), and they behave to all honest parties identically as in E^0_{2j+1}. Party $G_R[j]$ is Byzantine, and it behaves to all honest parties except p identically as in E^0_{2j+1}, and to party p honestly except that it pretends receiving no message from G_3 sent after round 1. Any round-2 or round-3 message from $G_3[1]$ to parties in $G_R[i], i = 1, ..., j - 1$ are delayed and received only after round 3.
- E^0_{2j+3} for $j = 1, 2, ..., |G_R| = 2f - 1$. The broadcaster is Byzantine, it behaves to G_L identically as in E^0_1, and to G_R identically as in E^1_1. Parties in G_3 are Byzantine, they behave to other parties identically as in E^0_{2j+2}, but they send no message to $G_R[j]$ after round 1.

  ▷ Claim. Any honest party in $G_L \setminus \{p\}$ cannot distinguish E^0_{2j+2} and E^0_{2j+1} in 3 rounds, and it will commit 0 in round 3 in E^0_{2j+2}. Then, by assumption, party p will also commit 0 in round 3 in E^0_{2j+2}. Similar to the previous claim, honest parties receive the same round-1 messages. For round 2, the Byzantine parties in E^0_{2j+2} behave identically to all honest parties, including party p since the difference from $G_R[j]$ to p is reflected only after round 2. Hence, honest parties in $G_L \setminus \{p\}$ will also receive the same messages in round 3, thus cannot distinguish E^0_{2j+2} and E^0_{2j+1} in 3 rounds.

  ▷ Claim. Party p cannot distinguish E^0_{2j+3} and E^0_{2j+2} in 3 rounds, and thus will commit 0 in round 3 in E^0_{2j+3}. Then, by assumption, all honest parties in $G_L \cup G_R$ also commit 0 in round 3. Similar to previous claim, honest parties receive the same round-1 and round-2 messages. For round 3, since Byzantine parties in G_3 send no message to $G_R[j]$ after round 1 in E^0_{2j+3}, the honest party $G_R[j]$ in E^0_{2j+3} will behave the same to p as the Byzantine party $G_R[j]$ which pretends to p that it receives no message from G_3 in E^0_{2j+2}. Hence, p cannot distinguish E^0_{2j+3} and E^0_{2j+2} in 3 rounds.

By the above constructions, we finally have an execution $E^0_{2j+3,j=2f-1} = E^0_{4f+1}$ where the Byzantine broadcaster behaves to G_L with input 0, and to G_R with input 1, and the Byzantine parties in G_3 send no message to G_R, but party p has to commit 0 in 3 rounds. Similarly, we can construct a series of symmetric executions of the above executions including E^1_1, i.e., $E^1_1, E^1_2, ..., E^1_{4f+1}$, and have the execution E^1_{4f+1} where the Byzantine broadcaster also behaves to G_L with input 0, and to G_R with input 1, and the Byzantine parties in G_3 send no message to G_L, but party q has to commit 1 in 3 rounds.

Contradiction. Now we construct another middle execution E_m, where the Byzantine broadcaster behaves to G_L with input 0, and to G_R with input 1, and Byzantine parties in G_3 behave to G_L identically as in E^0_{4f+1} and to G_R identically as in E^1_{4f+1}. It is easy to see that party p cannot distinguish E_m and E^0_{4f+1} in 3 rounds, and thus will commit 0 in round 3, while party q cannot distinguish E_m and E^1_{4f+1} in 3 rounds, and thus will commit 1 in round 3. This violates the agreement property of BRB, and hence such BRB protocol Π does not exist. ◀

C 3δ Unauthenticated Byzantine Broadcast under Synchrony

For completeness, we show an unauthenticated BB protocol in Figure 7 with good-case latency of 3δ under synchrony and $n \geq 3f + 1$, inspired by Bracha's reliable broadcast [5].

▶ **Theorem 19.** *The protocol in Figure 7 solves Byzantine broadcast under synchrony with resilience $n \geq 3f + 1$ and good-case latency of 3δ.*

The correctness proof is similar to that of Theorem 18, and we omit it here for brevity.

Initially, every party i starts the protocol at most δ time apart with a local clock and sets $\texttt{lock} = \perp$, $\sigma = \Delta$.

1. **Propose.** The designated broadcaster L with input v sends $\langle \texttt{propose}, v \rangle$ to all parties.
2. **Echo.** When receiving the first proposal $\langle \texttt{propose}, v \rangle$ from the broadcaster, a party sends an `echo` message for v to all parties in the form of $\langle \texttt{echo}, v \rangle$.
3. **Vote.**
   - When receiving $\langle \texttt{echo}, v \rangle$ from $n - f$ distinct parties, a party sends a `vote` message for v to all parties in the form of $\langle \texttt{vote}, v \rangle$ and sets $\texttt{lock} = v$ if not yet sent `vote` for any value.
   - When receiving $\langle \texttt{vote}, v \rangle$ from $f + 1$ distinct parties, a party sends a `vote` message for v to all parties in the form of $\langle \texttt{vote}, v \rangle$ and sets $\texttt{lock} = v$ if not yet sent `vote` for any value.
4. **Commit.** When receiving $\langle \texttt{vote}, v \rangle$ from $n - f$ distinct parties at time t, a party sets $\texttt{lock} = v$. If $t \leq 3\Delta + \sigma$, the party commits v.
5. **Byzantine agreement.** At local time $4\Delta + 2\sigma$, a party invokes an instance of Byzantine agreement with `lock` as the input. If not committed, the party commits on the output of the Byzantine agreement. Terminate.

Figure 7 3δ unauthenticated BB protocol under synchrony and $n \geq 3f + 1$.