Brief Announcement: From Few to Many Faults: Adaptive Byzantine Agreement with Optimal Communication

Andrei Constantinescu
ETH Zürich & DFINITY

Marc Dufay
ETH Zürich

Anton Paramonov
ETH Zürich

Roger Wattenhofer
ETH Zürich

Abstract

We study the problem of Strong Byzantine Agreement and establish tight upper and lower bounds on communication complexity, parameterized by the actual number of Byzantine faults. Specifically, for a system of n parties tolerating up to t Byzantine faults, out of which only $f \leq t$ are actually faulty, we obtain the following results:

In the partially synchronous setting, we present the first Byzantine Agreement protocol that achieves *adaptive* communication complexity of $\mathcal{O}(n + t \cdot f)$ words, which is asymptotically optimal. Our protocol has an optimal resilience of $t < n/3$.

In the asynchronous setting, we prove a lower bound of $\Omega(n + t^2)$ on the expected number of messages, and design an almost matching protocol with an optimal resilience that solves agreement with $\mathcal{O}((n + t^2) \cdot \log n)$ words. Our main technical contribution in the asynchronous setting is the utilization of a bipartite expander graph that allows for low-cost information dissemination.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Byzantine Agreement, Communication Complexity, Adaptive Communication Complexity, Resilience

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.48

Category Brief Announcement

Related Version *Full Version*: `arXiv:2505.19989` [10]

1 Introduction

Achieving agreement in a distributed setting is fundamental to blockchain systems. Modern blockchains often comprise thousands of nodes attempting to reach consensus. Given the widely accepted resilience model where $t \approx n/3$ validators can be Byzantine, and considering the classical Dolev-Reischuk lower bound of $\Omega(n^2)$ messages for agreement [11], such systems must exchange millions of messages. This communication overhead results in high latency and limited scalability. This work addresses this challenge through two key contributions:

We bypass the Dolev-Reischuk lower bound by introducing the first partial-synchrony algorithm for Byzantine Agreement that requires only $\mathcal{O}(n+t\cdot f)$ messages. This approach is significantly faster compared to the $\mathcal{O}(n^2)$ suggested by [11], since often in practice the actual number of misbehaving nodes is small. Our algorithm achieves asymptotically optimal communication complexity, as well as optimal resilience.

Furthermore, we highlight that system robustness is determined not by the relative fraction of possible corruptions but by the absolute number of failures it can tolerate. For instance, it's easier for

an adversary to corrupt a system with $3,001$ nodes that tolerates $1,000$ faults than one with $10,000$ nodes tolerating $2,000$ faults. Building on this insight, we design an algorithm that decouples n and t, achieving superior performance when $t \ll n$. Specifically, instead of the known $\mathcal{O}(n \cdot t)$ communication complexity, in the asynchronous setting, we achieve agreement with a communication complexity of $\mathcal{O}((n + t^2) \cdot \log n)$. This allows system designers to first choose a maximal threshold t of tolerated failures that would still allow for efficient dissemination of t^2 messages, and then scale n up to t^2 without introducing additional latency. Notably, our algorithm is non-constructive, since it internally utilizes a bipartite expander whose existence has been proven, but no explicit construction is known; when restricted to explicit constructions, the best-known guarantees are slightly weaker.

Finally, we establish that achieving adaptive communication complexity is infeasible in asynchronous settings. Specifically, we prove a lower bound of $\Omega(n+t^2)$ for any algorithm that guarantees almost-sure termination. This result demonstrates that our algorithm is optimal up to a logarithmic factor.

2 Related Work

Synchrony. In synchronous networks, deterministic Byzantine Broadcast (BB) and Weak Byzantine Agreement (BA) achieve cheap communication: Cohen et al. [9] give $\mathcal{O}(n(f+1))$ for Weak BA/BB, and [7] give deterministic BA with $\mathcal{O}(nf)$ at resilience $t < (\frac{1}{2} - \varepsilon)n$. Civit et al. [5] extend to Multi-Valued BA (MVBA) and Interactive Consistency (IC) with bit complexity $\mathcal{O}(L_o n + n(f+1)\kappa)$. Lower bounds include $\Omega(f^2)$ messages [11], $\Omega(n + tf)$ bits [15], and even randomized expected $\Omega(f^2)$ under adaptive adversaries [1].

Partial synchrony. In partial synchrony, BA attains $\mathcal{O}(n^2)$ communication complexity, e.g., Oper [6] Introduced by Civit et al., lifts any synchronous BA algorithm to partial synchrony while preserving per-process bit costs (yielding total $\mathcal{O}(n^2)$ for [5]). For State Machine Replication (SMR) in partial synchrony, HotStuff with robust view synchronization reaches $\mathcal{O}(nf)$ after GST [17, 14]. As for the lower bounds, Spiegelman [15] showed that for any partially synchronous algorithm, the number of messages sent before GST can be unbounded.

Asynchrony. In asynchrony, the classical FLP [12] impossibility rules out deterministic BA. However, randomness enables expected $\mathcal{O}(n^2)$ communication complexity to achieve an almost sure termination [4] and subquadratic $\mathcal{O}(n \log^2 n)$ for with high probability guarantees [8]. When allowing for an initial setup, expected $\mathcal{O}(n\kappa^4)$ can be achieved [2].

3 Preliminaries

We consider a set of n parties running a distributed protocol in a fully connected network. Channels are authenticated, meaning that when receiving a message, a party knows who sent it.

Network Setting: We consider 3 different network settings:

- *Synchronous network*: We assume there is a known $\Delta > 0$ such that when sending a message, a party has the guarantee that this message will be received within Δ units of time. This allows protocols to be expressed in rounds where each round takes Δ units of time.
- *Asynchronous network*: There is no guarantee on the delay between a party sending a message and it being received. The only guarantee is that a message sent is eventually received.
- *Partially synchronous network*: We assume the setting with a known $\Delta > 0$ and an unknown *Global Stabilization Time* (GST), such that a messages sent at time t is received by the time at most $\max\{t, GST\} + \Delta$.

Byzantine Behavior: We assume that a set of at most t parties can become faulty when running the protocol. In particular, an adversary is able to control these parties, and the protocol must be

Paper	Problem	Communication	Resilience	Det.	PKI	Properties
Synchrony						
Cohen et al. [9]	BB	$\mathcal{O}(n \cdot f)$	$t < n/2$	Yes	Yes	A;T;V
Civit et al. [7]	MVBA	$\mathcal{O}(n \cdot f)$	$t < (\frac{1}{2} - \varepsilon)n$	Yes	Yes	A;T;V
Civit et al. [5]	MVBA, IC	$\mathcal{O}(L_o n + n \cdot f)$	$t < \frac{n}{2}$	Yes	Yes	A;T;V;IC
This paper	BA	$\mathcal{O}(n + tf)$	$t < \frac{n}{2}$	Yes	Yes	A;T;V
Spiegelman et al. [15]	BA	$\Omega(n + tf)$	Any	Yes	No	A;T;V
Ittai et al.[1]	BA	$\mathbb{E}(\Omega(f^2))$	Any	No	Yes	A;T;V
Partial Synchrony						
Maofan et al. [17, 14]	SMR	$\mathcal{O}(n \cdot f)$	$t < n/3$	Yes*	Yes	A;T;EV
Civit et al. [6]	BA	$\mathcal{O}(n^2)$	$t < n/3$	Yes	No	A;T;V
This paper	BA	$\mathcal{O}(n + t \cdot f)$	$t < n/3$	Yes	Yes	A;T;V
Spigeleman et al. [15]	BA	$\Omega(\infty)^{\S}$	Any	Yes	No	A;T;V
Asynchrony						
Cachin et al. [4]	BA	$\mathbb{E}(\mathcal{O}(n^2))$	$t < n/3$	No	Yes	A;T‡;V
Cohen et al. [8]	BA	$\mathcal{O}(n \cdot \log^2 n)^{\dagger}$	$t < (\frac{1}{3} - \varepsilon)n$	No	Yes	A†;T†;V†
Blum et al. [2]	BA	$\mathbb{E}[\mathcal{O}(n \cdot \kappa^4)]$	$t < (\frac{1}{3} - \varepsilon)n$	No	Yes	A††;T††;V††
This paper	BA	$\mathbb{E}(\mathcal{O}((n + t^2) \cdot \log n))$	$t < n/3$	No	Yes	A;T‡;V
This paper	BA	$\mathbb{E}(\Omega(n + t^2))$	Any	No	Yes	A;T‡;V

Table 1 Comparison of Byzantine Agreement Protocols. When a communication is stated with Ω, it indicates a lower bound result.
$*$ - [14] uses random leader election, but can be derandomized, maintaining the mentioned characteristics.
§ - unlimited messages before GST., † - with high probability., †† - with overwhelming probability. , ‡ - with probability 1., E - in expectation.

resilient to this setting. Parties that are not corrupted are called honest. In the rest of this paper, we will also consider f, the number of actual byzantine parties during an execution of a protocol, we have $0 \leq f \leq t \leq n$.

Moreover, we assume that for the asynchronous setting as well as the partially synchronous setting before GST, the adversary has full control over the scheduler and can delay or reorder any messages as long as they are eventually received. Our protocols are resilient against a dynamic adversary. This means that the adversary is allowed to observe messages being sent before deciding which parties to corrupt.

Byzantine Agreement: In this paper, we focus on solving the problem of Strong Binary Byzantine Agreement, which we call just Byzantine Agreement for simplicity. In this problem, each honest party can *propose* a value (either 0 or 1) and can *decide* a value. The Byzantine Agreement is defined by the following properties.

- *Termination*: Every honest party eventually decides a value.
- *Agreement*: If two honest parties p_1 and p_2 respectively decide values v_1 and v_2, then $v_1 = v_2$.
- *Strong Unanimity*[1]: If all honest parties propose the same value, then this value must be decided.
- *Probabilistic termination*: Every honest party eventually decides a value almost surely.

For the synchronous and partially synchronous setting, we say that a protocol satisfies Byzantine Agreement (BA) if it satisfies Termination, Agreement, and Strong Unanimity. Because of the FLP impossibility result [12] BA cannot be solved deterministically in asynchrony. Therefore, for the asynchronous setting, we replace *Termination* with *Probabilistic Termination*:

[1] In literature, this property is also called Strong Validity.

Cryptographic Primitive: In some of our protocols, we will assume the presence of a public-key infrastructure (PKI). This allows for *threshold* signature schemes [3] which are used for combining multiple signatures into one and hence reducing the communication complexity of the algorithm.

Communication Complexity. We define the *communication complexity* of an algorithm as the number of *words* sent by honest nodes, where each word contains a constant number of signatures and a constant number of bits. As pointed out by Spiegelman et al. [15], there doesn't exist a partially synchronous algorithm that solves BA sending a bounded number of messages if messages sent before GST are counted. Therefore, when speaking about the communication complexity of partially synchronous algorithms, we refer only to messages sent *after* GST.

4 Results Overview

In this section, we formally state our results and describe the main components used to obtain them. On the feasibility side, we devise protocols with adaptive communication complexity for all major time models.

In partial synchrony, our algorithm achieves asymptotically optimal communication complexity.

▶ **Theorem 1.** *There exists a deterministic algorithm that tolerates up to $t < n/3$ Byzantine faults and, given a PKI, solves Byzantine Agreement in partial synchrony with communication complexity of $\mathcal{O}(n + t \cdot f)$.*

In an asynchronous setting, we show the existence of an algorithm that achieves optimal resilience and a communication complexity that is asymptotically optimal up to a logarithmic factor.

▶ **Theorem 2.** *There exists a randomized algorithm that tolerates up to $t < n/3$ Byzantine faults and, given a PKI, solves Byzantine Agreement in asynchrony with expected communication complexity of $O((n + t^2) \cdot \log n)$.*

In synchrony, our algorithm has asymptotically optimal adaptive communication and optimal resilience.

▶ **Theorem 3.** *There exists a deterministic algorithm that tolerates up to $t < n/2$ Byzantine faults and, given a PKI, solves Byzantine Agreement in synchrony with communication complexity of $\mathcal{O}(n + t \cdot f)$.*

All of our results follow a common high-level strategy. First, we execute an adaptive Byzantine Agreement protocol on a subset of parties of size $\Theta(t)$, which we refer to as the *quorum*. The communication complexity here only depends on t and f. We note that for this step we rely on existing Byzantine Agreement protocols in the synchronous and asynchronous settings, specifically those of Civit et al.[5] and Cachin et al.[4]. In contrast, the $\mathcal{O}(n + n \cdot f)$ protocol we use for the partially synchronous setting is our own contribution, described in Section 5. While inspired by Spiegelman's synchronous protocol for external validity [15], our algorithm advances it in two key ways: it operates in the partially synchronous setting and guarantees strong unanimity.

In the second step, we execute a Quorum-to-All Broadcast (QAB)—a communication primitive introduced in this work—where the quorum nodes disseminate the decided value to the rest of the network. We design QAB protocols for both partially synchronous and asynchronous settings. In particular, our asynchronous QAB leverages bipartite expander graphs to enable efficient communication, constituting one of the key technical contributions of this paper.

On the infeasibility side, we have the following versatile lower bound that shows the impossibility of achieving an adaptive communication complexity in the asynchronous setting.

▶ **Theorem 4.** *Let $\mathcal{A}$ be a protocol solving asynchronous byzantine agreement resilient to t byzantine parties. $\mathcal{A}$ is allowed to have access to PKI, shared randomness, and an initial setup phase before receiving values. Let M be the number of messages exchanged after setup to reach agreement, then there exists an input configuration and a message scheduling protocol such that $\mathbb{E}(M) = \Omega(t^2)$ without any byzantine party. This lower bound holds even if the adversary is static.*

The main contribution of this result — beyond the fact that it holds under a very weak adversary and a strong algorithmic setting — is that, to the best of our knowledge, it establishes the first lower bound of $\Omega(t^2)$, in contrast to the $\Omega(f^2)$ bounds shown in [11, 1]. This bound allows us to claim that the algorithm stated by Theorem 2 has an almost optimal communication complexity, since $\Omega(n)$ is a trivial lower bound.

5 Adaptive Byzantine Agreement in Partial Synchrony

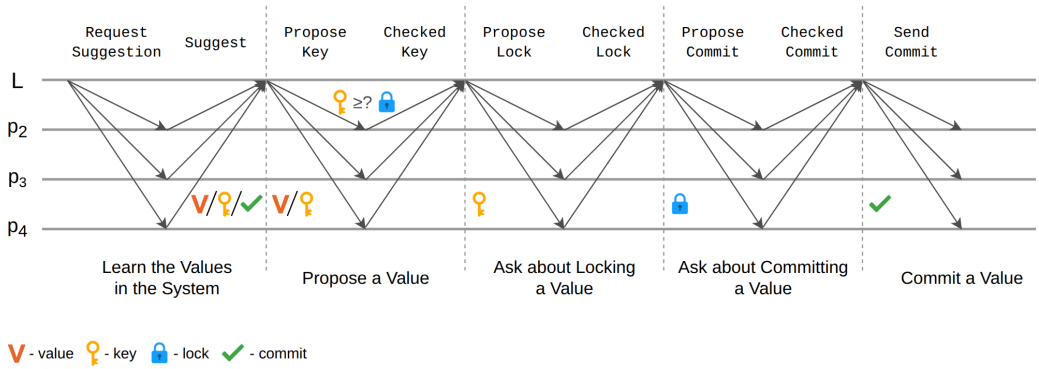

Figure 1 Evolution of a single view in our partially synchronous Byzantine Agreement protocol, assuming an honest leader L and that the system is after GST.

In this section, we present a partially synchronous algorithm that achieves an adaptive communication complexity of $\mathcal{O}(n + n \cdot f)$. Our algorithm is based on a modified version of Spiegelman's byzantine agreement protocol for external validity [15], which itself uses Hotstuff's view synchronization [17] technique. While inspired by Spiegelman's protocol for network-agnostic settings [15], our work departs significantly in both design and guarantees. We eliminate the asynchronous and randomized components of the original protocol and instead strengthen the synchronous part with additional checks, enabling it to function effectively in partial synchrony after GST. This modification allows us to achieve $\mathcal{O}(n \cdot f)$ message complexity for external validity without relying on full synchrony. Crucially, by allowing parties to selectively share their commit proofs upon request, we ensure termination without incurring additional message overhead. Moreover, we extend the protocol to achieve binary strong unanimity—rather than just external validity—by leveraging threshold signatures.

The protocol is organized in views. Each view has a leader, who is chosen in a round-robin way. In the full version of the paper [10], we show that the algorithm is guaranteed to succeed after at most n views after GST. Moreover, if there are no byzantine parties, everyone is guaranteed to decide within constant time after GST.

The algorithm for a single view consists of 4 phases of a leader asking and other parties responding. A leader proceeds to the next phase once they get $n - t$ responses. On each phase, the leader tries to acquire a new threshold signature, namely the key signature, then the lock and finally the commit signature, each with stronger properties than the other. After 4 phases pass successfully, the leader

obtains broadcasts his value and is guaranteed than only this value can be decided in the future (see Figure 1). We note that if the leader is byzantine or the network is not yet synchronous, the last phase may not be reached by the end of the view. Even then, the threshold values obtained so far are used in subsequent rounds to ensure agreement. A complete description of our protocol as well as a detailed proof for it is given in the full version of our paper [10].

This algorithm, coupled with the QAB primitive, solves Byzantine agreement while tolerating up to $t < n/3$ parties and using $\mathcal{O}(n + t \cdot f)$ messages. Therefore, it satisfies Theorem 1.

6 Broadcasting from a Quorum

To decrease the message complexity, we will rely on a primitive which we call Quorum-to-All Broadcast (abbreviated QAB). Briefly, this primitive allows a small group (of size $\Theta(t)$) of parties all having the same input value v_{in} to broadcast it to everyone. QAB is defined by the following property that encapsulates classical Agreement, Termination, and Validity.

- *Complete correctness:* Every honest node eventually decides v_{in}.

We next describe our QAB for Asynchrony at a high level. For formal description, as well as the partial synchrony QAB, please refer to the full version of the paper [10].

▶ **Theorem 5.** *For a Quorum of size $3t + 1$, there exists a deterministic algorithm that, given a PKI, solves QAB in asynchrony in the presence of at most t Byzantine faults with communication complexity of $\mathcal{O}((n + t^2) \cdot \log n)$.*

In order to achieve the stated communication complexity, we restrict the communication between nodes to a specific graph, which we call a communication graph. There are three roles in the algorithm (one node can have multiple roles): *quorum nodes* - nodes that have a common value to broadcast, *relayers* - nodes that help quorum nodes disseminate the value and *parties* - all the nodes, those that need to learn the value. First, quorum nodes agree on a value based on their own proposals, then disseminate it through relayers. Parties then confirm that they have received the value by sending a signed confirmation to relayers who in turn aggregate these confirmations and relay them to quorum nodes. If a quorum node does not receive a confirmation from some party, it sends a value to it directly.

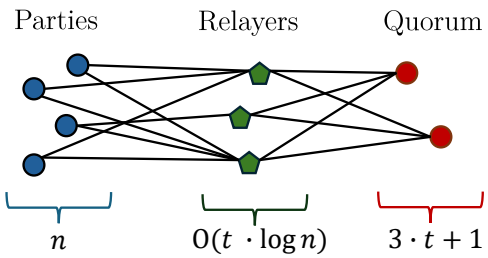

Figure 2 The communication graph for stages 1 and 2 of the algorithm. Nodes are assigned three roles (non-exclusive): *parties*, *relayers* and *quorum* with respective sizes of n, $O(t \log n)$ and $3t + 1$. Each party is only linked to $O(\log n)$ relayers, and each relayer is linked to each quorum node.

The crux of our result is the utilization of a bipartite (*parties, relayers*) graph that (I) has few edges (II) an adversary cannot disconnect many parties from the quorum. Please see the full paper [10] for a formal statement of its properties.

We remark that the existence of a bipartite expander with desired properties is only shown non-constructively in [16]. Known explicit bipartite expanders with polynomial time construction [13], when utilized instead, would imply a message complexity of $(n + t^{2+\varepsilon}) \cdot \log^{\mathcal{O}(1/\varepsilon)} n$.

References

1 Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In Peter Robinson and Faith Ellen, editors, *38th ACM PODC*, pages 317–326. ACM, July / August 2019. doi:10.1145/3293611.3331629.

2 Erica Blum, Jonathan Katz, Chen-Da Liu-Zhang, and Julian Loss. Asynchronous byzantine agreement with subquadratic communication. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 353–380. Springer, Cham, November 2020. doi:10.1007/978-3-030-64375-1_13.

3 Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 416–432. Springer, Berlin, Heidelberg, May 2003. doi:10.1007/3-540-39200-9_26.

4 Christian Cachin, Klaus Kursawe, and Victor Shoup. Random oracles in constantipole: practical asynchronous byzantine agreement using cryptography (extended abstract). In Gil Neiger, editor, *19th ACM PODC*, pages 123–132. ACM, July 2000. doi:10.1145/343477.343531.

5 Pierre Civit, Muhammad Ayaz Dzulfikar, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, and Manuel Vidigueira. DARE to agree: Byzantine agreement with optimal resilience and adaptive communication. In Ran Gelles, Dennis Olivetti, and Petr Kuznetsov, editors, *43rd ACM PODC*, pages 145–156. ACM, June 2024. doi:10.1145/3662158.3662792.

6 Pierre Civit, Muhammad Ayaz Dzulfikar, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, Manuel Vidigueira, and Igor Zablotchi. Partial synchrony for free: New upper bounds for byzantine agreement. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4227–4291. SIAM, 2025. doi:10.1137/1.9781611978322.144.

7 Pierre Civit, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, and Manuel Vidigueira. Strong byzantine agreement with adaptive word complexity, 2023. doi:10.48550/arXiv.2308.03524.

8 Shir Cohen, Idit Keidar, and Alexander Spiegelman. Not a coincidence: Sub-quadratic asynchronous byzantine agreement whp. In *34th International Symposium on Distributed Computing (DISC 2020)*, pages 25:1–25:17, 2020. doi:10.4230/LIPIcs.DISC.2020.25.

9 Shir Cohen, Idit Keidar, and Alexander Spiegelman. Brief announcement: Make every word count: Adaptive byzantine agreement with fewer words. In Alessia Milani and Philipp Woelfel, editors, *41st ACM PODC*, pages 421–423. ACM, July 2022. doi:10.1145/3519270.3538458.

10 Andrei Constantinescu, Marc Dufay, Anton Paramonov, and Roger Wattenhofer. From few to many faults: Adaptive byzantine agreement with optimal communication, 2025. Full version of this paper. doi:10.48550/arXiv.2505.19989.

11 Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. In Robert L. Probert, Michael J. Fischer, and Nicola Santoro, editors, *1st ACM PODC*, pages 132–140. ACM, August 1982. doi:10.1145/800220.806690.

12 Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985. doi:10.1145/3149.214121.

13 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009. doi:10.1145/1538902.1538904.

14 Andrew Lewis-Pye and Ittai Abraham. Fever: optimal responsive view synchronisation. In *27th International Conference on Principles of Distributed Systems (OPODIS 2023)*, pages 14:1–14:16, 2024. doi:10.4230/LIPIcs.OPODIS.2023.14.

15 Alexander Spiegelman. In Search for an Optimal Authenticated Byzantine Agreement. In *35th International Symposium on Distributed Computing (DISC 2021)*, pages 38:1–38:19, 2021. doi:10.4230/LIPIcs.DISC.2021.38.

16 Ola Svensson. Lecture 3: Bipartite expander graphs. https://theory.epfl.ch/courses/topicstcs/Lecture3.pdf, 2021. Accessed: 2025-04-16.

17 Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan-Gueta, and Ittai Abraham. HotStuff: BFT consensus with linearity and responsiveness. In Peter Robinson and Faith Ellen, editors, *38th ACM PODC*, pages 347–356. ACM, July / August 2019. doi:10.1145/3293611.3331591.