

An Analysis of Avalanche Consensus*

Ignacio Amores-Sesar^{1,2}, Christian Cachin^{1,3}, and Philipp Schneider^{1,4}

¹University of Bern

²ignacio.amores@unibe.ch

³christian.cachin@unibe.ch

⁴philipp.schneider2@unibe.ch

Abstract

A family of leaderless, decentralized consensus protocols, called Snow consensus was introduced in a recent whitepaper by Yin et al. These protocols address limitations of existing consensus methods, such as those using proof-of-work or quorums, by utilizing randomization and maintaining some level of resilience against Byzantine participants. Crucially, Snow consensus underpins the Avalanche blockchain, which provides a popular cryptocurrency and a platform for running smart contracts.

Snow consensus algorithms are built on a natural, randomized routine, whereby participants continuously sample subsets of others and adopt an observed majority value until consensus is achieved. Additionally, Snow consensus defines conditions based on participants' local views and security parameters. These conditions indicate when a party can confidently finalize its local value, knowing it will be adopted by honest participants.

Although Snow consensus algorithms can be formulated concisely, there is a complex interaction between randomization, adversarial influence, and security parameters, which requires a formal analysis of their security and liveness. Snow protocols form the foundation for Avalanche-type blockchains, and this work aims to increase our understanding of such protocols by providing insights into their liveness and safety characteristics. First, we analyze these Snow protocols in terms of latency and security. Second, we expose a design issue where the trade-off between these two is unfavorable. Third, we propose a modification of the original protocol where this trade-off is much more favorable.

1 Introduction

Establishing consensus is one of the most fundamental tasks in distributed computing, for instance to implement atomic broadcast, to synchronize processes, or to elect leaders. Distributed blockchains and in particular cryptocurrencies rely on consensus to ensure proper operation, and therefore trust in these systems, which has put increased focus on new kinds of consensus algorithms. In the consensus problem we consider n parties of which some are potentially faulty. Every party has some input value, which we often refer to as opinion in this article. We say that a protocol that coordinates communication and local computations of all parties solves consensus when everyone agrees on a single opinion, which was also the input of at least one party.

The consensus problem becomes challenging in the presence of Byzantine faults, i.e., parties that can deviate from the protocol. In particular, reaching consensus is impossible deterministically in the asynchronous setting with even one fault [12] and in the synchronous setting with one third or more Byzantine faults [11],

*Ignacio Amores-Sesar has been supported by the Swiss National Science Foundation (SNSF) under grant agreement Nr. 200021_188443 (Advanced Consensus Protocols). Philipp Schneider has been supported by a grant from Avalanche, Inc. to the University of Bern.

except if one would use cryptographic signatures. Real protocols that solve consensus in the context of blockchains have to navigate around these impossibility results, while also optimizing other criteria; chief among them are the latency until a transaction is finalized, the throughput of transactions, resource consumption, scalability, and resiliency against adversarial parties, which often necessitates trade-offs [5, 14].

One particularly simple consensus design sacrifices determinism and works along the following principle. Each party continually samples random subsets of other parties and adjusts its opinion based on the observed sample according to certain rules. There has been extensive research that explores such mechanisms, see the recent survey [2]. It has been shown for a subset of protocols of this type that they can be expected to converge very rapidly to a state of stable consensus (with a limited adversary, a bounded number of opinions, and in a synchronous network) [3, 4, 9, 10, 13]. Additionally, such mechanisms have the advantage that parties need only few such samplings to be relatively certain what the consensus opinion will be, resulting in near-linear message complexity.

A whitepaper [16] released in 2019 exploits this design and introduces the Avalanche protocol, which forms the basis of the Avalanche blockchain infrastructure and its services. Avalanche gained popularity and reach due to competitive characteristics in the performance spectrum of latency, throughput, scalability, and resource consumption [15].¹ In particular, [16] introduces the *Snow family* of binary consensus protocols that build on this principle of random samplings, which can be adapted to maintain consistency of the corresponding Avalanche blockchain network.

The simplest protocol of this family, known as *Slush*, works as follows. Each party continuously samples the opinion of $k \geq 2$ others; if such a sampling contains an opinion different from its own at least α times (for some $\alpha > k/2$) then the party adopts this opinion as its own. Slush can be considered as a self-organizing mechanism that it is likely to converge to a stable consensus relatively quickly and remain there, even in the presence of a limited number of parties that deviate from the protocol. The whitepaper also introduces the *Snowflake* and *Snowball protocols*, which add mechanisms to finalize an opinion of a node based on past queries which reflects how stable the observed majority is. The level of confidence in a majority can be controlled with a security parameter β .

The complex interaction between performance characteristics, security level, and the involved parameters k , α , and β makes the analysis of Snow-type consensus protocols challenging. The whitepaper [16] relies primarily on empirical observations and informal explanations to motivate its design choices. Currently, a formal understanding of the performance and security characteristics of Snow protocols is lacking.

Overview and Contributions. We focus on bridging the gap in understanding of Snow consensus protocols, which we consider as a necessary first step for an encompassing analysis of the complete Avalanche blockchain protocol, which builds upon Snow consensus (although the Avalanche protocol itself is beyond the scope of this work). First we explore the performance of Snow protocols, beginning with the self-organizing, binary consensus mechanism of Slush. In Section 3, we express the progress toward a stable consensus per round depending on the distribution of opinions and parameters $k \geq 2$ and $\alpha > \frac{k}{2}$, which gives insights into the evolution of the system (cf. Figure 1 for a visualization).

In Section 4, we show that coming close to a consensus already requires a minimum of $\Omega\left(\frac{\log n}{\log k}\right)$ rounds, even in the absence of adversarial influence (see Theorem 1, a simpler, weaker form is given in Corollary 4.4). Furthermore, we generalize upper bounds from the so-called Median and 3-Majority consensus protocols [3, 9] in the Gossip model (discussed in Section 4) and establish that Slush reaches a stable consensus in $O(\log n)$ rounds (see Theorem 2), which holds even when an adversary can influence up to $O(\sqrt{n})$

¹For instance, through its AVAX token, Avalanche ranks in the top 10 among the “Layer-1” blockchains by market capitalization (as of December 2023).

parties.

We interpret these results in the following way. Even assuming that the performance of Slush matches the lower bound, increasing the parameter k yields only a limited speed-up of $O(\log k)$ (usually $k \ll n$). Furthermore, since the message complexity per round is $\Theta(kn)$ Slush has the advantage of near-linear message complexity for small k , which is negated if k increases significantly (e.g., sampling sizes close to n). We conclude that values higher than $k = 20$ as suggested originally [16] have diminishing benefit and an unfavorable trade-off in terms of message complexity. In Section 5 we show that the lower bound $\Omega\left(\frac{\log n}{\log k}\right)$ rounds extends to Snowflake and Snowball.

In Section 6 we analyze the security mechanisms of the Snow protocols, that deal with the possibility of failing to achieve consensus due to the randomized nature of the algorithm or adversarial influence. The protocol provides a security parameter β to control the probability of such a failure. This has an unfavorable trade-off as we show in Section 6, specifically, a negligible probability of failure (w.r.t. β) and a latency to finalize a value that is at most polynomial in β are mutually exclusive (see Theorem 3). In Section 7 we propose a solution for this issue by introducing an alternative protocol. It replaces the security mechanism of Snowball with a simple mechanism that achieves security with all but negligible probability (w.r.t. β) in $O(\beta + \log n)$ rounds (see Theorem 4 and Corollary 7.4).

2 Preliminaries

Before moving to the technical parts of the analysis, we introduce the definitions and modeling assumptions that we use throughout the paper. In part, this work aims to be a supplementary of the whitepaper [16] in the style of other theoretical works that study randomized, self-stabilizing consensus protocols [3, 4, 9, 10, 13]. Therefore our nomenclature, definitions and modeling assumptions are a composition of those.

2.1 Model

Communication. We consider a fully connected network of n parties with identifiers $\mathcal{N} = \{1, \dots, n\}$. Parties communicate by sending point to point messages. For the message transfer we assume the synchronous message passing model, where there is a fixed period of time until any given message is delivered. In fact, the synchronous setting allows to assume that time is slotted into discrete rounds and all messages sent in the previous round have arrived by the next round. This also allows us to use the number of rounds as a proxy for algorithm running time.

Consensus. In the general problem setup there are m opinions in the network and each party has an initial opinion, however note that in the context of Snow protocols we typically have $m = 2$. Snow protocols can be seen as self-stabilizing protocols and we define a stable state where almost all parties have the same opinion and the likelihood to revert from this state is low (see Section 7 for some properties of this stable state).

Definition 2.1 (State of Stable Consensus). *The system is in a state of stable consensus if at least $n - o(n)$ parties have the same opinion.*

Randomization or the presence of an adversary implies that at any point in time there is a non-zero chance that a stable consensus is reverted. Therefore, in the context of blockchain applications, parties need to eventually finalize or *decide* on an opinion. In that sense we define the consensus problem as follows.

Definition 2.2 (Consensus Problem). *A protocol solves this problem if the following conditions are satisfied.*

Termination: *Every party eventually decides on some opinion.*

Validity: *If all parties propose the same value, then all parties decide on that value.*

Integrity: *No party decides twice.*

Agreement: *No two parties decide differently.*

We consider this consensus problem under the influence of the following adversary.

Definition 2.3 (*F*-Bounded Adversary). *An F -bounded adversary can set the opinion of up to F (undecided) parties at the beginning of each round to one of the m opinions.*

Randomization, Security and Latency. The consensus protocols we consider in this work are randomized, and we work with some standard definitions that we summarize in Appendix A. In particular we also consider an F -bounded adversary, which introduces a non-zero chance to delay a consensus for any fixed period of time. Hence, we can only hope to make the probability of failure of such a protocol negligibly small, while at the same time maintaining a reasonable latency (i.e., number of rounds until consensus). To connect the notions of failure probability and latency we make the following provisions.

Let E be an event or condition during the execution of some protocol \mathcal{P} , e.g., E describes the event that \mathcal{P} successfully establishes consensus, see Definition 2.2. The protocols we are investigating typically depend on so called security parameters. Increasing the security parameter increases the likelihood of success but typically has detrimental effects on the running time. To quantify this, we formally define further below what it means that E holds with all but negligible probability with respect to protocol \mathcal{P} . Roughly speaking, as we increase the running time of a protocol measured in the security parameter λ , the probability that some condition E (e.g., consensus) has not been established yet, decreases super-polynomially in λ .

Definition 2.4 (Negligible Probability, Security Parameter). *An event E holds with all but negligible probability or equivalently its complement \bar{E} has negligible probability in a protocol \mathcal{P} , if the following holds. There exists a polynomial ρ such that for any polynomial π there exists a value $\lambda_0 \geq 0$, such that for all $\lambda \geq \lambda_0$ the following holds. If \mathcal{P} is executed for at least $\rho(\lambda)$ rounds it holds that $\mathbb{P}(\bar{E}) \leq 1/\pi(\lambda)$, i.e., $\mathbb{P}(E) > 1 - 1/\pi(\lambda)$. We call the value $\lambda \geq \lambda_0$ a security parameter.*

In this article, we often talk about randomly sampling a set of k parties, which means that we take a uniform random sample of size k from the set of all n parties *with repetition*, which is modeled by the binomial distribution. Note that sampling *without repetition*, represented by the hyper-geometric distribution, approaches the binomial distribution as the ratio n/k becomes larger. Therefore, our results approximate the case without repetition well for $n \gg k$ (the usual case). The assumption of uniform sampling with repetition makes our analysis much more feasible, in particular it removes undesirable marginal cases (e.g. k -samples for $n < k$) allowing use to use continuous functions to describe certain aspects of the system.

2.2 The Snow Family

The Snow family consists of three consensus protocols based on random sampling instead of the traditional quorum intersection. This approach allows the snow family to reduce the message complexity by sending messages to a constant number of parties each round. Here, we provide an informal description of each protocol, whereas a more detailed pseudocode can be found in Appendix C.

Slush. The first protocol in the Snow family is Slush (Algorithm 2). An honest party j runs the Slush protocol in local rounds, however in the general protocol every party may have a different round value. Party j starts the round by randomly sampling k parties for their opinion, with k a constant value. If at least α parties respond with the same opinion b , party j adopts it as its own opinion and starts a new round. The value α must be strictly larger than $\frac{k}{2}$. If there is no α -majority, party j keeps its opinion and starts a new round. The Slush algorithm has a hard-coded number of rounds defining the protocol's end. When party j reaches the maximum round, it *decides*(b) its candidate value b . Note that in our analysis in Section

3 and 4 we analyze the time until Slush reaches a state of stable consensus (Definition 2.1) and assume that this hard coded maximum round does not exist or is sufficiently large to not play a role.

Snowflake. The main limitation of the Slush algorithm is the hard-coded number of rounds. The number of rounds needs to be relatively high to guarantee consensus even in the worst case (which the case when the network starts in a bivalent state: half of the parties *proposes*(0) and the other half *proposes*(1)). Snowflake aims to address this issue by modifying the termination condition of Slush.

In the Snowflake protocol (Algorithm 3), party j counts the number of consecutive queries with an α -majority for opinion b . If j observes β consecutive rounds with α -majority for b , party j *decides*(b). The intuition behind this termination rule is the following: the probability of obtaining β consecutive α -majorities for opinion b is small when expressed as a function of β , unless almost every party has b as candidate value in the network. This termination rule allows for an adaptive running time based on the state of the network. Looking ahead, we show how this termination rule forces the Snowflake protocol to choose between a high confidence in the agreement property and polynomial running time (Theorem 3).

Snowball. The Snowball protocol (Algorithm 4) introduces another modification how parties change their opinion. In Snowflake, the change of opinion is only based on the outcome of the last query, i.e., it is stateless. By contrast, in Snowball, party j considers the past queries in order to decide whether to change its opinion or not. Party j changes its opinion value from b to b' when the number of α -majorities for b' surpasses the number of α -majorities for value b since the beginning of the execution. The idea behind considering the whole history of the protocol is to make it less likely for a party with opinion b to switch to b' when the prevalent opinion in the network is b , thus possibly reducing the number of rounds until termination. Looking ahead, we show that this routine does not reduce the number of rounds until termination in expectation (Lemma 5.5).

Avalanche. The Snow consensus protocols serve as foundation for the Avalanche consensus [1, 16]. Avalanche employs a classification system to group transactions into conflicting sets and subsequently applies a tailored adaptation of the Snowball algorithm to each of these conflict sets. To optimize communication efficiency, Avalanche establishes connections between distinct instances of the Snowball consensus, enabling the reuse of messages and, consequently, reducing message complexity. However, due to these interdependencies, Avalanche is unable to inherit the liveness properties from the Snow family [1]. Nevertheless, it is noteworthy that the security of Avalanche remains equivalent to that of the Snowball protocol [16].

2.3 Related Work

The Avalanche protocol was introduced fairly recently thus research into this protocol is limited. The whitepaper [16] gives an iterative presentation of its algorithms and concepts, in particular the Snow protocols for binary consensus on which it then builds its Avalanche protocol in the UTXO model. This is supplemented with explanations about the design decisions and empiric data that highlights the protocols' performance in terms of latency, throughput and Byzantine resilience. Subsequently, the article [1] provided a formal description of the Avalanche protocol. They also showcase a vulnerability (that has since been addressed by subsequent versions of Avalanche) that is specific to the Avalanche protocol, where a single malicious party can delay acceptance of a transaction and proposes a modification that prohibits this attack. Part of the reason that an encompassing analysis of Avalanche is outside the scope for this work, is that it is currently still evolving, for instance recently transitioning from a directed acyclic graph (DAG) to a chain, providing a total order for transactions as opposed to a partial order.

Self-stabilizing consensus protocols based on random samplings have been investigated much earlier in message passing models, motivated by the so called GOSSIP model.² A particular strain of such protocols that attained some focus in the past are the so called 3-Majority, the 2-choices and the Median protocols [2]. In the 3-Majority protocol parties sample 3 random others and adopt the majority opinion using the first sampled parties' opinion in case of a tie. The 2-Choices protocol works similar, but only 2 parties are sampled with the third being the party itself which also provides the default opinion. In the Median protocol a party samples 2 others and adopts the median value among theirs and their own (which requires a total order on the opinions). There has been a plethora of work on the analysis of these and similar light weight protocols based on random sampling, a selection of those are [3, 4, 6, 8, 9, 13], see also the survey [2] for an overview. These works usually focus on analyzing the time to consensus with respect to the initial number of opinions in the network, sometimes also on the required initial bias of the network in case a consensus on the initial majority is desired (plurality consensus).

In contrast to this work, these articles focus on samplings of size at most 3, analysis of the dynamics for the whole spectrum of k, α have, to the best of our knowledge, so far not been attempted.³ Crucially, in case the number of opinions is constant, all these works arrive at $O(\log n)$ rounds until a state of stable consensus is attained with high probability. Interestingly, in the binary case the 3-Majority, the 2-choices and the Median protocols can be all be related to special cases of Slush. Besides analyzing security aspects of snow protocols, one of the main contributions of this work is to show how the dynamics of such sampling based protocols behave in the size k of those samplings.

3 Dynamics of Slush

The whitepaper [16] observes that the Slush consensus protocol converges to a stable consensus very fast in practice. Concrete claims are made pertaining to the time to consensus, but no conclusive proof is given. In this section we analyze the the rate of convergence of Slush towards a consensus, which will later also inform the rate of convergence of Snowflake and Snowflake (Section 5).

3.1 Expected Rate of Progress of Slush

We start by investigating the expected rate of progress of Slush, which characterizes the dynamics of Slush and how it depends on the parameters α and k . We will later show that other Snow protocols (Snowflake, Snowball) behave similar in terms of the required number of rounds to consensus.

We make the following definitions and assumptions. First we assume that all parties have an initial opinion 0 or 1, so no party has initially the opinion \perp (the case where there exist parties with opinion \perp can be disregarded for the lower bound and is handled separately for the upper bound). Recall that the set of parties \mathcal{N} is numbered from 1 to n and assume rounds are numbered $0, 1, 2, \dots$

- Let $X_{ij} \in \{0, 1\}$ be the current opinion of party j after round i of the Slush protocol was executed ($X_{0,j}$ describes the initial opinion of party j).
- Let Y_{ij} be the number of replies with opinion 1 that party j obtains in its sample of k parties in round i . Note that $Y_{ij} \sim \text{Bin}(k, p_i)$.
- Let the state of the network be $S_i := \sum_{j=1}^n X_{ij}$, which describes the total number of parties whose current opinion is 1.
- Let $p_i := S_i/n$ be the relative share of parties with opinion 1 in round i , which corresponds the probability that a sampled party has state 1.

²In the GOSSIP model, nodes can contact a few random neighbors in a graph.

³Metrics other than the round complexity have been considered for the binary k -Majority protocol which relates to Slush [7].

- For $i \geq 1$, we define as $\Delta_i := S_i - S_{i-1}$, i.e., the absolute progress to 1-consensus (or 0-consensus for negative values).
- Let $\delta_i := \mathbb{E}(\Delta_i)/n$ be the expected relative progress in round i . We will later show that δ_i can also be expressed as a function $\delta : [0, 1] \rightarrow \mathbb{R}$ that only depends on p_i (when viewing k, α as fixed values), such that $\delta_i = \delta(p_i)$. Subsequently, we establish a relation between the $\delta(p_i)$ for varying parameters k, α , in which case we denote it as $\delta^{k, \alpha}(p_i)$ (however, for conciseness we will refrain using this superscript whenever possible, in particular when only single values for k and α are involved).

Note that for $i \geq 1$, the quantities $X_{ij}, Y_{ij}, S_i, \Delta_i$, are random variables. This is not the case for the *expected* relative progress δ_i , which, for fixed α, k , can be expressed only in terms of p_i , i.e., δ_i can be expressed as a function $\delta(p_i)$ that depends only on p_i .

Lemma 3.1. *Let $k/2 < \alpha \leq k$. Then*

$$\delta_i = \delta(p_i) := \sum_{\ell=\alpha}^k \binom{k}{\ell} \left[p_i^\ell (1-p_i)^{k-\ell+1} - (1-p_i)^\ell p_i^{k-\ell+1} \right].$$

Proof. In a given round i a party j with can observe at least α times the opposite opinion in its query, in which case it switches, or not (observing at least α of both opinions is precluded due to $\alpha > k/2$). Note that for δ_i only events where parties switch their opinion are relevant. For some party j we have

$$\begin{aligned} \mathbb{P}(X_{ij} = 1 \mid X_{i-1,j} = 0) &= \mathbb{P}(Y_{ij} \geq \alpha) \\ &= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell}, \\ \mathbb{P}(X_{ij} = 0 \mid X_{i-1,j} = 1) &= \mathbb{P}(Y_{ij} \leq k - \alpha) \\ &= \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} = \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell}, \end{aligned}$$

where the last step is due to symmetry (see also Appendix B). Let δ_i^0 be the probability that a randomly selected party switches to from 1 to 0 and let δ_i^1 be analogous probability for switching from 0 to 1. These can also be interpreted as expected portions of parties switching from 1 to 0 and vice versa. We obtain

$$\begin{aligned} \delta_i^0 &= \mathbb{P}(X_{ij} = 0 \cap X_{i-1,j} = 1) = (1-p_i) \mathbb{P}(X_{ij} = 0 \mid X_{i-1,j} = 1) = \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell+1}, \\ \delta_i^1 &= \mathbb{P}(X_{ij} = 1 \cap X_{i-1,j} = 0) = p_i \cdot \mathbb{P}(X_{ij} = 1 \mid X_{i-1,j} = 0) = \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1}. \end{aligned}$$

The random variable Δ_i is determined by the number of parties that switch from 0 to 1 minus those that switch from 1 to 0. Since the events $X_{ij} = 0 \cap X_{i-1,j} = 1$ and $X_{ij} = 1 \cap X_{i-1,j} = 0$ are disjoint, we have that $E(\Delta_i) = n\delta_i^1 - n\delta_i^0$. Thus $\delta_i = nE(\Delta_i) = \delta_i^1 - \delta_i^0$ and the claim follows. \square

3.2 Mapping out the Dependency of δ_i on k and α

Recall that we define $\delta^{k, \alpha}(p_i)$ as the function defining the rate of progress for parameters α, k depending on p_i . It is interesting that for $k = 2\alpha - 1$ we have $\delta^{k, \alpha} = \delta^{k, \alpha+1}$ as the first summand in the expression $\delta^{k, \alpha}(p_i)$ from Lemma 3.1 is zero. Another interesting observation is that in the marginal case $k = \alpha = 1$, we have $\delta^{k, \alpha} = 0$ (see Figure 1a), i.e., there is no expected progress and Slush essentially degenerates into a random walk, and it is not too hard to show that in this case Slush takes $\Omega(n^2)$ rounds in expectation.

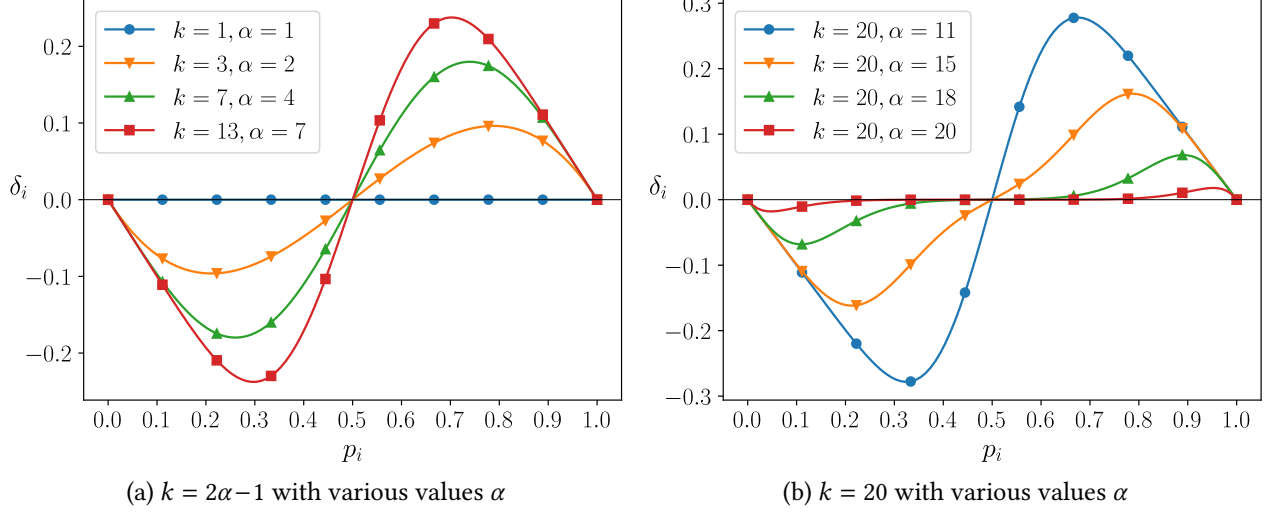


Figure 1: Plots of $\delta(p_i)$ for different parameters k and α . For $k = 2\alpha - 1$ the expected progress for larger α dominates those for smaller (note that in the extreme case $k = \alpha = 1$ there is no expected progress). For fixed k the opposite is true. The combination $k = 20, \alpha = 15$ was suggested by the whitepaper [16]. Note that $\delta(p_i)$ is point-symmetric with respect to the point $(\frac{1}{2}, 0)$.

In this section we establish non-trivial relations and claims for $\delta^{k,\alpha}(p_i)$. First, we show that $\delta^{k,\alpha}(p_i)$ is always larger (in absolute value) than $\delta^{k,\alpha'}(p_i)$ for $\alpha' > \alpha$ (see Figure 1b for a visualized example of this claim). This means that choosing the smallest α with $\alpha > k/2$ (i.e., $\alpha = \lceil \frac{k+1}{2} \rceil$) is best in terms of the expected rate of progress $\delta^{k,\alpha}(p_i)$. While this is useful on its own regarding the choice of α in practice, we utilize this later to essentially eliminate α from the of analysis for the lower bound of the rate of convergence to consensus.

Lemma 3.2. *For fixed k , let $k/2 < \alpha \leq k$ and consider $\alpha' > \alpha$. Then for any p_i we have $|\delta^{k,\alpha}(p_i)| \geq |\delta^{k,\alpha'}(p_i)|$.*

Proof. The lemma is a corollary of Lemma 3.1, where we notice that all summands of $\delta(p_i) = \delta^{k,\alpha}(p_i)$ are either positive for $p_i > 1/2$ or negative for $p_i < 1/2$ or 0 for $p_i = 1/2$. Since all summands of $\delta^{k,\alpha'}(p_i)$ occur in $\delta^{k,\alpha}(p_i)$ and all have the same sign for some fixed p_i , we have the claim. \square

In particular, for $\alpha = \lceil \frac{k+1}{2} \rceil$ we can express $\delta^{k,\alpha}(p_i)$ in a different form, which we will use frequently in subsequent lemmas to simplify (or enable) subsequent proofs.

Lemma 3.3. *Let $\alpha = \lceil \frac{k+1}{2} \rceil$. Then*

$$\begin{cases} \delta^{k,\alpha}(p_i) = \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] - p_i, & k \text{ odd} \\ \delta^{k,\alpha}(p_i) = \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] - p_i + \binom{k}{\alpha-1} p_i^\alpha (1-p_i)^{\alpha-1}, & k \text{ even.} \end{cases}$$

Proof. We start out with the expression from Lemma 3.1.

$$\begin{aligned}
\delta^{k,\alpha}(p_i) &= \sum_{\ell=\alpha}^k \binom{k}{\ell} \left[p_i^\ell (1-p_i)^{k-\ell+1} - (1-p_i)^\ell p_i^{k-\ell+1} \right] \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell+1} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} \left[p_i^\ell (1-p_i)^{k-\ell} - p_i^{\ell+1} (1-p_i)^{k-\ell} \right] - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell+1} (1-p_i)^{k-\ell} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1}
\end{aligned}$$

The proof forks into cases by the parity of k . Consider the case that k is odd. Then we have $k = 2\alpha - 1$. Continuing from above, we substitute $\ell' := k - \ell$ in the middle sum, i.e., ℓ' goes from 0 to $k - \alpha = \alpha - 1$.

$$\begin{aligned}
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - \sum_{\ell'=0}^{\alpha-1} \binom{k}{k-\ell'} p_i^{k-\ell'+1} (1-p_i)^{\ell'} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - \sum_{\ell'=0}^{\alpha-1} \binom{k}{\ell'} p_i^{k-\ell'+1} (1-p_i)^{\ell'} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - p_i \underbrace{\sum_{\ell=0}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell}}_{=1} = \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - p_i.
\end{aligned}$$

In the case that k is even we have $k = 2\alpha - 2$. Again we substitute $\ell' := k - \ell$ in the middle sum, that is, ℓ' goes from 0 to $k - \alpha = \alpha - 2$.

$$\begin{aligned}
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - \sum_{\ell'=0}^{\alpha-2} \binom{k}{k-\ell'} p_i^{k-\ell'+1} (1-p_i)^{\ell'} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - \sum_{\ell'=0}^{\alpha-2} \binom{k}{\ell'} p_i^{k-\ell'+1} (1-p_i)^{\ell'} - \sum_{\ell=\alpha}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell+1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - p_i \sum_{\ell=0}^k \binom{k}{\ell} (1-p_i)^\ell p_i^{k-\ell} + \binom{k}{\alpha-1} p_i^\alpha (1-p_i)^{\alpha-1} \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} - p_i + \binom{k}{\alpha-1} p_i^\alpha (1-p_i)^{\alpha-1}. \quad \square
\end{aligned}$$

The next two lemmas show, perhaps surprisingly, that for odd $k = 2\alpha - 1$ the expected progress functions $\delta^{k,\alpha}(p_i)$ and $\delta^{k-1,\alpha}(p_i)$ coincide. When combined with Lemma 3.2, this will later, when we cover lower bounds, allow us to focus our analysis solely on odd values of the form $k = 2\alpha - 1$ as the corresponding claim for even $k = 2\alpha - 2$ is implied.

Lemma 3.4. *For $k > \alpha \geq 1$ the following equation holds*

$$\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} = \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell-1} \right] + \binom{k-1}{\alpha-1} p_i^\alpha (1-p_i)^{k-\alpha}.$$

Proof.

$$\begin{aligned}
& \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \\
&= \left[\sum_{\ell=\alpha}^{k-1} \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \left\{ \binom{k-1}{\ell} + \binom{k-1}{\ell-1} \right\} p_i^\ell (1-p_i)^{k-\ell} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \left\{ \binom{k-1}{\ell} + \binom{k-1}{\ell-1} \right\} p_i^\ell (1-p_i)^{k-\ell} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] + \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell-1} p_i^\ell (1-p_i)^{k-\ell} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] + \left[\sum_{\ell=\alpha-1}^{k-2} \binom{k-1}{\ell} p_i^{\ell+1} (1-p_i)^{k-\ell-1} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell-1} \right] - \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^{\ell+1} (1-p_i)^{k-\ell-1} \right] \\
&\quad + \left[\sum_{\ell=\alpha-1}^{k-2} \binom{k-1}{\ell} p_i^{\ell+1} (1-p_i)^{k-\ell-1} \right] + p_i^k \\
&= \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell-1} \right] + \binom{k-1}{\alpha-1} p_i^\alpha (1-p_i)^{k-\alpha} - p_i^k + p_i^k \quad \square
\end{aligned}$$

Lemma 3.5. *Let $k = 2\alpha - 1$ and $\alpha \geq 2$. Then $\delta^{k,\alpha} = \delta^{k-1,\alpha}$.*

Proof. The proof is implied by combining Lemma 3.3 and Lemma 3.4. We start with the expression of $\delta^{k,\alpha}(p_i)$ derived in Lemma 3.3.

$$\begin{aligned}
\delta^{k,\alpha}(p_i) &\stackrel{\text{Lem.3.3}}{=} \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] - p_i \\
&\stackrel{\text{Lem.3.4}}{=} \left[\sum_{\ell=\alpha}^{k-1} \binom{k-1}{\ell} p_i^\ell (1-p_i)^{k-\ell-1} \right] + \binom{k-1}{\alpha-1} p_i^\alpha (1-p_i)^{\alpha-1} - p_i \\
&\stackrel{\text{Lem.3.3}}{=} \delta^{k-1,\alpha}(p_i) \quad \square
\end{aligned}$$

As our final structural claim in this subsection, we show that for $k = 2\alpha - 1$ or $k = 2\alpha$, the expected progress towards consensus $|\delta^{k,\alpha}(p_i)|$ does not decrease as α (and thereby k) increases, an example is given in Figure 1a. We will use this lemma later to extend an upper bound for the number of rounds to a consensus from small values of α to large (where $k = 2\alpha$ or $k = 2\alpha - 1$).

Lemma 3.6. *Let $k = 2\alpha - 1$ and $k' = 2\alpha' - 1$ for $\alpha > \alpha' > 1$. Then for any p_i we have $|\delta^{k,\alpha}(p_i)| \geq |\delta^{k',\alpha'}(p_i)|$. The same holds for even $k = 2(\alpha - 1)$ and $k' = 2(\alpha' - 1)$.*

Proof. We show the bound for $p_i \geq \frac{1}{2}$, the claim for $p_i \leq \frac{1}{2}$ holds by symmetry. Let $k = 2\alpha - 1$. We show that $\delta^{k,\alpha}(p_i) = \delta^{k-1,\alpha}(p_i) \geq \delta^{k-2,\alpha-1}(p_i) \geq \delta^{k-3,\alpha-1}(p_i)$, which implies the lemma for $\alpha' = \alpha - 1$ and the

rest follows by induction.

$$\begin{aligned}
\delta^{k,\alpha}(p_i) &\stackrel{\text{Lem.3.5}}{=} \delta^{k-1,\alpha}(p_i) \\
&\stackrel{\text{Lem.3.4}}{=} \left[\sum_{\ell=\alpha}^{k-2} \binom{k-2}{\ell} p_i^\ell (1-p_i)^{k-\ell-2} \right] + \binom{k-2}{\alpha-1} p_i^\alpha (1-p_i)^{k-\alpha} - p_i \\
&\stackrel{p_i \geq 1/2}{\geq} \left[\sum_{\ell=\alpha}^{k-2} \binom{k-2}{\ell} p_i^\ell (1-p_i)^{k-\ell-2} \right] + \binom{k-2}{\alpha-1} p_i^{\alpha-1} (1-p_i)^{k-\alpha+1} - p_i \\
&= \left[\sum_{\ell=\alpha-1}^{k-2} \binom{k-2}{\ell} p_i^\ell (1-p_i)^{k-\ell-2} \right] - p_i \\
&\stackrel{\text{Lem.3.3}}{=} \delta^{k-2,\alpha-1}(p_i) \stackrel{\text{Lem.3.5}}{=} \delta^{k-3,\alpha-1}(p_i). \quad \square
\end{aligned}$$

Note that while increasing α with $k = 2\alpha-1$ or $k = 2\alpha$ does in fact strictly increase $|\delta^{\alpha,k}(p_i)|$ thereby speeding up the expected time to consensus, this effect is rather limited, as we shall see in the next section.

4 Bounding the Time to Consensus for Slush

We will now show how the expected progress $\delta(p_i)$ can be used to obtain bounds for the number of rounds required to obtain consensus.

4.1 Lower Bound

As the system converges to consensus, arguably the most critical phase is when the network is roughly in balanced state, i.e., where fractions of parties with opinion 0 and 1 are roughly equal ($p_i \approx 1/2$) and where progress $\delta(p_i)$ towards consensus is close to 0, see Figure 1.

To lead us out of a potential perfect balance, the system can only rely pure randomness to gain some small initial imbalance, as the expected progress is 0. (The best one can hope for is a deviation of $p_i \approx 1/2 + c/\sqrt{n}$ for some constant c within reasonable time bounds, due to the central limit theorem). After that initial perturbation the convergence to consensus crucially depends on how fast the expected progress for the next round grows in parameter p_i .

Indicative for the change in progress is the derivative of $\delta(p_i)$, whose upper bound is useful to analyze the case where the system moves to a 1-consensus (w.l.o.g., due to a symmetry argument). Intuitively, this limits how fast the expected progress increases from an almost balanced state. We will first restrict ourselves to the case $k = 2\alpha-1$, as the previous section gives us all tools to extend this result to general k and α , as will be shown formally afterwards.

Lemma 4.1. *Let $k = 2\alpha - 1$. For $p_i \geq 1/2$ it holds that $\frac{\partial \delta(p_i)}{\partial p_i} \leq k - 1$.*

Proof. We use the expression from Lemma 3.3 and obtain the following derivative.

$$\begin{aligned}
\frac{\partial \delta(p_i)}{\partial p_i} &= \frac{\partial}{\partial p_i} \left(\left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \right] - p_i \right) \\
&= \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} \left(\ell \cdot p_i^{\ell-1} (1-p_i)^{k-\ell} - (k-\ell) \cdot p_i^\ell (1-p_i)^{k-\ell-1} \right) \right] - 1
\end{aligned}$$

We evaluate at $p_i = 1/2$, then

$$\begin{aligned}
\frac{\partial \delta(\frac{1}{2})}{\partial p_i} &= \frac{1}{2^{k-1}} \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} (2\ell - k) \right] - 1 \leq \frac{1}{2^{k-1}} \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} (2\ell - k) \right] - 1 \\
&= \frac{1}{2^{k-1}} \left[2 \sum_{\ell=\alpha}^k \binom{k}{\ell} \ell - \frac{k}{2} \sum_{\ell=0}^k \binom{k}{\ell} \right] - 1 && \text{due to } k = 2\alpha - 1 \text{ and } \binom{k}{\ell} = \binom{k}{k-\ell} \\
&\leq \frac{1}{2^{k-1}} \left[2 \sum_{\ell=0}^k \binom{k}{\ell} \ell - \frac{k}{2} \sum_{\ell=0}^k \binom{k}{\ell} \right] - 1 \\
&= \frac{1}{2^{k-1}} \left[2k \cdot 2^{k-1} - \frac{k}{2} \cdot 2^k \right] - 1 = k-1.
\end{aligned}$$

In the following we will also show that the second derivative $\frac{\partial \delta(p_i)}{\partial^2 p_i}$ is at most 0 for $p_i \geq \frac{1}{2}$. Combined with the above, this implies that the first derivative is $\frac{\partial \delta(p_i)}{\partial p_i} \leq k-1$ for any $p_i \geq \frac{1}{2}$. It remains to prove the claim about $\frac{\partial \delta(p_i)}{\partial^2 p_i}$. This is a bit tedious and involves modifying binomial coefficients on the level of the definition

$\binom{k}{\ell} = \frac{n!}{k!(n-k)!}$ (third step) and then shifting sum indices (fourth step).

$$\begin{aligned}
\frac{\partial \delta(p_i)}{\partial^2 p_i} &= \frac{\partial}{\partial p_i} \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} \left(\ell \cdot p_i^{\ell-1} (1-p_i)^{k-\ell} - (k-\ell) \cdot p_i^{\ell} (1-p_i)^{k-\ell-1} \right) \right] \\
&= \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell-2} (1-p_i)^{k-\ell-2} \\
&\quad \cdot \left(\ell(\ell-1)(1-p_i)^2 - 2\ell(k-\ell)p_i(1-p_i) + (k-\ell)(k-\ell-1)p_i^2 \right) \\
&= \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell-2} (1-p_i)^{k-\ell} \ell(\ell-1) \right] \\
&\quad + \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell} (1-p_i)^{k-\ell-2} (k-\ell)(k-\ell-1) \right] \\
&\quad - 2 \cdot \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell-1} (1-p_i)^{k-\ell-1} \ell(k-\ell) \right] \\
&= \left[\sum_{\ell=\alpha}^k \binom{k}{\ell-1} p_i^{\ell-2} (1-p_i)^{k-\ell} (\ell-1)(k-\ell+1) \right] \\
&\quad + \left[\sum_{\ell=\alpha}^k \binom{k}{\ell+1} p_i^{\ell} (1-p_i)^{k-\ell-2} (\ell+1)(k-\ell-1) \right] \\
&\quad - 2 \cdot \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell-1} (1-p_i)^{k-\ell-1} \ell(k-\ell) \right] \\
&= \left[\sum_{\ell=\alpha-1}^k \binom{k}{\ell} p_i^{\ell-1} (1-p_i)^{k-\ell-1} \ell(k-\ell) \right] \\
&\quad + \left[\sum_{\ell=\alpha+1}^k \binom{k}{\ell} p_i^{\ell-1} (1-p_i)^{k-\ell-1} \ell(k-\ell) \right] \\
&\quad - 2 \cdot \left[\sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^{\ell-1} (1-p_i)^{k-\ell-1} \ell(k-\ell) \right] \\
&= \binom{k}{\alpha-1} p_i^{\alpha-2} (1-p_i)^{\alpha-1} (\alpha-1) \alpha - \binom{k}{\alpha} p_i^{\alpha-1} (1-p_i)^{\alpha-2} (\alpha-1) \alpha \\
&= \binom{k}{\alpha} p_i^{\alpha-2} (1-p_i)^{\alpha-2} (\alpha-1) \alpha \cdot (1-2p_i) \quad \text{since } \binom{k}{\alpha} = \binom{k}{\alpha-1} \\
&\leq 0 \quad \text{for } p_i \geq \frac{1}{2}. \quad \square
\end{aligned}$$

Next, we show that the progress towards consensus in a single round is limited, in particular around the balanced state. Here we utilize two tools. First of all, we employ Lemma 4.1 that bounds the progress around an almost balanced state but only for the "well behaved" case $k = 2\alpha - 1$. Second, we use Lemma 3.5 to extend this to even Lemma 3.2 to extend it to any k and α for which $\frac{k}{2} < \alpha \leq k$.

Lemma 4.2. *Let $k \geq 2$ and $\frac{k}{2} < \alpha \leq k$ and $S_i \geq \frac{n}{2}$ (w.l.o.g.). Then $\Delta_{i+1} > (k-1)(S_i - \frac{n}{2}) + t\sqrt{n}$ with probability at most $\frac{1}{t^2}$, for any $t \geq 1$.*

Proof. We prove the claim for $k = 2\alpha - 1$ for $\alpha \geq 2$, and generalize it further below. Since $\delta(p_i)(\frac{1}{2}) = 0$ (cf.

Lemma 3.1 or Figure 1 and $\frac{\partial \delta(p_i)}{\partial p_i} \leq k$ by Lemma 4.1 we obtain $\mathbb{E}(\Delta_{i+1}) = \delta(p_i)(p_i) \cdot n \leq (k-1)n(p_i - \frac{1}{2})$.

To also obtain the claim with the stated probability, let us look at the variance $\sigma^2 := \text{Var}(S_{i+1}) = \text{Var}(\sum_{j=1}^n X_{i+1,j})$. After round i before round $i+1$ we have $\text{Var}(\Delta_{i+1}) = \text{Var}(S_{i+1} - S_i) = \text{Var}(S_{i+1}) = \sigma^2$, since S_i is a constant offset.

The $X_{i+1,j} \in \{0, 1\}$ are independent and identically distributed with $\text{Var}(X_{i+1,j}) \leq 1$, thus $\sigma^2 = n \cdot \text{Var}(X_{i+1,j}) \leq n$. Using the Chebyshev inequality we obtain

$$\begin{aligned} \mathbb{P}\left(\Delta_{i+1} \geq (k-1)n(p_i - \frac{1}{2}) + t\sqrt{n}\right) &\leq \mathbb{P}\left(\Delta_{i+1} \geq \mathbb{E}(\Delta_{i+1}) + t\sigma\right) \\ &\leq \mathbb{P}\left(|\Delta_{i+1} - \mathbb{E}(\Delta_{i+1})| \geq t\sigma\right) \leq \frac{1}{t^2}. \end{aligned}$$

It remains to generalize the claim for k, α . Let $\Delta_i^{k,\alpha}$ and $\delta_i^{k,\alpha}$ denote the absolute and relative expected progress in round i for these specific parameters. Let $k = 2\alpha - 1, \alpha \geq 2$ as before. By Lemma 3.5 we have $\mathbb{E}(\Delta_i^{k,\alpha}) = n \cdot \delta^{k,\alpha}(p_i) = n \cdot \delta^{k-1,\alpha}(p_i) = \mathbb{E}(\Delta_i^{k-1,\alpha})$, hence

$$\begin{aligned} \mathbb{P}\left(\Delta_{i+1}^{k-1,\alpha} \geq (k-1)n(p_i - \frac{1}{2}) + t\sqrt{n}\right) &\leq \mathbb{P}\left(\Delta_{i+1}^{k-1,\alpha} \geq \mathbb{E}(\Delta_{i+1}^{k,\alpha}) + t\sigma\right) \\ &= \mathbb{P}\left(\Delta_{i+1}^{k-1,\alpha} \geq \mathbb{E}(\Delta_{i+1}^{k-1,\alpha}) + t\sigma\right) \leq \frac{1}{t^2}. \end{aligned}$$

Note that this extends the claim from $k = 2\alpha - 1$ (odd) to the case $k = 2\alpha - 2$ (even).

Let now k be arbitrary (odd or even) and $\frac{k}{2} < \alpha' \leq k$. By Lemma 3.2 we have that

$$\mathbb{E}(\Delta_i^{k,\alpha'}) = n \cdot \delta(p_i)^{k,\alpha'} \leq n \cdot \delta^{k,\alpha}(p_i) = \mathbb{E}(\Delta_i^{k,\alpha}).$$

Applying the Chebyshev bound once more gives us

$$\begin{aligned} \mathbb{P}\left(\Delta_{i+1}^{k,\alpha'} \geq (k-1)n(p_i - \frac{1}{2}) + t\sqrt{n}\right) &\leq \mathbb{P}\left(\Delta_{i+1}^{k,\alpha'} \geq \mathbb{E}(\Delta_{i+1}^{k,\alpha}) + t\sigma\right) \\ &\leq \mathbb{P}\left(\Delta_{i+1}^{k,\alpha'} \geq \mathbb{E}(\Delta_{i+1}^{k,\alpha'}) + t\sigma\right) \leq \frac{1}{t^2}. \quad \square \end{aligned}$$

Building on the previous lemma, we can give the following probabilistic bound for the number of parties that have opinion 1 after i rounds.

Lemma 4.3. *Let $k \geq 2, \frac{k}{2} < \alpha \leq k$. Assume the system is in a roughly balanced state with $S_0 \leq \frac{n}{2} + f(n)$ for $f(n) = \sqrt{n \log n}$. Then for any $i \leq \frac{\log n}{c}$ it holds $S_i > \frac{n}{2} + (k+1)^i f(n)$ with probability at most $1/c$ for any $c \geq 1$.*

Proof. Let us first assume that each round the statement $\Delta_{i+1} \leq (k-1)(S_i - \frac{n}{2}) + f(n)$ is true and assess the overall progress we make at most towards a 1 consensus starting from S_0 .

$$S_1 = \Delta_1 + S_0 \leq (k-1)(S_0 - \frac{n}{2}) + f(n) + S_0 \leq \frac{n}{2} + (k-1)f(n) + 2f(n) = \frac{n}{2} + f(n)(k+1).$$

This satisfies the lemma for the first round and serves as our induction base. For the induction step we obtain

$$\begin{aligned} S_{i+1} = \Delta_{i+1} + S_i &\leq (k-1)(S_i - \frac{n}{2}) + f(n) + S_i && \text{(Lemma 4.2)} \\ &\leq (k-1)(k+1)^i f(n) + f(n) + \frac{n}{2} + (k+1)^i f(n) && \text{(induction hypothesis)} \\ &= \frac{n}{2} + k(k+1)^i f(n) + f(n) \\ &\leq \frac{n}{2} + (k+1)^{i+1} f(n) \end{aligned}$$

To conclude the proof, we compute the probability that the claim is true. By Lemma 4.2 we have that $\Delta_i > (k-1)(S_i - \frac{n}{2}) + \sqrt{n \log n}$ with probability at most $\frac{1}{\log n}$. Union bounding this for at most $\frac{\log n}{c}$ rounds, we obtain that the claim is true with probability at most $\frac{1}{c} \frac{\log n}{\log n} = \frac{1}{c}$. \square

We have all tools to deduce the lower bound for the number of rounds of Slush that is required even to get moderately close to a consensus state with some moderate probability.

Theorem 1. *For $k \geq 2$ and any $\frac{k}{2} < \alpha \leq k$ and sufficiently large n , running Slush for at most $\frac{\log n}{3 \log(k+1) \log \gamma}$ rounds there is a majority opinion with at least $\frac{n}{2} + \frac{n}{\gamma}$ parties with probability at most $\frac{1}{\log(k+1) \log \gamma}$ for any constant $\gamma \geq 2$.*

Proof. We will assume that there are initially no parties with opinion \perp (cf. the explanation of Slush in Section 2.2), which only strengthens the lower bound. Furthermore, we restrict ourselves to bound the probability of a 1-majority, since the same claim for a 0-majority holds by symmetry (and then we apply a union bound for the probability of one of either consensus happening).

The scenario for our lower bound for a 1 majority is the start state $S_0 \leq \frac{n}{2} + f(n)$ for $f(n) = \sqrt{n \log n}$ with $n > 16$, which conforms to the preconditions of Lemma 4.3. We choose the parameter c from Lemma 4.3 as $c = 3 \log(k+1) \log \gamma$. Then, by Lemma 4.3, after at most $i \leq \frac{\log n}{c}$ rounds it is $S_i > \frac{n}{2} + (k+1)^i f(n)$ with probability at most $\frac{1}{c}$. Furthermore we have

$$(k+1)^i f(n) \leq f(n) \cdot (k+1)^{\frac{\log n}{3 \log(k+1) \log \gamma}} = f(n) \cdot (n/\gamma)^{1/3} \leq \frac{n}{\gamma}.$$

In the last step we use that $f(n) \leq (n/\gamma)^{2/3}$ for sufficiently large n . Note that the same holds for obtaining a majority in the opinion 0 by starting in a state $\frac{n}{2} - f(n) \leq S_0 \leq \frac{n}{2} + f(n)$ due to symmetry. We obtain the claim of the theorem (for $k = 2\alpha - 1$) through a union bound on the number of rounds to either a 0-consensus or a 1-consensus with $\frac{2}{c} \leq \frac{1}{\log(k+1) \log \gamma}$. \square

We express the theorem above in a simpler, albeit weaker form.

Corollary 4.4. *For $k \geq 2$ and any $\frac{k}{2} < \alpha \leq k$, Slush takes $\Omega\left(\frac{\log n}{\log k}\right)$ rounds in expectation to reach a stable consensus (as defined in Definition 2.1).*

4.2 Upper Bound

We show how to use the structural insights about Slush with respect to parameters k and α to extend known upper bounds for the so called Median protocol, the 3-Majority protocol and the 2-Choices protocol. These are usually conceptualized for the case of multiple (> 2) opinions and are defined as follows.

Definition 4.5 (cf. [2]). *The Median protocol assumes some globally known total order among opinions. In each round, each party samples the opinion of two others and adopts the median among those two and its own.*

In the 3-Majority protocol, in each round, each party samples the opinion of three others and adopts the majority opinion, or picking a random opinion among the three in case of a tie.

In the 2-Choices protocol, in each round, each party samples the opinion of two others and then applies the 3-Majority rule, defaulting to its own opinion in case of a tie.

We make the following observation.

Remark 4.6. *Assume that all parties have initially only one of two opinions (i.e., the binary case, in particular, there are no parties with opinion \perp). Then the Median protocol, 2-choices protocol and Slush for $k = 2$ and $\alpha = 2$ are all equivalent. This is because in the binary case, in all three protocols a given party will switch its own opinion if and only if it samples two parties that both have a different opinion from its own. Under the same circumstances and for the same reason, the 3-Majority protocol is equivalent to Slush for $k = 3$ and $\alpha = 2$ (exploiting that there can never be a tie in the binary case).*

There has been extensive research on the dynamics of the Median, 2-Choices and 3-Majority protocol (Definition 4.5) and the techniques are for the most part analogous or at least quite similar if the number of opinions is kept constant. We have already established the lower bound of $\Omega\left(\frac{\log n}{\log k}\right)$, i.e., the *additional* speed-up one can gain by increasing the query size k diminishes very fast. Therefore, we do not deem it particularly worthwhile to show an upper bound that strictly improves on the $O(\log n)$ bound for the aforementioned cases.

Furthermore, it is *not* the scope of this paper to give detailed proofs of slight generalizations of those for the protocols from Definition 4.5. To keep this paper reasonably self-contained we showcase how these proofs generalize to Slush with arbitrary $k \geq 2$ and $\alpha = \lceil \frac{k+1}{2} \rceil$. We will give an extended proof sketch that shows how the existing proof techniques generalize to obtain the following theorem. For more details we refer to the according sources (in particular [3, 9]).

Theorem 2. *Let $k \geq 2$ and $\alpha = \lceil \frac{k+1}{2} \rceil$. Then Slush reaches a state where all but $n - O(\sqrt{n})$ have the same opinion in $O(\log n)$ rounds with high probability, even in the presence of a \sqrt{n} -bounded adversary.*

Proof Sketch. Our proof rests on the structure of the according proof for the $k = \alpha = 2$ and $(k, \alpha) = (3, 2)$ from [3, 9] for the binary case. The proof is divided into a constant number of phases. The phases range from the worst case where the distribution of opinions is roughly in an equilibrium, to the state of a stable consensus. We will show that it always takes at most $O(\log n)$ rounds to arrive in the corresponding next phase, even when starting from the worst case of an equilibrium. Due to symmetry we assume $S_i \geq 0$ w.l.o.g. We will at first make the main argument without considering the \sqrt{n} -bounded adversary and the opinion \perp (see Section 2.2) and argue why the proof holds for these cases at the end.

Phase 1: $\frac{n}{2} \leq S_i \leq \frac{n}{2} + c_1 \sqrt{n \log n}$. To lift S_i over the threshold of $\frac{n}{2} + c_1 \sqrt{n \log n}$ to the next phase, anti-concentration bounds are used. In particular, because $\alpha = \lceil \frac{k+1}{2} \rceil$ there is always a majority for either 0 or 1 in each query, thus we have that $X_{ij} \sim \mathcal{B}(q_i)$ is Bernoulli distributed with probability $q_i \approx p_i \approx \frac{1}{2}$ (since the system is close to balanced state). Hence, $S_i = \sum_{j=1}^n X_{ij}$ is a sum of independent, identically distributed Bernoulli variables. By the Central Limit Theorem, for sufficiently large n the variable S_i approximates a normal distribution with deviation of $\Omega(\sqrt{n})$ around the expectation $\mathbb{E}(S_i) \approx \frac{n}{2}$. The property of the normal distribution implies that there is a constant probability for $|S_i - \frac{n}{2}| \geq c\sqrt{n}$ in a single round. Applying concentration (Chernoff) bounds and considering that by symmetry we are allowed to escape Phase 1 in either direction, one can ensure $|S_i - \frac{n}{2}| > c_1 \sqrt{n \log n}$ after $O(\log n)$ rounds w.h.p. (cf. [9] for more details).

Phase 2: $\frac{n}{2} + c_1 \sqrt{n \log n} < S_i \leq \frac{n}{2} + \frac{n}{c_2}$. We explain the idea in a style that is akin to the proof by [3], which shows the argument for $(k, \alpha) = (3, 2)$ and then extend it to the general case. Note that expected progress is $\mathbb{E}(\Delta_i) = n \cdot \delta^{3,2}(p_i)$. The next step is to show that in the Phase 2 interval of S_i (here $c_2 = 3$ in general, c_2 depends on k, α) the function $\delta^{3,2}(p_i)$ can be lower bounded by a linear function of constant positive gradient through the point $(\frac{1}{2}, 0)$ (see Figure 1a for a visual representation). This implies that in each round the expected progress increases linearly in the progress that was made in the previous round. Intuitively, this corresponds to a situation of “compounding interest” in expected progress $\mathbb{E}(\Delta_i)$ with each round i , i.e., exponential growth in i . Hence, it takes at most $O(\log n)$ rounds until $\Psi_i > \frac{n}{c_2}$.

The caveat is, that this only works if we can guarantee that the system makes progress that is at least some constant fraction of the expected progress. Due to randomness the expected progress could be undershot or the system could even backslide towards the equilibrium. To show that this does not happen w.h.p., we exploit the intuition that in Phase 2 the system is already relatively far advanced into a majority, such that, the expected progress $\mathbb{E}(\Delta_i)$ exceeds the standard deviation of Δ_i . Concretely, one can use concentration bounds and union bounds, to guarantee a constant fraction of the expected progress $\mathbb{E}(\Delta_i)$ w.h.p. for each round in this phase.

By Lemma 3.5 we have that $\delta^{3,2}(p_i) = \delta^{2,2}(p_i)$, so the argument also extends to the 2-Choices protocol. Finally, Lemma 3.6 shows that the expected progress $\delta^{k,\alpha}$ for $k \geq 4$ and $\alpha = \lceil \frac{k+1}{2} \rceil$ dominates that for

$k \in \{2, 3\}$, and consequently the argument above applies for Slush in general.

Phase 3: $\frac{n}{2} + \frac{n}{c_2} < S_i \leq \frac{n}{2} + \frac{n}{c_3}$. This is arguably the simplest phase, since the system is a constant fraction of parties from both the equilibrium state and the consensus state. This implies that $\delta^{2,2}(p_i)$ ($= \delta^{3,2}(p_i)$ by Lemma 3.5) can be lower bounded by a constant (cf. Figure 1a), therefore a constant fraction of all parties will switch to the majority opinion in expectation. It is not hard to show that this also holds w.h.p. and by Lemma 3.6, also for any $k \geq 4$ and $\alpha = \lceil \frac{k+1}{2} \rceil$.

Phase 4: $\frac{n}{2} + \frac{n}{c_3} < S_i \leq n - c_4\sqrt{n}$. Although the expected progress $\delta^{2,2}(p_i)$ tends to 0 as p_i approaches 1 (see Figure 1a), one can apply a similar argument as in Phase 2 but “backwards”. In particular, in the corresponding interval of S_i the expected progress given by $\delta^{2,2}(p_i)$ can be lower bounded by a linear function with constant negative slope through the point $(1, 0)$ (see Figure 1a for a visual confirmation). This implies that the expected number of parties holding the minority opinion shrinks by a constant fraction in each round (cf. [3]). Thus it takes at most $O(\log n)$ rounds until S_i passes the threshold $n - c_4\sqrt{n}$ given that the progress is at least a constant fraction of the expected progress. As in Phase 2, the latter can be guaranteed w.h.p. using concentration bounds. The argument generalizes to $k \geq 3$ and $\alpha = \lceil \frac{k+1}{2} \rceil$ by Lemmas 3.5 and 3.6.

\sqrt{n} -Bounded Adversary: The optimal strategy of the adversary to avoid a consensus is to flip \sqrt{n} parties of the majority to the minority opinion. The main argument is that the ability of adversary to influence S_i is asymptotically not more than the standard deviation of S_i . In Phase 1 the random deviation from the expectation is a desired effect to lift the system out of an equilibrium and one can show that w.h.p., the given adversary can not inhibit this. In the subsequent phases, the random deviation is undesired as it may reduce the progress below its expectation. The intuition is that since the influence of the adversary is actually less than the standard deviation, we can deal with both the standard deviation and the effect of the adversary as before, by adjusting constants in the running time.

Dealing with parties with opinion \perp : It remains to argue that the asymptotic running time does not change if we introduce the special opinion \perp , which is relatively straight forward. Whenever a party j_1 with opinion \perp receives a query for an opinion by another party j_2 then j_1 adopts b (see Algorithm 2). Let $n_{0,1}$ and n_{\perp} be the number of parties that have opinion 0, 1 or \perp respectively ($n = n_{0,1} + n_{\perp}$). As long as $n_{0,1} \leq n_{\perp}$ it can be shown that $n_{0,1}$ doubles every $O(1)$ rounds w.h.p. Similarly, if $n_{0,1} \geq n_{\perp}$, the number of parties halves every $O(1)$ rounds w.h.p. Ultimately, this implies that the opinion \perp will die out after $O(\log n)$ rounds. \square

We can translate the above result into the notion of concentration with all but negligible probability (Definition A.3) by adding a factor of β to the running time that gives more control over the level of security in particular for small n (see Lemma A.5 and Remark A.6 for the details). Specifically, the corollary conforms to Definition 2.4, as the runtime is polynomial in β .

Corollary 4.7. *Let $k \geq 2$ and $\alpha = \lceil \frac{k+1}{2} \rceil$. Then Slush reaches a stable consensus in $O(\log n + \beta)$ rounds with all but negligible probability (with respect to β), even in the presence of a \sqrt{n} -bounded adversary.*

5 Dynamics of Snowflake and Snowball

In this section we are going to extend the lower bound for Slush derived in Section 4 to the Snowflake and Snowball protocols. Note that the quantities $S_i, p_i, \Delta_i, \delta_i$ can be defined the same as in Slush, see Section 3, since the variables only depend on the opinion attribute of parties, which is present in all three protocols. However, the actual (expected) changes of these quantities in this section can and will differ from those in Slush. We will denote these quantities with superscripts (slush, flake, ball) in case we compare them over protocols (but avoid this whenever possible). Moreover, we condition the results of this section on the assumption that no node decides (finalizes) their opinion, before the system reaches a stable majority and

consider the repercussions at the end.

Snowball, as explained in Section 2.2, augments the consensus mechanism from Slush with the concept of *confidence* associated to the current value, which influences the decision of a party to change its opinion. In a nutshell, a node changes opinion in the Snowball protocol when the cumulative number of queries with majority for the new opinion exceeds that for the old opinion. In the Slush protocol, the variable S_i was sufficient to describe the expected progress required to predict the evolution of the system, which is not the case in Snowball anymore, since aforementioned confidence levels play a crucial role.

Definition 5.1. Define the set L_i^c to be the set of parties in round i such that $\text{cnt}(1) - \text{cnt}(0) = c$ (where $\text{cnt}(b)$ is the number of queries of a given party that had a majority of opinion b). We further divide the set L_i^0 in two subsets $L_i^{0,0}$ and $L_i^{0,1}$. Parties in L_i^0 ($L_i^{0,1}$) that have opinion 0 (1) belong to $L_i^{0,0}$ ($L_i^{0,1}$).

The variable S_i can be reconstructed as follows: $S_i = |L_i^{0,1}| + \sum_{c>0} |L_i^c|$. Given a round $i > 0$, the set of parties \mathcal{N} is contained in $\bigcup_{c=-i}^i L_i^c$ as the end of round i , since every party performed i queries by the end of round i , thus the difference in counts c is bounded between $-i$ and i .

Remark 5.2. Consider the collection $\mathcal{L}_i := \{L_i^c\}_{c=-i}^i \cup \{L_i^{0,0}, L_i^{0,1}\}$ of disjoint sets. The evolution of the system in the next round $i + 1$ can now be described using this set \mathcal{L}_i . After a query is performed a party in L_i^c moves to set L_i^{c+1} if the query had a majority for 1, to L_i^{c-1} if the query had a majority for 0, or L_i^c if the query had no majority. The only parties that can change value after round i are the parties contained in L_i^0 .

Definition 5.3. For $i \geq 1$, we define the absolute progress as $\Delta_i := S_i - S_{i-1} = |L_i^{0,1}| - |L_{i-1}^{0,1}| + \sum_{c=1}^i (|L_i^c| - |L_{i-1}^c|)$, i.e., the number of parties with 1 in their view in round i minus the number of parties with 1 in their view in round $i - 1$. As before, we define the expected relative progress as $\delta_i := \mathbb{E}(\Delta_i)/n$.

The following Lemma shows that in Snowball, Δ_i is only affected by parties migrating from $L_{i-1}^{0,1}$ or $L_{i-1}^{0,0}$ in round $i-1$ to L_i^{-1} or L_i^1 in round i , respectively.

Lemma 5.4. The absolute progress can be expressed as $\Delta_i = |\Lambda_i^1| - |\Lambda_i^0|$, where $\Lambda_i^0 := L_i^{-1} \cap L_{i-1}^{0,1}$ and $\Lambda_i^1 := L_i^1 \cap L_{i-1}^{0,0}$.

Proof. Recall $\Delta_i = |L_i^{0,1}| - |L_{i-1}^{0,1}| + \sum_{c=1}^i (|L_i^c| - |L_{i-1}^c|)$ and consider a party j . We distinguish the following cases:

- If $j \in L_{i-1}^c$ for $c < 0$, then Remark 5.2 guarantees that $j \in L_i^{c-1} \cup L_i^c \cup L_i^{c+1} \cup L_i^{0,0}$. None of the previous sets are involved in the definition of Δ_i , thus the value of Δ_i is independent from party j .
- If $j \in L_{i-1}^c$ for $c > 0$, then Remark 5.2 guarantees that $j \in L_i^{c-1} \cup L_i^c \cup L_i^{c+1} \cup L_i^{0,1}$. In the definition of Δ_i , the terms $|L_i^{c'}|$ for $c' \in \{c-1, c, c+1, \{0, 1\}\}$ and $|L_{i-1}^c|$ appear with opposite signs. Thus, the contribution of party j cancels out.
- If $j \in L_{i-1}^{0,0}$, then Remark 5.2 guarantees that $j \in L_i^{-1} \cup L_i^{0,0} \cup L_i^1$. If $j \in L_i^{-1}$ or $j \in L_i^{0,0}$, the terms $L_{i-1}^{0,0}$, $j \in L_i^{0,0}$, and L_i^{-1} do not appear in the definition of Δ_i . If $j \in L_i^1$, then $j \in L_{i-1}^{0,0} \cap L_i^1 = \Lambda_i^1$ and j contributes with $+1$ to Δ_i .
- If $j \in L_{i-1}^{0,1}$, then Remark 5.2 guarantees that $j \in L_i^{-1} \cup L_i^{0,0} \cup L_i^1$. If $j \in L_i^1$ or $j \in L_i^{0,1}$, the term $L_{i-1}^{0,1}$ appears with coefficient -1 , whereas the terms $j \in L_i^1$ and $j \in L_i^{0,1}$ appear with coefficient 1. Thus, their contribution cancel out. If $j \in L_i^{-1}$, then $j \in L_{i-1}^{0,1} \cap L_i^{-1} = \Lambda_i^0$ the term $j \in L_{i-1}^{0,1}$ appears with coefficient -1 , whereas the term $j \in L_i^{-1}$ does not appear.

We conclude that $\Delta_i = |\Lambda_i^1| - |\Lambda_i^0|$. □

An interesting interpretation of Lemma 5.4 is the following. Since the parties contained in the set L_i^0 are the only parties that can change their opinion, the expected progress of Snowball in a given round is the same as the expected progress of Slush restricted to the parties in L_i^0 . We formalize this intuition in the following lemma.

Lemma 5.5. *The expected absolute progress of Snowball is at most as high as in Slush, i.e., $\delta_i^{\text{ball}} \leq \delta_i^{\text{slush}}$.*

Proof. Lemma 5.4 states that only parties in L_{i-1}^0 can modify the state S_i . Given a party $j \in L_{i-1}^{0,0}$ (respectively $j \in L_{i-1}^{0,1}$) The probability that j changes to value 1 (respectively 0) is given by

$$\begin{aligned}\mathbb{P}(X_{ij} = 1 \mid j \in L_{i-1}^{0,0}) &= \mathbb{P}(Y_{ij} \geq \alpha) = \sum_{\ell=\alpha}^k \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell}. \\ \mathbb{P}(X_{ij} = 0 \mid j \in L_{i-1}^{0,1}) &= \mathbb{P}(Y_{ij} \leq k - \alpha) = \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell},\end{aligned}$$

where the quantities X_{ij}, Y_{ij} are defined the same as in Section 3, i.e., we deal with exactly the same probabilities as in Lemma 3.1. Consequently, we are able to apply Lemma 3.1 and conclude that the expected rate of progress is $\delta_i^{\text{ball}} = \delta(p_i)$ when restricted to L_{i-1}^0 .

The latter is an important caveat, since what changes in Snowball is the set of parties on which the expected rate of progress is applied. Then the way the expected absolute progress of Snowball and Snowflake relate to each other is given as follows

$$\mathbb{E}(\Delta_i^{\text{ball}}) = |L_i^0| \cdot \delta_i^{\text{ball}} = |L_i^0| \cdot \delta(p_i) = |L_i^0| \cdot \delta_i^{\text{slush}} = \frac{|L_i^0|}{n} \cdot \mathbb{E}(\Delta_i^{\text{slush}}) \stackrel{|L_i^0| \leq n}{\leq} \mathbb{E}(\Delta_i^{\text{slush}}).$$

Dividing by n on both sides and using $\delta_i^{\text{ball}} = \frac{\mathbb{E}(\Delta_i^{\text{ball}})}{n}$, $\delta_i^{\text{slush}} = \frac{\mathbb{E}(\Delta_i^{\text{slush}})}{n}$ (see start of Section 3), yields the desired result. \square

Lemma 5.5 states that the expected progress of the Snowball protocol is upper-bounded by the expected progress of the Slush protocol. We conclude that the expected number of rounds even to reach majority of a constant fraction of nodes of one opinion of the Snowball protocol is lower-bounded by the Slush protocol, if no node decides prematurely. Note that the same is clearly true for as Snowflake which is essentially equal to Slush if no node decides prematurely.

Corollary 5.6 (cf. Corollary 4.4). *For $k \geq 2$ and any $\frac{k}{2} < \alpha \leq k$, Snowball and Snowflake take $\Omega\left(\frac{\log n}{\log k}\right)$ rounds in expectation to reach a state of stable consensus for any constant $\gamma \geq 2$, assuming that nodes do not decide before such a state is reached.*

Note that since the decision mechanism in Snowflake and Snowball implies that no node can decide before β rounds have passed the corollary implies a lower bound of $\Omega\left(\min\left(\frac{\log n}{\log k}, \beta\right)\right)$ rounds. Furthermore, we will see in Section 6 that the dependence of the runtime on β behaves much worse than $\Omega(\beta)$ as the adversary can exploit the decision mechanism to delay a decision *super-polynomially* in β .

6 Security of Snowflake and Snowball

We show that in Snowflake and Snowball, has a vulnerability towards an adversary that intends to delay consensus (as defined in Definition 2.2). In particular, there might is an unfavorable trade-off between confidence of success with and latency. In particular, the mechanic that Snowflake and Snowball protocols use introduces a security parameter β to control the probability of failure of obtaining a consensus (according to Definition 2.2). We show that this mechanism to make decision allows an adversary to delay the decision of any given party when using the consensus mechanisms of Snowflake and Snowball for a super-polynomial number of rounds in β . This is independent of the current state of the system, i.e., the claim is true even if the system is in a state of a stable consensus (see Definition 2.1) and is true for a weaker notion of the F -bounded adversary from Definition 2.3 for a small F .

Definition 6.1. A weak F -bounded adversary controls up to F undecided parties whose state (opinion) it can set once each round. We call these influenced parties and in particular we assume that the adversary can reset any decision on some opinion made by those.

We start by giving a lower bound for the probability that some party samples a majority of influenced parties.

Lemma 6.2. The probability that a random sample of k parties contains at least α that are influenced by a weak F -bounded adversary is at least $\left(\frac{F}{n}\right)^k$.

Proof. Recall that we nodes are sampled uniformly at random with repetition. We can lower bound q_j with the probability that j samples at least α Byzantine parties as follows

$$q_j = \sum_{\ell=\alpha}^k \binom{k}{\ell} \left(\frac{F}{n}\right)^\ell \left(\frac{n-F}{n}\right)^{k-\ell} \geq \sum_{\ell=k}^k \binom{k}{\ell} \left(\frac{F}{n}\right)^\ell \left(\frac{n-F}{n}\right)^{k-\ell} = \left(\frac{F}{n}\right)^k. \quad \square$$

Note that even though the probability above decreases with k , the parameter k is considered a small, constant sized tuning parameter [16] and is not a proper security parameter, particularly since the message complexity scales in $\Omega(kn)$.

Interestingly, the lemma shows that even in a stable consensus (according to Definition 2.1), i.e., where almost all nodes share the same opinion, even a weak adversary can create a small but inherent “background noise”, i.e., an expected fraction $\left(\frac{F}{n}\right)^k$ of all parties can be reverted by the adversary to the minority opinion each round, because it gains a majority in a sample.

This situation is what the security mechanic of Snowflake and Snowball protocols is intended for as it makes parties decide and finalize an opinion by introducing an according mechanism with a security parameter β . One condition for some party to decide an opinion, is that it must have at least β consecutive queries with an α -majority of the same opinion, see Section 2.2 or Section C for detailed pseudocode. (Note that Snowball imposes an additional condition for parties to decide based on the history of queries, which, however, only delays the decision of a party down even further).

The idea behind this mechanism is to reduce the probability that an adversary can make some party accept the *minority* opinion, since sampling the minority opinion β times in a row has a probability that is negligible with respect to β . This mechanism is flawed in the sense that even an adversary that influences just a single party can abuse it to introduce a delay to the decision of a party that scales badly in β .

Lemma 6.3. In the Snowflake and Snowball protocols, there exists a value $c > 1$ which is constant in β , such that a weak F -bounded adversary (Def. 6.1, for any $F \geq 2$) can ensure that the probability that a given party decides within $c^{\beta-1}/2$ queries (rounds) is at most $4/c^{2(\beta-1)}$. To enforce this, the adversary needs no information on the current state of the network.

Proof. In the Snowflake and Snowball protocols, a variable $cnt \geq 1$ maintains the length of the most recent sequence of consecutive queries which all had a majority of the same opinion b (see Section C). To decide on some opinion b (which we call “success” in the following), it is necessary that $cnt \geq \beta$.

Consider the following adversary strategy, where it splits its influenced parties into two roughly equally sized groups whose opinions it sets to 0 and 1, respectively (here we need $F \geq 2$, round group sizes if necessary). We will now compute the probability q_β of the event E_β (failure in the sequence of length β) that some given sequence of β queries of some party j contains either a query which had no opinion or at least one of each opinion, 0 or 1.

Assume, that the first opinion in such a sequence had a majority of 1. This is w.l.o.g., firstly, since the case of no majority in the first query already satisfies the condition above (by the law of total probability, neglecting this case gives us a lower bound for q_β). Secondly, because the case where a sequence starts

with a 0 majority is analogous, if we treat the two groups of the adversary as two separate adversaries of size at least $F' \geq \lfloor \frac{F}{2} \rfloor \geq 1$.

This means q_β is lower bounded by the probability that the next $\beta - 1$ queries have a majority from the F' -bounded adversary. Thus $q_\beta \geq (1 - q_j)^{\beta-1}$, where q_j is the lower bound of the probability that at least α Byzantine parties are sampled in a given query from Lemma 6.2. Note that q_j is lower bounded by a non-zero value that is independent of β .

The number of queries crucially depends on the probability q_β of failure of deciding in a sequence of length β . To measure the number of queries until success we introduce a r.v. Z_j that lower bounds the number counter resets because event E_β occurs (with probability q_β). This means that Z_j follows a geometric distribution $Z_j \sim \mathcal{G}(q_\beta)$, describing the number of failures (i.e., counter resets) until success (deciding on a value).

Note that Z_j is a conservative lower bound for the overall number of queries that j has to make. First, we significantly underestimate the probability of failure q_β . Second, we consider only counter resets caused by sampling a Byzantine majority. Third, we do not account for queries (that increase the counter) in between counter resets.

The expectation of $Z_j \sim \mathcal{G}(q_\beta)$ is $\mu = 1/q_\beta$ and the variance is $\sigma := (1 - q_\beta)/q_\beta^2$. Then the probability that X is at most half its mean can be bounded with the Chebyshev inequality:

$$\begin{aligned} \mathbb{P}(X \leq \frac{\mu}{2}) &\leq \mathbb{P}(|X - \mu| \geq \frac{\mu}{2}) \\ &= \mathbb{P}(|X - \mu| \geq k \cdot \sigma) && \text{where } k := \frac{\mu}{2\sigma} \\ &\leq \frac{1}{k^2} = \frac{4(1-q_\beta)^2}{q_\beta^2} \leq \frac{4}{q_\beta^2}. \end{aligned}$$

Let $c := (1 - p_j)^{-1} (> 1)$, thus $q_\beta \geq c^{-(\beta-1)}$. The claim follows from the fact that $\mu = 1/q_\beta = c^{\beta-1}$ and $\mathbb{P}(X > \frac{\mu}{2}) = 1 - \mathbb{P}(X \leq \frac{\mu}{2}) \geq 1 - 4/c^{2(\beta-1)}$. \square

Using the lemma above, we show that Snowflake and Snowball cannot satisfy Definition 2.4, which states the conditions for a mechanism that provides a decent trade-off between security and performance. Specifically, the following theorem shows that having consensus with all but negligible probability w.r.t., β and a polynomial runtime in β are mutually exclusive.

Theorem 3. *In the Snowflake and Snowball protocol with a weak F -bounded adversary ($F \geq 2$) the following properties are mutually exclusive*

- *The protocol ensures consensus with all but negligible probability with respect to β (cf. Def. 2.2).*
- *Parties decide with less than $\pi(\beta)$ queries with all but negligible probability with respect to β , for any fixed polynomial π .*

Note that this holds even when the definition of consensus is restricted to those parties which are not influenced by the adversary.

Proof. By Lemma 6.3, the probability that at least $c^{\beta-1}/2$ queries are required is at least $1 - 4/c^{2(\beta-1)}$. Intuitively, this means that the Snow protocols are likely to fail if not enough queries are used, unless β is small. If, on the other hand, β is indeed small, then there is a non-negligible chance that some party decides the minority value, thus prohibiting agreement.

Formally, assume that β is a security parameter as in Definition 2.4. Consider the threshold value $B := \frac{2}{\log c} + 1$ (constant in β). Consider the case that β is small, i.e., $\beta \leq B$. By Lemma 6.2 there is a non zero probability q_j (which does not depend on β) that an adversary controls at least α parties that have been queried by some party j . Then the probability that the adversary can make j decide on the minority value is at least q_j^β , which is constant in β , i.e., non-negligible.

Now consider $\beta > B$. Then the probability that the protocol fails is more than $3/4$ if less than $c^{\beta-1}/2$ queries are conducted because a party did not yet decide. To show the mutual exclusivity for β above this threshold, assume that the protocol decides with all but negligible probability. Then (much) more than $c^{\beta-1}/2$ queries are required by Lemma 6.3, which is larger than $\pi(\beta)$ for sufficiently large β . \square

7 Reconciling Security and Fast Consensus

Theorem 3 shows that the Snowflake and Snowball protocols cannot achieve consensus within a polynomial number of rounds with all but negligible probability with respect to security parameter β , due to the termination condition. We propose in Algorithm 5 a modification of the Slush protocol that we call *Blizzard* and which incorporates confidence levels as a termination criterion. Importantly, Blizzard leaves the basic dynamics of the Slush protocol intact, in particular Blizzard neglects the mechanic of Snowball, where parties change their current opinion depending not only on the current but also on past queries, which is in general detrimental for the time it takes to converge to a stable consensus as shown in Section 5.

This modification is arguably simpler than the corresponding mechanisms in Snowball and works as follows. Each party maintains two counters, which track the total number of queries that contained at least α of opinion 0 or 1, respectively (an α -majority). A decision in favor of one opinion is made if the corresponding counter has a decisive lead over the other. The lead that is required is of the order $O(\log n + \beta)$ and we show that this also corresponds to the number of rounds until consensus is established with all but negligible probability (w.r.t. β).

The idea is that within the given time frame, the network will reach a state of a stable consensus, and it will remain close to this state for a sufficiently long time such that each party can establish a lead in the counter for the opinion which is in the majority. Furthermore, unanimity is ensured because the required lead is large enough such that no party can accidentally make a “premature” decision, i.e., reaching the threshold even when no stable majority has been established yet.

Note that as time progresses and given an adversary that controls at least α parties, there is always a small but non-zero chance that a system reverts from a state of stable consensus and even switches majorities. However, we show that once the system is in a stable consensus, it will not “slide back” too far within a given time frame, i.e., one opinion retains an overwhelming majority for a sufficient amount of time, even in the presence of an adversary. We start with a lemma about the probabilities to sample an α -majority of an opinion given that one opinion has a majority.

Lemma 7.1. *Let $S_i \geq \frac{15n}{16}$ and $\alpha = \lceil \frac{k+1}{2} \rceil$. Then $\mathbb{P}(Y_{ij} \geq \alpha) \geq p_i$ and $\mathbb{P}(Y_{ij} \leq k - \alpha) \leq 4(1 - p_i)^2$.*

Proof. Note that $1 - p_i \leq \frac{1}{16}$. From the choice of α , it follows that there is always an α majority for either 0 or

1. Thus, it suffices to show that $\mathbb{P}(Y_{ij} \leq k-\alpha) \leq 4(1-p_i)^2$ the other inequality follows due to $4(1-p_i)^2 \leq 1-p_i$.

$$\begin{aligned}
\mathbb{P}(Y_{ij} \leq k-\alpha) &= \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} p_i^\ell (1-p_i)^{k-\ell} \leq \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} (1-p_i)^{k-\ell} \\
&\leq \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} (1-p_i)^\alpha = (1-p_i)^\alpha \sum_{\ell=0}^{k-\alpha} \binom{k}{\ell} \\
&\leq (1-p_i)^\alpha \sum_{\ell=0}^{\lfloor k/2 \rfloor} \binom{k}{\ell} \leq (1-p_i)^\alpha 2^{k-1} \\
&\stackrel{k \leq 2\alpha-1}{\leq} (1-p_i)^\alpha 2^{2\alpha-2} = (1-p_i)^2 (1-p_i)^{\alpha-2} 2^{2\alpha-2} \\
&\leq (1-p_i)^2 \left(\frac{1}{16}\right)^{\alpha-2} 2^{2\alpha-2} = (1-p_i)^2 \left(\frac{1}{2}\right)^{4\alpha-8} 2^{2\alpha-2} \\
&= (1-p_i)^2 \left(\frac{1}{2}\right)^{2\alpha-6} = 4(1-p_i)^2 \left(\frac{1}{2}\right)^{2\alpha-4} \stackrel{\alpha \geq 2}{\leq} 4(1-p_i)^2 \quad \square
\end{aligned}$$

The next lemma shows that once the network is in a stable consensus state, it will very likely conserve a majority for a certain time frame.

Lemma 7.2. *Let $s, t \geq 1$ with $s \leq \frac{t}{2}$ and $t \leq \sqrt{n}/16$. Assume the number of parties with opinion 1 is currently $S_0 \geq n - s\sqrt{n}$. Then $S_i \geq n - t \cdot \sqrt{n}$ for at least $i \leq T := \min\left(\frac{\sqrt{n}}{32t}, \frac{t}{4}\right)$ rounds with probability at least e^{-t^2} . This holds even in the presence of a \sqrt{n} -bounded adversary.*

Proof. We assume that the condition $p_i \geq 1 - t/\sqrt{n}$ is true and show for how long it can be maintained starting from round $i = 0$ in state S_0 . The claim is initially true due to $s \leq \frac{t}{2}$. Since $t \leq \sqrt{n}/16$, this assumption entails $p_i \geq 15/16$, which satisfies the precondition of Lemma 7.1.

Let B_{ij} the indicator variable that some party changes its opinion from 1 to 0 in round i , i.e., $B_{ij} = 1$ if that is the case, $B_{ij} = 0$ else. Then $B_i := \sum_{j=1}^n B_{ij}$ is the "backslide", i.e., the number of nodes that change opinion from 1 to 0, which is a sum of independent, identically distributed Bernoulli variables. We obtain

$$\mathbb{E}(B_i) = \sum_{j=1}^n \mathbb{E}(B_{ij}) = n\mathbb{P}(X_{ij}=0 \mid X_{i-1,j}=1) \leq \sum_{j=1}^n \mathbb{P}(Y_{ij} \geq \alpha) = \sum_{j=1}^n (1-p_i) \stackrel{\text{Lem.7.1}}{\leq} 4n \left(\frac{t}{\sqrt{n}}\right)^2 = 4t^2.$$

We will assume that in each round $i \leq T$ the backslide is $B_i \leq 2\mathbb{E}(B_i) = 8t^2$ and show that this is the case with high probability later. Considering this upper bound for the backslide, that the adversary can change the opinion of at most \sqrt{n} parties in each round and exploiting the bounds for s and T , then in round $i = T$ we obtain

$$\begin{aligned}
S_T &\geq n - s \cdot \sqrt{n} - T \cdot 8t^2 - T \cdot \sqrt{n} \\
&\geq n - \frac{t}{2} \cdot \sqrt{n} - \frac{\sqrt{n}}{32t} \cdot 8t^2 - \frac{t}{4} \cdot \sqrt{n} \\
&= n - \frac{t}{2} \cdot \sqrt{n} - \frac{t}{4} \cdot \sqrt{n} - \frac{t}{4} \cdot \sqrt{n} = n - t \cdot \sqrt{n}
\end{aligned}$$

In particular, this also implies $S_i \geq n - t \cdot \sqrt{n}$ thus $p_i \geq 1 - t/\sqrt{n}$ for all $i \leq T$ since $B_i \leq 2\mathbb{E}(B_i)$ for all $i \leq T$. It remains to compute the probability for the latter. We apply a Chernoff Bound (given in Lemma A.9) and

obtain $\mathbb{P}(B_i > 2\mathbb{E}(B_i)) \leq \exp(-\frac{4t^2}{3})$. A union bound shows that this holds for all $i \leq T$ with probability

$$\begin{aligned} \mathbb{P}\left(\bigcup_{i \leq T} B_i \leq 2\mathbb{E}(B_i)\right) &= 1 - \mathbb{P}\left(\bigcap_{i \leq T} B_i > 2\mathbb{E}(B_i)\right) \geq 1 - \sum_{i \leq T} \mathbb{P}(B_i > 2\mathbb{E}(B_i)) \\ &\geq 1 - T \cdot \exp\left(-\frac{4t^2}{3}\right) \geq 1 - \frac{t}{4} \cdot \exp\left(-\frac{4t^2}{3}\right) \\ &\geq 1 - \exp\left(\frac{t}{4}\right) \cdot \exp\left(-\frac{4t^2}{3}\right) = 1 - \exp\left(\frac{t}{4} - \frac{4t^2}{3}\right) \stackrel{t \geq 1}{\geq} 1 - e^{-t^2}. \quad \square \end{aligned}$$

While Lemma 7.2 captures the stability of a state of almost consensus in a more general way, we will use it in the following form, by specifying some parameters.

Lemma 7.3. *Let $s \geq 1$ be a constant and assume the number of parties with opinion 1 is $S_0 \geq n - s\sqrt{n}$. For an arbitrary constant $\beta \geq s/2$, let $T \geq \beta$ and $T \in o(n^{1/4})$. Then $S_i \geq \frac{15n}{16}$ for at least T rounds with probability at least $1 - e^{-\beta^2}$. This holds for sufficiently large n even with a \sqrt{n} -bounded adversary.*

Proof. Let $t := 4T$, i.e., $T \leq \frac{t}{4}$ by definition. Since $T \in o(n^{1/4})$ we have $t \leq \sqrt{n}/16$ for large enough n . Moreover, we have $\sqrt{n}/32t \in \Omega(\sqrt{n}/T) \subseteq \Omega(n^{1/4})$ and therefore $T \leq \sqrt{n}/32t$ for sufficiently large n . Furthermore, $s \leq 2\beta \leq 2T \leq \frac{t}{2}$. This satisfies all the requirements for Lemma 7.2 and we conclude that $S_i \geq n - t\sqrt{n} \geq 15n/16$ with probability at least $1 - e^{-t^2} = 1 - e^{-16T^2} \geq 1 - e^{-\beta^2}$. \square

We will now use the stability properties of a network that is in a state of stable consensus (Definition 2.1) to show that Blizzard ensures consensus (Definition 2.2) with all but negligible probability after $O(\beta + \log n)$ rounds. Recall that Blizzard essentially corresponds to running an instance of Slush, where each node maintains counters that track how often an α -majority was observed for opinion 0 and 1, respectively. If the difference in counters reaches a (sufficiently large) threshold τ , then a decision for the opinion with the larger counter is made. In the following theorem we use a variable running time for Slush (due to the fact that as speedups up to $\log k$ over our upper bound can not be excluded, see Theorems 1 and 2).

Theorem 4. *Algorithm 5 (Blizzard) with a threshold $\tau := 2T_{\text{Slush}}$ ensures consensus with all but negligible probability (w.r.t. β) after at most $7T_{\text{Slush}}$ rounds, assuming that T_{Slush} is the number of rounds until Slush reaches a state where at least $n - O(\sqrt{n})$ parties have the same opinion and T_{Slush} is a sufficiently large multiple of β . This holds even with a \sqrt{n} -bounded adversary.*

Proof. Let $s \geq 1$ such that at least $S_0 \geq n - s\sqrt{n}$ parties have opinion 1 (w.l.o.g.) with all but negligible probability after at most T_{Slush} rounds. Note that we have $T_{\text{Slush}} \in O(\beta + \log n)$ by Theorem 2.

Purely syntactically, we reinitialize the round counter to $i = 0$ when we reach such a state S_0 for the first time. Starting from round $i = 0$, by Lemma 7.3 we have $p_i \geq \frac{15}{16}$ for at least $T \in o(n^{1/4})$ rounds, even with a \sqrt{n} -bounded adversary.

Let $Z_{j,0}$ and $Z_{j,1}$ be the number of times party j observed an α -majority of 0 and 1 respectively in any query made from round $i = 0$ to round $i = T$. For the expectations we have.

$$\begin{aligned} \mathbb{E}(Z_{j,0}) &= \mathbb{P}(Y_{ij} \leq k - \alpha) \cdot T \stackrel{\text{Lem.7.1}}{\leq} 4(1 - p_i)^2 \cdot T \leq 4 \cdot \frac{1}{16^2} \cdot T \leq \frac{T}{16}. \\ \mathbb{E}(Z_{j,1}) &= \mathbb{P}(Y_{ij} \geq \alpha) \cdot T \stackrel{\text{Lem.7.1}}{\geq} p_i \cdot T \geq \frac{15T}{16}. \end{aligned}$$

Note that the variables $Z_{j,0}$ and $Z_{j,1}$ can be seen as sums of independent Bernoulli variables, which allows us to apply the following Chernoff bounds.

$$\begin{aligned} \mathbb{P}(Z_{j,0} \geq (1 + 1) \cdot \frac{T}{16}) &\leq \exp\left(-\frac{T}{3 \cdot 16}\right) \stackrel{T \geq 48\beta}{\leq} e^{-\beta} \\ \mathbb{P}(Z_{j,1} \leq (1 - \frac{1}{3}) \cdot \frac{15T}{16}) &\leq \exp\left(-\frac{15T}{9 \cdot 16}\right) \stackrel{T \geq 48\beta/5}{\leq} e^{-\beta}. \end{aligned}$$

This implies that $Z_{j,0} < \frac{T}{8}$ and $Z_{j,1} > \frac{5T}{8}$ with all but negligible probability given that $T \geq 48\beta$. Let $D_j(T) := Z_{j,1} - Z_{j,0}$ be the difference in the counters after T rounds, then we have $D_j(T) \geq \frac{5T}{8} - \frac{T}{8} = \frac{T}{2}$ after T rounds with all but negligible probability. We can now show that the properties of consensus in Definition 2.2 are met after $7T_{\text{Slush}} + O(\beta)$ rounds with all but negligible probability.

Termination. We have to show that for any given party j the absolute value of the difference in the counters $D'_j = \text{cnt}[1] - \text{cnt}[0]$ reaches the threshold τ eventually so that it decides (cf. Algorithm 5). We assume that after T_{Slush} rounds we reach a stable consensus where S_0 parties have opinion 1, whereas the opposite case with a stable consensus for opinion 0 is analogous. Note that in T_{Slush} rounds we have $D'_j \geq -T_{\text{Slush}}$, as in each round the counter $\text{cnt}[0]$ increases by at most one. After $T = 6T_{\text{Slush}}$ additional rounds starting from S_0 we obtain $D'_j \geq -T_{\text{Slush}} + D_j(T) \geq -T_{\text{Slush}} + 3T_{\text{Slush}} = 2T_{\text{Slush}} = \tau$. Therefore, any given party decides and the algorithm terminates after $7T_{\text{Slush}}$ rounds, with all but negligible probability, given that $T \geq 48\beta$ and $T \in o(n^{1/4})$. The former means that we require $T_{\text{Slush}} \geq 8\beta$. The latter is fulfilled as $T_{\text{Slush}} \in O(\log n)$.

Validity. Suppose all parties propose 1 (w.l.o.g.). Then there is also a stable consensus for 1 even if we assume that initially \sqrt{n} parties are flipped to 0 by an adversary. We showed above that from a stable consensus state, for any given node, the threshold $\text{cnt}[1] - \text{cnt}[0] = \tau$ will be reached with all but negligible probability.

Integrity. Follows from the construction of the algorithm.

Agreement. The idea is that the threshold τ is so large that no party can decide before a stable consensus is reached, and from there on every party must decide the same. Since in each round a counter of any party j can increase by only one, no party can decide before $\tau = 2T_{\text{Slush}}$ rounds have passed. Already after T_{Slush} rounds we will reach a stable consensus with a S_0 majority for opinion 1 (w.l.o.g.). When we arrive at the state of stable consensus we have $D'_j \geq -T_{\text{Slush}}$, hence reaching the $-\tau = -2T_{\text{Slush}}$ threshold for deciding 0 would require at least T_{Slush} additional rounds. We already showed that starting from S_0 the change in D'_j is strictly positive with all but negligible probability (provided that T_{Slush} is a sufficiently large multiple of β). Consequently for any party j the probability to attain the threshold $-\tau$ within the next $6T_{\text{Slush}}$ rounds is negligible. \square

The theorem above is given in a more general way, depending on the number of rounds until Slush reaches a stable consensus. Given that we have already shown an upper bound of $O(\log n + \beta)$ for this (Corollary 4.7), we can rephrase the theorem as follows.

Corollary 7.4. *Algorithm 5 (Blizzard) ensures consensus with all but negligible probability after at most $O(\log n + \beta)$ rounds. This holds even with a \sqrt{n} -bounded adversary.*

8 Conclusion

With the goal of improving latency in mind, we deduce two main recommendations for changes to Snow-style consensus protocols as they are deployed in the Avalanche network today. First, for a given k we recommend to choose $\alpha > k/2$ as small as possible, i.e., $\alpha = \lceil \frac{k+1}{2} \rceil$, as this promises a better performance compared to α closer to k (see Lemma 3.2 or Figure 1b). Second, we propose to change the termination condition of the Snowflake and Snowball protocols where we observe an unfavorable trade-off between security and latency (see Corollary 3) to the simpler one of the Blizzard protocol (Algorithm 5). This modification will resolve the observed issue (see Theorem 4 and Corollary 7.4). We caveat our recommendations by noting that there might be other considerations than the asymptotic performance aspects analyzed in this paper.

From a theoretical point of view, we see our results on the performance of Slush (Theorem 1 and 2) as a

natural continuation of the corresponding analysis of the randomized, self-stabilizing consensus protocols in the GOSSIP model where the sample size is at most 3 (such as the Median Protocol, the 2-Choices Protocol and the 3-Majority Protocol, see Definition 4.5). These protocols have been analyzed with respect to performance as a function of the initial number of opinions and it was shown that in “most” conditions they converge quite fast (cf. related work Section 2.3). An interesting avenue of future research is the adaptation of Slush to multiple opinions, a party adopts a new opinion if it has a simple majority in a sampling of size k , i.e., a “ k -Majority protocol”. We believe that our technical work on Slush gives insights how such a k -Majority protocol would perform on multiple opinions.

References

- [1] Ignacio Amores-Sesar, Christian Cachin, and Enrico Tedeschi. “When Is Spring Coming? A Security Analysis of Avalanche Consensus”. In: *26th International Conference on Principles of Distributed Systems, OPODIS 2022, December 13-15, 2022, Brussels, Belgium*. Ed. by Eshcar Hillel, Roberto Palmieri, and Etienne Rivière. Vol. 253. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 10:1–10:22. URL: <https://doi.org/10.4230/LIPIcs.OPODIS.2022.10>.
- [2] Luca Becchetti, Andrea E. F. Clementi, and Emanuele Natale. “Consensus Dynamics: An Overview”. In: *SIGACT News* 51.1 (2020), pp. 58–104. URL: <https://doi.org/10.1145/3388392.3388403>.
- [3] Luca Becchetti et al. “Simple dynamics for plurality consensus”. In: *Distributed Comput.* 30.4 (2017), pp. 293–306. URL: <https://doi.org/10.1007/s00446-016-0289-4>.
- [4] Luca Becchetti et al. “Stabilizing Consensus with Many Opinions”. In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. Ed. by Robert Krauthgamer. SIAM, 2016, pp. 620–635. URL: <https://doi.org/10.1137/1.9781611974331.ch46>.
- [5] Vitalik Buterin. *The Scalability Trilemma — Why Sharding is Great: Demystifying the Technical Properties*. Available online, <https://vitalik.ca/general/2021/04/07/sharding.html>. 2017.
- [6] Colin Cooper, Robert Elsässer, and Tomasz Radzik. “The Power of Two Choices in Distributed Voting”. In: *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*. Ed. by Javier Esparza et al. Vol. 8573. Lecture Notes in Computer Science. Springer, 2014, pp. 435–446. URL: https://doi.org/10.1007/978-3-662-43951-7_37.
- [7] Emilio Cruciani et al. “Phase Transitions of the k -Majority Dynamics in a Biased Communication Model”. In: *ICDCN ’21: International Conference on Distributed Computing and Networking, Virtual Event, Nara, Japan, January 5-8, 2021*. ACM, 2021, pp. 146–155. URL: <https://doi.org/10.1145/3427796.3427811>.
- [8] James R. Cruise and Ayalvadi Ganesh. “Probabilistic consensus via polling and majority rules”. In: *Queueing Syst. Theory Appl.* 78.2 (2014), pp. 99–120. URL: <https://doi.org/10.1007/s11134-014-9397-7>.
- [9] Benjamin Doerr et al. “Stabilizing consensus with the power of two choices”. In: *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*. Ed. by Rajmohan Rajaraman and Friedhelm Meyer auf der Heide. ACM, 2011, pp. 149–158. URL: <https://doi.org/10.1145/1989493.1989516>.
- [10] Robert Elsässer et al. “Brief Announcement: Rapid Asynchronous Plurality Consensus”. In: *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*. Ed. by Elad Michael Schiller and Alexander A. Schwarzmann. ACM, 2017, pp. 363–365. URL: <https://doi.org/10.1145/3087801.3087860>.

- [11] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. “Easy Impossibility Proofs for Distributed Consensus Problems”. In: *Distributed Comput.* 1.1 (1986), pp. 26–39. URL: <https://doi.org/10.1007/BF01843568>.
- [12] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. “Impossibility of Distributed Consensus with One Faulty Process”. In: *J. ACM* 32.2 (1985), pp. 374–382. URL: <https://doi.org/10.1145/3149.214121>.
- [13] Mohsen Ghaffari and Johannes Lengler. “Nearly-Tight Analysis for 2-Choice and 3-Majority Consensus Dynamics”. In: *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*. Ed. by Calvin Newport and Idit Keidar. ACM, 2018, pp. 305–313. URL: <https://dl.acm.org/citation.cfm?id=3212738>.
- [14] Seth Gilbert and Nancy A. Lynch. “Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services”. In: *SIGACT News* 33.2 (2002), pp. 51–59.
- [15] Vincent Gramoli et al. “Diablo: A Benchmark Suite for Blockchains”. In: *Proceedings of the Eighteenth European Conference on Computer Systems, EuroSys 2023, Rome, Italy, May 8-12, 2023*. Ed. by Giuseppe Antonio Di Luna et al. ACM, 2023, pp. 540–556. URL: <https://doi.org/10.1145/3552326.3567482>.
- [16] Team Rocket et al. “Scalable and Probabilistic Leaderless BFT Consensus through Metastability”. In: *CoRR* abs/1906.08936 (2019). arXiv: 1906.08936. URL: <http://arxiv.org/abs/1906.08936>.

A Probabilistic Concepts

We give a few basic definitions and principles pertaining the probabilistic security properties of some protocol (used in Definition 2.4) that we use throughout the paper.

Definition A.1 (Negligible Function). *A function f is negligible if for any polynomial π there is a constant $\lambda_0 \geq 0$, s.t., for any $\lambda \geq \lambda_0$ it is $f(\lambda) \leq \pi(\lambda)$.*

Remark A.2. *We often use that for any constant $c > 0$, the function $f(\lambda) = e^{-c \cdot \lambda}$ is a negligible w.r.t. λ .*

We usually aim for a certain security threshold given by a variable γ .

Definition A.3 (All But Negligible Probability). *An event is said to occur with all but negligible probability with respect to some parameter λ if the probability of the event not happening is a function in λ that is negligible w.r.t. λ .*

In the literature, randomized consensus protocols are often shown to be successful *with high probability*, which expresses the probability of failure as a function that decreasing inversely with the input size n of the problem (here n is the number of parties). This is often quite convenient as it eliminates any other variable from the analysis, compared to defining some fixed failure threshold.

Definition A.4 (With High Probability). *An event is said to hold with high probability (w.h.p.), if there exists a constant $c \geq 1$ such that the event occurs with probability at least $1 - n^{-c}$ for sufficiently large n .*

The disadvantage of the notion w.h.p. is that it usually looks at the asymptotic behavior of a system (i.e., for large n), which does not provide a fixed security level for small n , which can be a requirement in practice. The following lemma provides an interface between the two notions.

Lemma A.5. *Let E be an event such that the probability that E does not occur within any interval of T consecutive rounds is at most $\frac{1}{n}$ (which is guaranteed if E occurs w.h.p., see Definition A.4). Then E occurs with probability $1 - e^{-\lambda}$ (all but negligible, see Definition A.3) after $T' = T(\frac{\lambda}{\ln n} + 1)$ rounds for security parameter $\lambda > 0$.*

Proof. We consider the cases $\lambda < \ln n$ and $\lambda \geq \ln n$ starting with the former. Note that $T' \geq T$, thus the

probability that E does not occur is at most

$$\frac{1}{n} = e^{-\ln n} < e^{-\lambda}.$$

In the second case, we use that $T' \geq \frac{\lambda}{\ln n} T$. The probability that E does not occur after $\frac{\lambda}{\ln n} T$ rounds is at most

$$\frac{1}{n^{\frac{\lambda}{\ln n}}} = \frac{1}{e^{\frac{\lambda \ln n}{\ln n}}} = e^{-\lambda}.$$

□

Remark A.6. In Lemma A.5, given that some event E occurs w.h.p., within $T \in O(\log n)$ rounds, then $T' = T(\frac{\lambda}{\ln n} + 1) \in O(\lambda + \log n)$ rounds are sufficient that E occurs with all but negligible probability.

Lemma A.7 (Closure of Negligible Functions). Let $\lambda > 0$ and $k \leq \rho(\lambda)$ for some polynomial ρ . Let E_1, \dots, E_k be outcomes (events) of some algorithm \mathcal{A} . Assume that any individual event E_i takes place with all but negligible probability (Def. 2.4). Then $E := \bigcap_{i=1}^k E_i$ takes place with all but negligible probability.

Proof. There is a $\lambda_0 \geq 0$ such that $\rho(\lambda) \leq e^{\lambda/2}$ for all $\lambda \geq \lambda_0$. Let $\lambda_1, \dots, \lambda_k \in \mathbb{N}$ such that for all $i \in \{1, \dots, k\}$ we have $\mathbb{P}(\overline{E}_i) \leq e^{-\lambda}$ for $\lambda > \lambda_i$. With Boole's inequality (a.k.a. "Union Bound") we have that

$$\mathbb{P}(\overline{E}) = \mathbb{P}\left(\bigcup_{i=1}^k \overline{E}_i\right) \leq \sum_{i=1}^k \mathbb{P}(\overline{E}_i) \leq \sum_{i=1}^k e^{-\lambda} \leq \rho(\lambda) e^{-\lambda} \leq e^{-\lambda/2} = e^{-\lambda'}$$

for $\lambda' \geq 2 \max(\lambda_0, \dots, \lambda_k)$. □

Remark A.8. Lemma A.7 implicitly shows that a series of events all occur with all but negligible probability w.r.t. λ , given that the number of events is not too large in λ . In our proofs we often apply this mechanic without specifically mentioning the lemma. Sometimes we also have a number of events that scales in n , for this we (necessarily have to) assume that $n \leq \pi(\lambda)$ for some fixed but arbitrary polynomial π , which allows us to apply Lemma A.7 for such a number of events as well.

We use the following forms of Chernoff bounds in some of our proofs.

Lemma A.9 (Chernoff Bound). Let $X = \sum_{i=1}^n X_i$ for i.i.d. random variables $X_i \in \{0, 1\}$ and $\mathbb{E}(X) \leq \mu_H$ and $\delta \geq 1$, then

$$\mathbb{P}(X \geq (1+\delta)\mu_H) \leq \exp\left(-\frac{\delta\mu_H}{3}\right),$$

Similarly, for $\mathbb{E}(X) \geq \mu_L$ and $0 \leq \delta \leq 1$ we have

$$\mathbb{P}(X \leq (1-\delta)\mu_L) \leq \exp\left(-\frac{\delta^2\mu_L}{2}\right).$$

B Sums over Binomial Coefficients

We will summarize a few equations for sums over binomial coefficients of the form $\sum_{\ell=i}^j \binom{k}{\ell}$ that we use throughout this paper.

Remark B.1 (Some Identities). Starting with the definition $\binom{k}{\ell} := \frac{k!}{\ell!(k-\ell)!}$, the first observation is that $\binom{k}{\ell} = \binom{k}{k-\ell}$ by symmetry. By reordering summands we obtain

$$\sum_{\ell=0}^m \binom{k}{\ell} = \binom{k}{0} + \dots + \binom{k}{m} = \binom{k}{k} + \dots + \binom{k}{k-m} = \binom{k}{k-m} + \dots + \binom{k}{k} = \sum_{\ell=k-m}^k \binom{k}{\ell}.$$

Note that a closed form for partial sums of the form $\sum_{\ell=i}^j \binom{k}{\ell}$ is currently not known. In this article we make use a few known special cases.

Remark B.2 (Closed Forms). *The binomial theorem implies $\sum_{\ell=0}^k \binom{k}{\ell} = 2^k$. Using the equality from the previous remark, for uneven k we obtain*

$$\sum_{\ell=0}^{\lfloor k/2 \rfloor} \binom{k}{\ell} = \sum_{\ell=\lceil k/2 \rceil}^k \binom{k}{\ell} = 2^{k-1}.$$

For any k and any $m < k/2$ we have at least the following estimations

$$\sum_{\ell=0}^m \binom{k}{\ell} = \sum_{\ell=k-m}^k \binom{k}{\ell} \leq 2^{k-1}.$$

C Pseudocode

Algorithm 1 State

Global parameters and state	
1:	\mathcal{N} // set of parties
2:	$newRound \in \{\text{FALSE}, \text{TRUE}\}$ // variable indicating when to start a round
3:	$decided \in \{\text{FALSE}, \text{TRUE}\}$ // variable indicating when to finish the protocol
4:	$b \in \{0, 1, \perp\}$ // current estimate for decision, initially \perp
5:	$k \in \mathbb{N}$ // number of parties queried in each poll
6:	$\alpha \in \mathbb{N}$ // majority threshold for queries
7:	$\mathcal{S} : \text{HashMap}[\mathcal{T} \rightarrow \mathcal{N}]$ // set of sampled parties to be queried
8:	$votes : \text{HashMap}[\{0, 1\} \rightarrow \mathbb{N}]$ // number of votes for a value
9:	$cnt \in \mathbb{N}$ // counter for acceptance Snowflake and Snowball
10:	$cnt[0] \in \mathbb{N}$ // counter for acceptance Blizzard
11:	$cnt[1] \in \mathbb{N}$ // counter for acceptance Blizzard
12:	$\beta \in \mathbb{N}$ // threshold for acceptance
13:	$d : \text{HashMap}[\{0, 1\} \rightarrow \mathbb{N}]$ // confidence value of a transaction
14:	$round \in \mathbb{N}$ // current round
15:	$maxRound \in \mathbb{N}$ // maximum round

Algorithm 2 Slush (party j)

```
16: upon  $propose(b')$  do
17:    $decided \leftarrow \text{FALSE}$ 
18:    $newRound \leftarrow \text{TRUE}$ 
19:    $b \leftarrow b'$ 

20: upon  $newRound \wedge \neg decided$  do // still not decided
21:    $newRound \leftarrow \text{FALSE}$ 
22:    $votes[*] \leftarrow 0$ 
23:    $round \leftarrow round + 1$ 
24:   if  $b \neq \perp$  then
25:      $S \leftarrow \text{sample}(\mathcal{N} \setminus \{j\}, k)$  // sample  $k$  parties
26:      $send\ message\ [QUERY, b]$  to all parties  $k \in S$ 

27: upon  $votes[b'] \geq \alpha$  do //  $b' = 0$  or  $b' = 1$ 
28:    $b \leftarrow b'$ 
29:    $newRound \leftarrow \text{TRUE}$ 

30: upon  $n = k \wedge votes[0] < \alpha \wedge votes[1] < \alpha$  do // no majority
31:    $newRound \leftarrow \text{TRUE}$ 

32: upon receiving message  $[QUERY, b']$  from party  $k$  do
33:   if  $b = \perp$  then
34:      $decided \leftarrow \text{FALSE}$ 
35:      $b \leftarrow b'$ 
36:    $send\ message\ [VOTE, b]$  to party  $k$  // reply with the local value of  $b$ 

37: upon receiving message  $[VOTE, b']$  from a party  $k \in S$  do // collect the vote  $b'$ 
38:    $votes[b'] \leftarrow votes[b'] + 1$ 

39: upon  $round = maxRound \wedge \neg decided$  do // end of the protocol
40:    $decide(b)$ 
41:    $decided \leftarrow \text{TRUE}$ 
```

Algorithm 3 Snowflake (party j)

```
42: upon propose( $b'$ ) do
43:    $decided \leftarrow \text{FALSE}$ 
44:    $newRound \leftarrow \text{TRUE}$ 
45:    $b \leftarrow b'$ 

46: upon  $newRound \wedge \neg decided$  do // still not decided
47:    $newRound \leftarrow \text{FALSE}$ 
48:    $votes[*] \leftarrow 0$ 
49:   if  $b \neq \perp$  then
50:      $S \leftarrow \text{sample}(\mathcal{N} \setminus \{j\}, k)$  // sample  $k$  random parties
51:      $\text{send message [QUERY, } b \text{] to all parties } k \in S$ 

52: upon  $votes[b'] \geq \alpha$  do //  $b' = 0$  or  $b' = 1$ 
53:   if  $b = b'$  then // majority for our proposal
54:      $cnt \leftarrow cnt + 1$ 
55:   else
56:      $b \leftarrow b'$ 
57:      $cnt \leftarrow 1$ 
58:    $newRound \leftarrow \text{TRUE}$ 

59: upon  $n = k \wedge votes[0] < \alpha \wedge votes[1] < \alpha$  do // no majority
60:    $cnt \leftarrow 0$ 
61:    $newRound \leftarrow \text{TRUE}$ 

62: upon receiving message [QUERY,  $b'$ ] from party  $k$  do
63:   if  $b = \perp$  then
64:      $decided \leftarrow \text{FALSE}$ 
65:      $b \leftarrow b'$ 
66:    $\text{send message [VOTE, } b \text{] to party } k$  // reply with the local value of  $b$ 

67: upon receiving message [VOTE,  $b'$ ] from a party  $k \in S$  do // collect the vote  $b'$ 
68:    $votes[b'] \leftarrow votes[b'] + 1$ 

69: upon  $cnt = \beta \wedge \neg decided$  do // there is enough confidence for  $B$ 
70:    $decide(b)$ 
71:    $decided \leftarrow \text{TRUE}$ 
```

Algorithm 4 Snowball (party j)

```
72: upon propose( $b'$ ) do
73:    $decided \leftarrow \text{FALSE}$ 
74:    $newRound \leftarrow \text{TRUE}$ 
75:    $b \leftarrow b'$ 

76: upon  $newRound \wedge \neg decided$  do // still not decided
77:    $newRound \leftarrow \text{FALSE}$ 
78:    $votes[*] \leftarrow 0$ 
79:   if  $b \neq \perp$  then
80:      $S \leftarrow \text{sample}(\mathcal{N} \setminus \{j\}, k)$  // sample  $k$  random parties
81:     send message [QUERY,  $b$ ] to all parties  $k \in S$ 

82: upon  $votes[b'] \geq \alpha$  do //  $b' = 0$  or  $b' = 1$ 
83:    $d[b'] \leftarrow d[b'] + 1$ 
84:   if  $b = b'$  then // majority for our proposal
85:      $cnt \leftarrow cnt + 1$ 
86:   else
87:     if  $d[b'] > d[b]$  then
88:        $b \leftarrow b'$ 
89:      $cnt \leftarrow 1$ 
90:    $newRound \leftarrow \text{TRUE}$ 

91: upon  $n = k \wedge votes[0] < \alpha \wedge votes[1] < \alpha$  do // no majority
92:    $cnt \leftarrow 0$ 
93:    $newRound \leftarrow \text{TRUE}$ 

94: upon receiving message [QUERY,  $b'$ ] from party  $k$  do
95:   if  $b = \perp$  then
96:      $decided \leftarrow \text{FALSE}$ 
97:      $b \leftarrow b'$ 
98:   send message [VOTE,  $b$ ] to party  $k$  // reply with the local value of  $b$ 

99: upon receiving message [VOTE,  $b'$ ] from a party  $k \in S$  do // collect the vote  $b'$ 
100:   $votes[b'] \leftarrow votes[b'] + 1$ 

101: upon  $cnt = \beta \wedge \neg decided$  do // there is enough confidence for  $B$ 
102:   $decide(b)$ 
103:   $decided \leftarrow \text{TRUE}$ 
```

Algorithm 5 Blizzard (party j)

```
104: upon propose( $b'$ ) do
105:   decided  $\leftarrow$  FALSE
106:   newRound  $\leftarrow$  TRUE
107:    $b \leftarrow b'$ 

108: upon newRound  $\wedge$   $\neg$ decided do // not yet decided
109:   newRound  $\leftarrow$  FALSE
110:   votes[*]  $\leftarrow$  0
111:   if  $b \neq \perp$  then
112:      $\mathcal{S} \leftarrow \text{sample}(\mathcal{N} \setminus \{j\}, k)$  // sample  $k$  parties
113:     send message [QUERY,  $b$ ] to all parties  $k \in \mathcal{S}$ 

114: upon votes[ $b'$ ]  $\geq \alpha$  do //  $b' = 0$  or  $b' = 1$ 
115:    $b \leftarrow b'$ 
116:   cnt[ $b$ ] ++
117:   newRound  $\leftarrow$  TRUE

118: upon  $n = k \wedge \text{votes}[0] < \alpha \wedge \text{votes}[1] < \alpha$  do // no majority
119:   newRound  $\leftarrow$  TRUE

120: upon receiving message [QUERY,  $b'$ ] from party  $k$  do
121:   if  $b = \perp$  then
122:     decided  $\leftarrow$  FALSE
123:      $b \leftarrow b'$ 
124:   send message [VOTE,  $b$ ] to party  $k$  // reply with the local value of  $b$ 

125: upon receiving message [VOTE,  $b'$ ] from a party  $k \in \mathcal{S}$  do // collect the vote  $b'$ 
126:   votes[ $b'$ ]  $\leftarrow$  votes[ $b'$ ] + 1

127: upon  $\text{cnt}[1] - \text{cnt}[0] = \tau \wedge \neg$ decided do // threshold  $\tau \in O(\log n + \beta)$ , see Thm. 4 and Cor. 7.4
128:   decide(1)

   upon  $\text{cnt}[0] - \text{cnt}[1] = \tau \wedge \neg$ decided
129:   decide(0)
```
