Validity in Responsive Byzantine Agreement

Diana Ghinea[1], Simon Holmgaard Kamp[2], and Chen-Da Liu-Zhang[1]

[1]Lucerne University of Applied Sciences and Arts
[2]Ruhr University Bochum

Abstract

Byzantine Agreement (BA) protocols must ensure not only agreement and termination, but also validity: the value agreed upon should meaningfully reflect the honest parties' inputs. The choice of validity condition can change the exact resilience threshold at which BA is solvable. Tight characterizations for BA with general validity conditions are known in the partially synchronous model (PODC'23), in the synchronous model (PODC'24), and in the network-agnostic model (DISC'25). We focus on general validity for synchronous BA with *responsive* termination. Such protocols remain secure against up to t_s byzantine corruptions, but incur a running time that depends on the actual network delay δ, rather than the conservative delay bound $\Delta \gg \delta$, whenever at most $t_r \leq t_s$ parties are corrupted.

We present a tight characterization of the validity properties solvable in this setting. We prove that every non-trivial validity property requires $n > 2t_r + t_s$ in authenticated settings, where a public-key infrastructure and digital signatures are available, and $n > 3t_s$ in unauthenticated settings. These threshold conditions are accompanied by a validity-dependent requirement, the *responsive similarity condition*: roughly, for any concrete configuration of honest inputs, there is a value that is valid for any view that a protocol could obtain from this initial configuration. We then present matching protocols in both settings, showing that these conditions are sufficient.

The main technical contribution is an authenticated responsive Core-Set Agreement protocol requiring $n > 2t_r + t_s$. This threshold may place the protocol in an honest-minority regime, where prior constructions for general validity do not apply and where standard Synchronous Broadcast is not responsive.

Finally, we instantiate the characterization for several standard validity notions – weak validity, strong unanimity, convex validity, and honest-input validity.

1 Introduction

Byzantine Agreement (BA) is a fundamental primitive in distributed computing, enabling a set of n parties to agree on a common output even when up to t of the n parties are byzantine (malicious). Beyond agreement and termination, BA also requires a *validity* condition, ensuring that the decided value is meaningfully related to the honest parties' inputs.

The classical validity condition is *strong unanimity*: if all honest parties start with the same input v, then the output agreed upon must be v. While this captures the right guarantee for binary input/output sets, it is often too weak for more general spaces. For example, if parties want to agree on metrics that have an inherent noise such as a temperature or a location, whenever honest inputs differ even slightly, strong unanimity allows arbitrary outputs. This has motivated the study of more general validity properties, such as convex validity [15, 20, 23, 24, 33, 37], median validity [14, 32, 41], intrusion tolerance [17, 23, 24], and social-choice-inspired validity conditions [31]. Recent work has made significant progress toward understanding, for an arbitrary predefined validity property, which resilience conditions are required to design a BA protocol that solves the given validity property. This question has been addressed in the synchronous model [10] (where all messages get delivered within a predefined amount of time), partially synchronous model [11] (where there is a bound on the message delivery time, but this bound is unknown), and network-agnostic model [12] (where the network may be synchronous with high corruption thresholds or asynchronous with lower corruption thresholds, and the type of network is not known a priori).

At the same time, modern distributed protocols are often expected to be not only correct, but also fast when the network behaves well. Current BA protocols for general validity conditions in the synchronous and network-agnostic models [10,12] incur a running time that is proportional to the known worst-case delay upper bound Δ. Note that Δ must be chosen conservatively, to enable protocols to tolerate rare periods of congestion or transient failures, even though the actual delay δ in a typical execution may be much smaller.

In contrast, partially synchronous protocols [11] benefit from having *responsive termination* [2,27,28,38,39], which captures a protocol's ability to avoid this unnecessary waiting: the runtime of a responsive protocol depends on the actual delay δ, rather than the conservative bound Δ. Unfortunately, this comes at the cost of tolerating fewer corruptions (for example, for strong unanimity, partially synchronous protocols tolerate only up to $n/3$ corruptions, while synchronous authenticated protocols can tolerate up to $n/2$).

In this paper, we investigate synchronous BA protocols for general validity that achieve security against a large number t_s of byzantine corruptions, but, if the number of corruptions is lower, namely $t_r \leq t_s$, we also want to achieve responsive termination. This dual threshold captures a best-of-both-worlds guarantee: strong safety in the worst case, while achieving fast termination in favorable executions.

While the problem of synchronous BA with responsiveness has been studied in the context of classical BA, State Machine Replication (SMR), and Multi-Party Computation (MPC) [2, 27, 28, 38, 39], the question of providing a characterization of the resilience for general validity remains open.

Under what conditions can a validity property be solved by synchronous protocols that are secure against t_s corruptions and responsive when at most t_r parties are corrupted?

1.1 Our Contribution

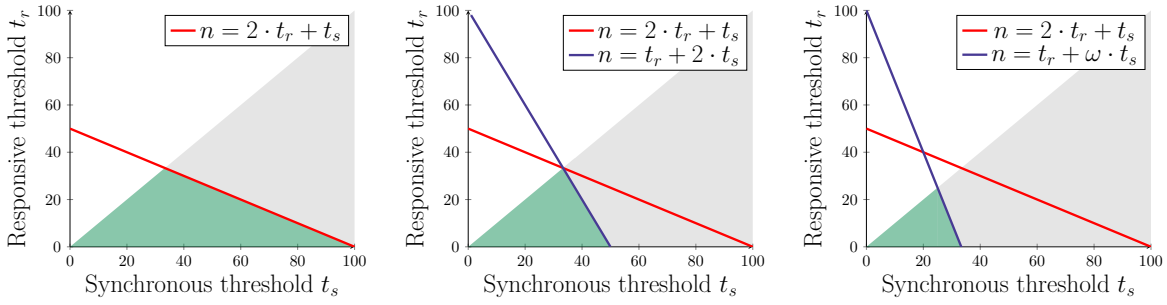
We present a tight characterization of responsive synchronous BA with general validity conditions assuming that: (i) the input space is at most infinitely countable, and (ii) the given validity property admits a finite number of outputs for some *view* containing all parties. Our lower bounds can be matched with deterministic protocols, but hold even for randomized protocols.

Theorem 1 (Informal)**.** *A* non-trivial *validity property can be solved by a synchronous* BA *protocol that is secure against t_s byzantine corruptions and additionally achieves responsive termination up to $t_r \leq t_s$ byzantine corruptions if and only if:*

- *$n > 2 \cdot t_r + t_s$ in authenticated settings, where a public-key infrastructure and digital signatures are available, and $n > 3 \cdot t_s$ in unauthenticated settings;*
- *the validity property satisfies the "responsive similarity condition": roughly, for any concrete configuration of honest inputs, there is a value that is valid for any view that a protocol could obtain from this initial configuration.*

We note that, by setting $t_r := 0$, our results match the tight feasibility conditions for the synchronous model [10, 12]. Hence, responsiveness gives a strict best-of-both-worlds improvement over classical synchronous protocols: the protocol keeps the same worst-case guarantees of security up to t_s corruptions, while additionally terminating at the speed of the actual network in fast, low-corruption executions.

Concrete instantiations. In Section 6, we make the tight conditions described by Theorem 1 concrete for standard validity notions. We display some of these in Figure 1.



(a) Weak Validity (b) Strong Unanimity (c) Convex Validity, $\omega = 3$

Figure 1: We illustrate our necessary and sufficient conditions in authenticated settings for three validity conditions: Weak Validity in Figure 1a, Strong Unanimity in Figure 1b, and Convex Validity for a space with Helly number $\omega = 3$ in Figure 1c. Each of the figures assumes $n = 100$. The red lines represent the condition $n = 2 \cdot t_r + t_s$, and the blue lines represent an additional condition imposed by the responsive similarity condition. The gray area represents the range $t_r \leq t_s$. Finally, the green area represents the choices of t_s, t_r for which the corresponding validity condition is solvable by a t_s-secure t_r-responsive BA protocol.

Weak Validity, which requires that if all parties are honest and have input v, the output agreed upon must be v, incurs only the general thresholds $n > 2 \cdot t_r + t_s$ in authenticated

settings, and $n > 3 \cdot t_s$ in unauthenticated settings. The tight conditions for Weak Validity in authenticated settings are displayed in Figure 1a.

For *stronger* validity definitions, the responsive similarity condition imposes stronger requirements that override the general thresholds described in Theorem 1.

Then, the traditional validity definition, Strong Unanimity (if all honest parties have input v, the output agreed upon must be v) yields the threshold $n > 2 \cdot t_s + t_r$ in authenticated settings and $n > 3 \cdot t_s$ in unauthenticated settings. The authenticated result matches that of [29], which focuses on t_r-responsive BA with this particular validity condition. We illustrate the tight conditions for Strong Unanimity in authenticated settings in Figure 1b.

For Convex Validity, which requires the output to be in the honest inputs' convex hull, the exact threshold is given by a feature of the space called the Helly number ω. For spaces with $\omega = 2$, such as integers or rational numbers, or trees with minimal-path convexity, Convex Validity has the same thresholds as Strong Unanimity: $n > 2 \cdot t_s + t_r$ in authenticated settings and $n > 3 \cdot t_s$ in unauthenticated settings. For spaces with $\omega > 2$, the threshold becomes $n > \omega \cdot t_s + t_r$ in both authenticated and unauthenticated settings. We illustrate the tight conditions for Convex Validity on a convex space with Helly number $\omega = 3$.

Finally, Honest-Input Validity over a finite domain of size $m > 2$, which requires that output is an honest party's input, requires $n > m \cdot t_s + t_r$.

Necessary conditions. Our impossibility results build on the indistinguishability arguments of [12], but the responsive setting introduces a different source of ambiguity. Instead of comparing synchronous and asynchronous executions, we compare executions in which some honest parties' messages are delayed beyond the responsive decision time, making them indistinguishable from byzantine parties that remain silent. We present the impossibility results formally in Section 3.

Sufficient conditions. We present a protocol matching the necessary conditions in Section 4. Similarly to [12, 15], the protocol first enables the honest parties to agree on an identical view over the inputs, and afterwards the parties apply a deterministic function provided by the responsive similarity condition to obtain their output. The common view is obtained using an Agreement on a Core-Set (ACS) protocol [4] that achieves Responsive Termination. In addition, this provides some stronger guarantees on the view obtained: it either contains values from at least $n - t_r$ parties, which would refer to the responsive case, or it contains all honest parties' inputs, which would refer to the case where we have at least $t_s \geq t_r$ corruptions. Depending on how one of the building blocks is instantiated, the protocol is either deterministic or randomized: randomization enables faster responsive termination.

Main technical contribution: authenticated responsive ACS. While our definition differs from prior work on ACS through the common view guarantees, the unauthenticated setting enables a construction by making minor adjustments to the asynchronous ACS mechanism of [4], which we present in Section B. On the other hand, designing a suitable protocol in the authenticated setting introduces new challenges, as the threshold $n > 2 \cdot t_r + t_s$ requires us to work with an *honest-minority*: the threshold t_s may be larger than $n/2$.

Prior work on BA with general validity involves an honest majority or does not achieve responsiveness at all. In the network-agnostic model of [12], the feasibility requirements imply an honest majority, which enables the authenticated ACS protocol to employ BA with Strong Unanimity as a building block. In our setting, BA with Strong Unanimity

requires a stronger threshold, so such an approach is not feasible. Similarly, in the partially synchronous model of [11] the resilience threshold is $t < n/3$. Honest-minority executions do arise in the purely synchronous model of [10]. However, in these protocols, the parties obtain a common view using synchronous protocols that do not achieve responsiveness (even though they may tolerate dishonest majority).

Consequently, the main technical difficulty is to construct an ACS protocol that provides security up to $t_s < n - 2 \cdot t_r$ corruptions in the worst case, and responsiveness up to t_r corruptions. Our construction consists of two components that separate the responsive and worst-case requirements – at a high level, this design is similar to that of prior works [27,28,39].

The first component is *optimistic*: it terminates responsively whenever the number of corruptions is up to $t_r \leq t_s$, and it achieves *safety* guarantees even if there are up to t_s corruptions. In line with [26], this can be seen as an ACS protocol that achieves *safety* up to t_s corruptions, and *liveness* up to t_r corruptions, whenever $n > 2 \cdot t_r + t_s$. This will be followed by a *worst-case* component, which parties execute only if they did not terminate via the optimistic component. In such case, it is safe to assume that the number of corruptions exceeds t_r, and hence the worst-case component only aims to achieve non-responsive termination while maintaining safety – we may use the Synchronous Broadcast (BC) protocol of [19] as a building block. As it is possible that only *some* of the honest parties terminate via the optimistic component, we make sure that, in such case, any honest party joining the worst-case component obtains an output that matches the outputs obtained optimistically. We present the complete protocol, along with the technical challenges that arise in the process, in Section 5.

1.2 Related Work

BA with general validity conditions. The question of which conditions are required to solve BA with a general validity property was first studied by Civit et al. [11] in the partially synchronous model, and later in the synchronous model [10]. Both works are concerned with deterministic protocols. More recently, the work of [12] studied the same question in the network-agnostic model, where a single protocol must provide security against t_s corruptions in a synchronous network, and up to $t_a \leq t_s$ in an asynchronous network. The work of [12] covers randomized protocols as well since, due to [21], the network-agnostic model requires randomized BA protocols whenever $t_a > 0$, and additionally extends some of the necessary conditions of [10] in the synchronous model to randomized protocols.

Our work is closest in spirit to [12], but focuses on a different axis: responsiveness in the synchronous model. As described above, our impossibility results largely adapt those of [12] from the network-agnostic model to our responsive setting and hold even for randomized protocols. On the other hand, our work diverges through the feasibility results, notably through the authenticated ACS construction. In addition, unlike in the network-agnostic model, randomization is not necessary for our feasibility results.

We note that the works of [10,11] extend beyond feasibility and also focus on optimal message complexity. This is outside our scope. In addition, similarly to [10–12], we restrict our attention to deterministic validity properties. Probabilistic validity notions (e.g., qualitative validity [25]) are outside the scope of our characterization.

Responsive protocols. Responsiveness was defined by Pass and Shi [38] as the property

that the running time of a protocol only depends on the actual network delay δ, rather than some known upper bound on the network delay $\Delta \geq \delta$. In other words, δ refers to the maximum delivery time of any messages between two honest parties in an execution, matching the period of an execution in asynchronous protocols [9].

The work of [38] is concerned with a variant of BA called State-Machine Replication (SMR), which assumes that parties receive inputs *continuously* from the environment and enables the parties to observe outputs in the same order, and shows that $t < n/3$ is required to achieve SMR with responsiveness. Their later work [39] separates the security threshold t_s and responsive threshold t_r and shows that SMR and BC can be achieved if $n > 2 \cdot t_r + t_s$, relying on a predetermined party to remain honest in order to terminate responsively. Note that this BC protocol does not imply responsive BA or ACS by simply having each party broadcast its input: this would only be responsive when all parties are honest.

The work of [29] showed the feasibility of BA with strong unanimity under the condition $2 \cdot t_s + t_r < n$. Our work generalizes this result to further validity conditions, which require us to deal with validity conditions that are possible in the dishonest majority regime.

Optimistic responsiveness was further extended to Multi-Party Computation (MPC) [27] under the threshold $t_s + 2 \cdot t_r < n$ for unanimous abort, and for the case of guaranteed output delivery (requiring in addition $t_s < n/2$). Both protocols make use of fully-homomorphic encryption, while our protocols consider settings without cryptographic assumptions, or settings that only make use of a PKI for signatures. Note that in the context of MPC parties jointly compute an arbitrary function on the parties' secret inputs, but the function *does not* depend on the corrupted set. Our setting is incomparable, since our work explores the feasibility of general validity conditions (which depend on which set of parties is honest).

The work of [2] focuses on deterministic BA with responsive termination, in settings with up to t corruptions: whenever $n > 3t$ and the network delay is δ, termination within $O(t \cdot \delta)$ time can be achieved deterministically. Moreover, in settings with $n \leq 3 \cdot t$, they prove that the runtime needs to depend on the message delay upper bound Δ, i.e., that $\lfloor (3 \cdot t - n)/2 \rfloor \cdot \delta + \Delta$ time is necessary.

Early-terminating protocols. A line of work [1, 5, 13, 16, 30, 40] has focused on achieving a round complexity that depends on the *actual* number of corruptions f instead of the upper bound t. We note that these protocols are not responsive, since every round has length Δ.

Agreement on a Core Set. ACS is the core of our feasibility result. This was introduced by Ben-Or, Canetti and Goldreich [3] as a building block in asynchronous MPC, and it allows a set of n parties among which t may be corrupted to agree on a view consisting of $n - t$ inputs transmitted by each party. It is therefore a natural building block towards constructing BA regardless of the exact validity condition. Classical asynchronous ACS [4, 6, 34] achieve responsiveness up to t_r corruptions, but lose security when there are t_s corruptions. In recent work, there have been variants of network-agnostic ACS [7, 15], but they do not achieve responsiveness.

The form of ACS that our work introduces achieves simultaneously 1) responsiveness up to t_r corruptions, 2) a consistent core-set that includes at least $n - t_r$ inputs or all honest inputs whenever there are up to t_s corruptions. In particular, the view provided by ACS must be useful in two different regimes: if the protocol terminates responsively, the view must be large enough to compensate for honest inputs that have not yet arrived; otherwise,

the view must contain all honest inputs.

We note that prior authenticated ACS protocols are designed in the honest-majority setting. In contrast, our authenticated ACS protocol works at the threshold required for responsive BA, which may allow an honest minority. This honest-minority case is the main new technical challenge in our feasibility result.

2 Preliminaries

We present our model, a few key concepts, and definitions, assuming a security parameter λ.

Model. We consider a set $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ of $n = \mathsf{poly}(\lambda)$ parties running a protocol in a fully connected network, where links model authenticated channels. The network is synchronous, i.e., the parties hold perfectly synchronized clocks and every message gets delivered within a publicly known amount of time Δ. Each party is modeled as a probabilistic Turing machine specified by the protocol: once its input, received messages, random tape are fixed, its behavior is deterministic and can be simulated by a Turing machine.

Adversary. We assume a central adversary that may corrupt up to $t < n$ parties: corrupted parties become byzantine, i.e., they may deviate arbitrarily from the protocol. In addition, the adversary may control the schedule of the message delivery subject to the bound Δ. Our protocols assume that the adversary is adaptive: it may choose which parties to corrupt at any time during the execution of the protocol. Our impossibility results hold even against a static adversary, that has to choose which parties to corrupt before the execution.

Cryptographic assumptions. We consider both settings with and without a trusted setup and cryptographic primitives. We use the term *unauthenticated setting* to refer to settings where no cryptographic assumptions are made. In contrast, we use the term *authenticated setting* to refer to a setting where a public key infrastructure and a secure digital signature scheme are available. For simplicity of presentation, we assume that signatures are unforgeable. When replaced with real-world instantiations, our feasibility results in the authenticated setting still hold except for negligible probability (in the scheme's security parameter) against computationally-bounded adversaries.

Byzantine Agreement and Validity. BA assumes that each party holds an input $v_{\mathsf{in}} \in V_{\mathsf{in}}$, and that honest parties output a value $v_{\mathsf{out}} \in V_{\mathsf{out}}$ satisfying a predefined validity property VAL. Concretely, VAL maps an *input configuration* to a set of admissible outputs in V_{out}.

An input configuration is a set $I \in V_{\mathsf{in}} \times \mathcal{P}$ containing pairs of *honest* parties and their inputs, where each party appears at most once: if $(v, P) \in I$, then P is an honest party with input v. For a static adversary, I contains exactly the inputs of the honest parties. For an adaptive adversary, we use the following convention: I contains exactly the inputs of parties that remain uncorrupted throughout the entire execution. We use the notation $\text{PARTIES}(I)$ to refer to the set of (honest) parties in the input configuration I. Note that, if $P \notin \text{PARTIES}(I)$, then P is corrupted. Moreover, we say that an input configuration I is *maximal* if $\big|I\big| = n$. We denote the set of all possible input configurations by $\mathcal{I} = \{I \subseteq V_{\mathsf{in}} \times \mathcal{P} : \big|I\big| \geq n - t_s\}$. Then, a validity property is defined by a function $\text{VAL} : \mathcal{I} \to 2^{V_{\mathsf{out}}}$.

We say that the validity property VAL is *trivial* if there is a value $v \in \bigcap_{I \in \mathcal{I}} \text{VAL}(I)$.

We assume that V_{in} is at most infinitely countable, it admits a (computable) total order,

and that parties' inputs can be represented on a Turing Machine. Consequently, the set of input configurations is at most countable. Moreover, we need to assume that there is a maximal input configuration I_0 so that $\text{VAL}(I_0)$ is finite.

Then, BA can be defined as follows:

Definition 1 (BA solving VAL)**.** *Consider a protocol* Π *where every party initially holds an input value* $v_{\mathsf{in}} \in V_{\mathsf{in}}$ *and every honest party may output a value* $v_{\mathsf{out}} \in V_{\mathsf{out}}$*. We say that* Π *is a* t*-secure* BA *protocol solving a validity property* VAL *if it achieves the following properties even when up to* t *of the* n *parties involved are byzantine:*

- ***Termination****: The probability that some honest party has not terminated by time* $\Delta \cdot f(\lambda)$ *is negligible in* λ*, where* $f \in \mathsf{poly}$ *and independent of* δ *and* Δ*. Moreover, as time goes to infinity, the probability that some honest party has not yet terminated is* 0*.*
- ***Agreement****: If two honest parties output* v_{out} *and* v'_{out}*, then* $v_{\mathsf{out}} = v'_{\mathsf{out}}$*.*
- ***Validity****: If* $I \in \mathcal{I}$ *is the input configuration defined by the honest parties and their inputs, then no honest party outputs* $v_{\mathsf{out}} \notin \text{VAL}(I)$*.*

Note that the polynomial f witnessing Termination may depend on n and t.

Responsiveness. We focus on protocols where the responsiveness condition is characterized by the number of corruptions. Specifically, we study the possible values of $t_r \leq t_s < n$ for which BA protocols with various validity conditions can be secure against t_s byzantine corruptions, and responsive whenever $t_r \leq t_s$ parties are corrupted.

Definition 2 (Responsive Protocols)**.** *We say that a protocol* Π *is* t_r*-responsive if it achieves the following property even when up to* t_r *of the parties involved are corrupted:*

- ***Responsive Termination:*** *The probability that some honest party has not terminated by time* $\delta \cdot f(\lambda)$ *is negligible in* λ*, where* $f \in \mathsf{poly}$ *and independent of* δ *and* Δ*.*

3 Necessary Conditions

We establish the necessary conditions for responsive BA, assuming a static adversary. We first show that every non-trivial validity property requires $n > 2 \cdot t_r + t_s$, even in authenticated settings. In unauthenticated settings, a stronger lower bound holds: $n > 3 \cdot t_s$, which comes directly from [12]. These conditions will not yet be sufficient – the missing piece is an additional validity-dependent requirement, namely the *responsive similarity condition*, that captures the fact that a protocol may not be able to distinguish between several possible input configurations. Such requirements appear in prior work as *the similarity condition* [11, 12] or as *the containment condition* [10].

General condition. We establish that solving any non-trivial validity condition VAL requires $n > 2 \cdot t_r + t_s$. This holds both in authenticated and unauthenticated settings.

Theorem 2. *Consider a validity property* VAL*. If there is a* t_s*-secure* t_r*-responsive* BA *protocol solving* VAL *when* $n \leq 2 \cdot t_r + t_s$*, then* VAL *is trivial.*

Assuming that there are two maximal input configurations I and J such that $\text{VAL}(I) \cap \text{VAL}(J) = \varnothing$, the lower bound $n > 2 \cdot t_r + t_s$ follows from a standard indistinguishability argument. However, non-triviality of VAL only says that there is no v in $\bigcap_{I \in \mathcal{I}} \text{VAL}(I)$: this does not imply that there is a pair of input configurations I, J such that $\text{VAL}(I) \cap \text{VAL}(J) = \varnothing$, let alone such a pair of maximal input configurations. Then, similarly to [12], we strengthen the validity property so that it already accounts for the possibility that some parties appearing honest may instead be byzantine, but following the protocol correctly. Concretely, given a validity condition VAL, we define a stronger condition $\text{VAL}^\star(I) = \bigcap_{J \subseteq I, |J| \geq n - t_s} \text{VAL}(J)$. As in [12, Lemma 3], the strengthening is without loss: Lemma 1 shows that every protocol solving VAL also solves $\text{VAL}^\star$. For proof, see Section A.1.

Lemma 1. *Consider a validity property* VAL*, and let* $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ |J| \geq n - t_s} \text{VAL}(J)$ *for any input configuration* I*. If there is a* t_s*-secure* BA *protocol* Π *solving* VAL*, then* Π *is also a* t_s*-secure* BA *protocol solving* $\text{VAL}^\star$*.*

Then, Lemma 2 shows that, if $\text{VAL}^\star$ is solvable when $n \leq 2 \cdot t_r + t_s$, then all maximal input configurations share a common valid output: there is a value $v^\star \in V_{\mathsf{out}}$ such that $v^\star \in \text{VAL}^\star(I)$ for every maximal I. The proof is analogous in spirit to Corollary 4 and Lemma 5 of [12]. The main difference will be caused by the Responsive Termination condition, which only requires all honest parties to terminate *quickly* except with probability $\text{negl}(\lambda)$ as opposed to *almost surely*. In contrast, [12] considers termination properties where all honest parties terminate *almost surely*. This is why our proof will require that there is a maximal input configuration I_0 such that $\text{VAL}(I_0)$ is finite. We defer the proof to Section A.1.

Lemma 2. *Let* VAL *be a validity property, and define* $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ \left|J\right| \geq n - t_s} \text{VAL}(J)$*. If there is a* t_s*-secure* t_r*-responsive* BA *protocol solving* $\text{VAL}^\star$ *when* $n \leq 2 \cdot t_r + t_s$ *and* $t_r > 0$*, then there is a value* $v^\star \in V_{\mathsf{out}}$ *such that, for every maximal input configuration* I*,* $v^\star \in \text{VAL}^\star(I)$*.*

Finally, Lemma 3, proven in Section A.1, shows that, if there is a value $v^\star \in V_{\mathsf{out}}$ such that $v^\star \in \text{VAL}^\star(I)$ for every maximal I, then VAL is trivial: this is similar to [12, Corollary 4].

Lemma 3. *Let* VAL *be a validity property, and define* $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ \left|J\right| \geq n - t_s} \text{VAL}(J)$*. If there is a value* $v^\star \in V_{\mathsf{out}}$ *such that, for every maximal input configuration* I*,* $v^\star \in \text{VAL}^\star(I)$*, then* VAL *is trivial.*

We may now return to Theorem 2 and present its proof.

Proof of Theorem 2. Assume that there is a t_s-secure t_r-responsive BA protocol Π solving VAL when $n \leq 2 \cdot t_r + t_s$. If $t_r = 0$, then $n \leq t_s$, contradicting the standing assumption that $t_s < n$. Hence, we may assume $t_r > 0$.

By Lemma 1, the protocol Π also solves $\text{VAL}^\star$. Therefore, by Lemma 2, there is a value $v^\star \in V_{\mathsf{out}}$ such that $v^\star \in \text{VAL}^\star(I)$ for every maximal input configuration I. Then, using Lemma 3, this implies that VAL is trivial. □

Unauthenticated condition. For unauthenticated settings, [12] provides a stronger lower bound for synchronous protocols. As [12] relies on a weaker termination guarantee (all honest parties terminate eventually almost surely), we may directly use this result.

Theorem 3 (Theorem 11 of [12]). *Consider a validity property* VAL*. If there is an unauthenticated* BA *protocol solving* VAL *when* $n \leq 3 \cdot t_s$*, then* VAL *is trivial.*

Responsive similarity condition. We conclude the section with the validity-dependent requirement. Recall that, as suggested by Lemma 1, any protocol solving VAL actually solves a stronger condition $\text{VAL}^\star$: if the input configuration appears to be I, it could be that the concrete input configuration is actually some $J \in \text{SIMILAR}_S(I) := \{J \subseteq I : \left|J\right| \geq n - t_s\}$, and hence the protocol needs to output a value in $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. Notice that, by definition, $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J) = \text{VAL}^\star(I)$. This suggests that the parties need to be able to find a value in this intersection. For general deterministic synchronous BA protocols, [10, Lemma 8] proves that, if VAL is solvable, then there is a Turing-computable function $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ such that $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. The lemma below, proven in Section A.2 extends this result to randomized protocols, similarly to the proof of [12, Lemma 14] in the network-agnostic model.

Lemma 4. *Assume there is a* t_s*-secure* BA *protocol* Π *solving a validity property* VAL*. Then there is a Turing-computable function* $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ *such that, for every input configuration* $I \in \mathcal{I}$*,* $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$*.*

Lemma 4 is not yet sufficient for responsive BA protocols – we need one more requirement, which follows from a similar reasoning to that of [12] for asynchronous executions. In executions where the protocol terminates responsively, some honest parties' messages may be delayed beyond the responsive termination time, and some honest parties may obtain outputs before learning all honest inputs. That is, if the input configuration appears to be I with $\left|I\right| \geq n - t_r$, the concrete input configuration might have been $J \in \text{SIMILAR}_R(I) := \{J \in \mathcal{I} : \left|J\right| \geq n - t_s$ and J is compatible with $I\}$ – we say that two input configurations $I, J \in \mathcal{I}$ are *compatible* if they assign the same value to every party appearing in both configurations; that is, whenever $(v_I, P) \in I$ and $(v_J, P) \in J$, we have $v_I = v_J$. Hence, if the input configuration appears to be I with $\left|I\right| \geq n - t_r$, the output agreed upon must be in $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$. This is proven by the lemma below. The proof is deferred to Section A.2, and is also similar to [12, Lemma 14], but adapted to Responsive Termination.

Lemma 5. *Assume there is a* t_s*-secure* t_r*-responsive* BA *protocol* Π *solving a non-trivial validity property* VAL*. Then, there is a Turing-computable function* $\sigma : \{I \in \mathcal{I} : \left|I\right| \geq n - t_r\} \to V_{\mathsf{out}}$ *such that, for every input configuration* $I \in \mathcal{I}$ *with* $\left|I\right| \geq n - t_r$*,* $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$*.*

Using Lemma 4 and Lemma 5, we obtain the theorem below. For proof, see Section A.2.

Theorem 4. *Assume there is a* t_s*-secure* t_r*-responsive* BA *protocol* Π *solving a non-trivial validity property* VAL*. Then* VAL *satisfies* the responsive similarity condition*: there is a Turing-computable function* $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ *such that, for every input configuration* $I \in \mathcal{I}$*:*

- *if* $\left|I\right| \geq n - t_r$*, then* $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$*;*
- *otherwise,* $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$*.*

4 Sufficient Conditions

We now show that the necessary conditions from Section 3 are also sufficient by presenting a matching protocol. At a high-level, the protocol follows the outline of prior works on BA with general validity conditions [10–12]: the parties first agree on a common view over the inputs, i.e., a set of value-sender pairs. This is done by using a variant of Core-Set Agreement (ACS) [4]. Once such a common view is obtained, the parties use the Turing-computable function σ guaranteed by the responsive similarity condition: they output σ of that view. To match the impossibility results in Section 3, we need the common view to satisfy a few properties that are tailored to our setting. In contrast to existing asynchronous ACS [6, 34] or variants of network-agnostic ACS [7, 15], our responsive ACS achieves simultaneously: 1) responsiveness up to t_r corruptions, 2) a consistent core-set that includes at least $n - t_r$ inputs or all honest inputs whenever there are up to t_s corruptions. More precisely, if the actual input configuration is $I^\star$, we need ACS to provide the honest parties with a set of value-sender pairs I ($I \in V_{\mathsf{in}} \times \mathcal{P}$, and I contains each party at most once) of size at least $n - t_s$ such that: $I^\star \in \text{SIMILAR}_R(I)$ if $\left|I\right| \geq n - t_r$, and $I^\star \in \text{SIMILAR}_S(I)$ otherwise. This way, we obtain that $\sigma(I) \in \text{VAL}(I^\star)$, we are guaranteed to achieve Validity. We present our responsive ACS definition below.

Definition 3 (Responsive Agreement on a Core Set)**.** *Consider an n-party protocol Π where every party initially holds an input value $v_{\mathsf{in}} \in V_{\mathsf{in}}$ and every honest party may output a set of value-sender pairs I, i.e., $I \subseteq V_{\mathsf{in}} \times \mathcal{P}$ where each party appears at most once. We say that Π is a t_s-secure t_r-responsive* ACS *protocol if it satisfies the following properties when up to t_s of the parties involved are corrupted:*

- ***Termination**: The probability that some honest party has not terminated by time $\Delta \cdot f(\lambda)$ is negligible in λ, where $f \in \mathsf{poly}$ and independent of δ and Δ. Moreover, as time goes to infinity, the probability that some honest party has not yet terminated is 0.*
- ***Responsive Termination:** If there are up to t_r corrupted parties involved, the probability that some honest party has not terminated by time $\delta \cdot f(\lambda)$ is negligible in λ, where $f \in \mathsf{poly}$ and independent of δ and Δ.*
- ***Agreement:** If two honest parties output I and I', then $I = I'$.*
- ***Validity:** If an honest party outputs I such that $(v, P) \in I$ for some honest party P with input v_P, then $v = v_P$.*
- ***Core Quality:** If an honest party outputs I, then either: $\left|I\right| \geq n - t_r$, or $(\cdot, P) \in I$ for every honest party P.*

Theorem 5 describes an unauthenticated ACS protocol, which follows the outline of the asynchronous ACS protocol of [4] with minor adjustments. We present this construction in Section B. Afterwards, Theorem 6 describes our authenticated ACS protocol. Note that the authenticated setting requires us to work with honest-minority, and therefore our protocol deviates from prior work. We present this protocol in detail in Section 5.

Theorem 5. *There are deterministic and randomized unauthenticated t_s-secure t_r-responsive* ACS *protocols Π_{ACS} whenever $n > 3 \cdot t_s$. For the deterministic protocol, the polynomial*

witnessing both Termination and Responsive Termination is $f_{\text{ACS}}(\lambda) = O(t_s)$. For the randomized protocol, this is $f_{\text{ACS}}(\lambda) = O(\lambda)$.

Theorem 6. *There are deterministic and randomized authenticated t_s-secure t_r-responsive* ACS *protocols Π_{ACS} whenever $n > t_s + 2 \cdot t_r$. For the deterministic protocol, the polynomials witnessing Termination and Responsive Termination are $f_{\text{ACS}}(\lambda) = n + O(t_s)$ and $f'_{\text{ACS}}(\lambda) = O(t_s)$, respectively. For the randomized protocol, they are $f_{\text{ACS}}(\lambda) = n + O(\lambda)$ and $f'_{\text{ACS}}(\lambda) = O(\lambda)$, respectively.*

We may then present our protocol Π_{BA}, which solves any given validity condition VAL using the function σ from the responsive similarity condition. Theorem 7 describes its guarantees. For proof, see Section C.

Π_{BA} solving VAL

Code for party P with input v

1: Join Π_{ACS} with input v and obtain the output set I.
2: Output $\sigma(I)$ and terminate.

Theorem 7. *Consider a validity property* VAL *satisfying the responsive similarity condition. Then, there are deterministic and randomized t_s-secure t_r-responsive* BA *protocols solving* VAL *if $n > t_s + 2 \cdot t_r$ in authenticated settings, and if $n > 3 \cdot t_s$ in unauthenticated settings.*

For the deterministic protocol, the polynomials witnessing Termination and Responsive Termination in authenticated settings are $f(\lambda) = n + O(t_s)$ and $f'(\lambda) = O(t_s)$, respectively; in unauthenticated settings, they are both $f(\lambda) = O(t_s)$.

For the randomized protocol, the polynomials witnessing Termination and Responsive Termination in authenticated settings are $f(\lambda) = n + O(\lambda)$ and $f'(\lambda) = O(\lambda)$, respectively; in unauthenticated settings, they are both $f(\lambda) = O(\lambda)$.

5 Authenticated ACS

We now present the ACS construction behind Theorem 6. We split the design of Π_{ACS} in two components. The first one is an *optimistic* component, called $\Pi_{\mathsf{opt\text{-}case}}$. This is designed to achieve t_r-responsive t_r-secure ACS, but to retain safety guarantees even if the number of corruptions exceeds t_r: if an honest party obtains an output set in $\Pi_{\mathsf{opt\text{-}case}}$, this satisfies the requirements of our ACS definition. Roughly, parties that did not obtain outputs in $\Pi_{\mathsf{opt\text{-}case}}$ will join the second component, namely the *worst-case* component, denoted by $\Pi_{\mathsf{worst\text{-}case}}$: as $\Pi_{\mathsf{opt\text{-}case}}$ is t_r-responsive, if an honest party did not obtain output by time $\tau := f_{\mathsf{opt\text{-}case}}(\lambda) \cdot \Delta$ for some publicly known polynomial $f_{\mathsf{opt\text{-}case}}$, we may safely assume that the number of corruptions is $t_s > t_r$. That is, $\Pi_{\mathsf{worst\text{-}case}}$ only needs to achieve Termination, along with the safety requirements of our ACS definition.

Assuming no honest party obtains an output in $\Pi_{\mathsf{opt\text{-}case}}$, $\Pi_{\mathsf{worst\text{-}case}}$ could consist of n parallel invocations of the Dolev-Strong BC protocol [19] – this would enable all parties to obtain the views required by Definition 3. However, it is possible that $\Pi_{\mathsf{opt\text{-}case}}$ succeeds for *some* of the honest parties with output I, and not for all – in such case, we need to ensure that parties joining $\Pi_{\mathsf{worst\text{-}case}}$ also output I.

Our threshold $n > 2 \cdot t_r + t_s$ prevents us from deciding whether $\Pi_{\text{opt-case}}$ succeeded using, for instance, a BA protocol with Strong Unanimity (as that would require $n > 2 \cdot t_s$). Instead, if $\Pi_{\text{opt-case}}$ provided some honest parties with output I, we enable the remaining honest parties to learn I. Hence, if an honest party P obtains output I in $\Pi_{\text{opt-case}}$, it first collects a certificate C of $n - t_r$ signatures for I. If P collects this certificate before time $\tau + \Delta$, it forwards it to all parties and terminates with output I. Note that our resilience threshold implies that there is at most one such certified set I. In addition, if such a set exists, then it is indeed an honest party's output from $\Pi_{\text{opt-case}}$, and hence it satisfies the requirements of our ACS definition. Finally, if there are up to t_r corruptions and hence $\Pi_{\text{opt-case}}$ succeeded, all honest parties obtain such a certificate C and terminate without joining $\Pi_{\text{worst-case}}$.

Then, $\Pi_{\text{worst-case}}$ allows parties to join either with a certified set (I, C) as input, or with their original input values v_{in}. An honest party that has not terminated by time $\tau + 2 \cdot \Delta$ joins $\Pi_{\text{worst-case}}$ – with input (I, C) if it has received such a certified set, or with its own input value. $\Pi_{\text{worst-case}}$ will ensure that, if any party joins with a certified set (I, C), then this will be the output. Otherwise, in fact, *all* honest parties join $\Pi_{\text{worst-case}}$, and $\Pi_{\text{worst-case}}$ returns a set I containing all honest parties and their inputs. Either way, ACS is achieved.

Protocol Π_{ACS}

Code for party P **with input** v_{in}

Attempting $\Pi_{\text{opt-case}}$**:**

Join $\Pi_{\text{opt-case}}$ with input v_{in}.

Obtaining certificates if $\Pi_{\text{opt-case}}$ **succeeded:**

If P obtains output I in $\Pi_{\text{opt-case}}$ before time $\tau := f_{\text{opt-case}}(\lambda) \cdot \Delta$, P computes its signature σ_i for I and sends (I, σ) to all parties.

Terminating via $\Pi_{\text{opt-case}}$**:**

If you have received a set C of $n - t_r$ signatures for a set I by time $\tau + \Delta$, send (I, C) to all parties. Afterwards, output I and terminate.

Terminating via $\Pi_{\text{worst-case}}$**:**

At time $\tau + 2 \cdot \Delta$:

    If you have received a set I with valid signatures from $n - t_r$ parties:
        Let C be the set of valid signatures for I received.
        Join $\Pi_{\text{worst-case}}$ with input (I, C).
    Otherwise, join $\Pi_{\text{worst-case}}$ with input v_{in}.

Upon obtaining output (I, C) in $\Pi_{\text{worst-case}}$, output I.

In the following, we present the two components $\Pi_{\text{opt-case}}$ and $\Pi_{\text{worst-case}}$ in detail. Afterwards, we discuss the guarantees of Π_{ACS} and therefore prove Theorem 6.

Optimistic component. We first focus on the optimistic component $\Pi_{\text{opt-case}}$: this achieves *safety* (Validity, Agreement, Core Quality) up to t_s corruptions, and *liveness* (Responsive Termination) up to $t_r \leq t_s$ corruptions whenever $n > 2 \cdot t_r + t_s$. We note that we only need to achieve liveness properties under $t_r < n/3$ corruptions: as $t_r \leq t_s$, we get that $n > 2 \cdot t_r + t_s \geq 3 \cdot t_r$. We instantiate $\Pi_{\text{ACS}}^{\text{live}}$ with the unauthenticated protocol described by Theorem 5, with corruption thresholds set to t_r. We use either its randomized instantiation, whose Termination and Responsive Termination polynomials are both $O(\lambda)$, or its deterministic instantiation, whose corresponding polynomials are both $O(t_r)$. Under t_r corruptions, $\Pi_{\text{ACS}}^{\text{live}}$ ensures safety and liveness. However, if the number of corruptions exceeds t_r, $\Pi_{\text{ACS}}^{\text{live}}$ offers no guarantees. We therefore need a mechanism to ensure that, if an honest party obtains an

output in $\Pi_{\mathsf{opt\text{-}case}}$ via $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$, then the safety properties hold.

A starting point is for the honest parties to sign their inputs to $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$. Then, if an honest party P receives an output I, it verifies whether this is well-formed, and that the signatures are correct. However, even in an execution with up to t_r corruptions, $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ could return a set I of at least $n - t_r$ value-party pairs that contains pairs without signatures corresponding to byzantine senders: this still satisfies Core Quality. We would have to accept such a set. However, we break Validity: a pair $(v', P') \in I$ where the signature is missing does not directly imply that P' is corrupted. It could be that the number of corruptions exceeds t_r and $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ failed.

We then use an initial step before the parties execute $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$, where parties exchange their signed inputs. If party P receives $n - t_r$ signed inputs, these would form an appropriate output I. P collects these in a vector, and this vector will be its input to $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$. Then, if party P receives an output I via Π_{ACS}, it checks that this can be parsed as at least $n - t_r$ value-sender pairs. If this is the case, P considers each pair $(v, P) \in I$ such that v is, in fact, a vector of n components that could be obtained in the initial exchange step. Then, P considers the signed values in this vector *believable* and saves them in a multiset B. Then, it determines its potential output set I by essentially removing duplicate parties in B.

After this step, Core Quality and Validity hold. Agreement will require a final step where the parties exchange their potential output sets I. We present the code of $\Pi_{\mathsf{opt\text{-}case}}$ below, and Lemma 6 describes its guarantees.

Protocol $\Pi_{\mathsf{opt\text{-}case}}$

Code for party P_i with input v_i

Obtaining a verifiable input for $\Pi_{\mathbf{ACS}}^{\mathsf{live}}$:

1: Compute signature σ_i on v_i and send (v_i, σ_i) to all parties.
2: Upon receiving signed inputs (v_j, σ_j) where $v_j \in V_{\mathsf{in}}$ and σ_j is a valid signature from P_j on v_j from $n - t_r$ different parties:
3: P_i defines $\textsc{vec}_i$ as a vector with n components where: $\textsc{vec}_i[j] = (v_j, \sigma_j)$ for each party P_j where such a pair was received, and otherwise $\textsc{vec}_i[j] = \perp$.
4: P_i joins $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ with input $\textsc{vec}_i$.

Checking the potential output from $\Pi_{\mathbf{ACS}}^{\mathsf{live}}$:

5: Upon receiving output I_{live} from $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$:
6: P_i parses I_{live} as a set of at least $n - t_r$ value-sender pairs, and defines multiset $B = \varnothing$.

Finding *believable* input values in I_{live}:

7: For each $(\textsc{vec}_j, P_j) \in I_{\mathsf{live}}$, check if $\textsc{vec}_j$ can be parsed as a vector of n components. Then, for each $k \in [n]$, check that $\textsc{vec}_j[k]$ is either $\perp$ or (v_k, σ_k), where σ_k is a valid signature on v_k under the public key of P_k – if this is the case, add (v_k, P_k) to B.

Determining a *potential* output set:

8: Let I be the set obtained by selecting for each party P_j in B the pair $(v_j, P_j) \in B$ with the lowest v_j in lexicographic order.
9: If $|I| \geq n - t_r$, send I to all parties.

Final output:

10: Upon receiving the same set I from $n - t_r$ parties, output I.

Lemma 6. *Assume $n > 2 \cdot t_r + t_s$. Then, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Validity, Agreement, Core Quality up to t_s corruptions and Responsive Termination up to t_r corruptions. The polynomial witnessing Responsive Termination is $f_{\mathsf{opt\text{-}case}}(\lambda) = O(\lambda)$ when $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is randomized and*

$f_{\mathsf{opt\text{-}case}}(\lambda) = O(t_r) = O(t_s)$ *when* $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ *is deterministic.*

We split the proof of Lemma 6 in multiple lemmas, each concerned with one of the properties. Assuming up to t_s corruptions, Lemma 7 focuses on Validity, Lemma 8 focuses on Agreement, and Lemma 9 proves Core Quality. Finally, Lemma 10 is concerned with Responsive Termination under up to t_r corruptions.

Lemma 7. *If up to t_s parties are byzantine, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Validity.*

Proof. Assume an honest party P outputs a set I in $\Pi_{\mathsf{opt\text{-}case}}$. Then, P has received I from $n - t_r > t_s$ parties, hence from at least one honest party P'. As P' has sent I, it computed I from a multiset S containing only pairs (v_i, P_i) extracted from vector entries with valid signatures by P_i on v_i. Therefore, for every $(v_i, P_i) \in I$, the value v_i has a valid signature by P_i. If P_i is honest and $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is an unauthenticated protocol, unforgeability implies that P_i signed v_i in Line 1 of $\Pi_{\mathsf{opt\text{-}case}}$, and hence v_i is P_i's input v_{in}.

As this holds for every honest party in $\textsc{parties}(I)$, we have obtained that $\Pi_{\mathsf{opt\text{-}case}}$ achieves Validity. □

Lemma 8. *If up to t_s parties are byzantine, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Agreement.*

Proof. Assume that two honest parties P and P' obtain outputs $I \neq I'$ in $\Pi_{\mathsf{opt\text{-}case}}$. Then, P has received I from $n-t_r$ parties, while P' has received I' from $n-t_r$ parties. Note that these two sets of senders intersect in at least t_s+1 parties: $(n-t_r)+(n-t_r)-n = n-2\cdot t_r > t_s$. Hence, there is an honest party that has sent both I and I', which contradicts the protocol. □

Lemma 9. *If up to t_s parties are byzantine, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Core Quality.*

Proof. By the protocol's description, if an honest party outputs I, then $\left|I\right| \geq n - t_r$, which ensures that Core Quality holds. □

Lemma 10. *If up to t_r parties are byzantine, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Responsive Termination with $f_{\mathsf{opt\text{-}case}}(\lambda) = O(\lambda)$ when $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is randomized and $f_{\mathsf{opt\text{-}case}}(\lambda) = O(t_r) = O(t_s)$ when $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is deterministic.*

Proof. Assume the network delay is δ and that up to t_r parties are byzantine. Then, every honest party receives $n - t_r$ signed inputs by time δ, and hence all honest parties join $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ by time δ. By Theorem 5, $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is t_r-responsive, with polynomial $f_{\mathrm{ACS}}^{\mathsf{live}}(\lambda) = O(\lambda)$ for the randomized instantiation and $f_{\mathrm{ACS}}^{\mathsf{live}}(\lambda) = O(t_r) = O(t_s)$ for the deterministic instantiation. Thus, except with probability $\mathrm{negl}(\lambda)$, all honest parties complete the execution of $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ by time $\delta + f_{\mathrm{ACS}}^{\mathsf{live}}(\lambda) \cdot \delta$.

Moreover, by Theorem 5, $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ is t_r-secure as $n > 2 \cdot t_r + t_s \geq 3 \cdot t_r$. Hence, as $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ achieves Core Quality, every honest party obtains a set of I_{live} that either has $n - t_r$ pairs or contains all honest parties. Consequently, every honest party that completes the execution of $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ obtains a set I_{live} of size at least $n - t_r$. Moreover, $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ achieves Agreement – all honest parties that complete the execution of $\Pi_{\mathrm{ACS}}^{\mathsf{live}}$ obtain the same output I^{live}. We next note that the honest parties that have obtained I^{live} define B and I identically. This follows because they obtain the same I^{live} and the parsing, extraction, and lexicographic tie-breaking rules are deterministic.

Let P be an honest party that has obtained I_{live} in $\Pi^{\mathsf{live}}_{\text{ACS}}$. As I^{live} has size at least $n - t_r$ and there are up to t_r corruptions, I^{live} contains at least one pair corresponding to an honest party P_i. As $\Pi^{\mathsf{live}}_{\text{ACS}}$ achieves Validity, for every honest party P_i that appears in the set I_{live} of party P, P obtains VEC_i equal to the input of P_i to $\Pi^{\mathsf{live}}_{\text{ACS}}$. P_i has defined VEC_i based on the signed values received from the first step of $\Pi_{\mathsf{opt\text{-}case}}$. Hence, VEC_i contains at least $n - t_r$ value-signature pairs (v_j, σ_j), where v_j was correctly signed by P_j. Consequently, P obtains a multiset B containing value-sender pairs for at least $n - t_r$ distinct parties, and therefore obtains a set I of size at least $n - t_r$. This argument applies to all honest parties that complete the execution of $\Pi^{\mathsf{live}}_{\text{ACS}}$.

Hence, except with probability $\text{negl}(\lambda)$, all honest parties obtain the same set I by time $\delta + f^{\mathsf{live}}_{\text{ACS}}(\lambda) \cdot \delta$. Then, there are at least $n - t_r$ honest parties that distribute I, and hence every honest party receives the same set I from $n - t_r$ distinct parties within δ additional time. Consequently, except with probability $\text{negl}(\lambda)$, all honest parties terminate in $\Pi_{\mathsf{opt\text{-}case}}$ by time $(2 + f^{\mathsf{live}}_{\text{ACS}}(\lambda)) \cdot \delta$. By Theorem 5, this is $O(\lambda) \cdot \delta$ time for the randomized instantiation and $O(t_r) \cdot \delta = O(t_s) \cdot \delta$ time for the deterministic instantiation, as claimed. □

Worst-case component. We now describe the worst-case component $\Pi_{\mathsf{worst\text{-}case}}$. We recall that this is only executed by honest parties that did not obtain a certified set (I, C) by time $\tau + 2 \cdot \Delta$ in Π_{ACS}. Hence, it assumes that, on top of up to t_s byzantine corruptions, any number of honest parties may *crash*. Every honest party that joins $\Pi_{\mathsf{worst\text{-}case}}$ either joins with a certified set (I, C) or with its input value v_{in}. We recall that the certificate C is constructed in Π_{ACS}, and it is a *valid* certificate if it consists of signatures for I from $n - t_r$ distinct parties. $\Pi_{\mathsf{worst\text{-}case}}$ will require one precondition to achieve Agreement, namely that every honest party signs at most one set I before the execution of $\Pi_{\mathsf{worst\text{-}case}}$, which implies that there is at most one such certified set I. This precondition is fulfilled in Π_{ACS}.

We first want to provide the (joining) honest parties with an identical view over the parties' inputs. We achieve this with the help of a Synchronous Broadcast (BC), as described by Definition 4. We make use of the BC protocol of Π_{BC} [19], provided by Theorem 8.

Definition 4 (Broadcast)**.** *Consider an n-party protocol Π where a designated sender S initially holds an input value v_{in} and every honest party may output a value v_{out}. We say that Π is a t-secure broadcast protocol if it achieves each of the following properties even when up to t parties are corrupted:*

- ***Termination:*** *Every honest party outputs within $f(\lambda) \cdot \Delta$ time.*
- ***Agreement****: If two honest parties output v_{out} and v'_{out}, then $v_{\mathsf{out}} = v'_{\mathsf{out}}$.*
- ***Validity****: If S is honest and an honest party outputs v_{out}, then $v_{\mathsf{out}} = v_{\mathsf{in}}$.*

Theorem 8 ([19])**.** *If $t < n$, there is an authenticated t-secure* BC *protocol Π_{BC}. The polynomial witnessing Termination is $f_{\text{BC}}(\lambda) = t + 1$.*

Hence, each party distributes its input via Π_{BC}, and the parties obtain an identical view over the parties' inputs. If the view contains some certified set of value-sender pairs (I, C), where C is a valid certificate for I, then the honest parties output any such I in $\Pi_{\mathsf{worst\text{-}case}}$ – we have the guarantee that there is at most one such I. Otherwise, $\Pi_{\mathsf{worst\text{-}case}}$ returns the set of value-sender pairs containing the values received via Π_{BC} and their corresponding senders. We present the code of $\Pi_{\mathsf{worst\text{-}case}}$ below, and Lemma 11 describes its guarantees.

$\Pi_{\text{worst-case}}$

Code for party P holding either v_{in} or (I, C) as input

1: Send your input (either v_{in} or (I, C)) to all parties via Π_{BC}.
2: When all Π_{BC} invocations have terminated:
3: If any Π_{BC} invocation has returned output (I, C) such that C consists of valid signatures for I from $n - t_r$ distinct parties:
4: Output any such (I, C).
5: Otherwise:
6: $I :=$ the set of value-sender pairs (v', P') where v' was received from P' via Π_{BC}.
7: Output $(I, \perp)$.

Lemma 11. *Assume that at most t_s parties are corrupted, and that there is no honest party that has signed two sets $I \neq I'$ before the execution of $\Pi_{\text{worst-case}}$. Then, every honest party that joins $\Pi_{\text{worst-case}}$ obtains an output within $n \cdot \Delta$ time. In addition, all joining honest parties output the same set of value-sender pairs I, and each joining honest party outputs some C such that either $C = \perp$ or C is a valid certificate for I. Furthermore:*

1. *If an honest party joins with input $(I^\star, C^\star)$ where $C^\star$ is a valid certificate for $I^\star$, then $I = I^\star$ and every joining honest party outputs a certificate $C \neq \perp$ that is valid for I.*
2. *If some honest party outputs $C = \perp$, then I contains every honest party that joined $\Pi_{\text{worst-case}}$ with a value input, paired with that input.*
3. *If some honest party outputs (I, C) with $C \neq \perp$, then there is an honest party that has signed I before the execution of $\Pi_{\text{worst-case}}$.*

Proof. If no honest party joins $\Pi_{\text{worst-case}}$, the claim holds trivially. Otherwise, consider the honest parties that do join. Parties that do not join can be treated as byzantine parties that simply send no messages. Since the total number of parties is n and fewer than n parties are faulty, each Π_{BC} invocation achieves BC by Theorem 8.

Thus every joining honest party obtains an output from every Π_{BC} invocation within $n \cdot \Delta$ time. As Π_{BC} achieves Agreement, all joining honest parties obtain the same output in each Π_{BC} invocation.

If the common view contains a pair (I, C) where C is a valid certificate for I, then there is a unique such set $I^\star$: otherwise, there are $(n - t_r) + (n - t_r) - n = n - 2 \cdot t_r > t_s$ parties that have signed two different sets, hence at least one honest party that has signed two different sets, which contradicts the assumption in the lemma's statement. Hence, all joining honest parties output $I^\star$ and some certificate $C \neq \perp$ that is valid for $I^\star$.

Otherwise, if the common view contains no pair (I, C) where C is a valid certificate for I, the protocol outputs $(I, \perp)$, where I consists of the value-sender pairs obtained from the Π_{BC} outputs. For every honest party P that joined $\Pi_{\text{worst-case}}$ with value input v_{in}, the Validity property of Π_{BC} implies that all joining honest parties receive v_{in} from P's Π_{BC} invocation. Hence $(v_{\text{in}}, P) \in I$.

Finally, if there is some honest party P that has joined with input $(I^\star, C^\star)$, then $(I^\star, C^\star)$ appears in the common view due to Π_{BC}'s Validity condition. Moreover, as described above, there is a unique such I with this property, hence all joining honest parties output $I = I^\star$ and a certificate $C \neq \perp$ that is valid for I.

For the third claim, assume that an honest party outputs (I, C) with $C \neq \perp$. Then, C is a valid certificate for I: it contains signatures for I from $n - t_r$ distinct parties, hence from $n - t_r - t_s \geq 1$ honest parties. Then, as the byzantine parties cannot forge signatures on the honest parties' behalf, I was indeed signed by some honest party P. Note that signatures in $\Pi_{\text{worst-case}}$ are only produced in the Π_{BC} invocations. However, honest parties' signatures in the Π_{BC} invocations are on broadcast-tagged messages and are not valid signatures on I. Consequently, P must have signed I before the execution of $\Pi_{\text{worst-case}}$. □

Putting it all together. We now return to Π_{ACS} and discuss its guarantees. Lemma 12 is concerned with Validity and Core-Quality, Lemma 13 handles Agreement, and Lemma 14 handles Termination. Afterwards, Lemma 15 shows that Π_{ACS} achieves Responsive Termination when the number of corruptions is up to t_r. Together, these lemmas show that Π_{ACS} is a t_s-secure t_r-responsive ACS protocol, therefore proving Theorem 6.

Lemma 12. *If up to t_s parties are corrupted, Π_{ACS} achieves Validity and Core Quality.*

Proof. Assume an honest party P obtains output I in Π_{ACS}. Then, by the protocol's description, P has either output I in Line 3 or in Line 7.

We first consider the case where P outputs in Line 3. Then, P has received a valid certificate for I. Since a valid certificate for I contains signatures for I from $n - t_r$ distinct parties, and $n - t_r > t_s$, at least one signer is honest. An honest signer signs a set only after obtaining it as output from $\Pi_{\text{opt-case}}$. Hence I was output by some honest party in $\Pi_{\text{opt-case}}$. By the safety properties of $\Pi_{\text{opt-case}}$, this set I satisfies Validity and Core Quality.

We now consider the case where P outputs in Line 7. Hence, P outputs I after running $\Pi_{\text{worst-case}}$, and let (I, C) be the output obtained by P in $\Pi_{\text{worst-case}}$. Note that, as every honest party signs at most one set I in Π_{ACS}, the precondition of $\Pi_{\text{worst-case}}$ described by Lemma 11 is satisfied.

If $C \neq \perp$, Lemma 11 ensures that C is a valid certificate for I, and C contains at least one signature for I that was produced by an honest party before the execution of $\Pi_{\text{worst-case}}$. By the protocol's description, I was output by an honest party in $\Pi_{\text{opt-case}}$. Therefore, I satisfies Validity and Core Quality.

It remains to consider the case $C = \perp$. We show that, in this case, no honest party has terminated in Line 3: if some honest party P' had terminated in Line 3, it has obtained a certified set $(I^\star, C^\star)$. Then, P' has forwarded $(I^\star, C^\star)$ to all parties before terminating, by time $\tau + \Delta$. It follows that every honest party that has not terminated by time $\tau + 2 \cdot \Delta$ receives $(I^\star, C^\star)$, and enters $\Pi_{\text{worst-case}}$ at time $\tau + 2 \cdot \Delta$ with a certified set (I, C) as input. We obtain a contradiction: Lemma 11 ensures that, if an honest party joins $\Pi_{\text{worst-case}}$ with input (I, C), then no honest party outputs $C = \perp$ in $\Pi_{\text{worst-case}}$. It follows that no honest party terminated before $\Pi_{\text{worst-case}}$, and all honest parties that join $\Pi_{\text{worst-case}}$ use their original input values as inputs to $\Pi_{\text{worst-case}}$. Then, as all honest parties join $\Pi_{\text{worst-case}}$, Lemma 11 ensures that the output set I contains every honest party paired with its input. Therefore I satisfies Validity and Core Quality. □

Lemma 13. *If up to t_s parties are corrupted, Π_{ACS} achieves Agreement.*

Proof. Assume that two honest parties P and P' complete Π_{ACS} with outputs I and I' respectively, such that $I \neq I'$.

If both P and P' have obtained their outputs in Line 3, then these I and I' were both signed by $n - t_r$ parties: there are $(n - t_r) + (n - t_r) - n = n - 2 \cdot t_r > t_s$ distinct parties that have signed both I and I', and hence at least one honest party that has signed both I and I'. This contradicts the protocol's description.

We note that the preconditions of $\Pi_{\mathsf{worst\text{-}case}}$ described in Lemma 11 are satisfied, as no honest party signs different sets I before the execution of $\Pi_{\mathsf{worst\text{-}case}}$. Then, if both P and P' have obtained their outputs in Line 7, we contradict Lemma 11, which ensures that all honest parties joining $\Pi_{\mathsf{worst\text{-}case}}$ obtain the same output.

It then remains to discuss the case where P obtains its output in Line 3, and P' obtains its output in Line 7. Since P has obtained its output in Line 3, it has obtained a certificate C for I consisting of $n - t_r$ signatures by time $\tau + \Delta$. Before terminating, P has sent (I, C) to all parties, hence P' has received (I, C) by time $\tau + 2 \cdot \Delta$ and has joined $\Pi_{\mathsf{worst\text{-}case}}$ with input (I, C). Then, Lemma 11 ensures that P' has obtained output $I' = I$, which concludes the proof. □

Lemma 14. *If up to t_s parties are corrupted, Π_{ACS} achieves Termination for polynomial $f_{\mathrm{ACS}}(\lambda) = n + 2 + f_{\mathsf{opt\text{-}case}}(\lambda)$. Hence, $f_{\mathrm{ACS}}(\lambda) = n + O(\lambda)$ when $\Pi^{\mathsf{live}}_{\mathrm{ACS}}$ is randomized and $f_{\mathrm{ACS}}(\lambda) = n + O(t_s)$ when $\Pi^{\mathsf{live}}_{\mathrm{ACS}}$ is deterministic.*

Proof. If an honest party P did not terminate by time $\tau + 2 \cdot \Delta = (2 + f_{\mathsf{opt\text{-}case}}(\lambda)) \cdot \Delta$, it joins $\Pi_{\mathsf{worst\text{-}case}}$ at time $\tau + 2 \cdot \Delta = (2 + f_{\mathsf{opt\text{-}case}}(\lambda)) \cdot \Delta$. Then, as no honest party signs distinct sets I by the protocol's description, the preconditions of $\Pi_{\mathsf{worst\text{-}case}}$ are satisfied and Lemma 11 ensures that P terminates within $n \cdot \Delta$ additional time, hence by time $(n + 2 + f_{\mathsf{opt\text{-}case}}(\lambda)) \cdot \Delta$, as claimed in the lemma's statement. □

Lemma 15. *Assume that there are up to t_r corruptions. Then, Π_{ACS} achieves Responsive Termination for polynomial $f'_{\mathrm{ACS}}(\lambda) := f_{\mathsf{opt\text{-}case}}(\lambda) + 1$. Hence, $f'_{\mathrm{ACS}}(\lambda) = O(\lambda)$ when $\Pi^{\mathsf{live}}_{\mathrm{ACS}}$ is randomized and $f'_{\mathrm{ACS}}(\lambda) = O(t_s)$ when $\Pi^{\mathsf{live}}_{\mathrm{ACS}}$ is deterministic.*

Proof. In this case, Lemma 6 ensures that, except with probability $\mathrm{negl}(\lambda)$, all honest parties obtain output I in $\Pi_{\mathsf{opt\text{-}case}}$ by time $f_{\mathsf{opt\text{-}case}}(\lambda) \cdot \delta$. Moreover, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Agreement – the honest parties obtaining output I obtain the same output.

Then, every honest party that obtains output I in $\Pi_{\mathsf{opt\text{-}case}}$ sends signed I to all parties. By Lemma 6, $\Pi_{\mathsf{opt\text{-}case}}$ achieves Agreement – these honest parties sign the same set I. Consequently, except with probability $\mathrm{negl}(\lambda)$, every honest party receives $n - t_r$ signatures by time $(f_{\mathsf{opt\text{-}case}}(\lambda) + 1) \cdot \delta \leq (f_{\mathsf{opt\text{-}case}}(\lambda) + 1) \cdot \Delta = \tau + \Delta$. It follows that, except with negligible probability $\mathrm{negl}(\lambda)$, every honest party outputs I and terminates in Π_{ACS} by time $(f_{\mathsf{opt\text{-}case}}(\lambda) + 1) \cdot \delta$. Therefore, Responsive Termination holds for polynomial $f'_{\mathrm{ACS}}(\lambda) := f_{\mathsf{opt\text{-}case}}(\lambda) + 1$, which is $O(\lambda)$ for the randomized instantiation and $O(t_s)$ for the deterministic instantiation. □

6 Concrete Instantiations

In this section, we instantiate our necessary and sufficient conditions for a few standard validity notions. For each of these validity notions, the stated thresholds enable us to define the Turing-computable function σ required by the responsive similarity condition: this implies sufficiency. Conversely, when the stated thresholds do not hold, we show that there is

no function σ enabling the responsive similarity condition to hold, which implies that our thresholds are tight. We defer the formal proofs to Section D.

We start with Weak Validity: this requires that, if all parties are honest and have input v, the parties must output v. Otherwise, any output is allowed. Weak Validity only incurs the general lower bound. We can define $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ as follows: if there is a value v such that every pair in I assigns value v, then set $\sigma(I) = v$; otherwise $\sigma(I)$ is a default value. This will satisfy the responsive similarity condition whenever the general threshold $n > 2 \cdot t_r + t_s$ holds, leading to the result below.

Corollary 1. *Weak Validity can be solved responsively if and only if $n > 2 \cdot t_r + t_s$ in an authenticated setting, and if and only if $n > 3 \cdot t_s$ in an unauthenticated setting.*

We now look into a stronger property, namely Strong Unanimity: if all honest parties have input v, the output agreed upon must be v; otherwise any output is allowed. Here, the responsive similarity condition leads to stronger requirements in authenticated settings. We define the Turing-computable function $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ that ensures that the responsive similarity condition holds whenever $n > 2 \cdot t_s + t_r$ as follows:

- if $\left|I\right| \geq n - t_r$ and there is a value v such that at least $\left|I\right| - t_s$ pairs in I assign value v, set $\sigma(I) = v$.
- Otherwise, if $\left|I\right| < n - t_r$ and there is a value v such that at least $n - t_s$ pairs in I assign value v, set $\sigma(I) = v$.
- Otherwise, set $\sigma(I) = v_0$, where v_0 is an arbitrary fixed value in V_{out}.

This implies that $n > 2 \cdot t_s + t_r$ is sufficient. In addition, this condition is tight: in Section D, we show that there is no Turing-computable function σ enabling Strong Unanimity to satisfy the responsive similarity condition when $n \leq 2 \cdot t_s + t_r$. We then obtain the following result. We note that this result was already proven in [29].

Corollary 2. *Strong Unanimity can be solved responsively if and only if $n > 2 \cdot t_s + t_r$ in an authenticated setting, and if and only if $n > 3 \cdot t_s$ in an unauthenticated setting.*

We now move towards Convex Validity, which requires the output to lie in the convex hull of the honest inputs. This considers a convex space over the input domain V_{in}: a family of subsets of V_{in} are declared convex, and $\langle X \rangle$, i.e., the convex hull of X, denotes the smallest convex set containing X. We need to assume that protocol inputs and outputs are finitely represented, and that whenever a finite family of finitely generated convex hulls has non-empty intersection, there is a Turing-computable function that returns a point in the intersection. We use a feature of the space called the Helly number, denoted by ω: this is the smallest integer such that, for every finite family of convex sets, if every ω of them have a non-empty intersection, then the whole family has non-empty intersection.

For Convex Validity, the Turing-computable function σ is similar to the *safe area* computation done in prior works [33, 37]. In Section D, we show that this is well-defined whenever $n > \omega \cdot t_s + t_r$, hence enabling Convex Validity to satisfy the responsive similarity condition. We also show that, if $n \leq \omega \cdot t_s + t_r$, there is no Turing-computable function σ enabling Convex Validity to satisfy the responsive similarity condition.

We separate the results for Convex Validity into spaces with Helly number $\omega = 2$ (e.g., integers, rational numbers, trees with shortest-path convexity), handled by Corollary 3, and spaces with Helly number $\omega > 2$ (e.g. graphs containing triangles, with minimal-path convexity), handled by Corollary 4.

Corollary 3. *Convex Validity on a convex space with Helly number $\omega = 2$ can be solved responsively if and only if $n > 2 \cdot t_s + t_r$ in the authenticated setting, and if and only if $n > 3 \cdot t_s$ in the unauthenticated setting.*

Corollary 4. *Convex Validity on a convex space with Helly number $\omega > 2$ can be solved responsively if and only if $n > \omega \cdot t_s + t_r$, both in authenticated and unauthenticated settings.*

Finally, we consider Honest-Input Validity [36]: this requires the outputs to be an honest party's input. As pointed out by [15], Honest-Input Validity is a particular case of Convex Validity, hence the proof relies on Corollary 4.

Corollary 5. *Honest-Input Validity on a space of size $m > 2$ can be solved responsively if and only if $n > m \cdot t_s + t_r$, both in authenticated and unauthenticated settings.*

7 Conclusions

In this work, we focused on investigating how validity properties can affect the feasibility of synchronous BA protocols that achieve security up to t_s corruptions and terminate *responsively* under t_r corruptions (i.e., without depending on the assumed network delay Δ).

We have presented a tight characterization: one can solve a *non-trivial* validity property responsively if and only if (1) $n > 2 \cdot t_r + t_s$ holds in authenticated settings, or $n > 3 \cdot t_s$ holds in unauthenticated settings, and (2) the responsive similarity condition holds. Our sufficiency result is constructive, and relies on a variant of ACS providing the honest parties with a suitable view over the honest inputs. The main technical challenge in this construction is building an appropriate authenticated ACS protocol matching the condition $n > 2 \cdot t_r + t_s$, which required maintaining security for thresholds with an honest-minority. Finally, we have presented corollaries instantiating our results for standard validity conditions.

Our work leaves a number of open questions. First, our constructions are designed to match the tight feasibility thresholds, but they do not aim to optimize communication. It would be interesting to see whether one can extend the message complexity tight results of non-responsive BA to responsive BA protocols, in line with [10, 18], or to obtain protocols with efficient/optimal bit complexity. Another interesting direction refers to designing responsive protocols for relaxed versions of the problem that we have considered by exploring weaker versions of consistency. For instance, Approximate Agreement (AA) is a relaxation of BA which allows the honest outputs to differ by at most ε.

8 Acknowledgements

We thank the anonymous DISC 2026 reviewers for their pointers to deterministic responsive BA protocols.

C. Liu-Zhang and D. Ghinea were supported in part by the SNSF grant number 2000-1-243069.

References

[1] Ittai Abraham and Danny Dolev. Byzantine agreement with optimal early stopping, optimal resilience and polynomial complexity. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 605–614, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2746539.2746581.

[2] Hagit Attiya and Jennifer L. Welch. Bounds on Worst-Case Responsiveness for Agreement Algorithms. 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 32:1–32:22, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2023.32, doi:10.4230/LIPIcs.OPODIS.2023.32.

[3] Michael Ben-Or, Ran Canetti, and Oded Goldreich. Asynchronous secure computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 52–61, New York, NY, USA, 1993. Association for Computing Machinery. doi:10.1145/167088.167109.

[4] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. Asynchronous secure computations with optimal resilience (extended abstract). In *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '94, page 183–192, New York, NY, USA, 1994. Association for Computing Machinery. doi:10.1145/197917.198088.

[5] Piotr Berman, Juan A Garay, and Kenneth J Perry. Optimal early stopping in distributed consensus. In *Distributed Algorithms: 6th International Workshop, WDAG'92 Haifa, Israel, November 2–4, 1992 Proceedings 6*, pages 221–237. Springer, 1992.

[6] 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, Heidelberg, November 2020. doi:10.1007/978-3-030-64375-1_13.

[7] Erica Blum, Chen-Da Liu Zhang, and Julian Loss. Always have a backup plan: Fully secure synchronous MPC with asynchronous fallback. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 707–731. Springer, Heidelberg, August 2020. doi:10.1007/978-3-030-56880-1_25.

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

[9] Ran Canetti and Tal Rabin. Fast asynchronous byzantine agreement with optimal resilience. In *25th ACM STOC*, pages 42–51. ACM Press, May 1993. doi:10.1145/167088.167105.

[10] Pierre Civit, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, Anton Paramonov, and Manuel Vidigueira. All byzantine agreement problems are expensive. In *Proceedings*

of the 43rd ACM Symposium on Principles of Distributed Computing, PODC '24, page 157–169, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3662158.3662780.

[11] Pierre Civit, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, and Manuel Vidigueira. On the validity of consensus. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing*, pages 332–343, 2023.

[12] 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.

[13] Andrei Constantinescu, Marc Dufay, Anton Paramonov, and Roger Wattenhofer. From few to many faults: Optimal adaptive byzantine agreement, 2026. URL: https://arxiv.org/abs/2505.19989, arXiv:2505.19989.

[14] 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. doi:10.4230/LIPIcs.OPODIS.2023.8.

[15] 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.

[16] Giovanni Deligios, Ivana Klasovita, and Chen-Da Liu-Zhang. Asymptotically optimal early termination for dishonest majority broadcast. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, pages 116–144, Cham, 2025. Springer Nature Switzerland.

[17] Giovanni Deligios and Mose Mizrahi Erbes. Closing the efficiency gap between synchronous and network-agnostic consensus. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 432–461, Cham, 2024. Springer Nature Switzerland.

[18] Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. *J. ACM*, 32(1):191–204, jan 1985. doi:10.1145/2455.214112.

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

[20] Marc Dufay, Diana Ghinea, and Anton Paramonov. General convex agreement with near-optimal communication, 2026. URL: `https://arxiv.org/abs/2602.21411`, `arXiv:2602.21411`.

[21] 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.

[22] 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`.

[23] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Brief Announcement: Communication-Optimal Convex Agreement. In *The 43rd ACM Symposium on Principles of Distributed Computing (PODC), Nantes, France*, June 2024.

[24] Diana Ghinea, Chen-Da Liu-Zhang, and Roger Wattenhofer. Communication-optimal convex agreement. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '25, page 39–49, New York, NY, USA, 2025. Association for Computing Machinery. `doi:10.1145/3732772.3733551`.

[25] Guy Goren, Yoram Moses, and Alexander Spiegelman. Probabilistic indistinguishability and the quality of validity in byzantine agreement. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, AFT '22, page 111–125, New York, NY, USA, 2023. Association for Computing Machinery. `doi:10.1145/3558535.3559789`.

[26] Martin Hirt, Ard Kastrati, and Chen-Da Liu-Zhang. Multi-threshold asynchronous reliable broadcast and consensus. In *24th International Conference on Principles of Distributed Systems*, page 1, 2021.

[27] Chen-Da Liu-Zhang, Julian Loss, Ueli Maurer, Tal Moran, and Daniel Tschudi. MPC with synchronous security and asynchronous responsiveness. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 92–119. Springer, Heidelberg, December 2020. `doi:10.1007/978-3-030-64840-4_4`.

[28] Julian Loss and Tal Moran. Combining asynchronous and synchronous byzantine agreement: The best of both worlds. Cryptology ePrint Archive, Report 2018/235, 2018. `https://eprint.iacr.org/2018/235`.

[29] Julian Loss and Tal Moran. Combining asynchronous and synchronous byzantine agreement: The best of both worlds. Cryptology ePrint Archive, Paper 2018/235, 2018. URL: `https://eprint.iacr.org/2018/235`.

[30] Julian Loss and Jesper Buus Nielsen. Early stopping for any number of corruptions. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 457–488, Cham, 2024. Springer Nature Switzerland.

[31] Darya Melnyk, Yuyi Wang, and Roger Wattenhofer. Byzantine Preferential Voting. In *14th Conference on Web and Internet Economics (WINE), Oxford, United Kingdom*, December 2018.

[32] Darya Melnyk and Roger Wattenhofer. Byzantine agreement with interval validity. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, pages 251–260, 2018. doi:10.1109/SRDS.2018.00036.

[33] Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.

[34] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of BFT protocols. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 31–42. ACM Press, October 2016. doi:10.1145/2976749.2978399.

[35] Achour Mostéfaoui, Hamouma Moumen, and Michel Raynal. Signature-free asynchronous byzantine consensus with t¡ n/3 and o (n2) messages. In *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, pages 2–9, 2014.

[36] Gil Neiger. Distributed consensus revisited. *Information Processing Letters*, 49(4):195–201, 1994. doi:10.1016/0020-0190(94)90011-6.

[37] 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. doi:10.4230/LIPIcs.DISC.2019.29.

[38] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 91. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[39] Rafael Pass and Elaine Shi. Thunderella: Blockchains with optimistic instant confirmation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 3–33. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78375-8_1.

[40] Kenneth J Perry and Sam Toueg. An authenticated byzantine generals algorithm with early stopping. Technical report, Cornell University, 1984.

[41] 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. doi:10.4230/LIPIcs.OPODIS.2015.22.

Appendix

A Necessary Conditions: Missing Proofs

We present the proofs for the results stated in Section 3 regarding our necessary conditions.

Similarly to [12], we model randomness by assigning each party an infinite local random tape, read sequentially during the execution. If the protocol uses shared randomness, we additionally include the shared random tapes in the randomness of the execution. A choice of random tapes is denoted by r.

An execution of a randomized protocol is determined by the input configuration, the set of corrupted parties and their behavior, the message schedule, and the random tapes. Once the random tapes are fixed, the behavior of every honest party is deterministic. Then, for a fixed choice of random tapes r, we may write $\epsilon(r)$ for the corresponding deterministic execution.

Our impossibility results rely on indistinguishability arguments, where we compare executions defined using the same choice of random tapes. This means that a party that is honest in more than one of the executions receives the same local random bits in each of them. Shared random tapes, if any, are also the same in all executions. When we say that a byzantine party behaves as in another execution, we mean that its strategy is prescribed to reproduce the corresponding messages.

We present the proofs from our necessary conditions described in Section 3.

A.1 General Condition

We first focus Lemma 1, which establishes that, any protocol solving VAL also solves $\text{VAL}^\star$.

Lemma 1. *Consider a validity property* VAL*, and let* $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ |J| \geq n - t_s} \text{VAL}(J)$ *for any input configuration* I*. If there is a* t_s*-secure* BA *protocol* Π *solving* VAL*, then* Π *is also a* t_s*-secure* BA *protocol solving* $\text{VAL}^\star$*.*

Proof. Consider a protocol Π solving VAL. Let I be an input configuration. We show that, in every terminating execution of Π with input configuration I, the value agreed upon belongs to $\text{VAL}^\star(I)$.

Let ε be an arbitrary terminating execution of Π with input configuration I, and let v be the agreed value output by the honest parties in ε.

Consider an arbitrary input configuration $I' \subseteq I$ such that $\left|I'\right| \geq n - t_s$. We define an execution ε' as follows: the input configuration is I', and all parties outside $\text{PARTIES}(I')$ are corrupted. Each party outside $\text{PARTIES}(I')$ reproduces its behavior from ε; in particular, parties in $\text{PARTIES}(I) \backslash \text{PARTIES}(I')$ send the same messages as in ε. All messages are scheduled exactly as in ε. Then every honest party in $\text{PARTIES}(I')$ has the same view in ε' as in ε, and therefore outputs the same value v in both executions. Since Π solves VAL, execution ε' must satisfy validity for input configuration I'. Hence, $v \in \text{VAL}(I')$.

As this holds for every input configuration $I' \subseteq I$ with $\left|I'\right| \geq n - t_s$, we obtain $v \in \bigcap_{I' \subseteq I,\ \left|I'\right| \geq n - t_s} \text{VAL}(I') = \text{VAL}^\star(I)$. Therefore, every terminating execution of Π with input configuration I outputs a value in $\text{VAL}^\star(I)$. Since I was arbitrary, Π solves $\text{VAL}^\star$. □

We next focus on Lemma 2, which shows that, if $\text{VAL}^\star$ can be solved when $n \leq 2 \cdot t_r + t_s$ (given that $t_r > 0$), then all maximal input configurations share a common $\text{VAL}^\star$-valid output.

Our proof makes use of the finite intersection property, stated below.

Lemma 16. *Let X be a finite set, and let $\{A_\alpha\}_{\alpha\in\Lambda}$ be a family of subsets of X. If every finite subfamily has non-empty intersection, i.e., for every finite $\Lambda' \subseteq \Lambda$, $\bigcap_{\alpha\in\Lambda'} A_\alpha \neq \varnothing$, then $\bigcap_{\alpha\in\Lambda} A_\alpha \neq \varnothing$.*

Proof. Assume that $\bigcap_{\alpha\in\Lambda} A_\alpha = \varnothing$. Then, for every $x \in X$, there is an index $\alpha_x \in \Lambda$ such that $x \notin A_{\alpha_x}$. Since X is finite, the set $\Lambda' = \{\alpha_x : x \in X\}$ is finite. Moreover, $\bigcap_{\alpha\in\Lambda'} A_\alpha = \varnothing$, because every $x \in X$ is excluded by A_{α_x}. This contradicts the assumption that every finite subfamily has non-empty intersection. Therefore, $\bigcap_{\alpha\in\Lambda} A_\alpha \neq \varnothing$. □

Lemma 2. *Let* VAL *be a validity property, and define* $\text{VAL}^\star(I) = \bigcap_{J\subseteq I,\ |J|\geq n-t_s} \text{VAL}(J)$. *If there is a t_s-secure t_r-responsive* BA *protocol solving* $\text{VAL}^\star$ *when $n \leq 2\cdot t_r + t_s$ and $t_r > 0$, then there is a value $v^\star \in V_{\mathsf{out}}$ such that, for every maximal input configuration I, $v^\star \in \text{VAL}^\star(I)$.*

Proof. Let Π be a t_s-secure t_r-responsive BA protocol solving $\text{VAL}^\star$: note that this implies that $\text{VAL}^\star(I) \neq \varnothing$ for any input configuration I. Let f be the polynomial from the Responsive Termination guarantee of Π. Choose δ and Δ such that $\tau := \delta \cdot f(\lambda) < \Delta$.

Since $n \leq 2 \cdot t_r + t_s$, we can partition $\mathcal{P}$ into three disjoint sets S_0, S_1, S_b such that $0 < |S_0| \leq t_r$, $0 < |S_1| \leq t_r$, $|S_b| \leq t_s$. Moreover, $|S_0 \cup S_b| \geq n - t_r$ and $|S_1 \cup S_b| \geq n - t_r$.

Fix a maximal input configuration I_0 such that $\text{VAL}(I_0)$ is finite: since I_0 appears in the intersection defining $\text{VAL}^\star(I_0)$, we have $\text{VAL}^\star(I_0) \subseteq \text{VAL}(I_0)$, and hence $\text{VAL}^\star(I_0)$ is finite. We first prove that, for every finite collection $\mathcal{F}$ of maximal input configurations, $\text{VAL}^\star(I_0) \cap \bigcap_{I\in\mathcal{F}} \text{VAL}^\star(I) \neq \varnothing$. Let $\mathcal{F} = \{I_1, \ldots, I_m\}$ be such a finite collection. For each $\ell \in \{1, \ldots, m\}$ and each choice of random tapes r, define the following executions:

- In execution $\varepsilon_0(r)$, the input configuration is I_0. Messages sent by parties in S_1 are delivered after exactly $\Delta > \tau$ time, while messages from parties in $S_0 \cup S_b$ are delivered within delay δ.
- In execution $\varepsilon'_0(r)$, the input configuration is the restriction of I_0 to parties in $S_0 \cup S_b$, and the parties in S_1 are byzantine and send no messages.
- In execution $\varepsilon_\ell(r)$, the input configuration is I_ℓ. Messages sent by parties in S_0 are delivered after exactly $\Delta > \tau$ time, while messages sent by parties in $S_1 \cup S_b$ are delivered within delay δ.
- In execution $\varepsilon'_\ell(r)$, the input configuration is the restriction of I_ℓ to $S_1 \cup S_b$, and the parties in S_0 are byzantine and send no messages.
- Finally, define an execution $\varepsilon^{I_\ell}(r)$ in which the honest parties are $S_0 \cup S_1$: parties in S_0 have the inputs assigned by I_0, parties in S_1 have the inputs assigned by I_ℓ, and parties in S_b are byzantine. Each byzantine party in S_b simulates two copies of itself, one behaving as in the execution with input configuration I_0 and one behaving as in the execution with input configuration I_ℓ. Messages between S_0 and the I_0-copies of S_b are scheduled as in $\varepsilon'_0(r)$, and messages between S_1 and the I_ℓ-copies of S_b are scheduled as in $\varepsilon'_\ell(r)$. All other messages are delivered after exactly $\Delta > \tau$ time.

For each $\ell \in \{1, \ldots, m\}$, Responsive Termination implies that, except with negligible probability over r, all honest parties output by time τ in both $\varepsilon'_0(r)$ and $\varepsilon'_\ell(r)$. Since m is finite, a union bound implies that, except with negligible probability over r, all of these termination events occur simultaneously for every $\ell \in \{1, \ldots, m\}$. In particular, for sufficiently large λ, this event has positive probability. Hence, there exists a choice of random tapes $r^\star$ for which all of these executions terminate by time τ. Fix such an $r^\star$.

Let v_0 be the common value output by the honest parties in $\varepsilon'_0(r^\star)$, and, for each ℓ, let v_ℓ be the common value output by the honest parties in $\varepsilon'_\ell(r^\star)$.

The parties in S_0 cannot distinguish $\varepsilon'_0(r^\star)$ from $\varepsilon_0(r^\star)$ or from $\varepsilon^{I_\ell}(r^\star)$ up to time τ. Hence, in both executions, the parties in S_0 output v_0 by time τ. Similarly, the parties in S_1 cannot distinguish $\varepsilon'_\ell(r^\star)$ from $\varepsilon_\ell(r^\star)$ or from $\varepsilon^{I_\ell}(r^\star)$ up to time τ. Hence, in both executions, the parties in S_1 output v_ℓ by time τ.

Since $\varepsilon^{I_\ell}(r^\star)$ is a valid execution with at most t_s byzantine parties, Agreement implies that $v_0 = v_\ell$.

Moreover, $\varepsilon_0(r^\star)$ is a valid execution with maximal input configuration I_0, so Validity for $\text{VAL}^\star$ gives $v_0 \in \text{VAL}^\star(I_0)$. Likewise, for each ℓ, $\varepsilon_\ell(r^\star)$ is a valid execution with maximal input configuration I_ℓ, so Validity for $\text{VAL}^\star$ gives $v_\ell \in \text{VAL}^\star(I_\ell)$. Since $v_0 = v_\ell$ for every ℓ, we obtain $v_0 \in \text{VAL}^\star(I_0) \cap \bigcap_{\ell=1}^{m} \text{VAL}^\star(I_\ell)$. This proves the finite-intersection claim.

We now use the fact that $\text{VAL}^\star(I_0)$ is finite. The finite-intersection claim says that the family of subsets $\{\text{VAL}^\star(I_0) \cap \text{VAL}^\star(I) : I \text{ maximal}\}$ has the finite intersection property inside the finite set $\text{VAL}^\star(I_0)$. We apply Lemma 16 with $X = \text{VAL}^\star(I_0)$ and $A_I = \text{VAL}^\star(I_0) \cap \text{VAL}^\star(I)$ for every maximal input configuration I. The finite-intersection claim shows that every finite subfamily of the sets A_I has non-empty intersection. Since $X = \text{VAL}^\star(I_0)$ is finite, Lemma 16 implies $\text{VAL}^\star(I_0) \cap \bigcap_{I \text{ maximal}} \text{VAL}^\star(I) \neq \varnothing$. Then, let $v^\star$ be any value in this intersection: $v^\star \in \text{VAL}^\star(I)$ for every maximal input configuration I, which completes the proof. □

Finally, we establish that, if there is a value $v^\star$ such that $v^\star \in \text{VAL}^\star(I)$ for every maximal configuration I, then the original validity property VAL is trivial.

Lemma 3. *Let* VAL *be a validity property, and define* $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ |J| \geq n - t_s} \text{VAL}(J)$. *If there is a value* $v^\star \in V_{\mathsf{out}}$ *such that, for every maximal input configuration* I, $v^\star \in \text{VAL}^\star(I)$, *then* VAL *is trivial.*

Proof. We show that the same value $v^\star$ is valid for every input configuration $J \in \mathcal{I}$. Fix an arbitrary $J \in \mathcal{I}$. Let I be a maximal input configuration extending J; that is, $J \subseteq I$. Since $J \in \mathcal{I}$, we have $|J| \geq n - t_s$. Therefore, J appears among the configurations in the intersection defining $\text{VAL}^\star(I)$, and hence $\text{VAL}^\star(I) = \bigcap_{J' \subseteq I, |J'| \geq n - t_s} \text{VAL}(J') \subseteq \text{VAL}(J)$. By assumption, $v^\star \in \text{VAL}^\star(I)$ for every maximal input configuration I. In particular, $v^\star \in \text{VAL}^\star(I)$ for the maximal extension I of J chosen above. Hence $v^\star \in \text{VAL}(J)$.

Since $J \in \mathcal{I}$ was arbitrary, we have $v^\star \in \bigcap_{J \in \mathcal{I}} \text{VAL}(J)$. It follows that $\bigcap_{J \in \mathcal{I}} \text{VAL}(J) \neq \varnothing$, and therefore VAL is trivial. □

A.2 Responsive Similarity Condition

We present the proofs for Lemma 4, Lemma 5, and Theorem 4, restated below.

Lemma 4. *Assume there is a t_s-secure* BA *protocol* Π *solving a validity property* VAL*. Then there is a Turing-computable function $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ such that, for every input configuration $I \in \mathcal{I}$, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.*

Proof. By Lemma 1, protocol Π also solves the stronger condition $\text{VAL}^\star$, where $\text{VAL}^\star(I) = \bigcap_{J \subseteq I,\ \left|J\right| \geq n - t_s} \text{VAL}(J)$. Since $\text{SIMILAR}_S(I) = \{J \subseteq I : \left|J\right| \geq n - t_s\}$, we have $\text{VAL}^\star(I) = \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.

It remains to define a Turing-computable function σ.

Given an input configuration I, consider a nondeterministic Turing machine that simulates an execution of Π with input configuration I where all parties outside $\text{PARTIES}(I)$ are byzantine and crash at the beginning of the execution, and messages between honest parties are delivered after exactly Δ time. If the protocol is authenticated, the simulation also fixes the PKI and gives each simulated honest party its own signing key. The honest parties use their signing keys in the natural way, while this is irrelevant for byzantine parties, as they send no messages. Whenever a simulated honest party requests a random bit, the nondeterministic machine branches into two computations, one in which the bit is 0 and one in which the bit is 1. Then, each computation branch corresponds to one possible choice of random tapes for the honest parties. If all honest parties output, the machine returns their common output value.

Since Π satisfies Termination, this simulated execution terminates with probability 1. Hence, there exists a choice of random tapes for which all honest parties terminate. Along such a terminating execution, only finitely many random bits are read before termination, so the nondeterministic machine has a halting branch. Moreover, by Agreement and Validity, every halting branch returns the output of a valid terminating execution with input configuration I. Since Π solves $\text{VAL}^\star$, every such output belongs to $\text{VAL}^\star(I)$.

Finally, since we only care about computability, the existence of a halting nondeterministic branch implies the existence of a deterministic Turing machine that finds one such branch and returns its output. This defines a Turing-computable function σ. Since every halting branch returns a value in $\text{VAL}^\star(I)$, we have $\sigma(I) \in \text{VAL}^\star(I) = \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. □

Lemma 5. *Assume there is a t_s-secure t_r-responsive* BA *protocol* Π *solving a non-trivial validity property* VAL*. Then, there is a Turing-computable function $\sigma : \{I \in \mathcal{I} : \left|I\right| \geq n - t_r\} \to V_{\mathsf{out}}$ such that, for every input configuration $I \in \mathcal{I}$ with $\left|I\right| \geq n - t_r$, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.*

Proof. Fix an input configuration $I \in \mathcal{I}$ with $\left|I\right| \geq n - t_r$. Choose δ and Δ such that $\tau := \delta \cdot f(\lambda) < \Delta$, where f is the polynomial witnessing the Responsive Termination guarantee of Π.

For each choice of random tapes r, let $\varepsilon_I(r)$ be the execution with input configuration I, where all parties outside $\text{PARTIES}(I)$ are byzantine and crash at the beginning of the execution. All messages between parties in $\text{PARTIES}(I)$ are delivered after exactly δ time. Since at most t_r parties are corrupted in $\varepsilon_I(r)$, Responsive Termination implies that, except with probability $\text{negl}(\lambda)$ over r, all honest parties in $\text{PARTIES}(I)$ output by time τ. Hence, for sufficiently large λ, there exists a choice of random tapes $r^\star$ for which this happens. Then, honest parties in $\text{PARTIES}(I)$ obtain outputs by time τ in $\varepsilon_I(r^\star)$, and, by Agreement, they output the same value v.

We show that $v \in \text{VAL}(J)$ for every $J \in \text{SIMILAR}_R(I)$. By definition, $\left|J\right| \geq n - t_s$, and J is compatible with I. Moreover, since $\left|I\right| \geq n - t_r$ and $\left|J\right| \geq n - t_s$, and since $n > 2 \cdot t_r + t_s$ by Theorem 2, we have $\text{PARTIES}(I) \cap \text{PARTIES}(J) \neq \varnothing$. We define an execution $\varepsilon_J(r^\star)$ with input configuration J. The parties in $\text{PARTIES}(I) \setminus \text{PARTIES}(J)$ are byzantine, but they follow Π correctly using the inputs assigned to them in I and the random tapes prescribed by $r^\star$. Parties outside $\text{PARTIES}(I) \cup \text{PARTIES}(J)$ are byzantine and crash at the beginning of the execution. Note that this is well-defined even if the protocol is authenticated: parties in $\text{PARTIES}(I) \setminus \text{PARTIES}(J)$ are byzantine in $\varepsilon_J(r^\star)$, and may therefore reproduce the messages they sent in $\varepsilon_I(r^\star)$, including signatures under their own keys. Messages among parties in $\text{PARTIES}(I)$ are scheduled exactly as in $\varepsilon_I(r^\star)$ up to time τ; all other messages are delayed for exactly Δ time – hence, until after time τ. Every party in $\text{PARTIES}(I) \cap \text{PARTIES}(J)$ has the same input, the same random tape, and receives the same messages up to time τ in $\varepsilon_J(r^\star)$ as in $\varepsilon_I(r^\star)$. Hence, such a party cannot distinguish $\varepsilon_J(r^\star)$ from $\varepsilon_I(r^\star)$ up to time τ, and therefore outputs v in $\varepsilon_J(r^\star)$. Since $\varepsilon_J(r^\star)$ is a valid execution with input configuration J and at most t_s byzantine parties, the Validity property of Π implies $v \in \text{VAL}(J)$. As this holds for every $J \in \text{SIMILAR}_R(I)$, we obtain $v \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.

As the above argument only uses the fact that $\varepsilon_I(r)$ outputs by time τ, every choice of random tapes with this property yields an output in $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.

It remains to define σ as a Turing-computable function. Given an input configuration I with $\left|I\right| \geq n - t_r$, consider a nondeterministic Turing machine that searches over security parameters $\lambda = 1, 2, \ldots$. Fix a delay δ, say $\delta = 1$. For each λ, let $\tau_\lambda = \delta \cdot f(\lambda)$ and choose a synchronous delay bound $\Delta_\lambda > \tau_\lambda$, for instance $\Delta_\lambda = 2\delta \cdot f(\lambda)$. The machine simulates the execution ε_I described above with parameters $\lambda, \delta, \Delta_\lambda$: all parties outside $\text{PARTIES}(I)$ are byzantine and crash, and all messages between parties in $\text{PARTIES}(I)$ are delivered after exactly δ time. If the protocol is authenticated, the machine also fixes the PKI for the simulated execution and gives each simulated honest party its own signing key. Since all parties outside $\text{PARTIES}(I)$ crash in this simulation, no byzantine party needs to produce any authenticated message. Whenever a simulated honest party requests a random bit, the machine branches into two computations, one in which the bit is 0 and one in which the bit is 1. Then, each computation branch corresponds to one possible choice of random tapes for the honest parties. For the current value of λ, the machine simulates the execution up to time τ_λ. If all honest parties output by time τ_λ, the machine returns their common output value.

For all sufficiently large λ, Responsive Termination guarantees that there is a choice of random tapes for which all honest parties output by time τ_λ. Hence the nondeterministic search has a halting branch. Moreover, by the argument above, every halting branch returns a value in $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.

Finally, since we only care about computability, a deterministic Turing machine can search for a halting branch of the nondeterministic computation and return its output. This defines a Turing-computable function σ. Since every halting branch returns a value in $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$, we have $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$. □

Theorem 4. *Assume there is a t_s-secure t_r-responsive* BA *protocol Π solving a non-trivial validity property* VAL*. Then* VAL *satisfies* the responsive similarity condition*: there is a Turing-computable function $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ such that, for every input configuration $I \in \mathcal{I}$:*

- *if $\left|I\right| \geq n - t_r$, then $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$;*

- *otherwise,* $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.

Proof. By Lemma 4, there is a Turing-computable function $\sigma_S : \mathcal{I} \to V_{\mathsf{out}}$ such that, for every input configuration $I \in \mathcal{I}$, $\sigma_S(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. By Lemma 5, there is a Turing-computable function $\sigma_R : \{I \in \mathcal{I} : \left|I\right| \geq n - t_r\} \to V_{\mathsf{out}}$ such that, for every input configuration $I \in \mathcal{I}$ with $\left|I\right| \geq n - t_r$, $\sigma_R(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$. We define $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ as follows: if $\left|I\right| \geq n - t_r$, let $\sigma(I) = \sigma_R(I)$; otherwise, let $\sigma(I) = \sigma_S(I)$. Since σ_S and σ_R are Turing-computable, and since the condition $\left|I\right| \geq n - t_r$ is decidable, the function σ is Turing-computable. Moreover, by construction, for every input configuration $I \in \mathcal{I}$:

- if $\left|I\right| \geq n - t_r$, then $\sigma(I) = \sigma_R(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$;
- otherwise, $\sigma(I) = \sigma_S(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.

Hence, VAL satisfies the responsive similarity condition. □

B Unauthenticated ACS

In this section, we present the construction behind Theorem 5, restated below. As mentioned in Section 4, the construction behind Theorem 5 relies on the asynchronous protocol of [4] (as presented in [34]), with minor adjustments.

Theorem 5. *There are deterministic and randomized unauthenticated t_s-secure t_r-responsive* ACS *protocols Π_{ACS} whenever $n > 3 \cdot t_s$. For the deterministic protocol, the polynomial witnessing both Termination and Responsive Termination is $f_{\text{ACS}}(\lambda) = O(t_s)$. For the randomized protocol, this is $f_{\text{ACS}}(\lambda) = O(\lambda)$.*

The only (potentially) randomized component in our unauthenticated ACS construction is a building block, namely a t_s-secure t_r-responsive BA protocol solving Strong Unanimity. It will therefore be useful to restate Theorem 5 to make the assumed BA protocol explicit:

Theorem 9. *Assume a t_s-secure t_r-responsive protocol Π_{BA} achieving* BA *with Strong Unanimity on $\{0,1\}$ whenever $n > 3 \cdot t_s$, and let $f_{\text{BA}}(\lambda)$ be the polynomial witnessing its Termination and Responsive Termination guarantees. Then, there is an unauthenticated t_s-secure t_r-responsive* ACS *protocol Π_{ACS} whenever $n > 3 \cdot t_s$. The polynomial witnessing Termination and Responsive Termination is $f_{\text{ACS}}(\lambda) = O(1) \cdot f_{\text{BA}}(\lambda)$.*

In the asynchronous ACS mechanism of [4], all honest parties first distribute their input values via a Reliable Broadcast protocol (RBC) [8], denoted by Π_{RBC} – this enables parties to distribute values in a consistent manner, but comes with the caveat that, if a value's sender is corrupted, then the parties may wait forever to obtain an output. We provide the formal definition below. The parties decide which RBC invocations are worth waiting for with the help of n invocations of a BA protocol solving Strong Unanimity, which we denote by Π_{BA} – there is one Π_{BA} invocation for each party. Once a party P receives a value from party P' via Π_{RBC}, it joins the Π_{BA} protocol corresponding to party P' with input 1. Once a party observes output 1 in $n - t$ such invocations (where $t = t_s = t_r$), it joins all Π_{BA} invocations that it has not yet joined with input 0. Once all invocations terminate, the view agreed upon

consists of the values received via Π_{rBC} from parties whose corresponding Π_{BA} invocations resulted in output 1.

The termination guarantees of this mechanism are inherited from Π_{BA}. However, returning to our synchronous setting, we need to recall the Core Quality property: if the number of corruptions is $t_r \leq t_s < n/3$, and the view agreed upon consists of fewer than $n - t_r$ values, then all honest parties must appear in the view.

We achieve this by making a minor adjustment to the mechanism of [4]. If the network delay is $\delta \leq \Delta$, Π_{rBC} ensures that honest senders' invocations terminate within $c_{\text{rBC}} \cdot \delta$ time, where $c_{\text{rBC}} := 3$. This way, parties may avoid joining Π_{BA} invocations with input 0 until either $n - t_r$ invocations have terminated with output 1, or $c_{\text{rBC}} \cdot \Delta$ time has passed – this time limit ensures that all honest values sent via Π_{rBC} were delivered. Then, the Core Quality property is achieved.

In the following, we first define the building blocks, and afterwards we present the code of the protocol behind Theorem 9.

Reliable Broadcast. We recall the definition of Reliable Broadcast (rBC) [8], making the runtime explicit.

Definition 5 (Reliable Broadcast)**.** *Let Π denote a protocol where a designated party S (the sender) holds a value v_S, and every party P may output a value v_P. We say that Π is a t_s-secure* rBC *protocol if it achieves Validity, Consistency, Honest Termination, and Conditional Termination up to t_s corruptions.*

- ***Validity**: If S is honest, and an honest party outputs v_P, then $v_P = v_S$.*
- ***Consistency**: If P and P' are honest and output v_P and resp. $v_{P'}$, then $v_P = v_{P'}$.*
- ***Honest Termination**: If S is honest and the network delay is up to δ, all honest parties obtain outputs within $c_{\text{rBC}} \cdot \delta$ time.*
- ***Conditional Termination**: If an honest party P outputs at some time τ and the network delay is up to δ, all honest parties obtain outputs by time $\tau + c'_{\text{rBC}} \cdot \delta$.*

We make use of Bracha's rBC protocol, described by the theorem below. The analysis regarding the timing guarantees follows from [22, Theorem 4.2].

Theorem 10 ([8])**.** *There is an unauthenticated t_s-secure t_r-responsive* rBC *protocol Π_{rBC} whenever $t_s < n/3$, with $c_{\text{rBC}} := 3$ and $c'_{\text{rBC}} := 2$.*

BA with Strong Unanimity. The next building block is BA with Strong Unanimity on $\{0, 1\}$: if all honest parties hold input b, no honest party outputs $b' \neq b$. We denote the protocol by Π_{BA}, and assume it is t_s-secure and t_r-responsive whenever $t_s < n/3$.

We are running Π_{BA} in a setting where honest parties may start actively participating at different times. We therefore use the following convention: an honest party *joins* an invocation of Π_{BA} once it provides its input to that invocation and starts executing the protocol. Messages addressed to an honest party before it joins are buffered until it joins, and its local execution and timers start when it joins. An asynchronous protocol adapts to this setting seamlessly. A synchronous t_s-secure t_r-responsive protocol does so as well, given that the difference between the honest parties' join times can be absorbed into the message delay.

Remark 1. *Consider a t_s-secure t_r-responsive* BA *protocol Π_{BA} with Strong Unanimity, and let $f_{\mathrm{BA}}(\lambda)$ be the polynomial witnessing its Termination and Responsive Termination guarantees. Assume that the synchronous delay upper bound is set to $\Delta' := (c+1)\cdot\Delta$.*

If all honest parties join an invocation between times τ and $\tau + c\cdot\Delta$, then all honest parties terminate by time $\tau + c\cdot\Delta + (c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\Delta$, except with negligible probability, and all honest parties terminate eventually almost surely. Similarly, if up to t_r parties are corrupted, the actual network delay is at most δ, and all honest parties join between times τ and $\tau + c\cdot\delta$, then all honest parties terminate by time $\tau + c\cdot\delta + (c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\delta$, except with negligible probability. Moreover, Π_{BA} still achieves BA *with Strong Unanimity under delayed starts.*

Proof. We assume that every message sent by an honest party is tagged with the sender's local send time. A receiving party buffers messages, and considers messages as received once the sending time is at least the time in the tag.

Let τ_P be the time at which an honest party P joins the invocation. We construct a virtual execution in which all honest parties start at virtual time 0.

Consider a message sent by an honest party P at its real time x to an honest party P'. In the original execution, the message is sent at time $\tau_P + x$ and reaches P' by time $\tau_P + x + \delta$. By the buffering convention, P' processes it at real protocol time $x' \leq \max(x, \tau_P + x + \delta - \tau_{P'})$. Since the honest join times differ by at most $c\cdot\delta$, we have $0 \leq x' - x \leq c\cdot\delta + \delta = (c+1)\cdot\delta$. Consequently, in the virtual execution, every message sent between honest parties is received after it is sent and within $(c+1)\cdot\delta$ time.

Messages sent by byzantine parties are unrestricted and can therefore be reproduced by the byzantine parties in the virtual execution. Hence, the virtual execution is a valid execution of Π_{BA} whose message delay is at most $(c+1)\cdot\delta \leq \Delta'$ and where honest parties start executing Π_{BA} at the same time. Hence, Π_{BA} achieves BA in the virtual execution.

By construction, every honest party's local execution is identical in the original and virtual executions. The Agreement and Strong Unanimity properties of Π_{BA} in the original execution follow from Π_{BA} achieving these in the virtual execution. If up to t_r parties are corrupted, Responsive Termination ensures that every honest party terminates by virtual time $(c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\delta$ except with negligible probability. This means that, in the original execution, every honest party P therefore terminates by time $\tau_P + (c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\delta$, hence by time $\tau + c\cdot\delta + (c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\delta$. For Termination, the virtual message delay is then at most $(c+1)\cdot\Delta = \Delta'$, so all honest parties terminate within $(c+1)\cdot f_{\mathrm{BA}}(\lambda)\cdot\Delta$ time after the last honest party joins in the original execution, except with negligible probability. Moreover, as Π_{BA} terminates almost surely in every execution with up to t_s corrupted parties in the virtual execution, the honest parties also terminate almost surely in the original execution. □

Since honest parties joining late can impact the termination time, as described in Remark 1, we want to ensure that honest parties join a Π_{BA} execution at most $O(1)\cdot\delta$ time away from each other in our ACS construction. To this end, it will be useful to bound the difference between the times at which honest parties obtain their outputs in Π_{BA} by $O(1)\cdot\delta$. To obtain this property, we define $\Pi_{\mathrm{BA}^\star}$ by adding one step in which the parties distribute their outputs from Π_{BA} via Π_{RBC} and wait until $n - t_s$ parties have confirmed the same output.

Protocol $\Pi_{\mathrm{BA}^\star}$

Code for party P with input v_{in}

1: Join Π_{BA} with input v_{in}.
2: Upon obtaining an output v_{out} from Π_{BA}, send v_{out} to all parties via Π_{rBC}.
3: Upon receiving the same v_{out} from $n - t_s$ parties via Π_{rBC}, output v_{out}.

Lemma 17. *Assume that $n > 3 \cdot t_s$ and that Π_{BA} is a t_s-secure t_r-responsive* BA *protocol achieving Strong Unanimity, and let $f_{\mathrm{BA}}(\lambda)$ be the polynomial witnessing its Termination and Responsive Termination guarantees. Then $\Pi_{\mathrm{BA}^\star}$ is a t_s-secure t_r-responsive* BA *protocol achieving Strong Unanimity, with Termination and Responsive Termination polynomial $f_{\mathrm{BA}^\star}(\lambda) = f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}$.*

Moreover, if there are up to t_s corruptions and an honest party obtains an output at time τ, then all honest parties obtain the same output by time $\tau + c'_{\mathrm{BA}^\star} \cdot \delta$, where $c'_{\mathrm{BA}^\star} = c'_{\mathrm{rBC}}$.

Proof. We first consider the termination guarantees.

Assume that there are up to t_s corruptions, and an honest party P has obtained an output v_{out} at time τ. Then, P has received v_{out} from $n - t_s$ invocations of Π_{rBC}: all honest parties obtain v_{out} in these $n - t_s$ invocations of Π_{rBC} by time $\tau + c'_{\mathrm{rBC}} \cdot \delta = \tau + c'_{\mathrm{BA}^\star} \cdot \delta$, due to Consistency and Conditional Termination of Π_{rBC}.

If up to t_s of the parties involved are corrupted, Π_{BA} achieves Termination and Agreement. This implies that, almost surely, all honest parties output v_{out} eventually. Hence, assuming that an honest party P never obtains an output in $\Pi_{\mathrm{BA}^\star}$, the claim above implies that no honest party obtains an output in $\Pi_{\mathrm{BA}^\star}$. Then, almost surely, all honest parties obtain v_{out} in Π_{BA} eventually. Then, at least the $n - t_s$ honest parties distribute v_{out} via Π_{rBC} eventually – these messages are eventually received, and hence all honest parties terminate in $\Pi_{\mathrm{BA}^\star}$. Moreover, the Termination property of Π_{BA} ensures that honest parties obtain v_{out} from Π_{BA} by time $f_{\mathrm{BA}}(\lambda) \cdot \Delta$ except with probability $\mathrm{negl}(\lambda)$. Then, if no honest party has obtained an output in $\Pi_{\mathrm{BA}^\star}$ by time $f_{\mathrm{BA}}(\lambda) \cdot \Delta$, all honest parties obtain v_{out} from Π_{BA} by time $f_{\mathrm{BA}}(\lambda) \cdot \Delta$ except with probability $\mathrm{negl}(\lambda)$, hence at least the $n - t_s$ honest parties distribute v_{out} via Π_{rBC} by time $f_{\mathrm{BA}}(\lambda) \cdot \Delta$ except with probability $\mathrm{negl}(\lambda)$. The Validity and Honest Termination properties of Π_{rBC} ensure that these messages are received within $c_{\mathrm{rBC}} \cdot \Delta$ additional time, hence all honest parties output v_{out} in $\Pi_{\mathrm{BA}^\star}$ by time $(f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}) \cdot \Delta$ except with negligible probability $\mathrm{negl}(\lambda)$, as claimed. Otherwise, if some honest party has obtained an output in $\Pi_{\mathrm{BA}^\star}$ by time $\tau < f_{\mathrm{BA}}(\lambda) \cdot \Delta$, the argument above shows that all honest parties obtain outputs in $\Pi_{\mathrm{BA}^\star}$ by time $\tau + c'_{\mathrm{rBC}} \cdot \Delta \leq (f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}) \cdot \Delta$. Consequently, $\Pi_{\mathrm{BA}^\star}$ achieves Termination for polynomial $f_{\mathrm{BA}^\star}(\lambda) = f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}$.

We next consider Responsive Termination: if there are at most t_r corruptions, Π_{BA} achieves Responsive Termination and Agreement. Except with negligible probability, the honest parties obtain v_{out} in Π_{BA} by time $f_{\mathrm{BA}}(\lambda) \cdot \delta$. If no honest party obtains an output in $\Pi_{\mathrm{BA}^\star}$ by time $f_{\mathrm{BA}}(\lambda) \cdot \delta$, all honest parties obtain v_{out} in Π_{BA} by time $f_{\mathrm{BA}}(\lambda) \cdot \delta$ except with probability $\mathrm{negl}(\lambda)$. Hence, at least $n - t_s$ honest parties distribute v_{out} via Π_{rBC} by time $f_{\mathrm{BA}}(\lambda) \cdot \delta$ except with probability $\mathrm{negl}(\lambda)$, and these messages are received within $c_{\mathrm{rBC}} \cdot \delta$ additional time by Honest Termination and Validity of Π_{rBC}. Therefore, all honest parties output v_{out} in $\Pi_{\mathrm{BA}^\star}$ by time $(f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}) \cdot \delta$ except with negligible probability $\mathrm{negl}(\lambda)$. Otherwise, if some honest party obtains an output in $\Pi_{\mathrm{BA}^\star}$ by time $\tau < f_{\mathrm{BA}}(\lambda) \cdot \delta$, the

argument above shows that all honest parties obtain outputs in $\Pi_{\mathrm{BA}^\star}$ by time $\tau + c'_{\mathrm{rBC}} \cdot \delta \leq (f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}) \cdot \delta$. Consequently, $\Pi_{\mathrm{BA}^\star}$ is t_r-responsive for polynomial $f_{\mathrm{BA}^\star}(\lambda) = f_{\mathrm{BA}}(\lambda) + c_{\mathrm{rBC}}$.

We now show that $\Pi_{\mathrm{BA}^\star}$ achieves Agreement even up to t_s corruptions. Assume that two honest parties P and P' obtain outputs $v_{\mathsf{out}} \neq v'_{\mathsf{out}}$ in $\Pi_{\mathrm{BA}^\star}$. Then, P has received v_{out} in $n - t_s$ of the Π_{rBC} invocations. As Π_{rBC} achieves Consistency, there are at most $t_s < n - t_s$ invocations in which P' has received v'_{out}, which contradicts that P' has output $v'_{\mathsf{out}} \neq v_{\mathsf{out}}$.

Finally, we show that $\Pi_{\mathrm{BA}^\star}$ achieves Strong Unanimity even up to t_s corruptions. Assume that all honest parties hold input v and that an honest party P has output $v_{\mathsf{out}} \neq v$. Then, P has received v_{out} in $n - t_s > t_s$ invocations of Π_{rBC}, hence from at least one honest party P'. Then, as Π_{rBC} achieves Validity, P' has obtained $v_{\mathsf{out}'} = v_{\mathsf{out}}$ in Π_{BA}. However, honest parties join Π_{BA} with input v: as Π_{BA} achieves Strong Unanimity, $v = v_{\mathsf{out}'}$. □

We also note that $\Pi_{\mathrm{BA}^\star}$, or a t_s-secure t_r-responsive BA protocol with Strong Unanimity in general, never allows an honest party to output a value proposed only by byzantine parties. Since a party proposes a value to an invocation precisely when it joins that invocation, some honest party must have joined the invocation with input b before the output occurred. We may therefore state the following remark.

Remark 2. *If an honest party obtains output b in a $\Pi_{\mathrm{BA}^\star}$ invocation at time τ, then some honest party has joined that invocation with input b by time τ.*

Protocol. We are now ready to present the unauthenticated ACS protocol. Let $c_{\mathsf{join}} := \max\{c_{\mathrm{rBC}}, c'_{\mathrm{rBC}}, c'_{\mathrm{BA}^\star}\}$. Each invocation of $\Pi_{\mathrm{BA}^\star}$ uses synchronous delay upper bound $\Delta' := (c_{\mathsf{join}} + 1) \cdot \Delta$.

Protocol Π_{ACS} [4, 34]

Code for party P with input v_{in}

1: Send v_{in} to every party via Π_{rBC}.
2: When receiving a value v from P' via Π_{rBC}:
3: Join the invocation of $\Pi_{\mathrm{BA}^\star}$ for P' with input 1.
4: When at least $n - t_r$ of the $\Pi_{\mathrm{BA}^\star}$ invocations returned 1 or $c_{\mathrm{rBC}} \cdot \Delta$ time has passed:
5: Join the remaining $\Pi_{\mathrm{BA}^\star}$ invocations with input 0.
6: When all $\Pi_{\mathrm{BA}^\star}$ invocations have terminated:
7: $\mathcal{P}$:= parties whose corresponding $\Pi_{\mathrm{BA}^\star}$ invocations have terminated with output 1.
8: When all invocations of Π_{rBC} having senders in $\mathcal{P}$ have terminated:
9: I := the set of pairs (v', P'), where $P' \in \mathcal{P}$ and v' is the value P' sent via Π_{rBC}.
10: Output I.

Next, we show that Π_{ACS} achieves each of the ACS properties of Definition 3. We analyze the Responsive Termination and Termination properties of Π_{ACS} in Lemma 18 and Lemma 19 respectively. Afterwards, Lemma 20 discusses Agreement, Lemma 21 considers Validity, and Lemma 22 shows that Π_{ACS} achieves Core Quality. Then, Theorem 9 follows directly from these results.

Before presenting these lemmas, we recall that $n = \mathsf{poly}(\lambda)$. Hence, if an event fails with probability $\mathrm{negl}(\lambda)$ in each invocation, then the probability that it fails in at least one of n invocations is at most $n \cdot \mathrm{negl}(\lambda) = \mathrm{negl}(\lambda)$ by a union bound.

Lemma 18. *If up to t_r of the parties involved are corrupted, Π_{ACS} achieves Responsive Termination for polynomial $f'_{\text{ACS}}(\lambda) = c_{\text{rBC}} + c'_{\text{BA}^\star} + 2 \cdot (c_{\mathsf{join}} + 1) \cdot f_{\text{BA}^\star}(\lambda) + c'_{\text{rBC}} = O(1) \cdot f_{\text{BA}^\star}(\lambda)$.*

Proof. Assume that the network delay is at most $\delta \leq \Delta$ and that up to t_r parties are corrupted. We first show that all honest parties terminate in each of the n invocations of $\Pi_{\text{BA}^\star}$ by time $T_{\mathsf{join}} := c_{\text{rBC}} \cdot \delta + c'_{\text{BA}^\star} \cdot \delta + 2 \cdot (c_{\mathsf{join}} + 1) \cdot f_{\text{BA}^\star}(\lambda) \cdot \delta$, except with negligible probability. To this end, we first bound the time by which all honest parties have joined every invocation and then show that, in each invocation, their join times lie within an interval of length at most $c_{\mathsf{join}} \cdot \delta$.

First, assume that some honest party has obtained output 1 in at least $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$ by time $c_{\text{rBC}} \cdot \delta$. By Lemma 17, $\Pi_{\text{BA}^\star}$ ensures that all honest parties obtain output 1 in these invocations by time $(c_{\text{rBC}} + c'_{\text{BA}^\star}) \cdot \delta \leq T_{\mathsf{join}}$. Hence, by time T_{join}, all honest parties have joined every remaining invocation.

Otherwise, no honest party has obtained output 1 in $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$ by time $c_{\text{rBC}} \cdot \delta$. If the timeout $c_{\text{rBC}} \cdot \Delta$ has occurred, then all honest parties have joined the n invocations of $\Pi_{\text{BA}^\star}$ by time $c_{\text{rBC}} \cdot \delta$ – hence, we may assume the timeout has not occurred. Since no honest party has obtained output 1 in $n - t_r$ invocations either, no honest party has joined any invocation with input 0 by time $c_{\text{rBC}} \cdot \delta$. Honest Termination of Π_{rBC} ensures that all honest parties have obtained outputs in the Π_{rBC} invocations having honest senders by that time. Hence, all honest parties have joined with input 1 the at least $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$ corresponding to honest senders. Strong Unanimity ensures that these invocations output 1. Since the honest parties join each such invocation within an interval of length at most $c_{\text{rBC}} \cdot \delta \leq c_{\mathsf{join}} \cdot \delta$, Remark 1, applied to $\Pi_{\text{BA}^\star}$, ensures that all honest parties obtain output 1 in these invocations by time $c_{\text{rBC}} \cdot \delta + (c_{\mathsf{join}} + 1) \cdot f_{\text{BA}^\star}(\lambda) \cdot \delta \leq T_{\mathsf{join}}$, except with negligible probability. It follows that, by time T_{join}, all honest parties have joined every remaining invocation with input 0, unless they joined it earlier with input 1.

In the two cases above, we have obtained that all honest parties have joined every invocation of $\Pi_{\text{BA}^\star}$ by time T_{join}, except with negligible probability $\text{negl}(\lambda)$. We next focus on bounding the difference between honest parties' joining times in the $\Pi_{\text{BA}^\star}$ invocations. Fix an arbitrary party P.

Let P' be the first honest party to join the invocation of $\Pi_{\text{BA}^\star}$ corresponding to P, and let τ be the time at which it does so. Note that $\tau \leq T_{\mathsf{join}}$, except with probability $\text{negl}(\lambda)$.

If P' joins with input 1, then it has obtained an output in the invocation of Π_{rBC} having P as sender by time τ. By Conditional Termination of Π_{rBC}, every honest party obtains an output in this Π_{rBC} invocation by time $\tau + c'_{\text{rBC}} \cdot \delta$. Every honest party that has not already joined the invocation of $\Pi_{\text{BA}^\star}$ corresponding to P therefore joins it by that time. Since P' is the first honest party to join this invocation, no honest party joins it before time τ. Hence, all honest parties join the invocation corresponding to P between times τ and $\tau + c'_{\text{rBC}} \cdot \delta$.

Next, assume that P' joins with input 0 because it has obtained output 1 in $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$. Since P' joins the invocation corresponding to P at time τ, the condition that it has obtained 1 in $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$ becomes true at exactly time τ; otherwise, P' would have joined the invocation corresponding to P earlier. By Lemma 17, all honest parties obtain output 1 in these $n - t_r$ invocations by time $\tau + c'_{\text{BA}^\star} \cdot \delta$. Hence, all honest parties have joined the invocation corresponding to P by that time. Since P' is the first honest party to join this invocation, no honest party joins it before time τ. Therefore, all

honest parties join the invocation corresponding to P between times τ and $\tau + c'_{\mathrm{BA}^\star} \cdot \delta$.

Finally, assume that P' joins with input 0 because the timeout $c_{\mathrm{rBC}} \cdot \Delta$ has occurred. Then $\tau = c_{\mathrm{rBC}} \cdot \Delta$. Since P' is the first honest party to join the invocation corresponding to P, no honest party has joined this invocation before time τ. At time τ, every honest party joins every invocation that it has not yet joined. Therefore, all honest parties join the invocation corresponding to P at time τ.

Consequently, all honest parties join the invocation corresponding to P between times τ and $\tau + c_{\mathsf{join}} \cdot \delta$. Let $\tau_{\max}$ be the time at which the last honest party joins this invocation. Since all honest parties have joined every invocation by time T_{join}, we have $\tau_{\max} \leq T_{\mathsf{join}}$. By Remark 1, applied to $\Pi_{\mathrm{BA}^\star}$, all honest parties terminate in this invocation by time $\tau_{\max} + (c_{\mathsf{join}} + 1) \cdot f_{\mathrm{BA}^\star}(\lambda) \cdot \delta \leq T_{\mathsf{join}} + (c_{\mathsf{join}} + 1) \cdot f_{\mathrm{BA}^\star}(\lambda) \cdot \delta$, except with negligible probability.

It remains to wait for the Π_{rBC} invocations corresponding to outputs 1. Fix a party P whose corresponding invocation of $\Pi_{\mathrm{BA}^\star}$ outputs 1. By Remark 2, some honest party joined that invocation with input 1 by the time an honest party obtained output 1. Hence, that party had obtained an output in the invocation of Π_{rBC} having P as sender by that time. Conditional Termination of Π_{rBC} ensures that all honest parties obtain this value within $c'_{\mathrm{rBC}} \cdot \delta$ additional time. Consequently, every honest party outputs in Π_{ACS} by time $f'_{\mathrm{ACS}}(\lambda) \cdot \delta$, except with negligible probability. □

Lemma 19. *If up to t_s of the parties involved are corrupted, Π_{ACS} achieves Termination for polynomial $f_{\mathrm{ACS}}(\lambda) = c_{\mathrm{rBC}} + (c_{\mathsf{join}} + 1) \cdot f_{\mathrm{BA}^\star}(\lambda) + c'_{\mathrm{rBC}} = O(1) \cdot f_{\mathrm{BA}^\star}(\lambda)$.*

Proof. Every honest party joins every invocation of $\Pi_{\mathrm{BA}^\star}$ by time $c_{\mathrm{rBC}} \cdot \Delta$. Since the time difference between the first and last honest parties joining is at most $c_{\mathrm{rBC}} \cdot \Delta \leq c_{\mathsf{join}} \cdot \Delta$, Remark 1, applied to $\Pi_{\mathrm{BA}^\star}$, ensures that all honest parties terminate in every invocation by time $c_{\mathrm{rBC}} \cdot \Delta + (c_{\mathsf{join}} + 1) \cdot f_{\mathrm{BA}^\star}(\lambda) \cdot \Delta$, except with negligible probability $\mathrm{negl}(\lambda)$.

Afterwards, each honest party waits for every invocation of Π_{rBC} corresponding to an output 1. For each such invocation, Remark 2 ensures that some honest party joined the corresponding invocation of $\Pi_{\mathrm{BA}^\star}$ with input 1, and hence obtained an output in that invocation of Π_{rBC}. Conditional Termination of Π_{rBC} ensures that all honest parties obtain the same output within $c'_{\mathrm{rBC}} \cdot \Delta$ additional time. Hence, all honest parties terminate in Π_{ACS} by time $f_{\mathrm{ACS}}(\lambda) \cdot \Delta$, except with negligible probability $\mathrm{negl}(\lambda)$.

Almost-sure termination follows from the same argument. Every honest party joins all n invocations of $\Pi_{\mathrm{BA}^\star}$ by time $c_{\mathrm{rBC}} \cdot \Delta$, and each invocation terminates almost surely. Since there are finitely many invocations, they all terminate almost surely. For every invocation that outputs 1, Remark 2 and Conditional Termination of Π_{rBC} ensure that the corresponding Π_{rBC} invocation terminates for every honest party. Therefore, Π_{ACS} terminates almost surely. □

Lemma 20. *If up to t_s of the parties involved are corrupted, Π_{ACS} achieves Agreement.*

Proof. Assume that two honest parties output I and I' in Π_{ACS}. Then, as $\Pi_{\mathrm{BA}^\star}$ achieves Agreement, they have obtained the same outputs in each of the n invocations of $\Pi_{\mathrm{BA}^\star}$, hence $\textsc{parties}(I) = \textsc{parties}(I')$. Moreover, the value v assigned to each party $P \in \textsc{parties}(I)$ is that received in the Π_{rBC} invocation having P as sender. Π_{rBC} achieves Consistency, which implies that the two honest parties have assigned the same value to P, and hence $I = I'$. □

Lemma 21. *If up to t_s of the parties involved are corrupted, Π_{ACS} achieves Validity.*

Proof. Assume that an honest party P' obtains I in Π_{ACS}, and that $(v, P) \in I$ for an honest party P holding input v_{in}. Then, P' has obtained output v in the Π_{RBC} invocation having P as a sender. By the Validity property of Π_{RBC}, $v = v_{\mathsf{in}}$. □

Lemma 22. *If up to t_s of the parties involved are corrupted, Π_{ACS} achieves Core Quality.*

Proof. Assume an honest party P outputs I in Π_{ACS}. If $\left|I\right| \geq n - t_r$, then the property is achieved trivially. The remainder of the proof is therefore concerned with the case where $\left|I\right| < n - t_r$.

As P has obtained I with $\left|I\right| < n - t_r$, P has obtained output 1 in fewer than $n - t_r$ of the n invocations of $\Pi_{\text{BA}^\star}$. By Remark 2, for each of the at least $t_r + 1$ invocations of $\Pi_{\text{BA}^\star}$ where P has obtained output 0, there is an honest party that joined with input 0.

We assume an honest party P' has joined one of the $\Pi_{\text{BA}^\star}$ invocations with input 0, and we show that P' has joined the $\Pi_{\text{BA}^\star}$ invocations corresponding to honest parties with input 1.

Before joining any $\Pi_{\text{BA}^\star}$ invocation with input 0, P' has waited until one of these conditions became true: (i) it obtained output 1 in $n - t_r$ of the $\Pi_{\text{BA}^\star}$ invocations, or (ii) $c_{\text{RBC}} \cdot \Delta$ time has passed. Note that (i) contradicts that $\Pi_{\text{BA}^\star}$ achieves Agreement: if P' has obtained output 1 in at least $n - t_r$ invocations of $\Pi_{\text{BA}^\star}$, then Agreement implies that P should have also obtained output 1 in these at least $n - t_r$ invocations, which contradicts that $\left|I\right| < n - t_r$. This implies that only (ii) holds: P' has joined $\Pi_{\text{BA}^\star}$ invocations with input 0 once $c_{\text{RBC}} \cdot \Delta$ time has passed. As the network delay is up to Δ, Π_{RBC}'s Honest Termination property ensures that all Π_{RBC} invocations with honest senders have completed by time $c_{\text{RBC}} \cdot \Delta$, and therefore P' joins the $\Pi_{\text{BA}^\star}$ invocations corresponding to honest parties with input 1. Note that this holds for any honest party P' that has joined any of the $\Pi_{\text{BA}^\star}$ invocations with input 0.

Hence, we have obtained that no honest party joins the $\Pi_{\text{BA}^\star}$ invocations corresponding to honest parties with input 0. Since party P has obtained output I in Π_{ACS}, it has obtained an output in each of these $\Pi_{\text{BA}^\star}$ invocations. Then, by Remark 2, P has obtained output 1 in every $\Pi_{\text{BA}^\star}$ invocation corresponding to an honest party. Consequently, $\textsc{parties}(I)$ contains all honest parties. □

Deterministic ACS. We first instantiate the underlying protocol Π_{BA} in Theorem 9 with the deterministic protocol of [2, Corollary 5]: this achieves t_s-secure and t_r-responsive BA with Strong Unanimity on $\{0, 1\}$ whenever $n > 3 \cdot t_s$. The protocol of [2, Corollary 5] has Termination and Responsive Termination polynomial $f_{\text{BA}}(\lambda) = O(t_s)$. Hence, Lemma 17 gives $f_{\text{BA}^\star}(\lambda) = O(t_s)$ and $c'_{\text{BA}^\star} = c'_{\text{RBC}} = O(1)$. Consequently, $c_{\mathsf{join}} = O(1)$, and Theorem 9 implies that Π_{ACS} is a deterministic unauthenticated t_s-secure t_r-responsive ACS protocol whenever $n > 3 \cdot t_s$. The polynomials witnessing Termination and Responsive Termination are both $O(t_s)$, as stated in Theorem 5.

Randomized ACS. Alternatively, we instantiate the BA protocol Π_{BA} in Theorem 9 with the (terminating) protocol of [35]. The protocol is designed *in the asynchronous model*: it only assumes that every message gets delivered *eventually*. This is a t_s-secure BA protocol with Strong Unanimity on $\{0, 1\}$ whenever $n > 3 \cdot t_s$.

We first explain why this asynchronous protocol satisfies the termination guarantees required by Definition 1 when run in our synchronous network. Π_{BA} terminates almost surely:

as time goes to infinity, the probability that an honest party has not yet terminated is 0. Moreover, its proof gives a geometric tail bound on the number of asynchronous rounds. The protocol relies on a weak common coin with constant parameter $d \geq 2$, meaning that in each asynchronous round all honest parties receive the same prescribed coin value with probability at least $1/d$. This implies that the probability that the protocol has not terminated after r rounds is at most $(1 - 1/d)^{\Omega(r)}$. Then, after $O(\lambda)$ rounds, the probability that some honest party has not yet terminated is $\mathrm{negl}(\lambda)$.

Each asynchronous round of the protocol consists of a constant number of steps. Hence, if the actual message delay is at most δ, then each asynchronous round completes within $c_{\mathrm{BA}} \cdot \delta$ real time for a constant c_{BA}. Therefore, every honest party terminates within $f_{\mathrm{BA}}(\lambda) \cdot \delta$ real time except with probability $\mathrm{negl}(\lambda)$, where $f_{\mathrm{BA}}(\lambda) = c_{\mathrm{BA}} \cdot \lambda$. Since $\delta \leq \Delta$, the same execution also terminates within $f_{\mathrm{BA}}(\lambda) \cdot \Delta$ except with negligible probability. Together with almost-sure eventual termination, this means that Π_{BA} satisfies both Termination and Responsive Termination according to our definitions.

Hence, using this protocol in Theorem 9, Π_{ACS} becomes a randomized t_s-secure t_r-responsive ACS protocol whenever $n > 3 \cdot t_s$. The polynomials witnessing Termination and Responsive Termination are both $O(\lambda)$, as stated in Theorem 5.

C Sufficient Conditions: Missing Proof

We present the proof of Theorem 7, describing the guarantees of our protocol Π_{BA} solving the given validity property VAL.

Theorem 7. *Consider a validity property* VAL *satisfying the responsive similarity condition. Then, there are deterministic and randomized t_s-secure t_r-responsive* BA *protocols solving* VAL *if $n > t_s + 2 \cdot t_r$ in authenticated settings, and if $n > 3 \cdot t_s$ in unauthenticated settings.*

For the deterministic protocol, the polynomials witnessing Termination and Responsive Termination in authenticated settings are $f(\lambda) = n + O(t_s)$ and $f'(\lambda) = O(t_s)$, respectively; in unauthenticated settings, they are both $f(\lambda) = O(t_s)$.

For the randomized protocol, the polynomials witnessing Termination and Responsive Termination in authenticated settings are $f(\lambda) = n + O(\lambda)$ and $f'(\lambda) = O(\lambda)$, respectively; in unauthenticated settings, they are both $f(\lambda) = O(\lambda)$.

Proof. We instantiate Π_{ACS} with the protocol from Theorem 6 in authenticated settings, and with the protocol from Theorem 5 in unauthenticated settings. Under the corresponding threshold assumptions in the theorem's statement, Π_{ACS} satisfies the guarantees of Definition 3.

Let $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ be the Turing-computable function guaranteed by the responsive similarity condition for VAL. We note that, if an honest party obtains I from Π_{ACS}, $\left|I\right| \geq n - t_s$ due to the Core Quality property, hence $I \in \mathcal{I}$, which establishes that $\sigma(I)$ is well defined.

We first prove Termination. Fix either the deterministic or the randomized instantiation of Π_{ACS}, and let $f_{\mathrm{ACS}}(\lambda)$ be the polynomial witnessing its Termination guarantee. In any execution with at most t_s byzantine parties, Π_{ACS} ensures that honest parties terminate eventually almost surely. Then, parties terminate almost surely in Π_{BA} as well. Moreover, except with probability $\mathrm{negl}(\lambda)$, Π_{ACS} ensures that every honest party obtains an output set I from Π_{ACS} by time $f_{\mathrm{ACS}}(\lambda) \cdot \Delta$. Upon obtaining I, the party outputs $\sigma(I)$ and terminates. Hence,

Π_{BA} achieves Termination with the same polynomial $f_{\text{ACS}}(\lambda)$. In authenticated settings, this polynomial is $n + O(t_s)$ for the deterministic protocol and $n + O(\lambda)$ for the randomized protocol; in unauthenticated settings, it is $O(t_s)$ for the deterministic protocol and $O(\lambda)$ for the randomized protocol.

Responsive Termination follows similarly. Let $f'_{\text{ACS}}(\lambda)$ be the polynomial witnessing Responsive Termination of the chosen instantiation of Π_{ACS}. In any execution with at most t_r byzantine parties and actual network delay at most δ, except with probability $\text{negl}(\lambda)$, every honest party obtains an output set I from Π_{ACS} by time $f'_{\text{ACS}}(\lambda) \cdot \delta$, after which it outputs $\sigma(I)$ and terminates. Hence, Π_{BA} achieves Responsive Termination with the same polynomial $f'_{\text{ACS}}(\lambda)$. In both authenticated and unauthenticated settings, this polynomial is $O(t_s)$ for the deterministic protocol and $O(\lambda)$ for the randomized protocol.

Agreement follows directly from the Agreement property of Π_{ACS}: all honest parties obtain the same output set I from Π_{ACS}. Since σ is a deterministic function, all honest parties output the same $\sigma(I)$.

It remains to discuss Validity. We have established that the honest parties obtain the same output set $I \in \mathcal{I}$. Let $I^\star$ be the input configuration defined by the parties remaining honest throughout the whole execution and their inputs in this execution. We split the analysis into two cases based on the size of I. We first assume that $\left|I\right| \geq n - t_r$, and we show that $I^\star \in \text{SIMILAR}_R(I) = \{J \in \mathcal{I} : \left|J\right| \geq n - t_s \text{ and } J \text{ is compatible with } I\}$. $\left|I^\star\right| \geq n - t_s$ as there are at most t_s corruptions involved. Next, if $(v, P) \in I$ and $(v^\star, P) \in I^\star$, then P is an honest party with input $v^\star$, and the Validity property of Π_{ACS} ensures that $v^\star = v$, hence $I^\star$ is compatible with I. Consequently, $I^\star \in \text{SIMILAR}_R(I)$. Then, in this case, the common output is computed as $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) \subseteq \text{VAL}(I^\star)$. Next, we assume that $\left|I\right| < n - t_r$. By the Core Quality property of Π_{ACS}, I contains all honest parties, hence $\left|I\right| \geq n - t_s$. Moreover, by the Validity property of Π_{ACS}, each honest party appears with its actual input in I. We therefore obtain that $I^\star \subseteq I$, and hence $I^\star \in \text{SIMILAR}_S(I) := \{J \subseteq I : \left|J\right| \geq n - t_s\}$. Here, the common output is computed as $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J) \subseteq \text{VAL}(I^\star)$. In both cases, the value output by the honest parties belongs to $\text{VAL}(I^\star)$. Therefore, Π_{BA} satisfies Validity. Hence, Π_{BA} is a t_s-secure t_r-responsive BA protocol solving VAL. □

D Concrete Instantiations: Missing Proofs

We present the formal proofs for our corollaries in Section 6.

Corollary 1. *Weak Validity can be solved responsively if and only if $n > 2 \cdot t_r + t_s$ in an authenticated setting, and if and only if $n > 3 \cdot t_s$ in an unauthenticated setting.*

Proof. Weak Validity is the validity property $\text{VAL} : \mathcal{I} \to 2^{V_{\mathsf{out}}}$ defined by $\text{VAL}(I) = \{v\}$ if $\left|I\right| = n$ and every pair in I assigns value v, and $\text{VAL}(I) = V_{\mathsf{out}}$ otherwise.

We first establish that Weak Validity satisfies the responsive similarity condition whenever $n > 2 \cdot t_r + t_s$. Define $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ as follows: if there is a value v such that every pair in I assigns value v, then set $\sigma(I) = v$. Otherwise, set $\sigma(I)$ to an arbitrary fixed value in V_{out}. Note that σ is Turing-computable. It remains to show that $\sigma(I)$ is in $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$ if $\left|I\right| \geq n - t_r$, and in $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$ otherwise.

First, assume $\left|I\right| \geq n - t_r$, and consider any $J \in \text{SIMILAR}_R(I)$. If $\text{VAL}(J) = V_{\mathsf{out}}$, then $\sigma(I) \in \text{VAL}(J)$ trivially. Otherwise, by the definition of Weak Validity, $\left|J\right| = n$ and all pairs

in J assign the same value v. Since $\left|J\right| = n$, every party appearing in I also appears in J. As J is compatible with I, each party appearing in I must be assigned the same value in I and in J. Hence, every pair in I assigns value v. By the definition of σ, $\sigma(I) = v$, and therefore $\sigma(I) \in \text{VAL}(J) = \{v\}$. Then, since this holds for every $J \in \text{SIMILAR}_R(I)$, we obtain $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.

Next, assume $\left|I\right| < n - t_r$, and consider any $J \in \text{SIMILAR}_S(I)$. If $\text{VAL}(J) = V_{\mathsf{out}}$, then $\sigma(I) \in \text{VAL}(J)$ trivially. Otherwise, by the definition of Weak Validity, $\left|J\right| = n$ and all pairs in J assign the same v. Since $J \subseteq I$ and $\left|J\right| = n$, we must have $I = J$. Hence, every pair in I assigns value v. By the definition of σ, $\sigma(I) = v$, and therefore $\sigma(I) \in \text{VAL}(J) = \{v\}$. Since this holds for every $J \in \text{SIMILAR}_S(I)$, we obtain $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.

Consequently, Weak Validity satisfies the responsive similarity condition whenever $n > 2 \cdot t_r + t_s$. Then, by Theorem 7, Weak Validity can be solved responsively whenever $n > 2 \cdot t_r + t_s$ in authenticated settings, and whenever $n > 3 \cdot t_s$.

Finally, Weak Validity is non-trivial: two maximal configurations assigning distinct values to all parties have disjoint sets of valid outputs. Then, Theorem 2 implies that $n > 2 \cdot t_r + t_s$ is necessary even in authenticated settings, while Theorem 3 implies that $n > 3 \cdot t_s$ is necessary in unauthenticated settings, which concludes the proof. □

Corollary 2. *Strong Unanimity can be solved responsively if and only if $n > 2 \cdot t_s + t_r$ in an authenticated setting, and if and only if $n > 3 \cdot t_s$ in an unauthenticated setting.*

Proof. Strong Unanimity is the validity property $\text{VAL} : \mathcal{I} \to 2^{V_{\mathsf{out}}}$ defined by $\text{VAL}(I) = \{v\}$ if I assigns value v to every party in $\text{PARTIES}(I)$, and $\text{VAL}(I) = V_{\mathsf{out}}$ otherwise.

We first establish that Strong Unanimity satisfies the responsive similarity condition if and only if $n > 2 \cdot t_s + t_r$. Assuming that $n > 2 \cdot t_s + t_r$, we define a Turing-computable function $\sigma : \mathcal{I} \to V_{\mathsf{out}}$ as follows:

- if $\left|I\right| \geq n - t_r$ and there is a value v such that at least $\left|I\right| - t_s$ pairs in I assign value v, then set $\sigma(I) = v$. Note that there is at most one such value v with this property: if at least $\left|I\right| - t_s$ pairs assign value v, and at least $\left|I\right| - t_s$ assign value $v' \neq v$, then we have $\left|I\right| \geq 2 \cdot (\left|I\right| - t_s)$, hence $\left|I\right| \leq 2 \cdot t_s$. As $\left|I\right| \geq n - t_r$, this implies that $n - t_r \leq 2 \cdot t_s$, contradicting $n > 2 \cdot t_s + t_r$.

- Otherwise, if $\left|I\right| < n - t_r$ and there is a value v such that at least $n - t_s$ pairs in I assign value v, then set $\sigma(I) = v$. Similarly to the previous case, there is at most one value v with this property: assuming that at least $n - t_s$ pairs in I are assigned value v, and at least $n - t_s$ pairs in I are assigned value $v' \neq v$, we get that $n - t_r > \left|I\right| \geq 2 \cdot (n - t_s)$. However, this implies that $2 \cdot t_s > n + t_r \geq n$, contradicting that $n > 2 \cdot t_s + t_r$.

- Otherwise, set $\sigma(I) = v_0$, where v_0 is an arbitrary fixed value in V_{out}.

Next, we need to show that $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$ if $\left|I\right| \geq n - t_r$ and $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$ otherwise.

Assume that $\left|I\right| \geq n - t_r$, and consider any input configuration $J \in \text{SIMILAR}_R(I)$. If $\text{VAL}(J) = V_{\mathsf{out}}$, then $\sigma(I) \in \text{VAL}(J)$ trivially. Otherwise, by the definition of Strong Unanimity, all pairs in J assign the same value v. Since J is compatible with I, every pair in I assigning a value different from v must be absent from J. Since $\left|J\right| \geq n - t_s$, J omits at most t_s parties, hence at most t_s pairs from I. It follows that at most t_s pairs in I assign

a value different from v. Then, $\sigma(I) = v$, and therefore $\sigma(I) \in \text{VAL}(J)$. Since this holds for every $J \in \text{SIMILAR}_R(I)$, we obtain $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$.

Then, assume $\left|I\right| < n - t_r$, and consider any $J \in \text{SIMILAR}_S(I)$. If $\text{VAL}(J) = V_{\mathsf{out}}$, $\sigma(I) \in \text{VAL}(J)$ trivially. Otherwise, by the definition of Strong Unanimity, every pair in J assigns some common value v. Since $J \subseteq I$ and $\left|J\right| \geq n - t_s$, at least $n - t_s$ pairs in I assign value v. Then, $\sigma(I) = v$, and therefore $\sigma(I) \in \text{VAL}(J)$. Since this holds for every $J \in \text{SIMILAR}_S(I)$, we obtain $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$.

Consequently, Strong Unanimity satisfies the responsive similarity condition whenever $n > 2 \cdot t_s + t_r$. We now focus on the other direction: if $n \leq 2 \cdot t_s + t_r$, the responsive similarity condition is not satisfied: we consider two input values $v \neq v'$ and define an input configuration $I \in \mathcal{I}$ of size $n - t_r$ such that $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) = \varnothing$. Since $n - t_r \leq 2 \cdot t_s$, we may partition $\text{PARTIES}(I)$ into two (non-empty) sets A and B of size at most t_s each. I assigns value v to the parties in A and value v' to the parties in B. We then define two input configurations J_v and $J_{v'}$ in $\text{SIMILAR}_R(I)$. J_v assigns value v to all parties in $\mathcal{P} \setminus B$, while $J_{v'}$ assigns value v' to all parties in $\mathcal{P} \setminus A$. Note that, as $\left|A\right|, \left|B\right| \leq t_s$, $\left|J_v\right|, \left|J_{v'}\right| \geq n - t_s$. The configuration J_v is compatible with I as it contains the parties of A with value v and omits the parties of B. Similarly, $J_{v'}$ is compatible with I as it contains the parties of B with value v' and omits the parties of A. Hence, $J_v, J_{v'} \in \text{SIMILAR}_R(I)$. By Strong Unanimity, $\text{VAL}(J_v) = \{v\}$ and $\text{VAL}(J_{v'}) = \{v'\}$. Then, $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) \subseteq \text{VAL}(J_v) \cap \text{VAL}(J_{v'}) = \varnothing$, and no function σ can enable Strong Unanimity to satisfy the responsive similarity condition.

Therefore, Strong Unanimity satisfies the responsive similarity condition if and only if $n > 2 \cdot t_s + t_r$. Then, Theorem 7 establishes that there is a t_s-secure t_r-responsive BA protocol solving Strong Unanimity whenever $n > \max(2 \cdot t_s + t_r, 2 \cdot t_r + t_s) = 2 \cdot t_s + t_r$ in authenticated settings, and whenever $n > \max(2 \cdot t_s + t_r, 3 \cdot t_s) = 3 \cdot t_s$ in unauthenticated settings. Afterwards, Theorem 4 ensures that there is no t_s-secure t_r-responsive BA protocol solving Strong Unanimity when $n \leq 2 \cdot t_s + t_r$. Moreover, as Strong Unanimity is non-trivial (since all-v and all-v' configurations, with $v \neq v'$, lead to distinct valid outputs), Theorem 3 ensures that $n > 3 \cdot t_s$ is required in unauthenticated settings, completing the proof. □

Corollary 3. *Convex Validity on a convex space with Helly number $\omega = 2$ can be solved responsively if and only if $n > 2 \cdot t_s + t_r$ in the authenticated setting, and if and only if $n > 3 \cdot t_s$ in the unauthenticated setting.*

Proof. Convex Validity is the validity property $\text{VAL} : \mathcal{I} \to 2^{V_{\mathsf{out}}}$ defined by $\text{VAL}(I) = \langle I \rangle$, where $\langle I \rangle$ denotes the convex hull of the values in I.

We first show that Convex Validity satisfies the responsive similarity condition if $n > 2 \cdot t_s + t_r$. To define the function σ, we establish that the relevant intersections of valid output sets is non-empty: $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$ for every input configuration I of size at least $n - t_r$, and $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$ for input configurations I of size smaller than $n - t_r$. Consider an input configuration I.

First, assume $\left|I\right| \geq n - t_r$. Recall that we are working with a space with Helly number 2. For every $J \in \text{SIMILAR}_R(I)$, let $A_J := I \cap J$. Since $A_J \subseteq I$, the family $\mathcal{F}_I := \{\langle A_J \rangle : J \in \text{SIMILAR}_R(I)\}$ contains only finitely many distinct sets. Then, to show that $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) = \bigcap_{J \in \text{SIMILAR}_R(I)} \langle J \rangle \neq \varnothing$, it suffices to show that the finite family $\mathcal{F}_I$ has non-empty intersection, since $A_J \subseteq J$ implies $\langle A_J \rangle \subseteq \langle J \rangle$ for every $J \in \text{SIMILAR}_R(I)$. As the Helly number is 2, it is enough to show that any two sets in $\mathcal{F}_I$ intersect. Fix

$J, J' \in \text{SIMILAR}_R(I)$. We show that $\langle A_J \rangle \cap \langle A_{J'} \rangle \neq \varnothing$. Since $\left|J\right| \geq n - t_s$, the configuration J omits at most t_s parties, and hence omits at most t_s parties from $\text{PARTIES}(I)$. Therefore $\left|\text{PARTIES}(A_J)\right| = \left|\text{PARTIES}(J) \cap \text{PARTIES}(I)\right| \geq \left|I\right| - t_s$. Similarly, $\left|\text{PARTIES}(A_{J'})\right| \geq \left|I\right| - t_s$. If $\text{PARTIES}(A_J) \cap \text{PARTIES}(A_{J'}) = \varnothing$, then $\left|I\right| \geq 2 \cdot (\left|I\right| - t_s)$, and hence $2 \cdot t_s \geq \left|I\right| \geq n - t_r$, contradicting $n > 2 \cdot t_s + t_r$. Therefore $\text{PARTIES}(A_J) \cap \text{PARTIES}(A_{J'}) \neq \varnothing$.Let P be a party in this intersection. Since both J and J' are compatible with I, the value assigned to P in J and in J' is the same as the value assigned to P in I. Hence $A_J \cap A_{J'} \neq \varnothing$, which implies $\langle A_J \rangle \cap \langle A_{J'} \rangle \neq \varnothing$. It follows that every two sets in the finite family $\mathcal{F}_I$ have non-empty intersection. Since the Helly number is 2, we obtain $\bigcap_{J \in \text{SIMILAR}_R(I)} \langle A_J \rangle \neq \varnothing$. Finally, for every $J \in \text{SIMILAR}_R(I)$, we have $A_J \subseteq J$, and therefore $\langle A_J \rangle \subseteq \langle J \rangle$. Hence $\bigcap_{J \in \text{SIMILAR}_R(I)} \langle J \rangle \neq \varnothing$.

Next, assume $\left|I\right| < n - t_r$: we show that $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J) = \bigcap_{J \in \text{SIMILAR}_S(I)} \langle J \rangle \neq \varnothing$. $\text{SIMILAR}_S(I)$ is finite as it only contains subsets of I, hence it suffices to show that, for any $J, J' \in \text{SIMILAR}_S(I)$, $\langle J \rangle \cap \langle J' \rangle \neq \varnothing$. By the definition of $\text{SIMILAR}_S(I)$, $J, J' \subseteq I$ and $\left|J\right|, \left|J'\right| \geq n - t_s$. Assuming that J and J' are disjoint subsets of I, as each has size at least $n - t_s$, we get that $n - t_r > \left|I\right| \geq 2 \cdot (n - t_s)$, which implies that $2 \cdot t_s > n + t_r \geq n$ – this contradicts that $n > 2 \cdot t_s + t_r$. Therefore, $J \cap J'$ contains at least one value-sender pair. Hence $\langle J \rangle \cap \langle J' \rangle \neq \varnothing$. Again, since the Helly number is $\omega = 2$, the family $\{\langle J \rangle : J \in \text{SIMILAR}_S(I)\}$ has non-empty total intersection: $\bigcap_{J \in \text{SIMILAR}_S(I)} \langle J \rangle \neq \varnothing$.

Therefore, for every $I \in \mathcal{I}$, the intersection required by the responsive similarity condition is non-empty. We define $\sigma(I)$ as a Turing-computable choice of a point in the corresponding intersection (such function exists due to our assumptions on the convex space). Then, if $\left|I\right| \geq n - t_r$, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$, and otherwise, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. Consequently, Convex Validity satisfies the responsive similarity condition if $n > 2 \cdot t_s + t_r$. Theorem 7 ensures that Convex Validity is solvable in authenticated settings if $n > \max(2 \cdot t_s + t_r, 2 \cdot t_r + t_s) = 2 \cdot t_s + t_r$, and in unauthenticated settings if $n > \max(2 \cdot t_s + t_r, 3 \cdot t_s) = 3 \cdot t_s$.

Finally, we prove necessity, with the help of Corollary 2. We assume a t_s-secure t_r-responsive BA protocol Π solving Convex Validity for a space with Helly number $\omega = 2$, and we construct a t_s-secure t_r-responsive BA protocol Π' solving Strong Unanimity on $\{0, 1\}$.

Since the convex input space considered by Π has Helly number $\omega = 2$, it contains two non-empty convex sets C_0 and C_1 such that $C_0 \cap C_1 = \varnothing$. We may then consider $x_0 \in C_0$ and $x_1 \in C_1$, and construct Π' as follows: if a party has input 0, it joins Π with input x_0; otherwise it joins with input x_1. If the output agreed upon in Π is in C_0, the parties output 0 in Π', and otherwise they output 1 in Π'. The Agreement, Termination and Responsive Termination guarantees of Π' follow directly from those of Π. It remains to discuss Strong Unanimity. The case where honest parties do not join Π' with the same input bit is trivial, as Strong Unanimity enables any output in $\{0, 1\}$. If all honest parties have input b, they all join Π with input x_b, and hence they obtain an output in $\langle \{x_b\} \rangle$ in Π due to the Convex Validity property. Note that $\langle \{x_b\} \rangle \subseteq C_b$. Then, if $b = 0$, the honest parties output 0. If $b = 1$, as $C_0 \cap C_1 = \varnothing$, the honest parties output 1. Consequently, Strong Unanimity holds; and we have obtained that Convex Validity is no weaker than Strong Unanimity.

Then, as Corollary 2 states that any t_s-secure t_r-responsive BA protocol solving Strong Unanimity (even on $\{0, 1\}$ requires $n > 2 \cdot t_s + t_r$ in authenticated settings, and $n > 3 \cdot t_s$ in unauthenticated settings, these requirements apply to Convex Validity as well. □

Corollary 4. *Convex Validity on a convex space with Helly number $\omega > 2$ can be solved responsively if and only if $n > \omega \cdot t_s + t_r$, both in authenticated and unauthenticated settings.*

Proof. Convex Validity is the validity property $\text{VAL} : \mathcal{I} \to 2^{V_{\text{out}}}$ defined by $\text{VAL}(I) = \langle I \rangle$, where $\langle I \rangle$ denotes the convex hull of the values in I.

We first show that Convex Validity satisfies the responsive similarity condition if and only if $n > \omega \cdot t_s + t_r$. Similarly to the proof of Corollary 3, to define the function σ when $n > \omega \cdot t_s + t_r$, we show that the relevant intersections are non-empty: $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$ if $\big|I\big| \geq n - t_r$, and $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$ if $\big|I\big| < n - t_r$.

First, assume that $\big|I\big| \geq n - t_r$. In this case, we show that $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) = \bigcap_{J \in \text{SIMILAR}_R(I)} \langle J \rangle \neq \varnothing$. For every $J \in \text{SIMILAR}_R(I)$, let $A_J := I \cap J$. Since $A_J \subseteq I$, the family $\mathcal{F}_I := \{\langle A_J \rangle : J \in \text{SIMILAR}_R(I)\}$ contains only finitely many distinct sets. Hence, as we are working with a space with Helly number ω, it is enough to show that any ω sets in $\mathcal{F}_I$ have non-empty intersection. Fix $J_1, \ldots, J_\omega \in \text{SIMILAR}_R(I)$, and let $A_{J_k} := I \cap J_k$ for every $k \in [\omega]$. Since $\big|J_k\big| \geq n - t_s$ for every $k \in [\omega]$, each configuration J_k omits at most t_s parties, and hence omits at most t_s parties from $\text{PARTIES}(I)$. Therefore, the configurations $J_1, \ldots, J_\omega$ together omit at most $\omega \cdot t_s$ parties from $\text{PARTIES}(I)$, and so $\big|\bigcap_{k=1}^{\omega} \text{PARTIES}(A_{J_k})\big| = \big|\text{PARTIES}(I) \cap \big(\bigcap_{k=1}^{\omega} \text{PARTIES}(J_k)\big)\big| \geq \big|\text{PARTIES}(I)\big| - \omega \cdot t_s$. Since $\big|I\big| \geq n - t_r > \omega \cdot t_s$, there is a party P in this intersection. Then, as every J_k is compatible with I, the value assigned to P is the same in $I, J_1, \ldots, J_\omega$. Hence $\bigcap_{k=1}^{\omega} A_{J_k} \neq \varnothing$, and therefore $\bigcap_{k=1}^{\omega} \langle A_{J_k} \rangle \neq \varnothing$. It follows that every ω sets in the finite family $\mathcal{F}_I$ have non-empty intersection. Then, as the Helly number is ω, we obtain $\bigcap_{J \in \text{SIMILAR}_R(I)} \langle A_J \rangle \neq \varnothing$. Finally, for every $J \in \text{SIMILAR}_R(I)$, we have $A_J \subseteq J$, and therefore $\langle A_J \rangle \subseteq \langle J \rangle$. Hence $\bigcap_{J \in \text{SIMILAR}_R(I)} \langle J \rangle \neq \varnothing$.

Next, assume $\big|I\big| < n - t_r$: we show that $\bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J) = \bigcap_{J \in \text{SIMILAR}_S(I)} \langle J \rangle \neq \varnothing$. As we are working with a space of Helly number ω, and $\text{SIMILAR}_S(I)$ is finite as it only contains subsets of I, showing that any ω configurations $J_1, \ldots, J_\omega \in \text{SIMILAR}_S(I)$ have valid values in common suffices. By the definition of $\text{SIMILAR}_S(I)$, $J_k \subseteq I$ and $\big|J_k\big| \geq n - t_s$ for every k. Since $J_k \subseteq I$ and $\big|J_k\big| \geq n - t_s$, the complement $I \setminus J_k$ has size at most $\big|I\big| - (n - t_s)$. Assume, towards a contradiction, that $J_1, \ldots, J_\omega$ have an empty intersection. Then, for every $(v, P) \in I$, there is some k such that $(v, P) \notin J_k$. Since $(v, P) \in I$, this means $(v, P) \in I \setminus J_k$. That is, every element of I belongs to at least one of the sets $I \setminus J_1, \ldots, I \setminus J_\omega$, so these complements cover I. We obtain that $\big|I\big| \leq \sum_{k=1}^{\omega} \big|I \setminus J_k\big| \leq \omega(\big|I\big| - (n - t_s))$. Equivalently, $\omega(n - t_s) \leq (\omega - 1)\big|I\big|$. On the other hand, since $\big|I\big| < n - t_r$, we have $(\omega - 1)\big|I\big| < (\omega - 1)(n - t_r)$. Moreover, the assumption $n > \omega \cdot t_s + t_r$ implies $\omega(n - t_s) > (\omega - 1)(n - t_r)$. Therefore $\omega(n - t_s) > (\omega - 1)(n - t_r) > (\omega - 1)\big|I\big|$, contradicting $\omega(n - t_s) \leq (\omega - 1)\big|I\big|$. Consequently, $J_1, \ldots, J_\omega$ have a common value-sender pair, and therefore their convex hulls intersect. This implies that every ω sets in the family $\{\langle J \rangle : J \in \text{SIMILAR}_S(I)\}$ have non-empty intersection, and, as the Helly number is ω, we obtain $\bigcap_{J \in \text{SIMILAR}_S(I)} \langle J \rangle \neq \varnothing$.

Therefore, for every $I \in \mathcal{I}$, the intersections required by the responsive similarity condition are non-empty. Let $\sigma(I)$ be any Turing-computable choice of a point in the corresponding intersection: such function exists because of our assumptions on the convex space. Then, if $\big|I\big| \geq n - t_r$, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J)$, and otherwise, $\sigma(I) \in \bigcap_{J \in \text{SIMILAR}_S(I)} \text{VAL}(J)$. Hence, Convex Validity satisfies the responsive similarity condition if $n > \omega \cdot t_s + t_r$.

We now show that, if $n \leq \omega \cdot t_s + t_r$, there is an input configuration I of size at least $n - t_r$ such that $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) = \varnothing$, which will imply that Convex Validity does not satisfy the responsive similarity condition.

Since we are working with a space that has Helly number ω, this space has ω convex sets $C_1, \ldots, C_\omega$ such that $\bigcap_{i=1}^{\omega} C_i = \varnothing$, but every $\omega - 1$ of them have non-empty intersection. For

every $i \in \{1, \ldots, \omega\}$, choose a point $x_i \in \bigcap_{j \neq i} C_j$.

Let I be an input configuration of size $n - t_r$. As $n - t_r \leq \omega \cdot t_s$, we may partition $\text{PARTIES}(I)$ into ω sets $B_1, \ldots, B_\omega$, each of size at most t_s. Assign value x_i to every party in B_i. We define ω input configurations J_i with $1 \leq i \leq \omega$ such that each J_i is in $\text{SIMILAR}_R(I)$. Each J_i omits the parties in B_i, contains all parties in $\text{PARTIES}(I) \setminus B_i$ with their values from I, and assigns arbitrary values in C_i to the parties in $\mathcal{P} \setminus \text{PARTIES}(I)$. As J_i is compatible with I, and $\left|J_i\right| = n - \left|B_i\right| \geq n - t_s$, $J_i \in \text{SIMILAR}_R(I)$. Note that all parties in J_i are assigned values in C_i: for the parties in B_k with $k \neq i$, this follows from the definition of x_k. Then, since C_i is convex, this implies that $\langle J_i \rangle \subseteq C_i$.

Note that $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) \subseteq \bigcap_{J_i : 1 \leq i \leq \omega} \langle J_i \rangle$. As $\langle J_i \rangle \subseteq C_i$, $\bigcap_{i=1}^{\omega} \langle J_i \rangle \subseteq \bigcap_{i=1}^{\omega} C_i = \varnothing$. Then, $\bigcap_{J \in \text{SIMILAR}_R(I)} \text{VAL}(J) = \varnothing$. Consequently, no function σ can satisfy the responsive similarity condition.

Hence, we have obtained that Convex Validity satisfies the responsive similarity condition if and only if $n > \omega \cdot t_s + t_r$. Then, Theorem 7 establishes that there is an unauthenticated t_s-secure t_r-responsive BA protocol solving Convex Validity whenever $n > \max(\omega \cdot t_s + t_r, 3 \cdot t_s) = \omega \cdot t_s + t_r$. Moreover, Theorem 4 establishes that $n > \omega \cdot t_s + t_r$ is necessary (even in authenticated settings). □

Corollary 5. *Honest-Input Validity on a space of size $m > 2$ can be solved responsively if and only if $n > m \cdot t_s + t_r$, both in authenticated and unauthenticated settings.*

Proof. Honest-Input Validity is the validity property $\text{VAL} : \mathcal{I} \to 2^{V_{\mathsf{out}}}$ defined by $\text{VAL}(I) = \text{VALUES}(I)$, where $\text{VALUES}(I)$ denotes the set of input values appearing in I.

We view this validity property as Convex Validity over the convexity space on V_{in} in which every subset of V_{in} is convex. In this space, the convex hull of a set is the set itself: for every input configuration $I \in \mathcal{I}$, $\langle I \rangle = \text{VALUES}(I)$. Hence Convex Validity in this space coincides exactly with Honest-Input Validity.

It remains to identify the Helly number ω of this discrete convexity space on m points. Recall that this is the smallest integer such that, for every finite family of convex sets, if every ω of them have a non-empty intersection, then the whole family has non-empty intersection. We show that $\omega = m$.

First, $\omega \geq m$: we show that there is a family of convex sets such that the whole family has an empty intersection, while every $m - 1$ of the sets in the family intersect. This implies that the Helly number cannot be smaller than m. Consider the family of convex sets $\{V_{\mathsf{in}} \setminus \{v\} : v \in V_{\mathsf{in}}\}$. The intersection of all m sets in this family is empty. However, the intersection of any $m - 1$ of them is non-empty: if the set $V_{\mathsf{in}} \setminus \{v^\star\}$ is omitted, then $v^\star$ belongs to all remaining sets, which proves our claim.

Next, we show that $\omega \leq m$: we show that, every family whose m members intersect has a non-empty intersection, which implies that the Helly number is at most m. Let $\mathcal{F}$ be any finite family of subsets of V_{in} whose total intersection is empty. Since $\bigcap_{S \in \mathcal{F}} S = \varnothing$, every point $v \in V_{\mathsf{in}}$ is excluded by at least one set in $\mathcal{F}$. For each $v \in V_{\mathsf{in}}$, choose a set $S_v \in \mathcal{F}$ such that $v \notin S_v$. Then the subfamily $\{S_v : v \in V_{\mathsf{in}}\}$ has size at most m and already has empty intersection, because every point of V_{in} is excluded by one of its sets. It follows that every empty-intersection family contains an empty-intersection subfamily of size at most m, which is equivalent to saying that every family whose every m members intersect has non-empty total intersection.

Therefore, the Helly number of this convexity space is exactly m. Since $m > 2$, Corollary 4 implies that Honest-Input Validity can be solved responsively if and only if $n > m \cdot t_s + t_r$, both in authenticated and unauthenticated settings. □