SoK: Approximate Agreement*

Diana Ghinea[1] and Chen-Da Liu-Zhang[2]

[1] diana.ghinea@hslu.ch, Lucerne University of Applied Sciences and Arts
[2] chen-da.liuzhang@hslu.ch, Lucerne University of Applied Sciences and Arts

Abstract. Approximate Agreement ($\mathcal{AA}$) is a relaxation of consensus that requires honest parties to output values that are close and within the honest inputs' range. Introduced as a relaxation of exact consensus, $\mathcal{AA}$ has become a versatile primitive with applications from blockchain oracles to cyber-physical systems. This paper provides a systematization of knowledge (SoK) on byzantine-resilient $\mathcal{AA}$ in complete networks.
We mainly focus on the real-valued variant, and chart the feasibility frontiers in synchronous, asynchronous, and network-agnostic models. We compare protocols in terms of resilience, round complexity, and communication efficiency, while also clarifying overlooked details and gaps.
Beyond standard requirements on the outputs, we discuss stronger conditions, such as having the outputs *close* to the honest inputs' median. Moreover, we briefly situate the real-valued $\mathcal{AA}$ problem within the broader landscape of $\mathcal{AA}$, where other input domains such as higher-dimensional spaces and graphs introduce further challenges.

1 Introduction

Distributed systems frequently face the challenge of producing a common output from multiple inputs: whether maintaining synchronized clocks [59], aggregating price feeds in blockchain oracles [8], aligning gradients in distributed machine learning [23,39,40,69], or coordinating aviation control systems [60,68]. Such systems require a set of parties to reach reliable collective decisions. Yet, network asynchrony, hardware failures, and byzantine (malicious) faults bring difficulties. Byzantine Agreement ($\mathcal{BA}$) [57] has long stood as the canonical solution to this problem, ensuring that all honest parties agree on a value. However, $\mathcal{BA}$ is often either impossible (e.g., under asynchrony with byzantine faults [44]) or too strict for real-world tasks where slight deviations are tolerable: clocks only need to be approximately synchronized, or sensor readings may tolerate bounded imprecision. This motivates Approximate Agreement ($\mathcal{AA}$), a relaxation of $\mathcal{BA}$ introduced in [34] that requires outputs to be close, i.e., within a predefined error $\varepsilon > 0$, while remaining inside the range of honest inputs even if t out of the n parties involved are byzantine. Beyond its role as a relaxation of $\mathcal{BA}$, $\mathcal{AA}$ has been deployed (often implicitly) as a building block in practical distributed systems. Oracle networks are a prominent example. These are designed to aggregate data (e.g., prices) from multiple, potentially unreliable sources and output a single value for consumption by smart contracts [8, 21, 25].

* The full version of the paper can be found in [50].

This paper provides a *systematization of knowledge (SoK)* on byzantine-resilient $\mathcal{AA}$ in complete networks. Our focus is primarily on the real-valued variant, which has been the core setting of most of the literature and remains the best understood. We chart the feasibility frontiers in synchronous, asynchronous, and network-agnostic models, compare known protocols in terms of resilience, round complexity, and communication cost, and highlight common techniques and lower bounds. Table 1 summarizes the prior results we mainly focus on. Along the way, we fix several gaps in the literature, clarifying subtle points where existing results were incomplete or left implicit. Afterwards, we situate the real-valued case within the broader landscape of $\mathcal{AA}$, where other input domains such as higher-dimensional spaces and graphs introduce further challenges. Finally, we briefly describe $\mathcal{AA}$ from the practical lens of decentralized oracle networks. By unifying established results, addressing overlooked details, and identifying the gaps that remain, our goal is to provide both a reference for the theory of $\mathcal{AA}$ and a foundation for future work on $\mathcal{AA}$-based distributed protocols.

Table 1: Summary of real-valued $\mathcal{AA}$ protocols in prior works. In the model column, S refers to synchronous, A to asynchronous, and NA to network-agnostic. In the round and message complexities, D denotes the honest inputs' range size, M the honest inputs' magnitude (largest honest input in absolute value), and ε the agreement parameter.

Model	Resilience	Rounds	Messages	Ref.
S	$t < n/3^{\dagger}$	$O(\log(\mathsf{D}/\varepsilon))^{\S}$	$O\big(n^2 \cdot \log(\mathsf{D}/\varepsilon)\big)$	[34]
	$t < n/4$	$O\Big(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\Big)^{\ddagger\S}$	$O\Big(n^{\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}}\Big)$	[42]
	$t < n/3^{\dagger}$	$O\Big(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\Big)^{\ddagger}$	$O\Big(n^3 \cdot \frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\Big)$	[13]
S*	$t < n/2^{\dagger}$	$O(\log(\mathsf{D}/\varepsilon))^{\S}$	$O\big(n^4 \cdot \log(\mathsf{D}/\varepsilon)\big)$	[51, 59]
	$t = \big(\frac{1-c}{2}\big) \cdot n$ (where c is a constant)	$O\Big(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\Big)^{\ddagger}$	$O\Big(n^4 \cdot \frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\Big)$	[49]
A	$t < n/5$	$O(\log(\mathsf{D}/\varepsilon))^{\ddagger\S}$	$O\big(n^2 \cdot \log(\mathsf{D}/\varepsilon)\big)$	[34]
	$t < n/3^{\dagger}$	$O(\log(\mathsf{D}/\varepsilon))$	$O\big(n^4 + n^3 \cdot \log(\mathsf{D}/\varepsilon)\big)$	[1]
	$t < n/3^{\dagger}$	$O(\log(\mathsf{M}/\varepsilon))$	$O(n^2 \cdot \log(\mathsf{M}/\varepsilon))$	[65]
NA*	$n > 2 \cdot t_s + t_a{}^{\dagger}$	$O(\log(\mathsf{D}/\varepsilon))^{\S}$	$O\big(n^4 \cdot \log(\mathsf{D}/\varepsilon)\big)$	[51]

$\star$ Assuming a public-key infrastructure and digital signatures
† Optimal resilience
‡ Asymptotically optimal round complexity according to [41, 42]
§ Protocol terminates within the stated round complexity assuming knowledge of D.

2 Preliminaries

Model. We consider a setting of n parties $P_1, P_2, \ldots, P_n$ that are running a protocol in a fully-connected network, where links model authenticated channels.

Throughout the paper, we will make different assumptions about the network. Section 3 assumes a synchronous network: the parties' clocks are synchronized and every message is delivered within a publicly known amount of time Δ. Section 4 considers an asynchronous network: the only assumption is that messages get delivered eventually. Finally, Section 5 discusses results in the network-agnostic model: this is a bridge between the synchronous model and the asynchronous one, assuming that the network may be synchronous, or asynchronous, and the parties are not aware of what kind of network they are in.

Adversary. The protocols we discuss are secure against an adaptive adversary, which may corrupt at any point of the protocol's execution up to t of the n parties involved. Corrupted parties become byzantine: they may deviate arbitrarily (maliciously) from the protocol. In the network-agnostic model, we distinguish between the number of parties t_s that the adversary may corrupt when the network is synchronous, and the number of parties $t_a \leq t_s$ that the adversary may corrupt when the network is asynchronous.

While the protocols discussed assume an adaptive adversary, the impossibility results we include hold even against a static adversary, which has to choose which parties to corrupt at the beginning of the protocol's execution. Note that this is a weaker adversarial model, leading to stronger impossibility results.

Cryptographic assumptions. Some of the protocols we discuss rely on a public key infrastructure and a secure digital signature scheme. Such settings are called *authenticated*, while settings without cryptographic assumptions are called *unauthenticated*. For simplicity of presentation, we assume that signatures are unforgeable. When replaced with real-world instantiations, the results presented in authenticated settings still hold, except for negligible probability (in the scheme's security parameter) against computationally-bounded adversaries.

Approximate Agreement definition. We first recall the definition of $\mathcal{BA}$ [57].

Definition 1. *Let Π be an n-party protocol in which every party holds a value as input. We say that Π achieves $\mathcal{BA}$ if the following hold, even when up to t of the n parties involved are corrupted:* (Termination) *Every honest party produces an output;* (Validity) *If all honest parties hold the same input v, no honest party outputs $v' \neq v$.* (Agreement) *If two honest parties output v and v', then $v = v'$.*

We include the definition of $\mathcal{AA}$ [34]. $\mathcal{AA}$ relaxes the Agreement property by allowing a predefined error in the honest parties' outputs. The Validity condition, however, becomes stronger: the honest outputs must be in the honest inputs' range (between the lowest and the highest honest inputs). Note that this implies $\mathcal{BA}$'s Validity condition. We add that a variant of $\mathcal{BA}$ that requires the stronger Validity condition of $\mathcal{AA}$ also exists, under the name of Convex Agreement ($\mathcal{CA}$) [30, 53, 71]. $\mathcal{AA}$ is also closely related to the notion of Proxcensus [45, 49], which is a degraded version of $\mathcal{BA}$ with a weaker agreement condition, but with the same validity as $\mathcal{BA}$. Proxcensus can be seen as accompanying the output value with a grade value to express the degree of agreement achieved, generalizing notions such as crusader agreement and graded consensus [43].

Definition 2. *Let $\Pi(\varepsilon)$ be an n-party protocol in which every party holds a value in $\mathbb{R}$ as input. We say that $\Pi(\varepsilon)$ achieves $\mathcal{AA}$ if the following hold for any predefined $\varepsilon > 0$, even when up to t of the n parties involved are corrupted:* (Termination) *Every honest party produces an output in $\mathbb{R}$;* (Validity) *Honest parties' outputs lie within the range of the honest inputs;* (ε-Agreement) *If two honest parties output v and v', then $|v - v'| \leq \varepsilon$.*

Additional notations. In the remainder of the paper, we use the term *valid value* to refer to a value within the range of the honest inputs. We also add that we use $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ to denote the floor and ceiling functions, respectively.

3 $\mathcal{AA}$ in the Synchronous Model

In this section, we focus on $\mathcal{AA}$ in the synchronous model. We start with characterizing synchronous $\mathcal{AA}$ from a resilience perspective. Afterwards, we delve into efficiency metrics: round complexity and communication complexity.

3.1 Feasibility

We begin by showing that, similarly to $\mathcal{BA}$, $\mathcal{AA}$ requires $t < n/2$. We rely on a straightforward scenario argument.

Theorem 1. *There is no protocol achieving $\mathcal{AA}$ when $t \geq n/2$.*

Proof (Sketch). If $t \geq n/2$, we may split the parties into two sets of size at most t each: $S_{-\varepsilon}$ with input $-\varepsilon$, and $S_{+\varepsilon}$ with input $+\varepsilon$. An honest party in $S_{-\varepsilon}$ cannot distinguish between $S_{+\varepsilon}$ containing honest parties and $S_{+\varepsilon}$ being fully byzantine, following the protocol correctly with input $+\varepsilon$. By Validity and Termination, it must output $-\varepsilon$. A symmetric argument shows that an honest party in $S_{+\varepsilon}$ must output $+\varepsilon$. Hence, in a setting where both $S_{-\varepsilon}$ and $S_{+\varepsilon}$ contain honest parties, the honest outputs differ by $2 \cdot \varepsilon$, breaking ε-Agreement.

Note that the argument behind Theorem 1 does not rely on a complex adversarial strategy: the result follows from the mere presence of byzantine parties and their effect on the Validity condition. In contrast, unauthenticated settings enable more complex adversarial strategies which lead to stronger requirements. The result below, noted in [34], follows the proof of [57] on $n > 3t$ being necessary for $\mathcal{BA}$ in unauthenticated settings.

Theorem 2. *There is no protocol achieving $\mathcal{AA}$ in an unauthenticated setting when $t \geq n/3$.*

A simple protocol with $t < n/3$. The work of [34] presents an elegant protocol that matches the lower bound of Theorem 2, proceeding in iterations that gradually bring the honest parties' values closer. The parties join each iteration $\mathtt{it} \geq 1$ with values $v_{\mathtt{it}-1}$, and obtain new values $v_{\mathtt{it}}$ such that: (i) the values $v_{\mathtt{it}}$

are within the range of honest values $v_{\mathtt{it}-1}$, (ii) the values $v_{\mathtt{it}}$ are closer than the honest values $v_{\mathtt{it}-1}$. Validity always holds and, after sufficiently many iterations, ε-Agreement holds. This outline is followed by most $\mathcal{AA}$ protocols, even under different network assumptions, and even in variants on other input spaces.

Each iteration consists of three steps. First, every party announces its value $v_{\mathtt{it}-1}$ via a distributing mechanism. In the synchronous model with up to $t < n/3$ corruptions, simply sending $v_{\mathtt{it}}$ to all parties suffices. Each party then receives $n - t$ values $v_{\mathtt{it}}$ from honest parties, plus up to t values from byzantine parties. The honest parties fill in the missing values with a default value, e.g., 0. Then, out of the n values a party holds, up to t are outside the range of honest values $v_{\mathtt{it}-1}$: after discarding the lowest t and the highest t of these values, all remaining values lie within the range of honest values $v_{\mathtt{it}-1}$. Note that there are at least $n - 2t > t + 1$ values remaining. Finally, each party computes its new value $v_{\mathtt{it}}$ as the average of the undiscarded values: this is also within the range of honest values $v_{\mathtt{it}-1}$, which guarantees Validity. The honest parties' multisets of undiscarded values, in fact, pair-wise intersect: this implies that the new values get closer. Precisely, they get closer by a factor of $\left(\lceil \frac{n-2t-1}{t} \rceil + 1\right)^{-1}$: as $n > 3t$, the honest values' range gets halved. Consequently, if the honest inputs are D-close, ε-Agreement is achieved within $O\left(\log_{\lceil \frac{n-2t}{t} \rceil}(\mathsf{D}/\varepsilon)\right)$ iterations.

Detecting when ε-Agreement is reached. Given an upper bound D on the honest inputs' range size (which is ensured by common data types, e.g., `float32`), the protocol described so far already achieves $\mathcal{AA}$: we may use D to fix a sufficient number of iterations. Alternatively, we may enable the parties to estimate when ε-Agreement is reached. The protocol of [34] proposes a mechanism where each party estimates a sufficient number of iterations through an exchange preceding the first iteration: the parties announce their input value v_{IN}, and each party may consider the multiset of values it has received, denoted by V_{EST}, as an over-estimation of the honest inputs' range. Then, each party estimates a number of sufficient iterations $I_{\text{EST}} := \log_{\lceil \frac{n-2t}{t} \rceil}((\max V_{\text{EST}} - \min V_{\text{EST}})/\varepsilon)$: as V_{EST} includes the honest inputs, this is sufficient for the honest inputs as well. This way, the parties may join the first iteration with value $v_0 := v_{\text{IN}}$. As the honest parties may obtain different estimations I_{EST}, each party announces when its own estimation I_{EST} was reached. As one cannot distinguish between an honest party and a byzantine party making such an announcement, the parties wait until they see $t + 1$ such messages – until that point, they keep executing iterations assuming that parties that sent halting announcements maintained their values.

A caveat left open in [34] is that the runtime is, in fact, chosen by the adversary: byzantine parties may choose which values to send in the initial exchange. This issue was closed by the asynchronous $\mathcal{AA}$ protocol of [1], which proposes a mechanism that provides each party with a valid value v_0 (which may be different from its input) and a multiset V_{EST} such that (i) every honest party's value v_0 is in $[\min V_{\text{EST}}, \max V_{\text{EST}}]$, and (ii) V_{EST} is included in the honest inputs' range. Roughly, these properties are achieved through an initial exchange of values as well. However, the mechanism involves an additional step where the parties re-

port the values they have received. Then, each party defines a valid value v_0 for every party based on the values reported, and these values v_0 compose the multisets V_{EST}. Afterwards, similarly to [34], the parties estimate a sufficient number of iterations based on their multisets V_{EST}, and announce when they believe a sufficient number of iterations was reached. This mechanism leads to the result below.

Theorem 3 ([34], [1]). *There is a deterministic synchronous protocol achieving $\mathcal{AA}$ even when up to $t < n/3$ of the n parties involved are byzantine. For $\varepsilon > 0$ denoting the agreement parameter and D an upper bound on the honest inputs' range size, the protocol has round complexity $O(\log_{\lceil \frac{n-2t}{t} \rceil}(D/\varepsilon))$ and message complexity $O(n^4 + \log_{\lceil \frac{n-2t}{t} \rceil}(D/\varepsilon) \cdot n^2)$.*

A simple protocol in the authenticated setting, with $t < n/2$. We now consider the authenticated setting, and we show that Theorem 1 is tight. We present a simple protocol that follows from [51, 59], which we denote by $\mathsf{AA}^{\mathsf{auth}}$.

While $\mathsf{AA}^{\mathsf{auth}}$ follows the iteration-based outline of [34], increasing the resilience threshold from $t < n/3$ to $t < n/2$ introduces a few challenges. First, filling in missing values may prevent convergence. However, not filling in missing values disables the prior strategy for discarding outliers: discarding the lowest t and the highest t values received may lead to no values remaining. Instead, we may rely on the synchronous model's properties: honest values are received within a known amount of time: if a party receives $n-t+k$ values, with $0 \leq k \leq t$, at least $n - t$ of these values are honest, and discarding the lowest k and the highest k values received is sufficient.

The byzantine parties may still prevent the honest values from converging by sending different values: this can be avoided using digital signatures. Instead of simply sending their current values, each party may send its *signed* value to all parties. Then, the parties could exchange the signed values, and identify any attempts of equivocation. Hence, if two honest parties consider a value sent by party P, they consider the same value. This can be implemented as a *Weak Broadcast* ($\mathcal{WBC}$) protocol [33].

Then, the parties compute their new values as the average between the lowest and the highest undiscarded values – these are in the range of honest values at the start of the iteration. As opposed to the $t < n/3$ setting, now it is possible that the parties' multisets of undiscarded values do not intersect. However, the ranges do pair-wise intersect – this follows from [51, Lemma 3.7], hence the honest values' range is halved.

We still need to discuss Termination: the works of [51, 59] assume an upper bound on the honest inputs' range size. The mechanism proposed in [34] can be used directly, ensuring termination in a finite amount of time. The mechanism of [1] can also be adapted – however, the property that the multisets V_{EST} are included in the honest inputs' range is lost: byzantine parties' reported values may contain a corrupted majority, leading to values v_0 outside the honest inputs' range, and hence to an adversarially-chosen round complexity. Instead, we may

rely on the mechanism of [12], where the parties observe during the iterations that ε-Agreement was reached – with the cost of one more round per iteration. We replace $\mathcal{WBC}$ with Graded Broadcast ($\mathcal{GBC}$) [12,63]: instead of only receiving values (or $\perp$/nothing), the parties receive values with grades in $\{0, 1, 2\}$. Grades differ by at most 1, and, if the sender is honest, all parties receive its value with grade 2. Moreover, if two parties get v, v' with grades at least 1, $v = v'$. Using this mechanism, the parties compute their new values based on the values received with grade at least 1. Once a party P observes $n - t$ values with grade 2 that are ε-close, all parties have received these values with grade at least 1: their new values are in the range defined by these ε-close values. Then, in the next iteration, all parties observe that ε-Agreement was reached.

$\mathsf{AA}^{\mathsf{auth}}$

Code for party P with input v_{in}

1: `done` := `false`, $v_0 := v_{\text{IN}}$.
2: **For** `it` $= 1 \ldots \infty$:
3: Send $v_{\texttt{it}-1}$ via $\mathcal{GBC}$. If `done` := `true`, output $v_{\texttt{it}-1}$ and terminate.
4: Let V := the multiset of values received with grade at least 1.
5: If $n - t$ values received with grade 2 are ε-close, `done` := `true`.
6: Let S := the multiset obtained by discarding the lowest k and highest k values from V, where $k := \big|V\big| - (n - t)$, and let $v_{\texttt{it}} := \frac{1}{2} \cdot (\min S + \max S)$.

The next theorem (proven in [50]) states the guarantees of $\mathsf{AA}^{\mathsf{auth}}$.

Theorem 4. *There is a deterministic synchronous protocol achieving $\mathcal{AA}$ in an authenticated setting even when up to $t < n/2$ of the n parties involved are byzantine. For $\varepsilon > 0$ denoting the agreement parameter, and D an upper bound on the honest inputs' range size, the protocol has round complexity $O(\log(\mathsf{D}/\varepsilon))$ and message complexity $O(\log(\mathsf{D}/\varepsilon) \cdot n^4)$.*

3.2 Round Complexity

So far, we have discussed tight conditions regarding resilience, and we now shift our focus towards optimal round complexity. The protocols described so far require a number of rounds that is logarithmic in D/ε, where D is the honest inputs' range size. Although this round complexity is efficient, it is not optimal.

Lower bound. The work of [34] has addressed the optimality of their protocol in terms of *convergence factor*, i.e., *how close the honest parties' values can get after a fixed number of rounds*, and provided a lower bound for a particular class of protocols in *standard form*: in every iteration, the parties send their current values, and each party computes its new value as a function of the values received. Within this class, the protocol of [34] is optimal: no other function for computing a new value performs better. Fekete [42] has later provided a lower bound for the general case, stated below, leaving a gap in contrast to the protocol of [34].

Theorem 5 (Theorem 15 of [42]). *Let Π be an arbitrary deterministic R-round protocol that satisfies Validity and Termination on $[a, b] \subseteq \mathbb{R}$ even when up to t of the n parties involved are byzantine. Then, there is an execution of Π where the honest inputs are in $\{a, b\}$ and two honest parties output values v and v' satisfying $|v - v'| \geq \mathsf{D} \cdot K(R)$, where $K(R) \geq \frac{t^R}{R^R \cdot (n+t)^R}$ and $\mathsf{D} = b - a$.*

As described in [47], the proof of Theorem 5 relies on constructing a chain at most $s \leq \frac{(n+t)^R}{t_1 \cdot t_2 \cdot \ldots \cdot t_R}$ views for each distribution $\{t_1, t_2, \ldots, t_R \in \mathbb{N} : t_1 + t_2 + \cdots + t_R \leq t\}$. Roughly, t_i represents the number of byzantine parties that try to cause inconsistencies in the honest parties' views for the first time in round i. For any pair of views that are consecutive in the chain, there is an execution where two honest parties obtain those views. The chain is constructed such that, by Validity, the first view leads to output a, and the last view leads to output b. This implies the existence of two consecutive views in the chain that yield two honest outputs $v, v' \in \mathbb{R}$ with $|v - v'| \geq \mathsf{D}/s \geq \mathsf{D} \cdot K(R)$. The work of [47] restates Theorem 5 in terms of round complexity, as described by the corollary below. Note that this result is somewhat orthogonal to the well-known requirement of $t + 1$ rounds for deterministic $\mathcal{BA}$ [35].

Corollary 1 (Theorem 2 of [47]). *Assume that $n, t > 0$. Then, any deterministic protocol that achieves $\mathcal{AA}$ on $[a, b] \subseteq \mathbb{R}$ even when up to t of the n parties are byzantine requires at least $\Omega\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon) + \log\frac{n+t}{t}}\right)$ rounds, where $\mathsf{D} = b - a$.*

A first round-optimal protocol. Fekete [42] has also proved the lower bound of Theorem 5 to be asymptotically tight. Assuming $n = c \cdot t$ for a constant $c < 1/4$, [42] proposed a fixed-round full-information protocol – the parties send their complete view to all parties in each round, leading to a message complexity of $O(n^R)$ for R rounds. The parties then make a decision based on their final view: this allows them to identify byzantine parties that have deviated from the protocol, enabling a better convergence factor than that of [34].

Gradecast-based approach. Later, [12] has presented an $\mathcal{AA}$ protocol that asymptotically matches Theorem 5, with optimal resilience $t < n/3$, and efficient message complexity $O(R \cdot n^3)$ for R rounds. Internally, the protocol of [12] operates in iterations, similarly to [34], but allows the parties to identify byzantine parties that misbehave. In each iteration, the parties distribute their current values via $\mathcal{GBC}$. Each party fills in the missing values with some default value, and then discards the lowest t and the highest t values. Afterwards, it computes its new value as the average of undiscarded values. If a byzantine party $\tilde{P}$ attempts to introduce inconsistencies in the honest parties' views during iteration $\mathtt{it}$, the honest parties detect this (no honest party obtains grade 2 in $\tilde{P}$'s $\mathcal{GBC}$ invocation). The parties mark such parties as corrupted, and do not actively participate in any further $\mathcal{GBC}$ invocations having these parties as senders: $\tilde{P}$ will be ignored in all further iterations. Then, if $t_{\mathtt{it}}$ out of the t byzantine parties cause inconsistencies in the values received by honest parties in iteration $\mathtt{it}$, the protocol ensures that the honest values' range shrinks by a multiplicative factor of $\frac{t_{\mathtt{it}}}{n-2t}$. As a result, if

the honest inputs are D-close, their values become $\left(\frac{t^R}{R^R \cdot (n-2t)^R} \cdot \mathsf{D}\right)$-close after R iterations. The following theorem comes from [47], and extends the analysis of [12] from a fixed $\varepsilon := 1/n$ to any $\varepsilon > 0$.

Theorem 6 (Theorem 3 of [47]). *There is a deterministic synchronous protocol achieving $\mathcal{AA}$ on real values when even up to $t < n/3$ of the n parties involved are byzantine. For $\varepsilon > 0$ denoting the agreement parameter and D an upper bound on the honest inputs' range size, the protocol has round complexity $O\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\right)$ and message complexity $O\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)} \cdot n^3\right)$.*

Matching the lower bound under an honest majority. Theorem 5 is also asymptotically tight under an honest majority in authenticated settings. As pointed out in [47], with a few adjustments, the Proxcensus protocol of [49] becomes an $\mathcal{AA}$ protocol that matches Theorem 5 for $t = c \cdot n$ corruptions, for any constant $c \in [1/3, 1/2)$ – given an upper bound D on the honest inputs' range size. Roughly, the protocol of [49] operates in iterations enabling the honest parties' values to converge. It relies on $\mathcal{GBC}$ to identify corruptions similarly to [12], while also integrating the insights we have discussed for the $t < n/2$ setting. This leads to an $\mathcal{AA}$ protocol that, after R iterations, brings the honest values closer by a factor of $\frac{t^R}{R^R \cdot (n-2t)^R}$, asymptotically matching Theorem 5. As stated in [47, Theorem 9], ε-Agreement is reached within $O(1) \cdot \frac{\log_2(\mathsf{D}/\varepsilon)}{\log_2\left((1-2c)/c\right) + \log_2\log_2(\mathsf{D}/\varepsilon)}$ rounds.

We need, however, to highlight a few technical details. If $n - 2t \in o(n)$, the number of sufficient rounds depends on t. Whether there is a tighter analysis for the protocol of [49] or the lower bound of Theorem 5 can be improved is an open question. Another observation, pointed out in [47], is that $\log_2\left((1-2c)/c\right) \leq 0$ for any $c \in [1/3, 1/2)$, and hence the current analysis requires D/ε to be sufficiently large for a meaningful round complexity. To circumvent this issue, as discussed in [47], we may rely on the protocol described in Theorem 4 if $\log_2\left((1-2c)/c\right) + \log_2\log_2(\mathsf{D}/\varepsilon) \leq 0$ and hence $\mathsf{D}/\varepsilon = O(1)$. In this case, the protocol of Theorem 4 requires $O(1)$ rounds.

Theorem 7 (Theorem 8 of [47]). *If the honest inputs are D-close real values and c is a constant in $[1/3, 1/2)$, there is a deterministic synchronous protocol achieving $\mathcal{AA}$ in an authenticated setting even when up to $t = c \cdot n$ of the n parties are byzantine. If $\varepsilon > 0$ denotes the agreement parameter, the protocol has round complexity $O\left(\frac{\log_2(\mathsf{D}/\varepsilon)}{\log_2\log_2(\mathsf{D}/\varepsilon)}\right)$ and message complexity $O\left(\frac{\log_2(\mathsf{D}/\varepsilon)}{\log_2\log_2(\mathsf{D}/\varepsilon)} \cdot n^4\right)$.*

We add that one can remove the assumption that D is known in the adjusted protocol of [49] by adapting the mechanism of [12] to the honest-majority setting, as we have described for $\mathsf{AA}^{\mathsf{auth}}$. However, this approach prevents us from relying on another protocol if D/ε is constant. We leave as an open problem whether the adjusted Proxcensus protocol of [49] achieves, in fact, ε-Agreement within $O(1)$ rounds when D/ε is constant, or whether it is possible to design a termination mechanism that enables us to combine the protocol of [49] with another protocol.

We conclude the subsection by noting that each of the solutions we have discussed matches Theorem 5 and Corollary 1 only asymptotically. Whether the lower bounds can be improved, or whether one can design protocols with improved convergence guarantees is an open question.

3.3 Communication Complexity

Optimal communication complexity for $\mathcal{AA}$ is an open question – both in terms of message complexity, and in terms of bit complexity. Below we describe the existing results. We note that lower bounds from the literature of $\mathcal{BA}$ [2, 15, 35] imply that $\mathcal{AA}$ requires at least $O(n^2)$ messages overall.[3]

The protocol of [34] incurs $O(n^2)$ messages *per iteration*, where each message is a real value – taking into account the (efficient) termination mechanism we have described for Theorem 3, this leads to $O(n^4 + \log(\mathsf{D}/\varepsilon) \cdot n^2)$ messages, where each message is a real value. $\mathcal{GBC}$-based solutions involve n invocations of $\mathcal{GBC}$ per iteration, hence our protocol $\mathsf{AA}^{\mathsf{auth}}$ has a message complexity of $O(\log(\mathsf{D}/\varepsilon) \cdot n^4)$. The optimal-round protocols reach $O\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)} \cdot n^3\right)$ messages in the $t < n/3$ setting, and $O\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)} \cdot n^4\right)$ in the $t < n/2$ setting. Note that, if $n \in \Omega\left(\frac{\log(\mathsf{D}/\varepsilon)}{\log\log(\mathsf{D}/\varepsilon)}\right)$, the protocol of [34] achieves a lower message complexity than that of the round-optimal protocol of [12]. It would be interesting to see whether there is a trade-off between round complexity and communication complexity. Such trade-off is also suggested by the $\mathcal{CA}$ protocol of [53], which achieves asymptotically optimal bit complexity within $O(n \log n)$ rounds, in contrast to $t + 1$ rounds for higher-communication protocols.

We add that each of the messages contains a real value within the honest inputs' range. The concrete length of these messages may be, in fact, adversarially chosen (as the byzantine parties can send values in the honest range, but with arbitrary precision). Prior works have investigated approaches for reducing the bit complexity, i.e., [65] achieves (asynchronous) $\mathcal{AA}$ resilient against $t < n/3$ corruptions with round complexity $O(\log(\mathsf{M}/\varepsilon))$ and bit complexity $O(n^2 \cdot \log(\mathsf{M}/\varepsilon) \cdot \log\log(\mathsf{M}/\varepsilon))$, where M denotes the honest inputs' magnitude (i.e., the largest honest input in absolute value). We add that, in the synchronous model, we can achieve $\mathcal{AA}$ within $O(n \log n)$ rounds and with bit complexity $O(n \cdot \log(\mathsf{M}/\varepsilon) + \kappa \cdot n^2 \log^2 n)$ assuming a hash function with security parameter κ through a simple reduction to the protocol of [53] (see [50]).

4 $\mathcal{AA}$ in the Asynchronous Model

We now consider the other extreme in terms of network assumptions: the asynchronous model, which only assumes that messages get delivered eventually. Asynchronous protocols are therefore robust to network issues, in contrast to

[3] For $\mathcal{BA}$, randomized protocols with subquadratic communication are possible with static and weakly-adaptive adversaries [2, 14, 16, 55].

synchronous protocols. However, this comes with a cost regarding resilience: in the asynchronous model, parties cannot distinguish between honest parties whose messages are simply delayed and byzantine parties that have not sent those messages. This discards the property that *all honest values are delivered*, which played a key role in achieving $\mathcal{AA}$ even up to $t \in [n/3, n/2)$ corruptions.

4.1 Feasibility

Asynchronous $\mathcal{AA}$ requires $t < n/3$: in contrast to Theorem 2, this holds even in authenticated settings. Roughly, the messages of t honest parties may be delayed for a long time, and these honest parties are essentially replaced by t byzantine parties. If $n \leq 3t$, the remaining at most t honest parties will be forced to complete the protocol in a byzantine-majority setting. Then, similarly to the arguments behind Theorem 1, by trying to achieve Validity, ε-Agreement is broken.

Theorem 8. *There is no asynchronous protocol achieving $\mathcal{AA}$ when $t \geq n/3$.*

We note that this requirement applies to randomized asynchronous $\mathcal{BA}$ protocols as well [70]. On the other hand, there is no deterministic asynchronous protocol achieving $\mathcal{BA}$ even up to $t = 1$ corruptions [44].

A first asynchronous $\mathcal{AA}$ protocol. Along with introducing the $\mathcal{AA}$ problem and presenting its characterization in the synchronous model, [34] has also investigated its feasibility in the asynchronous model. Their asynchronous $\mathcal{AA}$ protocol requires $t > n/5$ and follows the synchronous one with minor adjustments: in each iteration, instead of waiting for one round to ensure that all honest parties' messages are received, the parties wait for $n - t$ messages. Afterwards, each party discards the lowest t and the highest t values received, and computes its new value as the average of the undiscarded values, leading to the honest values getting closer by a factor of $(\lfloor (n - 3t - 1)/2t \rfloor + 1)^{-1}$.

First approach for optimal resilience. Coan [26] has presented a compiler transforming a crash-resilient protocol into a byzantine-resilient protocol, leading to an asynchronous $\mathcal{AA}$ protocol resilient against $t < n/3$ corruptions. However, the resulting protocol comes with a cost in terms of how fast the honest values converge – from $(\lfloor (n - 3t - 1)/2t \rfloor + 1)^{-1}$ in the protocol of [34] to $\lfloor (n-1)/t \rfloor^{-1/3}$. We note that the protocol of [26] does not include a termination mechanism.

In the remainder of the subsection, we will be building towards the optimal-resilience solution of [1], which halves the honest values' range in each iteration.

An immediate improvement over [34]. A reason why $t < n/5$ is required by the asynchronous protocol of [34] is that the byzantine parties may send different values to different parties. As pointed out in [1], this can be easily avoided if the parties send their values via Reliable Broadcast ($\mathcal{RBC}$) [20]. However, this would only lead to resilience up to $t < n/4$ byzantine corruptions: while $\mathcal{RBC}$ ensures that the honest parties do not receive different values from the same byzantine

party, the asynchronous model still introduces challenges. If the honest parties wait until they receive $n-t$ values via $\mathcal{RBC}$ before computing new values, we may lose the guarantee that their ranges of undiscarded values have some intersection, which may prevent the honest parties' values from converging. Indeed, $\mathcal{RBC}$ ensures that the honest parties receive sufficient values to guarantee intersection *eventually* – the honest parties would need to wait *long enough*.

Optimal resilience via the Witness Technique. The work of [1] has introduced a mechanism called *the Witness Technique*, which forces the honest parties to wait *long enough* so that their ranges of undiscarded values indeed intersect. The parties report the first $n-t$ values received via $\mathcal{RBC}$, but keep waiting for later values. Once a party receives all $n-t$ values reported by some party P, it marks P as a witness. A party has waited *long enough* for values once it has marked $n-t$ parties as witnesses. This will ensure that each pair of parties has a common honest witness, and hence they have received $n-t$ common values. This enables [1] to achieve $\mathcal{AA}$ when up to $t < n/3$ of the parties involved are byzantine: due to the large pair-wise overlap in the multisets of the values obtained by the honest parties through the Witness Technique, they have common values after discarding the lowest t and the highest t values as well. Then, they compute their new values as the average between the lowest and the highest undiscarded values, which ensures that the honest parties' values stay in the honest inputs' range, and that honest values' range is halved with each iteration. If the honest inputs are D-close, $O(\log(\mathsf{D}/\varepsilon))$ iterations are sufficient to achieve $\mathcal{AA}$.

Termination mechanism. As described in Section 3, the protocol of [1] starts with an exchange where each party (i) estimates a multiset V_{EST} within the honest inputs' range, and (ii) obtains a (valid) value v_0 that is in every honest party's $[\min V_{\text{EST}}, \max V_{\text{EST}}]$. The parties then join the first iteration using their values v_0, and send halting messages announcing the number of iterations they believe to be sufficient. Once a party notes that the iteration corresponding to the $(t+1)$-th lowest number announced was executed, it outputs the value obtained in that iteration, as ε-Agreement was reached. This leads to the result below.

Theorem 9 (Theorem 2 of [1]). *There is a deterministic asynchronous protocol achieving $\mathcal{AA}$ even when up to $t < n/3$ of the n parties are byzantine. If $\varepsilon > 0$ is the agreement parameter and D an upper bound on the honest inputs' range size, the protocol has round complexity $O(\log(\mathsf{D}/\varepsilon))$ and message complexity $O(n^4 + \log(\mathsf{D}/\varepsilon) \cdot n^3)$.*

A technical note. If we instantiate the $\mathcal{RBC}$ protocol with Bracha's protocol [20] in the $\mathcal{AA}$ protocol of [1], the honest parties need to continue actively participating in prior iteration's $\mathcal{RBC}$ invocations even after they obtain an output. This is needed to ensure that honest parties that are still in earlier iterations indeed receive sufficient values via $\mathcal{RBC}$ to move forward. This issue is discussed in detail in the work of [64], which proposes an alternative to the Witness Technique that enables the honest parties to terminate upon obtaining an output.

4.2 Round Complexity

While Corollary 1 already gives us a lower bound, it leaves a gap in contrast to the round complexity of the protocols described in this section. Below we first present a stronger lower bound, which comes from [41], describing how close the honest parties' values may get after R asynchronous rounds of communication, given that up to t of the parties involved may exhibit omission failures. This is a weaker adversarial model in comparison to byzantine failures: parties exhibiting omission failures may choose whether to send a message or not, but otherwise follow the protocol correctly. Hence, the bound described in Theorem 10 holds for byzantine failures as well. The term *canonical-round* in the theorem statement refers to protocols that can be written in *full-information form*: in (asynchronous) round r, parties send their views from round $r-1$, and, upon receiving $n-t$ such views in round r, a party may advance to round $r+1$.

Theorem 10 (Theorem 2 of [41]). *Let Π be a deterministic canonical-round R-round protocol that satisfies Validity and Termination on $[a,b] \subseteq \mathbb{R}$ in the asynchronous model even when up to t of the n parties involved may exhibit omission failures. Then, there is an execution of Π where the honest inputs are in $\{a,b\}$ and two honest parties output values v and v' satisfying $|v-v'| \geq D \cdot K(R)$, where $K(R) \geq \left\lceil \frac{n-t}{t} \right\rceil^{-R}$ and $D = b-a$.*

We may convert this result into a lower bound regarding round complexity.

Corollary 2. *Assume that $n, t > 0$. Then, any deterministic canonical-round asynchronous protocol achieving $\mathcal{AA}$ when the honest inputs are in $[a,b] \subseteq \mathbb{R}$ and even when up to t of the n parties exhibit omission failures has round complexity $\Omega\left(\log_{\frac{n}{t}}(D/\varepsilon)\right)$, where $\varepsilon > 0$ denotes the agreement parameter and $D = b-a$.*

The asynchronous $\mathcal{AA}$ protocol of [34] is already in canonical-round form. After R rounds, this protocol brings the honest parties' values closer by a factor of $\left\lceil \frac{n-3t}{2t} \right\rceil^{-R}$. By Theorem 10, this protocol achieves optimal round complexity up to constants for any $t < n/5$ – which is not optimal resilience. The recent work of Attiya, Flam and Welch [5, 6] note that the optimally-resilient protocols in [1,26] do not fit the canonical-round form, and that protocols in canonical-round form tolerating $t \in [n/5, n/3)$ corruptions must incur an unbounded number of canonical rounds. Whether the lower bound of [41] can be extended to protocols beyond canonical-form is an open problem. Such an extension would make the protocol of [1] round-optimal up to constants whenever $t/n \in \Theta(1)$.

4.3 Communication Complexity

Communication complexity does not only provide open questions in the synchronous model, but also in the asynchronous model. Regarding sufficiency, in each of the iterations of the optimal-resilience protocol of [1], the honest parties send a total of $O(n^3)$ real values, plus $O(n^3)$ bits (to report values in the Witness

Technique) – in total, this is $O(\log(\mathsf{D}/\varepsilon) \cdot n^3)$ messages. The $\mathcal{RBC}$-based protocol we have briefly described also requires the honest parties to send $O(n^3)$ real values per iteration. The communication complexity was improved by a linear factor in [8], albeit convex validity is slightly relaxed, or in [34], at the cost of tolerating up to $t < n/5$ corruptions as opposed to $t < n/3$. The recent work of [65] investigates whether $\mathcal{AA}$ can be achieved with $O(n^2)$ messages per iteration in the asynchronous model when even up to $t < n/3$ of the parties involved are byzantine, and answers this question positively, presenting a protocol with round complexity $O(\log(\mathsf{M}/\varepsilon))$ and bit complexity $O(n^2 \cdot \log(\mathsf{M}/\varepsilon) \cdot \log\log(\mathsf{M}/\varepsilon))$, where M denotes the honest inputs' magnitude (largest input in absolute value).

5 Network-Agnostic $\mathcal{AA}$

So far, we have seen that the (authenticated) synchronous model enables protocols with high resilience thresholds, but relies on strong network assumptions: sporadic network issues may cause honest messages to be delayed longer than expected, and, in such cases, a synchronous protocol may not provide any guarantees. In contrast, asynchronous protocols are robust against network issues, with the cost of lower corruption thresholds. However, the assumptions behind the asynchronous model (messages get delivered *eventually*) are very weak and somewhat too pessimistic for a real-world network. The network-agnostic model, introduced in [17], aims to combine the advantages of the two extremes, providing the *best-of-both worlds*: protocols designed in this model are meant to reach optimal corruption thresholds in the synchronous model, while also providing security when running in an asynchronous network. More precisely, the parties are unaware of whether the network is synchronous or not. Up to t_s parties may be corrupted if the network is synchronous, and up to $t_a \leq t_s$ otherwise. For $\mathcal{AA}$, this leads to the following definition, presented in [51]:

Definition 3. *Let $t_s \geq t_a$, and let $\Pi(\varepsilon)$ be an n-party protocol in which every party holds a value in $\mathbb{R}$ as input. We say that $\Pi(\varepsilon)$ is a (t_s, t_a)-secure $\mathcal{AA}$ protocol if (i) it achieves $\mathcal{AA}$ when running in a synchronous network where up to t_s of the n parties involved are byzantine; (ii) it achieves $\mathcal{AA}$ when running in an asynchronous network where up to t_a of the n parties involved are byzantine.*

5.1 Feasibility

We first discuss the necessary and sufficient conditions for (t_s, t_a)-secure $\mathcal{AA}$.

Lower bound. We may already point out two constraints: $t_s < n/2$ from Theorem 1 and $t_a < n/3$ from Theorem 8. However, these are not tight. As shown in [51], a stronger condition is needed: $n > 2 \cdot t_s + t_a$. This matches the necessary condition for network-agnostic $\mathcal{BA}$ [7, 17, 27, 32] and network-agnostic $\mathcal{CA}$ [30], providing a trade-off between the synchronous and asynchronous bounds.

Theorem 11 (Theorem 5.1 of [51]). *There is no (t_s, t_a)-secure $\mathcal{AA}$ protocol if $n \leq 2 \cdot t_s + t_a$.*

The arguments behind this theorem are similar to those of Theorem 8 and Theorem 2: if $n \leq 2 \cdot t_s + t_a$, one cannot distinguish between a synchronous execution where t_s parties are byzantine and silent, and an asynchronous execution where the messages of t_s honest parties are delayed. Then, in an asynchronous execution, the t_a byzantine parties can follow the protocol with different inputs towards the two sets of at most t_s honest parties, forcing them to output values that are valid, but not ε-close. This argument would fail in a synchronous network with digital signatures. In an asynchronous network, however, the honest parties may only receive proof of equivocation after deciding their outputs.

Protocol. The work of [51] shows that the condition $n > 2 \cdot t_s + t_a$ is also sufficient. Their protocol follows the outline of the asynchronous protocol of [1], while incorporating the insights described in Section 3. Concretely, [51] introduces a network-agnostic variant of the Witness Technique of [1], called *Overlap All-to-All Broadcast* ($\mathcal{OBC}$). On top of preserving the Witness Technique property that the parties' views have a large overlap, it also ensures that, if the network is synchronous, all honest parties' values are delivered. If party P receives $n - t_s + k$ values via $\mathcal{OBC}$, then: if the network is synchronous, up to k of these values come from byzantine parties, and, otherwise, up to t_a. To compute its new value, P discards the lowest $\max(k, t_a)$ and the highest $\max(k, t_a)$ values received, and takes the average between the lowest and the highest values remaining. This way, in each iteration, the honest parties' new values stay within the range of values they held at the start of the iteration. In addition, after discarding outliers, the *ranges* obtained by the honest parties pair-wise intersect (similarly to the honest-majority synchronous case), hence the honest values' range gets halved.

Theorem 12 (Theorem 3.1 of [51]). *Given D such that the honest inputs are D-close, there is a deterministic protocol achieving (t_s, t_a)-secure $\mathcal{AA}$ in an authenticated setting if $n > 2 \cdot t_s + t_a$. If ε denotes the agreement parameter, the protocol has round complexity $O(\log(D/\varepsilon))$ and message complexity $O(\log(D/\varepsilon) \cdot n^4)$.*

Detecting when ε-Agreement is reached. Note that Theorem 12 requires an upper bound on the honest inputs' range size. This assumption can be removed through the termination mechanism presented in [52] (by instantiating the underlying $\mathcal{RBC}$ protocol with that of [66] instead of [20]). This mechanism is similar to that of the asynchronous $\mathcal{AA}$ protocol of [1]. However, as described in [52], the resulting runtime is finite, but may be adversarially chosen: due to the higher resilience threshold t_s when the network is synchronous, similarly to the honest-majority synchronous setting, the honest parties are unable to derive valid values v_0 for the byzantine parties, hence their multisets V_{EST} are not guaranteed to be in the honest inputs' range. Designing a termination mechanism tailored to the network-agnostic model that avoids this issue is an open question.

5.2 Round Complexity

While the design of an efficient termination mechanism remains open, the question of optimal round complexity is also unresolved – even when assuming an

upper bound D on the honest inputs' range. A network-agnostic $\mathcal{AA}$ protocol cannot bypass Theorem 5 in the synchronous setting. Beyond having a suitable asynchronous lower bound regarding round complexity, it is unclear whether one can design a network-agnostic protocol exploring the advantages of synchrony to accelerate convergence in synchronous networks. This question is compelling even in unauthenticated settings: is there an $\mathcal{AA}$ protocol secure against up to $t < n/3$ corruptions that matches the lower bound of Theorem 5 when running in a synchronous network and still achieves a reasonable runtime otherwise?

5.3 Communication Complexity

Similarly to synchronous $\mathcal{AA}$ and asynchronous $\mathcal{AA}$, optimal communication complexity for network-agnostic $\mathcal{AA}$ is an open problem. Regarding sufficient conditions, we note that each iteration of [51] incurs $O(n^4)$ messages, where each message consists of a real value and a signature. We add that the message complexity per iteration can be reduced to $O(n^3)$ by using threshold signatures.

6 Stronger Validity Conditions

While $\mathcal{AA}$ ensures that the honest outputs are in the honest inputs' range, this still leaves room for adversarial bias towards extremes of this range. This can be mitigated through stronger Validity notions. For instance, δ-Median Validity [68] requires the honest outputs to be *close* to the honest inputs' median: if the honest inputs are $v_1 \leq v_2 \leq \ldots \leq v_{n-t}$ and the median is at position μ, the outputs must be within $[v_{\max(1,\mu-\delta)}, v_{\min(\mu+\delta,n-t)}]$. A more general definition is (δ, m)-Interval Validity [60]: the outputs must be in $[v_{\max(1,m-\delta)}, v_{\min(m+\delta,n-t)}]$.

Tight conditions in the synchronous model. We first discuss lower bounds regarding how close the honest outputs may be to the m-th lowest honest input in synchronous $\mathcal{AA}$ with (δ, m)-Interval Validity: this depends on the actual value of m. These lower bounds below follow from a scenario-based argument presented in [60] for exact agreement that can be easily generalized to ε-Agreement. Roughly, the byzantine parties may follow the protocol correctly with inputs that are no higher than the lowest honest input, pushing the honest parties' estimation of the m-th lowest input $\delta \approx t/2$ positions away from the m-th lowest honest input, as stated below. In addition, values of m denoting extremes of the honest inputs' range require an even stronger condition on δ.

Theorem 13 (Theorem 1, Theorem 2 of [60]). *If t of the n parties are byzantine, there is no $\mathcal{AA}$ protocol that additionally achieves (δ, m)-Interval Validity for $\delta < \lceil t/2 \rceil$ if $m \in [\lceil t/2 \rceil + 1, n - \lfloor 3t/2 \rfloor]$, and for $\delta < t$ if $m \notin [\lceil t/2 \rceil + 1, n - \lfloor 3t/2 \rfloor]$.*

To achieve $\mathcal{AA}$ with (δ, m)-Interval Validity, one may use the setup sub-protocol (Phase 1) of [60] – through an exchange of inputs, this provides each party with a value v satisfying (δ, m)-Interval Validity for $\delta < \lceil t/2 \rceil$ if $m \in$

$[\lceil t/2 \rceil + 1, n - \lfloor 3t/2 \rfloor]$, and for $\delta < t$ if $m \notin [\lceil t/2 \rceil + 1, n - \lfloor 3t/2 \rfloor]$. The parties may subsequently run any $\mathcal{AA}$ protocol with inputs v. This provides ε-close outputs within the range of honest values v, maintaining (δ, m)-Interval Validity.

Tight conditions in the asynchronous model. The asynchronous model imposes the stricter requirement of $\delta \geq t$ for δ-Median Validity. Intuitively, in the synchronous model, the byzantine parties were only able to push the honest median up to t positions *to the right* (by claiming inputs lower than the honest median). In the asynchronous model, on the other hand, t of the honest inputs may be missing, enabling the byzantine parties to also push the honest median up to t positions *to the left*. Note that, if $n = 3t + 1$, Theorem 14 implies that one cannot do better than the standard Validity definition of $\mathcal{AA}$.

Theorem 14 (Lemma 6 of [28]). *If $n > 3t$ and $\delta < t$ there is no asynchronous protocol achieving Termination and δ-Median Validity.*

Theorem 14 is tight – [28] proposes a protocol where the parties have an initial exchange of values where they obtain values satisfying δ-Median Validity for $\delta < t$, similar to that of [60]. Afterwards, the parties run an $\mathcal{AA}$ protocol.

A network-agnostic approach. Due to the stronger requirements on δ regarding δ-Median Validity in the asynchronous model in comparison to the synchronous model, the work of [28] proposes a network-agnostic protocol that matches the lower bounds of the network it runs in. This is an $\mathcal{AA}$ protocol that tolerates up to $t < n/3$ byzantine corruptions and achieves δ-Median Validity for $\delta = \lceil t/2 \rceil$ if it runs in a synchronous network for sufficiently long (i.e., throughout an initial exchange of values), and for $\delta = t$ when it runs in an asynchronous network.

7 Variants of $\mathcal{AA}$ on Other Input Spaces

Extensions of $\mathcal{AA}$ beyond real values have been explored in several settings: multidimensional real values [52, 61, 62, 71], particular classes of graphs [3, 46, 47, 58, 65, 67], and semi-lattices [67]. Given a convexity notion (i.e., straight-line convexity for multidimensional real values), the Validity requirement becomes that the honest outputs are in the honest inputs' *convex hull*. In addition, in discrete settings such as graphs, the ε-Agreement requirement may be changed to accommodate the non-zero distances achievable, e.g., 1-Agreement.

7.1 Feasibility

Protocols. Regardless of the input space considered, the iterative outline of real-valued $\mathcal{AA}$ remains a powerful tool for designing protocols. Once communication mechanisms are in place, the core difficulty becomes a local problem: the parties are given consistent views (i.e., at least $n - t$ values/inputs) that (pair-wise) intersect in at least $n - t$ values. Each such view includes at least h honest values. Every party then needs to derive a value in the honest values' convex

hull such that the new values get *closer* – defining a function for computing a new value that satisfies these properties is the core difficulty. Once such a function is defined, repeating this process over sufficiently many iterations ensures $\mathcal{AA}$. This abstraction is highlighted in [67], under the name of *abstract* $\mathcal{AA}$.

Once a party obtains its view, it can estimate a *safe area* that is included in the honest values' convex hull by intersecting the convex hulls of each subset of h values in its view: this is a generalization of discarding outliers in $\mathcal{AA}$ on $\mathbb{R}$. More complex spaces require stronger conditions to ensure that the honest parties' safe areas are non-empty. These depend on a feature of the space called *Helly number* ω, i.e., the smallest w such that: for any finite family of convex sets of the space, if every w members of the family intersect, then the whole family has a non-empty intersection. Concrete conditions are then $t < n/\omega$ in the synchronous model [30, 67, 71], $t < n/(\omega + 1)$ in the asynchronous model [30, 61, 67, 71], and $n > \max(\omega \cdot t_s, \omega \cdot t_a + t_s)$ in the network-agnostic model [30, 52].

Once the honest parties hold non-empty safe areas, we would like the safe areas to have some (pair-wise) intersection. This way, computing the new values as some form of "middle point" of the safe area (how to do so varies across spaces) implies that the honest values converge. In the synchronous and asynchronous models, the ω-based conditions for non-empty safe areas often ensure such intersection. A crucial property in these settings is that the more values a party receives, the *larger* its safe area becomes. In the network-agnostic model, this property is lost, and current solutions rely on stronger conditions. Hence, protocols for $\mathbb{R}^d$ (where $\omega = d+1$) assume $t < n/(d+1)$ in the synchronous model [71], $t < n/(d+2)$ in the asynchronous model [61, 71], $n > (d+1) \cdot t_s + t_a$ in the network-agnostic model [52]. For particular classes of graphs (chordal graphs), where $\omega :=$ the size of the largest clique, the conditions become $t < n/\omega$ in the synchronous model [67], $t < n/(\omega + 1)$ in the asynchronous model [30, 67], and $n > \omega \cdot t_s + t_a$ in the network-agnostic model [30].

Impossibility results. The conditions stated so far are *sufficient*. These are only known to be necessary for $\mathbb{R}^d$ in the synchronous and asynchronous models [61, 62, 71]. We add that one can obtain tight necessary conditions for trees (where $\omega = 2$) following the arguments discussed for $\mathbb{R}$.

The sufficient conditions for network-agnostic $\mathcal{AA}$ on $\mathbb{R}^d$ [52] and chordal graphs [30] generalize the sufficient conditions in the synchronous and asynchronous models. However, it is unknown whether these are tight. As the weaker condition $n > \max(2 \cdot t_s + t_a, \omega \cdot t_s, \omega \cdot t_a + t_s)$ is tight for network-agnostic $\mathcal{CA}$, one could expect that this should be the case for $\mathcal{AA}$ as well (at least in $\mathbb{R}^d$).

7.2 Efficiency Metrics

Local computation. The extensions of $\mathcal{AA}$ leave many aspects of efficiency still open. The protocol outline we have described incurs exponential local computation complexity. While this can be avoided for trees [47], designing more efficient approaches for $\mathbb{R}^d$ or more general classes of graphs is an open question. For $\mathbb{R}^d$,

multiple relaxations of the problem have been considered: assuming that inputs can be *validated* [36], requiring the honest outputs to lie in the convex hull of the honest inputs' projections onto lower dimensions [72, 73] or to approximate the honest inputs' centroid [22] instead of requiring the outputs to lie in the honest inputs' convex hull. Such relaxations enable lower local communication complexity, and also higher resilience thresholds.

Round complexity. Optimal round complexity also raises several open questions. For $\mathbb{R}^d$, the work of [48] improved the round complexity compared to the initial solutions [61, 71]. However, the optimal round complexity for $\mathcal{AA}$ on $\mathbb{R}^d$ remains an open, especially in *message-passing* models (where communication occurs via a network) with byzantine failures. In the wait-free model, the recent work of [4] has established lower bounds and provided efficient protocols for $\mathcal{AA}$ on $\mathbb{R}^d$.

For trees, the asynchronous $\mathcal{AA}$ protocol of [67] follows the outline of [1] and achieves round complexity $O(\log \mathsf{D})$, where D is the diameter of the input space tree. In the synchronous model, the work of [47] has extended the lower bound of Theorem 5 to trees and provided an asymptotically matching protocol. We note that [47] uses a different approach, reducing $\mathcal{AA}$ on trees to $\mathcal{AA}$ on $\mathbb{R}$.

For chordal graphs, [67] proposed two protocols: one presented as a protocol for cycle-free semilattices with round complexity $O(\mathsf{D})$, and a more efficient protocol with $O(\log \mathsf{D})$ rounds, where D denotes the input space graph's diameter. However, the efficient protocol of [67] was shown to be incorrect in [30], which provided a network-agnostic protocol with $O(\mathsf{D})$ rounds. Whether $o(\mathsf{D})$ rounds are, in fact, sufficient for chordal graphs remains an open problem.

Communication complexity. Protocols following the standard outline [52, 61, 62, 67, 71] match the message-complexity per iteration of the real-valued variants, except that now messages contain input space elements as opposed to real values. We note the recent results regarding $\mathcal{AA}$ on trees. The work of [65] has provided an asynchronous $\mathcal{AA}$ protocol for trees with $O(n^2)$ messages per iteration, improving over $O(n^3)$ in the protocol of [67]. In addition, the reduction of [47] from $\mathcal{AA}$ on trees to $\mathcal{AA}$ on $\mathbb{R}$ in the synchronous model maintains the message complexity of the underlying real-valued protocol (this runs on D-close inputs, where D is the diameter of the tree), except for an (additive) overhead of $O(n^3)$ messages in unauthenticated settings and $O(n^4)$ in authenticated settings.

8 Applications of $\mathcal{AA}$

In this section we discuss applications of $\mathcal{AA}$ that have arisen in the literature.

Oracle Networks. One of the main current applications of $\mathcal{AA}$ is oracle networks, where parties (oracles) aggregate numerical data from multiple sources and output a single value to a blockchain or smart contract. Such values are required to be in the honest reports' range, hence these systems naturally relate to $\mathcal{CA}$ and, more broadly, to real-valued $\mathcal{AA}$. These oracle networks are used to bring off-chain information (e.g., asset prices, sensor data, external events) onto a blockchain, where smart contracts can use it to make decisions [56].

Chainlink [21] is the most widely deployed oracle solution. It assumes $n \geq 3t+1$ and operates in partial synchrony (a model that assumes the network is initially asynchronous but eventually becomes synchronous [29, 38]). Chainlink's protocol builds on partially-synchronous $\mathcal{BA}$ techniques [24], proceeding in epochs with a designated leader. The parties retrieve data from external sources and report their signed observations to the leader. Once the leader collects $2t+1$ signed observations, it announces them to all parties. Each party then aggregates these observations, computing the median as the candidate output. From an $\mathcal{AA}$ perspective, this step corresponds to discarding the lowest t and highest t values, yielding a value in the honest observations' range. Parties sign a report encoding the result of this aggregation, and send this back to the leader, who, upon observing $t+1$ signatures on the same report, forms a certificate and broadcasts it. Finally, the certified report is submitted to an aggregator smart contract on the blockchain. Chainlink also includes mechanisms for detecting a byzantine leader and triggering leader replacement. Assuming an honest leader, the protocol completes in $O(1)$ rounds and incurs $O(\ell n^3 + \kappa n^3)$ bits of communication, where ℓ denotes the bit-length of honest inputs and κ the security parameter of the signature scheme. The protocol incurs an additional $O(n)$ factor in communication and rounds when the adversary is adaptive (since the protocol needs to iterate until it finds an honest leader).

DORA [25] improves over Chainlink by tolerating an asynchronous network and reducing communication to $O(\ell n^2 + \kappa n^2)$. Moreover, if the honest inputs are δ-close, DORA tolerates up to $t < n/2$ corruptions. Instead of relying on a single leader, DORA uses a set of aggregators that collect signed values from the parties and post them directly on the blockchain; the first sufficiently large posted set is taken as the output. Unlike Chainlink, which requires $2t+1$ reports, DORA only requires a set containing $(t+1)$ δ-close values. The agreed-upon output is the mean of these values, which is δ-close to the honest inputs' range.

Delphi [8] further improves upon prior approaches by replacing the use of signatures as a core piece in achieving agreement with $\mathcal{AA}$. Similarly to [25], [8] assumes that honest inputs are drawn from a thin-tailed distribution, implying that they are δ-close with high probability. The work of [8] presents an asynchronous protocol achieving $\mathcal{AA}$ with a relaxed Validity guarantee (honest outputs are δ-close to the honest inputs' range). If the honest inputs consist of up to ℓ bits, this protocol incurs $O\big(\ell n^2 \cdot \frac{\delta}{\varepsilon} \cdot \big(\log\big(\frac{\delta}{\varepsilon}\log\frac{\delta}{\varepsilon}\big) + \log(\lambda \log n)\big)\big)$ bits of communication and $O\big(\log\big(\frac{\delta}{\varepsilon}\log\frac{\delta}{\varepsilon}\big) + \log(\lambda \log n)\big)$ rounds, where λ denotes a statistical security parameter. As mentioned in Section 4, this improves over the protocol of [1] by a linear factor of n in terms of communication complexity. Delphi then builds an oracle network on top of this $\mathcal{AA}$ protocol by adding one final communication round: once $\mathcal{AA}$ with agreement parameter ε is reached, parties distribute signed outputs rounded to the nearest multiple of ε. When a party observes $t+1$ signatures on the same value, this value is published to the blockchain. The published output is $(\delta+\varepsilon)$-close to the honest inputs' range.

Randomness Beacons. $\mathcal{AA}$ protocols have also been used in larger building blocks like randomness beacons [9], which have been used for more complex systems

like multi-party computation [11]. While many randomness beacon protocols make use of threshold cryptographic setups or incur high computational cost with the use of public-key cryptography [18,19,31,54], protocols like HashRand [9] generate a common random value via $\mathcal{AA}$, which is deterministic and only incurs basic arithmetic operations. At a high level, the protocol uses n parallel instances of asynchronous verifiable secret sharing, where each party distributes a random value, along with the $\mathcal{AA}$ protocol Delphi [8] to consolidate the values into a single approximately-agreed common random value that is unpredictable. The resulting protocol produces 78 beacons per minute on a geo-distributed environment with $n = 136$ nodes, improving over state of the art approaches that use discrete-log operations [31,54]. The protocol is also post-quantum secure.

Sensor Networks. A recent work [10] has considered $\mathcal{AA}$ in the context of sensor-equipped cyber-physical systems (CPS). In this setting, sensors are deployed over an area and they must collectively locate certain target locations. Sensors are resource-constrained and often deployed in adverse environments, which requires a fault-tolerant distributed protocol to agree on (an approximate coordinate of) the target location. The authors employ Delphi [8] along with 2D Voronoi diagrams to divide the area into different regions. In a sensor testbed with $n = 19$ sensors, their protocol reduces the energy consumption of the prior state-of-the-art protocol [37] to $\frac{2}{5}$-th, in exchange for a 2% loss in location accuracy.

9 Conclusions

In this work, we surveyed $\mathcal{AA}$ in the presence of byzantine faults, focusing on the well-understood case of real-valued inputs and situating it within broader domains such as higher dimensions and graphs. We described feasibility across synchronous, asynchronous, and network-agnostic models, highlighted the main techniques used in existing protocols, and compared them in terms of resilience, round complexity, and communication cost. In addition, we explored stronger Validity notions and broader input spaces. Beyond unifying existing results, this systematization exposes several fundamental open questions. While resilience thresholds for real-valued $\mathcal{AA}$ are largely settled, optimal efficiency guarantees remain unresolved: both in terms of round complexity and communication complexity. Termination mechanisms present an additional challenge, especially in the network-agnostic model. Finally, while several extensions beyond the $\mathbb{R}$ are known, we still lack a clear characterization of $\mathcal{AA}$ on general input spaces.

By clarifying what is known and where gaps lie, we hope this SoK provides a useful reference point and helps guide future work on $\mathcal{AA}$ and its applications.

10 Acknowledgments

We thank our anonymous shepherd and the anonymous reviewers for their helpful feedback and suggestions, which significantly improved this paper. We also thank Akhil Bandarupalli and Mose Mizrahi Erbes for drawing our attention to relevant related work.

References

1. Ittai Abraham, Yonatan Amit, and Danny Dolev. Optimal resilience asynchronous approximate agreement. In Teruo Higashino, editor, *Principles of Distributed Systems*, pages 229–239, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
2. 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.
3. Manuel Alcántara, Armando Castañeda, David Flores-Peñaloza, and Sergio Rajsbaum. The topology of look-compute-move robot wait-free algorithms with hard termination. *Distributed Computing*, 32(3):235–255, 2019. doi:10.1007/s00446-018-0345-3.
4. Hagit Attiya and Faith Ellen. The Step Complexity of Multidimensional Approximate Agreement. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:12, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2022.6, doi:10.4230/LIPIcs.OPODIS.2022.6.
5. Hagit Attiya, Itay Flam, and Jennifer L. Welch. Beyond Canonical Rounds: Communication Abstractions for Optimal Byzantine Resilience, 2025. URL: https://arxiv.org/abs/2510.04310, arXiv:2510.04310.
6. Hagit Attiya, Itay Flam, and Jennifer L. Welch. Brief Announcement: Communication Patterns for Optimal Resilience. In Dariusz R. Kowalski, editor, *39th International Symposium on Distributed Computing (DISC 2025)*, volume 356 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:7, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2025.46, doi:10.4230/LIPIcs.DISC.2025.46.
7. Renas Bacho, Daniel Collins, Chen-Da Liu-Zhang, and Julian Loss. Network-agnostic security comes (almost) for free in dkg and mpc. In *Annual International Cryptology Conference*, pages 71–106. Springer, 2023.
8. A. Bandarupalli, A. Bhat, S. Bagchi, A. Kate, C.-D. Liu-Zhang, and M. K. Reiter. Delphi: Efficient Asynchronous Approximate Agreement for Distributed Oracles. In *Proceedings of the 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 456–469, Brisbane, Australia, 2024. IEEE. doi:10.1109/DSN58291.2024.00051.
9. Akhil Bandarupalli, Adithya Bhat, Saurabh Bagchi, Aniket Kate, and Michael K Reiter. Random beacons in monte carlo: Efficient asynchronous random beacon without threshold cryptography. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2621–2635, 2024.
10. Akhil Bandarupalli, Adithya Bhat, Somali Chaterji, Michael K Reiter, Aniket Kate, and Saurabh Bagchi. Sensorbft: Fault-tolerant target localization using voronoi diagrams and approximate agreement. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pages 186–197. IEEE, 2024.
11. Akhil Bandarupalli, Xiaoyu Ji, Aniket Kate, Chen-Da Liu-Zhang, and Yifan Song. Computationally efficient asynchronous mpc with linear communication and low additive overhead. In *Annual International Cryptology Conference*, pages 261–294. Springer, 2025.

12. Michael Ben-Or, Danny Dolev, and Ezra N. Hoch. Brief announcement: Simple gradecast based algorithms. In Nancy A. Lynch and Alexander A. Shvartsman, editors, *Distributed Computing*, pages 194–197, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
13. Michael Ben-Or, Danny Dolev, and Ezra N. Hoch. Simple gradecast based algorithms. *CoRR*, abs/1007.1049, 2010. URL: `http://arxiv.org/abs/1007.1049`, `arXiv:1007.1049`.
14. Amey Bhangale, Chen-Da Liu-Zhang, Julian Loss, Kartik Nayak, and Sravya Yandamuri. Leader election with poly-logarithmic communication per party. In *Annual International Cryptology Conference*, pages 37–68. Springer, 2025.
15. Erica Blum, Elette Boyle, Ran Cohen, and Chen-Da Liu-Zhang. Communication lower bounds for cryptographic broadcast protocols. *Distributed Computing*, 38(1):1–17, 2025.
16. Erica Blum, Jonathan Katz, Chen-Da Liu-Zhang, and Julian Loss. Asynchronous byzantine agreement with subquadratic communication. In *TCC 2020, Part I*, LNCS, pages 353–380. Springer, Heidelberg, March 2020. `doi:10.1007/978-3-030-64375-1_13`.
17. Erica Blum, Jonathan Katz, and Julian Loss. Synchronous consensus with optimal asynchronous fallback guarantees. In *Theory of Cryptography Conference*, pages 131–150. Springer, 2019.
18. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Yvo G. Desmedt, editor, *Public Key Cryptography — PKC 2003*, pages 31–46, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
19. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22*, pages 416–432. Springer, 2003.
20. Gabriel Bracha. Asynchronous Byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987.
21. Lorenz Breidenbach, Christian Cachin, Alex Coventry, Yan Ji, Kostis Karantias, Philipp Schindler, Chrysoula Stathakopoulou, and Alexandru Topliceanu. Chainlink 3.0 Off-Chain Reporting (OCR3) Protocol. Technical report, Chainlink Labs, 2025. URL: `https://research.chain.link/ocr3.pdf`.
22. Melanie Cambus and Darya Melnyk. Improved Solutions for Multidimensional Approximate Agreement via Centroid Computation. *arXiv preprint arXiv:2306.12741*, 2023. URL: `https://arxiv.org/pdf/2306.12741`.
23. Mélanie Cambus, Darya Melnyk, Tijana Milentijević, and Stefan Schmid. Approximate Agreement Algorithms for Byzantine Collaborative Learning. In *Proceedings of the 37th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '25, page 89–100, New York, NY, USA, 2025. Association for Computing Machinery. `doi:10.1145/3694906.3743343`.
24. Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.
25. Prasanth Chakka, Saurabh Joshi, Aniket Kate, Joshua Tobkin, and David Yang. DORA: Distributed Oracle Agreement with Simple Majority, 2023. URL: `https://arxiv.org/abs/2305.03903`, `arXiv:2305.03903`.
26. B.A. Coan. A compiler that increases the fault tolerance of asynchronous protocols. *IEEE Transactions on Computers*, 37(12):1541–1553, 1988. `doi:10.1109/12.9732`.

27. Andrei Constantinescu, Marc Dufay, Diana Ghinea, and Roger Wattenhofer. Validity in Network-Agnostic Byzantine Agreement. In Dariusz R. Kowalski, editor, *39th International Symposium on Distributed Computing (DISC 2025)*, volume 356 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:23, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2025.24, doi:10.4230/LIPIcs.DISC.2025.24.
28. Andrei Constantinescu, Diana Ghinea, Lioba Heimbach, Zilin Wang, and Roger Wattenhofer. A Fair and Resilient Decentralized Clock Network for Transaction Ordering. In Alysson Bessani, Xavier Défago, Junya Nakamura, Koichi Wada, and Yukiko Yamauchi, editors, *27th International Conference on Principles of Distributed Systems (OPODIS 2023)*, volume 286 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2023.8, doi:10.4230/LIPIcs.OPODIS.2023.8.
29. Andrei Constantinescu, Diana Ghinea, Jakub Sliwinski, and Roger Wattenhofer. Brief Announcement: Unifying Partial Synchrony. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing (DISC 2024)*, volume 319 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:7, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2024.43, doi:10.4230/LIPIcs.DISC.2024.43.
30. Andrei Constantinescu, Diana Ghinea, Roger Wattenhofer, and Floris Westermann. Convex Consensus with Asynchronous Fallback. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing (DISC 2024)*, volume 319 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2024.15, doi:10.4230/LIPIcs.DISC.2024.15.
31. Sourav Das, Vinith Krishnan, Irene Miriam Isaac, and Ling Ren. Spurt: Scalable distributed randomness beacon with transparent setup. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2502–2517. IEEE, 2022.
32. Giovanni Deligios, Martin Hirt, and Chen-Da Liu-Zhang. Round-efficient byzantine agreement and multi-party computation with asynchronous fallback. In *Theory of Cryptography Conference*, pages 623–653. Springer, 2021.
33. Danny Dolev et al. The Byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.
34. Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33(3):499–516, May 1986. doi:10.1145/5925.5931.
35. Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for Byzantine agreement. *J. ACM*, 32(1):191–204, January 1985. doi:10.1145/2455.214112.
36. Maya Dotan, Gilad Stern, and Aviv Zohar. Validated byzantine asynchronous multidimensional approximate agreement. *arXiv preprint arXiv:2211.02126*, 2022. URL: https://arxiv.org/abs/2211.02126, doi:10.48550/arXiv.2211.02126.
37. Sisi Duan, Xin Wang, and Haibin Zhang. Fin: Practical signature-free asynchronous common subset in constant time. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 815–829, 2023.

38. Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 35(2):288–323, apr 1988. doi:10.1145/42282.42283.
39. El-Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.
40. El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis, Lê Nguyên Hoang, and Sébastien Rouault. Genuinely Distributed Byzantine Machine Learning. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC '20, page 355–364, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3382734.3405695.
41. Alan David Fekete. Asynchronous approximate agreement. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '87, page 64–76, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/41840.41846.
42. Alan David Fekete. Asymptotically optimal algorithms for approximate agreement. *Distributed Computing*, 4(1):9–29, 1990.
43. Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM Journal on Computing*, 26(4):873–933, 1997.
44. 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.
45. Matthias Fitzi, Chen-Da Liu-Zhang, and Julian Loss. A new way to achieve round-efficient byzantine agreement. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 355–362, 2021.
46. Marc Fuchs, Diana Ghinea, and Zahra Parsaeian. Brief Announcement: Towards Round-Optimal Approximate Agreement on Trees. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '25, page 54–57, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3732772.3733555.
47. Marc Fuchs, Diana Ghinea, and Zahra Parsaeian. Round-Optimal Approximate Agreement on Trees, 2025. URL: https://arxiv.org/abs/2502.05591, arXiv:2502.05591.
48. Matthias Függer and Thomas Nowak. Fast Multidimensional Asymptotic and Approximate Consensus. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing (DISC 2018)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.DISC.2018.27, doi:10.4230/LIPIcs.DISC.2018.27.
49. Diana Ghinea, Vipul Goyal, and Chen-Da Liu-Zhang. Round-optimal byzantine agreement. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 96–119, Cham, 2022. Springer International Publishing.
50. Diana Ghinea and Chen-Da Liu-Zhang. SoK: Approximate agreement. Cryptology ePrint Archive, Paper 2025/2339, 2025. URL: https://eprint.iacr.org/2025/2339.
51. Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Optimal Synchronous Approximate Agreement with Asynchronous Fallback. In *Proceedings of the 2022*

ACM Symposium on Principles of Distributed Computing, PODC'22, page 70–80, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519270.3538442.

52. Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Multidimensional Approximate Agreement with Asynchronous Fallback. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '23, page 141–151, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3558481.3591105.
53. Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Communication-Optimal Convex Agreement. In *The 44th ACM Symposium on Principles of Distributed Computing (PODC), Huatulco, Mexico*, June 2025.
54. Timo Hanke, Mahnush Movahedi, and Dominic Williams. Dfinity technology overview series, consensus system. *arXiv preprint arXiv:1805.04548*, 2018.
55. Valerie King and Jared Saia. Breaking the $O(n^2)$ bit barrier: scalable byzantine agreement with an adaptive adversary. *Journal of the ACM (JACM)*, 58(4):1–24, 2011.
56. Chainlink Labs. 77+ Smart Contract Use Cases Enabled By Chainlink. https://blog.chain.link/smart-contract-use-cases/#post-title, 2019.
57. Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.
58. Jérémy Ledent. Brief announcement: Variants of approximate agreement on graphs and simplicial complexes. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, page 427–430, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465084.3467946.
59. Christoph Lenzen and Julian Loss. Optimal Clock Synchronization with Signatures. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, PODC'22, page 440–449, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519270.3538444.
60. Darya Melnyk and Roger Wattenhofer. Byzantine Agreement with Interval Validity. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, pages 251–260, Salvador, Brazil, 2018. IEEE Computer Society. doi:10.1109/SRDS.2018.00036.
61. Hammurabi Mendes and Maurice Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 391–400, Palo Alto, CA, USA, June 2013. ACM Press. doi:10.1145/2488608.2488657.
62. Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in Byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.
63. Silvio Micali and Vinod Vaikuntanathan. Optimal and player-replaceable consensus with an honest majority. 2017.
64. Mose Mizrahi Erbes and Roger Wattenhofer. Quit-Resistant Reliable Broadcast and Efficient Terminating Gather. In Silvia Bonomi, Letterio Galletta, Etienne Rivière, and Valerio Schiavoni, editors, *28th International Conference on Principles of Distributed Systems (OPODIS 2024)*, volume 324 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:22, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2024.15, doi:10.4230/LIPIcs.OPODIS.2024.15.

65. Mose Mizrahi Erbes and Roger Wattenhofer. Asynchronous Approximate Agreement with Quadratic Communication. In Andrei Arusoaie, Emanuel Onica, Michael Spear, and Sara Tucci-Piergiovanni, editors, *29th International Conference on Principles of Distributed Systems (OPODIS 2025)*, volume 361 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:26, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2025.16`, `doi:10.4230/LIPIcs.OPODIS.2025.16`.
66. Atsuki Momose and Ling Ren. Multi-threshold byzantine fault tolerance. pages 1686–1699. ACM Press, 2021. `doi:10.1145/3460120.3484554`.
67. Thomas Nowak and Joel Rybicki. Byzantine Approximate Agreement on Graphs. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing (DISC 2019)*, volume 146 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2019/11336`, `doi:10.4230/LIPIcs.DISC.2019.29`.
68. David Stolz and Roger Wattenhofer. Byzantine Agreement with Median Validity. In Emmanuelle Anceaume, Christian Cachin, and Maria Potop-Butucaru, editors, *19th International Conference on Principles of Distributed Systems (OPODIS 2015)*, volume 46 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1–14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2016/6591`, `doi:10.4230/LIPIcs.OPODIS.2015.22`.
69. Lili Su and Nitin H. Vaidya. Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, page 425–434, New York, NY, USA, 2016. Association for Computing Machinery. `doi:10.1145/2933057.2933105`.
70. Sam Toueg. Randomized byzantine agreements. In Robert L. Probert, Nancy A. Lynch, and Nicola Santoro, editors, *3rd ACM PODC*, pages 163–178, Vancouver, BC, Canada, August 1984. ACM. `doi:10.1145/800222.806744`.
71. Nitin H. Vaidya and Vijay K. Garg. Byzantine vector consensus in complete graphs. In Panagiota Fatourou and Gadi Taubenfeld, editors, *32nd ACM PODC*, pages 65–73, Montreal, QC, July 2013. ACM. `doi:10.1145/2484239.2484256`.
72. Zhuolun Xiang and Nitin H Vaidya. Brief announcement: Relaxed byzantine vector consensus. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 401–403, 2016.
73. Zhuolun Xiang and Nitin H. Vaidya. Relaxed Byzantine Vector Consensus. In Panagiota Fatourou, Ernesto Jiménez, and Fernando Pedone, editors, *20th International Conference on Principles of Distributed Systems (OPODIS 2016)*, volume 70 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2016.26`, `doi:10.4230/LIPIcs.OPODIS.2016.26`.