

A Polylogarithmic Gossip Algorithm for Plurality Consensus

Mohsen Ghaffari
MIT
ghaffari@mit.edu

Merav Parter
MIT
parter@mit.edu

ABSTRACT

Consider n anonymous nodes each initially supporting an opinion in $\{1, 2, \dots, k\}$ and suppose that they should all learn the opinion with the largest support. Per round, each node contacts a random other node and exchanges B bits with it, where typically B is at most $O(\log n)$.

This basic distributed computing problem is called the *plurality consensus* problem (in the *gossip* model) and it has received extensive attention. An efficient plurality protocol is one that converges to the plurality consensus as fast as possible, and the standard assumption is that each node has memory at most polylogarithmic in n . The best known time bound is due to Becchetti et al. [SODA'15], reaching plurality consensus in $O(k \log n)$ rounds using $\log(k+1)$ bits of local memory, under some mild assumptions. As stated by Becchetti et al., achieving a poly-logarithmic time complexity remained an open question.

Resolving this question, we present an algorithm that with high probability¹ reaches plurality consensus in $O(\log k \log n)$ rounds, while having message and memory size of $\log k + O(1)$ bits. This even holds under considerably more relaxed assumptions regarding the initial bias (towards plurality) compared to those of prior work. The algorithm is based on a very simple and arguably natural mechanism.

1. INTRODUCTION AND RELATED WORK

Coordination problems in distributed systems require nodes to agree on a common action or value. These problems are referred to as consensus, agreement, or voting and are among the most fundamental problems in distributed computing [Lyn96, Tsi84, TBA86]. In this paper, we consider the following basic agreement problem, often called plurality consensus problem—also known as *proportionate agreement*

This work was partially funded by the following grants: AFOSR FA9550-13-1-0042, NSF CCF-1461559, CCF-1217506, CCF-0939370.

¹As standard, *with high probability* (w.h.p.) indicates that an event has probability $1 - 1/n^c$ for a constant $c \geq 2$.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](http://permissions.acm.org).

PODC'16, July 25-28, 2016, Chicago, IL, USA

© 2016 ACM. ISBN 978-1-4503-3964-3/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2933057.2933097>

or *majority agreement* when $k = 2$ —in the gossip model:

There are n nodes each supporting an opinion in $\{1, \dots, k\}$. Per round each node contacts one random other node and exchanges B bits with it, where typically B is at most $O(\log n)$. Eventually, all nodes should hold the opinion with the largest initial support.

This problem, or close variants of it, come up in computer networks such as sensor networks [AAD⁺06] and peer-to-peer networks [AFJ06], as well as in a wider range of settings that goes much beyond the classical distributed computing settings: e.g., social networks [MS], biological systems [CKFL05, SKJ⁺08, CCN12, BSDDS14], and chemical reaction networks [CDS⁺13, Dot14].

State of the Art: Gossip algorithms for plurality have received extensive attention, see e.g. [BCN⁺14, BCN⁺15a, BCN⁺15b, AAE08, PVV09]. The key performance measures of the algorithms are: (1) *time complexity*, and (2) space measured by the *message size* and *memory size* per node. In the standard regime where message and memory sizes are at most $\text{poly}(\log n)$, the best known algorithm is due to Becchetti et al. [BCN⁺15a], achieving a time complexity of $O(k \log n)$, under some mild assumptions, while having $\log(k+1)$ bit message/memory. A faster solution was known before, however at the expense of a significantly larger message/memory size. Particularly, the general approach of Kempe et al. [KDG03] can be used to obtain a plurality algorithm with $O(\log n)$ time-complexity but with $O(k \log n)$ message/memory size. Also, a number of other fast algorithms are known but all limited to very small k , e.g., $k = 2$ [AAE08, PVV09]. Given that each of the known solutions lacks in one of the dimensions, Becchetti et al. [BCN⁺15a] state:

“a major open question for the plurality consensus problem is whether a plurality protocol exists that converges in polylogarithmic time and uses only polylogarithmic local memory.”

Our Result: In this paper, we answer this question in the affirmative. We particularly present an algorithm that has nearly optimal bounds in both dimensions:

Theorem 1.1. *There is a gossip algorithm that solves plurality w.h.p. in $O(\log k \cdot \log n)$ rounds using $\log k + O(1)$ bits of local memory, assuming that the initial bias $p_1 - p_2 =$*

$\Omega(\sqrt{\log n/n})$. Here p_1 and p_2 are, respectively, the fractions of nodes holding the plurality and second largest opinions.

This initial bias assumption $p_1 - p_2 = \Omega(\sqrt{\log n/n})$ is due to the concentration considerations², and it is at least as relaxed as the assumptions in prior work. In fact under the stronger assumptions of [BCN⁺15a] (to be stated later), our algorithm converges in $O(\log k \log \log n + \log n)$ rounds.

As we will see, the algorithm is based on a quite simple and intuitive mechanism and one can even argue that dynamics with a similar flavor occur in nature. We also note that the result might be viewed as somewhat surprising, given the indications in the prior work and especially when put in contrast to what Becchetti et al. [BCN⁺15a] write in their conclusion:

“We believe that this distance [referring to the monochromatic distance, which can be as large as $\Omega(k)$] might represent a general lower bound on the convergence time of any plurality dynamics which uses only $\log k + O(1)$ bits of local memory.”

A Remark—Measuring Memory Size: We note that because of the practical scenarios in which gossip algorithms are applied, it is more standard to measure the memory size in terms of the *number of states* rather than the *number of bits*. Here we briefly discuss the reasons. One justification for considering the number of states is that many of these dynamics are often viewed as Finite-state Automata, for which minimizing the number of states is a classical optimization objective (see, e.g., [Hop71, Yu05]). In addition, states of nodes have a clear physical interpretation in practical settings such as in molecular programming: in particular, in chemical reaction networks, the encoding of the state is not binary but rather (closer to) unary [CSWB09], and hence minimizing the number of the states is a more sensible objective. Although these two ways of measuring are equivalent when talking about exact bounds—as the number of states is simply exponential in the number of bits—they differ when discussing relaxed notions of optimality. Particularly, a constant factor improvement in the bit complexity, which might be regarded negligible, translates to a polynomial improvement in the number of states, which would be considered significant.

In fact, when presenting our result, we start with a simpler algorithm with $\log k + O(\log \log k)$ bit complexity, and then present a more complex one with bit complexity of $\log k + O(1)$. Our first algorithm converges within $O(\log k \cdot \log n)$ rounds but uses $\log k + O(\log \log k)$ bits. Note that this already gives an answer to the open problem raised by Becchetti et al. [BCN⁺15a], stated above. Yet, even though the message size of this simple algorithm in terms of the number bits is optimal up to an *additive* $O(\log \log k)$ term, with respect to the *number of states*, its $O(k \log k)$ state complexity bound is an $O(\log k)$ factor off from the trivial lower bound k . In our second algorithm, we improve this to a *state complexity* of $O(k)$, which is optimal up to a constant.

²Even when flipping n fair coins, there is a $1/n^{0.1}$ probability that the number of heads is outside $n/2 \pm \Omega(\sqrt{n \log n})$, which means a relative deviation of $\pm \Omega(\sqrt{\log n/n})$. This is essentially why we need to assume $p_1 - p_2 = \Omega(\sqrt{\log n/n})$.

1.1 Our approach, in the context of the previous ones

Plurality protocols can be roughly divided into two main classes: what we call “*reading*” protocols and what we call *plurality amplification* protocols (or amplification protocols, for short).

A natural approach for solving plurality, in fact perhaps the first approach that comes to the mind from a standard algorithm design viewpoint, is to estimate the frequencies of the nodes of each opinion and pick the most frequent one. In a more relaxed definition, in a reading style protocol, for each opinion i , we would have at least one node v that learns some estimation for the frequency of this opinion in the population. Kempe et al. [KDG03] provided a general approach for designing reading type protocols for computing aggregate functions (e.g. sum or average). As noted by Becchetti et al. [BCN⁺15a], adopting the approach of Kemp et al. to the plurality problem yields an $O(\log n)$ time complexity but with $O(k \log n)$ message/memory size. Generally, it seems implausible that reading style algorithms would lead to a plurality algorithm for the random gossip model with polylogarithmic size messages and polylogarithmic time complexity³.

The second class of plurality amplification protocols do not try to estimate the initial distribution, but rather attempt to *modify* this initial distribution into another distribution of the same plurality opinion, but with a *larger* bias towards plurality. In other words, throughout the execution of an amplification protocol, there is a shift from the initial distribution (with possibly very weak bias) towards the target configuration (where we have consensus on plurality) by repetitively amplifying the bias towards the plurality opinion via some *positive feedback mechanism*. The standard plurality amplification protocols are based on simple majority rules. For example, Becchetti et al. [BCN⁺14] consider an update rule where each node polls the opinion of three random nodes and adopts the majority opinion among these. This algorithm uses only $\Theta(\log k)$ bits of local memory, but unfortunately has running time $O(\min\{k \log n, n^{1/3} \cdot \log^{2/3} n\})$. Other known time efficient algorithms exist for small values of k , e.g., $k = 2$ [PVV09, DGM⁺11]. Recently, Becchetti et al. [BCN⁺15a] analyzed the well-known *Undecided-State Dynamics*, in which each undecided node meeting a node of different opinion becomes undecided (i.e., forgets its opinion) and every undecided node meeting a decided node adopts its opinion. This protocol provides the current best bounds: they showed that the undecided-state dynamics converges within $O(k \log n)$ rounds, using $\log(k + 1)$ bits of local memory, assuming⁴ $k = O((n/\log n)^{1/6})$ and $p_1 - p_2 = \Omega(1/n^{1/6})$.

The high level idea of our algorithm: We want to create a protocol with a ‘*strong and fast positive feedback*’ so that, roughly speaking, the opinions with larger support get amplified while those with smaller support get attenuated. In more colloquial terms, we seek a (fast) ‘*rich get richer*’ effect. At the high level, our protocol consists of two main

³We note however that if the gossip model is relaxed to include non-random meetings, a rather simple “reading” style algorithm would achieve this objective, as shown in the full version.

⁴They assume $p_1 \geq (1 + \alpha)p_2$ for some constant α and $k \leq O(n^{1/6})$. This implies $p_1 - p_2 = \Omega(p_1) = \Omega(1/n^{1/6})$.

alternating steps: A *selection* step, and a sequence of *recovery* steps. The selection process imposes a constraint on the system that potentially may lead many nodes to lose their opinion and become undecided. Specifically, a node survives the selection process only if it meets a node of its *own* opinion. By that, we intentionally cut down the number of decided nodes significantly while opinions with larger support survive through this process slightly better. Particularly, the plurality opinion gets slightly less attenuation in comparison to the others, hence the proportion of the plurality opinion to that of any other opinion grows. Since repetitive applications of selections would eventually diminish the decided population, our system evolution incorporates some healing process between two consecutive selections. In the healing phase, the fraction of undecided nodes shrinks back to (almost) 0 but the ratios between the number of nodes holding different opinions is roughly preserved. We repeat through this cycle of selection and healing, thus continually *amplifying the plurality*, till reaching plurality consensus.

Other related work: The plurality problem bears some resemblance to problems studied in some other settings: The restricted case of $k = 2$ has been extensively studied in the setting of population protocols [AAD⁺06] and many “plurality amplification” type algorithms using small number of states (e.g., 3 or 4) have been suggested [AAE08, DV12, MNRS14, AGV15]. This case of binary consensus ($k = 2$) has been also studied in the voter model [DW83, HP01, WH04, Lig12]. The plurality problem also resembles the *heavy hitters* problem in the streaming model, see e.g. [BICS10].

2. THE ALGORITHM: TAKE 1

We next present a simpler variant of our algorithm, which reaches plurality in $O(\log k \log n)$ rounds, using messages of size $\log k + O(1)$ bits, but with per node memory size of $\log k + \log \log k + O(1)$ bits. Note that this result already answers the open question stated in the introduction.

2.1 The Algorithm Description

The algorithm is an extremely simple and clean dynamics, as follows: Let us call nodes holding one of opinions $\{1, 2, \dots, k\}$ *decided* while nodes holding no opinion are called *undecided*.

The algorithm works in phases, each having $R = O(\log k)$ rounds, as follows:

- **Round 1 of each phase:** - Relative Gap Amplification -
 - a decided node keeps its opinion *only if* it contacts a node with the same opinion,
 - undecided nodes remains undecided.
- **Rounds 2 to R of each phase:** - Healing -
 - decided nodes keep their opinion,
 - an undecided node v that contacts a decided node u adopts u 's opinion.

When implementing this algorithm, each message has size $\log(k+1)$ bits as it contains only one opinion in $\{0, 1, \dots, k\}$, where 0 represents being undecided. As for the memory, the only additional thing that a node needs to remember besides

its opinion is the round number in the phase, i.e. modulo $R = O(\log k)$, and that takes $\log \log k + O(1)$ bits.

We next provide some intuition for why this dynamic converges to plurality within $O(\log n)$ phases and then the next subsection presents the related formal analysis.

Before that, we note that it was pointed out to us recently that in a simultaneous and independent work, Berenbrink et al. [BFGK16] also present a similar algorithm and analysis as the take 1 protocol presented here. They prove that this algorithm terminates in $O(\log k \log \log_\gamma n + \log \log n)$ rounds, where γ is the relative gap at the beginning between the first and second largest opinions. In the worst case considered in our paper, $\gamma = 1 + \tilde{O}(1/\sqrt{n})$. Thus in this regime, the two complexities match. For larger initial bias, especially when γ is a constant, there is a difference between the two complexities as our statement is $O(\log k \log \log n + \log n)$. However, one can see that the arguments in Lemma 2.8 could be tightened easily to match the same complexity. Since our main interest is in cases with smaller bias, where the problem is more interesting technically, we keep the statements and analysis as they are.

Basic notations: To keep track of the state of the system, we use a vector $\mathbf{p} = (p_1, p_2, \dots, p_k) \in [0, 1]^k$ where each entry p_i denotes the fraction of the nodes holding opinion $i \in \{1, 2, \dots, k\}$. Thus, the fraction of undecided nodes is $1 - \sum_{i=1}^k p_i$. Without loss of generality, in our analysis, we assume that the opinions are renumbered such that initially $p_1 > p_2 \geq \dots \geq p_k$.

Convergence intuition: A rough intuition for why the algorithm is designed this way and why it should intuitively converge to plurality consensus fast, based merely on expectations, is as follows. We emphasize that this explanation is provided only for the sake of intuition and it is imprecise. However, it will be made precise in the sequel. Generally, one should be careful in using similar merely expectation-based arguments, as they might lead to incorrect claims.

In the first round of each phase, each node v keeps its opinion only if it contacted a node with the same opinion. Note that this means, even if v contacted an undecided node, v forgets its opinion and becomes undecided. At the end of this round, the fraction of nodes holding an opinion i changes from p_i to p_i^2 , in expectation. Thus, the ratio between the plurality opinion and an opinion $i \in \{2, \dots, k\}$ goes up from $\frac{p_1}{p_i}$ to $(\frac{p_1}{p_i})^2$. This already is the desired ‘rich get richer’ effect, in terms of the ratios. However, as a side-effect of this relative *gap amplification*, the fraction of decided nodes has dropped down significantly to about $\sum_{i=1}^k p_i^2$, which can be as small as $\Theta(1/k)$. During rounds 2 to R of the phase, we fix this issue by growing the fraction of decided nodes back to about 1. As we will show, during this *healing* period, the ratio between different opinions remains essentially the same. Thus, looking over phases, we can say that the ratio $\frac{p_1}{p_i}$ starts with at least $1 + 1/n$ and gets essentially squared in each phase. That means, this ratio reaches 2 within $O(\log n)$ phases and from there to n in $O(\log \log n)$ additional phases. Due to the integralities, once the ratio passes n , it actually means that $p_i = 0$. When this happens for all the opinions $i \in \{2, \dots, k\}$, all of them have been filtered out and only the plurality opinion remains alive⁵. From there, within ad-

⁵One case say this is virtually the same as ‘survival of the fittest’ in Herbert Spencer’s terminology for describing Dar-

ditional $O(\log n / \log k)$ phases, the system reaches plurality consensus where all nodes hold the plurality opinion.

2.2 Analysis

We next provide the formal analysis, proving that:

Theorem 2.1. *The above algorithm reaches plurality consensus w.h.p. within $O(\log k \cdot \log n)$ rounds, assuming that the initial bias $p_1 - p_2 \geq \sqrt{C \log n / n}$, for a sufficiently large constant C . Furthermore, if initially $p_1 / p_2 \geq 1 + \delta$ for some constant $\delta > 0$, then the convergence will happen in fact within $O(\log k \log \log n + \log n)$ rounds, w.h.p.*

Let $\text{bias} = p_1 - p_2$ be the difference between the fraction of the plurality opinion and the second largest opinion. Finally, to keep track of the the ratio between the plurality opinion and the second largest opinion, let us define

$$\text{gap} = \min\left\{\frac{p_1}{\sqrt{10 \log n / n}}, \frac{p_1}{p_2}\right\} \quad (1)$$

Note that when $p_2 \leq \sqrt{10 \log n / n}$, there is a large constant gap between p_1 and each of the other opinions as $p_1 \geq \sqrt{C \log n / n}$. Since once all other opinions are below $\sqrt{10 \log n / n}$, we will not be able to bound their deviations from expected by an $(1 - o(1))$ factor, in such cases, we will try to base the analysis mainly on the fast growth of p_1 , rather than the growth of p_1 / p_2 . The above definition of gap allows us to unify the two statements.

The three main transitions of the protocol: Our protocol can be characterized by three main transitions: The first transition occurs during the first $O(\log n)$ phases, by the end of which, the relative gap between the plurality opinion and the second largest opinion exceeds a 2 factor. The second transition occurs within $O(\log \log n)$ additional phases, where all the non-plurality opinions become extinct. At that point, we will have $p_1 \geq 2/3$ and $\sum_{i \geq 2} p_i = 0$. Finally, the last transition occurs within $O(\log n / \log k)$ additional phases where all nodes hold the plurality opinion. The second and third transitions are proven in [Lemma 2.5](#) to [Lemma 2.8](#).

We begin by showing that in every phase, three conditions hold: two of these are safety conditions guaranteeing that the fraction of decided nodes is sufficiently large and that the initial bias is maintained; the third condition is a progress condition that guarantees the growth of the gap.

Lemma 2.2. *Consider a phase j . Let $\mathbf{p} = (p_1, p_2, \dots, p_k)$ denote the vector of opinions at the beginning of this phase and \mathbf{p}^{new} to denote the same vector at the end of the phase. Assume that $\sum_{i=1}^k p_i \geq 2/3$, $\text{bias} \geq \sqrt{C \log n / n}$ and $p_1 \leq 2/3$. Then, with high probability, we have:*

- (S1) $\sum_{i=1}^k p_i^{\text{new}} \geq 2/3$,
- (S2) $\text{bias}^{\text{new}} = p_1^{\text{new}} - p_2^{\text{new}} \geq \sqrt{C \log n / n}$, and most importantly
- (P) either $p_1^{\text{new}} \geq 2/3$ or $\text{gap}^{\text{new}} \geq \text{gap}^{1.4}$ where gap^{new} is defined as in Eq. (1) with $p_1^{\text{new}}, p_2^{\text{new}}$.

Proof. We begin with (S1). We first claim that due to the gap-amplification step of phase j , the fraction of decided nodes has been dropped down to $\Omega(1/k + \log n / n)$, win's *natural selection* concept [Spe64]. In our setting, the largest opinion is effectively the fittest.

and then show that this drop is recovered during the $R = O(\log k)$ rounds of the healing step. To see this, note that the expected fraction of decided nodes at the end of the gap amplification round is $\sum_{i=1}^k p_i^2$. Since $\sum_{i=1}^k p_i \geq 2/3$, we have $\sum_{i=1}^k p_i^2 \geq k \cdot ((\sum_{i=1}^k p_i) / k)^2 \geq 4 / (9k)$. In addition, since $p_1 \geq \sqrt{\log n / n}$, $p_1^2 = \Omega(\log n / n)$. We thus get that $\sum_{i=1}^k p_i^2 = \Omega(1/k + \log n / n)$. We now claim that the $\Omega(1/k + \log n / n)$ fraction of decided nodes after the gap-amplification is increased to at least $2/3$ over the R rounds of the healing process. The argument is simple and similar to the usual process of gossiping a rumor. Particularly, in every round $r \in \{2, \dots, R\}$ of phase j , so long as the fraction of decided nodes is less than $2/3$, this fraction grows in expectation by a factor of at least $4/3$ and hence, it will grow by factor of $6/5$ with high probability. We therefore get that within $O(\log k)$ rounds, the fraction of decided nodes $\sum_{i=1}^k p_i$ reaches at least $2/3$ as required. Thus, property (S1) holds.

Next, we prove Claim (P) and then, we reuse some of the arguments to establish (S2). For (P), we distinguish between two cases depending on the magnitude of the second largest opinion p_2 .

Case (P.a), when $p_2 \geq \sqrt{10 \log n / n}$: After the first gap-amplification round, in expectation, there are np_1^2 nodes with opinion 1 and np_2^2 nodes with opinion 2. We want to show that the ratio between the actual numbers is very close to what we would have according to these expectations, i.e., $(\frac{p_1}{p_2})^2$, and remains so throughout rounds 2 to R . Since we might start with a bias as small as $\text{bias} = \Theta(\sqrt{\log n / n})$, the deviations from expectations should be analyzed carefully. Throughout, we focus on the two leading opinions of this phase and denote by the random variables x_r and y_r , the number of nodes in these two opinions at round $r \in \{1, \dots, R\}$ of this phase. By Chernoff, it holds that w.h.p.

$$\begin{aligned} x_1 &\in np_1^2 \left(1 \pm \frac{\sqrt{5 \log n / n}}{p_1}\right) \quad \text{and} \\ y_1 &\in np_2^2 \left(1 \pm \frac{\sqrt{5 \log n / n}}{p_2}\right). \end{aligned} \quad (2)$$

We now use this to bound the ratio x_1 / y_1 :

$$\begin{aligned} \frac{x_1}{y_1} &\geq \left(\frac{p_1}{p_2}\right)^2 \cdot \left(1 + \frac{3\sqrt{5 \log n / n}}{p_2}\right)^{-2} \\ &\geq \left(\frac{p_1}{p_2}\right)^2 \left(1 + \frac{p_1 - p_2}{p_2}\right)^{-0.2} = \left(\frac{p_1}{p_2}\right)^{1.8}, \end{aligned} \quad (3)$$

where the last inequality follows because at the start of the phase we have $\text{bias} = p_1 - p_2 \geq \sqrt{C \log n / n}$. We next argue that throughout rounds $r \in \{2, \dots, R\}$, this ratio between the plurality opinion and the second opinion remains almost the same, except for minimal deviations similar to above which would be tolerable and still leave the gap amplified to at least $(p_1 / p_2)^{1.4}$.

First, we provide an initial intuition about the change in the number of decided nodes of each of these opinions throughout rounds $r \in \{2, \dots, R\}$ and the potential deviations. Let us use (the random variable) q_r to denote the fraction of undecided nodes at the end of round r . For each r , we expect $q_r x_r$ and $q_r y_r$ new nodes to adopt, respectively,

the plurality opinion and the second largest opinion. Thus, $\mathbb{E}[x_{r+1}] = x_r(1 + q_r)$ and $\mathbb{E}[y_{r+1}] = y_r(1 + q_r)$. Here, the expectation is based on the randomness in round $r + 1$. If the random variables happen to be sharply around their expectations, the ratio remains essentially preserved. We next argue that this is true even despite the possible deviations.

We break the argument into two parts: Let r^* be the first round in which the fraction of undecided nodes is below $1/2$ —i.e., r^* is the minimum r satisfying that $q_r < 1/2$. We first show in [Claim 2.3](#) that the ratio remains essentially preserved by round r^* , and then we use a reasoning with a similar flavor to argue in [Claim 2.4](#) that a similar thing happens even for up to round R .

Claim 2.3. *W.h.p., we have $\frac{x_{r^*}}{y_{r^*}} \geq (\frac{p_1}{p_2})^{1.7}$.*

Proof. For each round $r \in \{1, \dots, r^*\}$, Chernoff bound shows that the random variables of how many nodes get added to each of the opinions can deviate from their expectations by at most $\pm(\sqrt{5x_r q_r \log n})$ and $(\pm\sqrt{5y_r q_r \log n})$. Here, we have used that $x_r q_r \geq x_1/2 \geq 5 \log n$ and $y_r q_r \geq y_1/2 \geq 5 \log n$. Hence, we can say w.h.p.,

$$\begin{aligned} \frac{x_{r+1}}{y_{r+1}} &\geq \frac{x_r(1 + q_r) - \sqrt{5x_r q_r \log n}}{y_r(1 + q_r) + \sqrt{5y_r q_r \log n}} \\ &\geq \frac{x_r}{y_r} \cdot \left(\frac{1 - \sqrt{5q_r \log n/x_r}}{1 + \sqrt{5q_r \log n/y_r}} \right). \end{aligned}$$

Let us define $\text{DEV}(x_r) = \sqrt{5q_r \log n/x_r}$ and $\text{DEV}(y_r) = \sqrt{5q_r \log n/y_r}$. Then, we can write

$$\frac{x_{r^*}}{y_{r^*}} \geq \frac{x_1}{y_1} \prod_{r=1}^{r^*} \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right).$$

Recall that we want to argue that despite the deviations of each round $r \in \{1, \dots, r^*\}$, the ratio remains essentially preserved. For that, we need to analyze the deviation product in the above inequality. Noting that since p_1 and p_2 are both at least $\Omega(\sqrt{\log n/n})$, we know that for each r , $\text{DEV}(x_r)$ and $\text{DEV}(y_r)$ are both smaller than 0.1. This allows us to write

$$\prod_{r=1}^{r^*} \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right) \geq 4^{-2(\sum_{r=1}^{r^*} \text{DEV}(x_r) + \sum_{r=1}^{r^*} \text{DEV}(y_r))}$$

Thus, we need to upper bound the summations $\sum \text{DEV}(x_r)$ and $\sum \text{DEV}(y_r)$. For convenience in doing this without repeating the argument, let us use $z \in \{x, y\}$ and consider the z_r for $r \in \{1, \dots, r^*\}$. We next show that $\sum_r \text{DEV}(z_r) = \sum_{r=1}^{r^*} \sqrt{5q_r \log n/z_r} = O(\frac{\sqrt{\log n}}{\sqrt{z_1}})$, with high probability. During rounds $r \in \{2, r^*\}$, in each round we expect z_{r+1} to be $z_r(1 + q_r) \geq 3z_r/2$. Since $z_r q_r = \Omega(\log n)$, with high probability, we will indeed have $z_{r+1} \geq 4z_r/3$. Hence, we have $\sum_{r=1}^{r^*} \sqrt{5q_r \log n/z_r} \leq \sum_{r=1}^{r^*} \sqrt{5 \log n/z_r} < O(\sqrt{5 \log n/z_1})$ where the last inequality holds because, during these rounds, the sum is a geometric series with a constant decay factor. Now we can use the inequality established in the above para-

graph to write:

$$\begin{aligned} \frac{x_{r^*}}{y_{r^*}} &\geq \frac{x_1}{y_1} \prod_{r=1}^{r^*} \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right) \\ &\geq \frac{x_1}{y_1} 4^{-2(\sum_{r=1}^{r^*} \text{DEV}(x_r) + \sum_{r=1}^{r^*} \text{DEV}(y_r))} \\ &\geq \frac{x_1}{y_1} 4^{-O(\sqrt{\log n/x_1} + \sqrt{\log n/y_1})} \\ &\geq \frac{x_1}{y_1} (1 - O(\sqrt{\log n/x_1} + \sqrt{\log n/y_1})) \\ &\geq \left(\frac{p_1}{p_2}\right)^{1.8} (1 - O(\sqrt{\log n/n p_1^2} + \sqrt{\log n/n p_2^2})) \\ &\geq \left(\frac{p_1}{p_2}\right)^{1.8} (1 + (p_1 - p_2)/p_2)^{-0.1} = \left(\frac{p_1}{p_2}\right)^{1.7} \end{aligned}$$

□

Claim 2.4. *W.h.p., we have $\frac{x_R}{y_R} \geq (\frac{p_1}{p_2})^{1.4}$.*

Proof. Notice that even in these rounds, it is true that per round r , $\mathbb{E}[x_{r+1}] = x_r(1 + q_r)$ and $\mathbb{E}[y_{r+1}] = y_r(1 + q_r)$. However, here $x_r q_r$ and $y_r q_r$ might drop below $O(\log n)$, which means every time when we write the additive deviations, we should still add a possible $O(\log n)$ term. Particularly, for each $r \in \{r^* + 1, \dots, R\}$ we can say w.h.p.,

$$\begin{aligned} \frac{x_{r+1}}{y_{r+1}} &\geq \frac{x_r(1 + q_r) - (\sqrt{3x_r q_r \log n} + 5 \log n)}{y_r(1 + q_r) + (\sqrt{5y_r q_r \log n} + 5 \log n)} \\ &\geq \frac{x_r}{y_r} \cdot \left(\frac{1 - (\sqrt{5q_r \log n/x_r} + 5 \log n/x_r)}{1 + \sqrt{5q_r \log n/y_r} + 5 \log n/y_r} \right). \end{aligned}$$

Let us define $\text{DEV}(x_r) = \sqrt{5q_r \log n/x_r} + 5 \log n/x_r$ and $\text{DEV}(y_r) = \sqrt{5q_r \log n/y_r} + 5 \log n/y_r$, for $r \in \{r^* + 1, \dots, R\}$. Then, we can write

$$\frac{x_R}{y_R} \geq \frac{x_{r^*}}{y_{r^*}} \prod_{r=r^*+1}^R \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right).$$

Noting that since p_1 and p_2 are both at least $\Omega(\sqrt{\log n/n})$, we know that for each r , $\text{DEV}(x_r)$ and $\text{DEV}(y_r)$ are both smaller than 0.1. This allows us to write

$$\prod_{r=r^*+1}^R \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right) \geq 4^{-(\sum_{r=r^*+1}^R \text{DEV}(x_r) + \sum_{r=r^*+1}^R \text{DEV}(y_r))}$$

Thus, we need to upper bound the summations $\sum \text{DEV}(x_r)$ and $\sum \text{DEV}(y_r)$ in the above inequality. Again, for convenience in doing this without repeating the argument, let us use $z \in \{x, y\}$ and consider the z_r for $r \in \{r^* + 1, \dots, R\}$. We next show that w.h.p.,

$$\begin{aligned} \sum_{r=r^*+1}^R \text{DEV}(z_r) &= \sum_{r=r^*+1}^R \sqrt{5q_r \log n/z_r} + 5 \log n/z_r \\ &= O\left(\frac{\sqrt{\log n}}{\sqrt{z_{r^*}}} + \frac{\log k \log n}{z_{r^*}}\right). \end{aligned}$$

First, note that the z_r values cannot decrease in the healing phase, and hence for each $r' \in \{r^*, \dots, R\}$, we have

$z_{r'} \geq z_{r^*}$. Thus,

$$\begin{aligned} & \sum_{r=r^*}^R (\sqrt{5 \log n q_r / z_r} + 5 \log n / z_r) \leq \\ & \sqrt{5 \log n / z_{r^*}} \cdot \sum_{r=r^*+1}^R (\sqrt{q_r}) + 5R \log n / z_{r^*}. \end{aligned}$$

We next bound the term $\sum_{r=r^*+1}^R (\sqrt{q_r})$. Note that in round $r^* + 1$, we expect the number of undecided nodes to change from $n/2$ to $n/4$ and with high probability, it will be at most $n/3$. Similarly, so long as $nq_r \geq \Omega(\log n)$, with high probability, we have $q_{r+1} \leq 3q_r/4$. Thus, the summation $\sum_{r=r^*+1}^R (\sqrt{q_r})$ is also a summation of a geometric series, so long as $q_r \geq \Omega(\log n/n)$. For rounds where $q_r \leq O(\log n/n)$, the summation of $\sqrt{q_r}$ is at most $O(\log^2 n/n) = o(1)$. Therefore, we get that $\sum_{r=r^*}^R \sqrt{q_r} \leq 4q_{r^*} + o(1) \leq 5$. This means: $\sum_{r=r^*+1}^R (\sqrt{5 \log n q_r / z_r} + 5 \log n / z_r) \leq 10\sqrt{\log n / z_{r^*}} + 5R \log n / z_{r^*}$.

From the above, it thus follows that the $\sum_{r=r^*+1}^R \text{DEV}(x_r) + \sum_{r=r^*+1}^R \text{DEV}(y_r) \leq O(\frac{\sqrt{\log n} + \log k \log n}{\sqrt{x_{r^*}}}) + O(\frac{\sqrt{\log n} + \log k \log n}{\sqrt{y_{r^*}}})$. We know that w.h.p. $x_{r^*} = \Omega(np_1)$ and $y_{r^*} = \Omega(np_2^2/p_1)$. The reason for this is as follows: Note that at round r^* , the total fraction of decided nodes has dropped below $1/2$, which means, ignoring a 2-factor, x_{r^*} is the same as the fraction among decided nodes that have opinion 1. Now, at the end of the gap amplification round, this fraction is w.h.p. $\frac{\Theta(np_1^2)}{\Theta(\sum_{i=1}^k np_i^2)} \geq \frac{\Theta(np_1^2)}{\Theta(np_1)} \geq \Omega(p_1)$. Now, during the rounds $r \in \{1, \dots, r^*\}$, even in a coarse and pessimistic estimate, this fraction deviates by at most a constant factor, because of the deviation factor analysis that we provided above for rounds $r \in \{1, \dots, r^*\}$. Hence, in round r^* , about $\Omega(p_1)$ fraction of decided nodes have opinion 1, which since at r^* more than $n/2$ nodes are decided, means $x_{r^*} = \Omega(np_1)$. A similar argument can be used to show that $y_{r^*} = \Omega(n \frac{p_2^2}{\sum_{i=1}^k p_i^2}) = \Omega(np_2^2/(p_1^2 + p_2)) = \Omega(\min\{np_2^2/p_1^2, np_2\})$.

For simplicity, let us now distinguish between two cases, regarding the value of p_1/p_2 . First, we consider the possibly sensitive regime when $p_1/p_2 \in [1, 500]$. In the complementary case, the concentrations we need will be far more relaxed. For the case $p_1/p_2 \in [1, 500]$, from above we get that

$$\begin{aligned} \frac{x_R}{y_R} & \geq \frac{x_{r^*}}{y_{r^*}} \cdot \prod_{r=r^*+1}^R \left(\frac{1 - \text{DEV}(x_r)}{1 + \text{DEV}(y_r)} \right) \\ & \geq \frac{x_{r^*}}{y_{r^*}} \cdot 4^{-(\sum_{r=r^*+1}^R \text{DEV}(x_r) + \sum_{r=r^*+1}^R \text{DEV}(y_r))} \\ & \geq \frac{x_{r^*}}{y_{r^*}} \cdot 4^{-O(\frac{\sqrt{\log n} + \log k \log n}{\sqrt{x_{r^*}}} + \frac{\sqrt{\log n} + \log k \log n}{\sqrt{y_{r^*}}})} \\ & \geq \frac{x_{r^*}}{y_{r^*}} \cdot 4^{-O(\frac{\sqrt{\log n} + \log k \log n}{\sqrt{np_1}} + \frac{\sqrt{\log n} + \log k \log n}{\sqrt{np_2}})} \\ & \geq \frac{x_{r^*}}{y_{r^*}} \cdot 4^{-O(\frac{\sqrt{\log n} + \log k \log n}{\sqrt{np_2}})} \geq \frac{x_{r^*}}{y_{r^*}} \cdot 4^{-O(\frac{\sqrt{\log n}}{\sqrt{np_2^2}})} \\ & \geq \frac{x_{r^*}}{y_{r^*}} \cdot (1 - O(\frac{\sqrt{\log n}}{\sqrt{np_2^2}})) \\ & \geq \left(\frac{p_1}{p_2}\right)^{1.7} \left(1 + \frac{p_1 - p_2}{p_2}\right)^{-0.3} = \left(\frac{p_1}{p_2}\right)^{1.4} \end{aligned}$$

Now suppose that $p_1/p_2 > 500$. Here, it suffices to bound the deviations by a small constant. It is easy to see that w.h.p. $y_R \leq 5y_{r^*} + O(\log^2 n)$, because during these rounds, as discussed above, q_r decays exponentially with r . Thus, $\frac{x_R}{y_R} \geq \frac{x_{r^*}}{y_{r^*}} \geq \frac{x_{r^*}}{5y_{r^*} + O(\log^2 n)} \geq \frac{x_{r^*}}{y_{r^*}} / (5 + o(1)) > \left(\frac{p_1}{p_2}\right)^{1.7} / 5.1 \geq \left(\frac{p_1}{p_2}\right)^{1.4}$. This completes the proof of (P3.a). Next, we consider the complementary case where p_2 is small. \square

Case (P.b), when $p_2 \leq \sqrt{10 \log n/n}$: Now we come back to the case that $p_2 = \max_{i=2}^k p_i \leq \sqrt{10 \log n/n}$. Note that in this case, the multiplicative factor deviations of the opinion 2 from its expected values might be large. Recall the definition $\text{gap} = \min\{\frac{p_1}{\sqrt{10 \log n/n}}, \min_{i=2}^k \frac{p_i}{p_i}\}$. Also, note that by the above arguments, the ratio between $\frac{p_1}{p_2}$ will grow to at least $\text{gap}^{1.4}$ because that would happen even if we manually raise the initial p_2 to $\sqrt{10 \log n/n}$, thus allowing us to use the above concentration arguments, and the whole process is clearly monotonic meaning that raising the initial p_2 cannot increase the ratio $\frac{p_1^{new}}{p_2^{new}}$. However, to show that gap^{new} increases to $\text{gap}^{1.4}$, in this regime of $p_2 \leq \sqrt{10 \log n/n}$, the minimizer in the definition of gap is the $\frac{p_1}{\sqrt{10 \log n/n}}$ term which means we should show that p_1 will grow accordingly.

Here, since $p_1 \geq \sqrt{C \log n/n}$, we have $\text{gap} \geq \frac{\sqrt{C \log n/n}}{\sqrt{10 \log n/n}} \gg 20$. On the other hand, by repeating the arguments given above, we know that p_1^{new} will be within at most a $1 \pm o(1)$ factor of its expectation, which is $\frac{p_1^2}{\sum_{i=1}^k p_i^2} = \frac{p_1^2}{p_1^2 + \sum_{i=2}^k p_i^2} \geq \frac{p_1^2}{p_1^2 + p_i^*}$. Thus, the growth factor from p_1 to p_1^{new} is at least $(1 - o(1)) \frac{p_1}{p_1^2 + p_2} \geq \frac{1 - o(1)}{p_1 + 1/\text{gap}}$. If $1/\text{gap} \leq 0.45p_1$, then $p_1^{new} > 2/3$. Otherwise, the growth is at least $\frac{1 - o(1)}{3.3/\text{gap}} \geq \text{gap}/4 \geq \text{gap}^{1/2}$, which finishes the proof of case (P.b) and thus also property (P).

Property (S2): Finally, we use the above concentration arguments to prove property (S2), by showing that w.h.p. $\text{bias}^{new} \geq p_1 - p_2$. First suppose that $\frac{p_1}{p_2} < 10$. Then, we know that p_1^{new} and p_2^{new} will be within $1 + o(1)$ factors of their expectations (as analyzed above), which are, respectively, $\frac{p_1^2}{\sum_{j=1}^k p_j^2}$ and $\frac{p_1^2}{\sum_{j=1}^k p_j^2}$. Thus,

$$\begin{aligned} p_1^{new} - p_2^{new} & \geq (1 - o(1)) \frac{p_1^2}{\sum_{j=1}^k p_j^2} - \frac{p_1^2}{\sum_{j=1}^k p_j^2} \\ & = (1 - o(1))(p_1 - p_2) \cdot \frac{(p_1 + p_2)}{\sum_{j=1}^k p_j^2} \\ & \geq (1 - o(1))(p_1 - p_2) \cdot \frac{(p_1 + p_i)}{p_1 \sum_{j=1}^k p_j} \\ & \geq (1 - o(1))(p_1 - p_2) \cdot \frac{(p_1 + p_2)}{p_1} \\ & \geq (1.1 - o(1))(p_1 - p_2). \end{aligned}$$

On the other hand, when $\frac{p_1}{p_2} > 10$, we know the $\frac{p_1^{new}}{p_2^{new}} \geq \text{gap}^{new} \geq \text{gap}^{1.4} > 2.5\text{gap}$. We also know that w.h.p. p_1 does not decrease in this case. This implies that $\text{bias} = p_1^{new} - p_2^{new}$ also does not decrease, thus proving property (S2). \square

Lemma 2.5. *After $O(\log n)$ phases from the start, we have $\text{gap} \geq 2$. If initially we had $p_1 > (1 + \delta)p_2$ for some constant δ , then in fact within $O(1)$ phases we would have $\text{gap} \geq 2$.*

Proof. Initially,

$$\text{gap} \geq \min\left\{\frac{p_1}{\sqrt{10 \log n/n}}, \min_{i=2}^k \frac{p_1}{p_i}\right\} \geq \min\left\{3, \frac{p_1}{p_2}\right\} > 1 + \frac{1}{\sqrt{n}}.$$

Note that once $p_1 \geq 2/3$, we also have $\text{gap} \geq 2$. Define $\gamma = \text{gap} - 1$. In every phase that $p_1^{new} \geq 2/3$ does not happen, [Lemma 2.2](#) gives $\text{gap}^{new} \geq \text{gap}^{1.4}$. This implies $\gamma^{new} \geq (1 + \gamma)^{1.4} - 1 \geq 6\gamma/5$. Hence, after $O(\log n)$ phases, γ becomes larger than 1 which means $\text{gap} \geq 2$. If initially we had $p_1 > (1 + \delta)p_2$, then initially $\text{gap} \geq 1 + \delta$ and thus it takes it only $O(1)$ phases to reach $\text{gap} \geq 2$. \square

Lemma 2.6. *If at the start of a phase $p_1 \geq \frac{2}{3}$, then w.h.p. we have $p_1^{new} \geq \frac{2}{3}$ also at the end of it.*

Proof. At the end of the gap amplification round, w.h.p., the fraction of decided nodes of opinion 1 is $(1 - o(1))4/9$ and the fraction of decided nodes of all other opinions is at most $(1 + o(1))1/9$. That is, w.h.p., the fraction among decided nodes that have opinion 1 is at least $(1 - o(1))\frac{4/9}{4/9 + 1/9} > 3/4$. During each round r of the healing rounds, both the fraction of plurality nodes and the total fraction of decided nodes are expected to grow by $(1 + q_r)$, where q_r is the fraction of undecided nodes at the start of that round. Since the number of plurality opinion nodes and also the total number of decided nodes are $\Omega(n)$, the actual growth in the two fractions will be within $1 \pm o(\frac{1}{\log n})$ factor of their expectations. Thus, the fraction among the decided node that have opinion 1 remains preserved up to at most a $1 \pm o(\frac{1}{\log n})$ factor. This means, even after all the $O(\log k)$ rounds of healing, the fraction among the decided nodes that have opinion 1 is at least $(1 - o(1/\log n))^{O(\log k)}3/4 = (1 - o(1))3/4$. On the other hand, at that point the probability for a node to remain undecided is at most $(1/3)^{O(\log k)} \leq 0.01/k$, which means w.h.p., at least 0.99 fraction of nodes will be decided. Hence, w.h.p., at least $0.98 \times (1 - o(1))3/4 > 2/3$ fraction of nodes will have opinion 1. \square

Lemma 2.7. *After $\text{gap} \geq 2$, within $O(\log \log n)$ phases, w.h.p., we have $p_1 \geq 2/3$ and $\sum_{i \geq 2} p_i = 0$.*

Proof. By [Lemma 2.2](#), in each phase j , either we get $p_1 \geq 2/3$ or $\text{gap}^{new} \geq \text{gap}^{1.4}$. Since we start with a $\text{gap} \geq 2$, within $O(\log \log n)$ phases, the latter condition cannot continue to hold which means we would have $p_1 \geq 2/3$. At this point, by [Lemma 2.6](#), it will always remain true that $p_1 \geq 2/3$.

Then, for any other opinion $i \in [2, k]$, in each phase, the change satisfies the following inequality: $\mathbb{E}[p_i^{new}] = \frac{p_i^2}{\sum_{i=1}^k p_i^2} \geq \frac{p_i^2}{4/9} = 2.25p_i^2$. Here, the expectation is based only on the randomness of this phase. Hence, within $O(\log \log n)$ phases, we expect p_i to fall below $1/n^c$, for any desirable constant c . By Markov's inequality, this means at that point, we actually have $p_i = 0$ w.h.p. \square

We proceed by showing that from that point on, the plurality opinion becomes the consensus.

Lemma 2.8. *After $p_1 \geq 2/3$ and $\sum_{i \geq 2} p_i = 0$, within $O(\log n / \log k)$ phases, w.h.p., $p_1 = 1$.*

Proof. Consider the first phase after we have $p_1 \geq 2/3$ and $\sum_{i \geq 2} p_i = 0$. First note that by [Lemma 2.6](#), w.h.p. $p_1 \geq 2/3$ from now on. Now, let q be the fraction of undecided nodes. Notice that after the first round, q increases slightly to q' where $\mathbb{E}[q'] = 1 - (1 - q)^2 \leq 2q$. During the next $R - 1 = O(\log k)$ rounds of this phase, a node remains undecided only if all nodes that it contacted where undecided and that has probability at most $(5/9)^{R-1} \leq 1/(4k)$. Thus, $\mathbb{E}[q^{new}] \leq \frac{q}{2k}$. This expectation relies only on the randomness of this phase. Thus, the expected value of q after $O(\log n / \log k)$ phases is $1/n^c$ —for any arbitrary c , by adjusting the constant in the number of phases. Hence, by Markov's inequality, the probability that $q \neq 0$, which would imply $q \geq 1/n$, is at most $1/n^{c-1}$. Thus, w.h.p., $q = 0$ and $p_1 = 1$. Clearly, from that point onwards, it remains true that $p_1 = 1$. \square

We are now ready to complete the proof of [Theorem 2.1](#)

Proof of Theorem 2.1. By [Lemma 2.5](#), within $O(\log n)$ phases from the beginning of the execution, $\text{gap} \geq 2$. In fact if we initially have $p_1 \geq (1 + \delta)p_2$ for a constant δ , this part takes $O(1)$ phases. Then, after additional $O(\log \log n)$ phases, by [Lemma 2.5](#), we have $p_1 \geq 2/3$ and all other nodes are undecided. Finally, by [Lemma 2.8](#), w.h.p., within $O(\log n / \log k)$ phases, plurality achieves totality. \square

3. THE ALGORITHM: TAKE 2

In the algorithm presented in the previous section, each node has a memory size of $\log k + \log \log k + O(1)$ bits, where $\log \log k + O(1)$ bits of it are used solely to remember the round number, modulo $R = O(\log k)$. Here, we explain how to remove this $\log \log k$ overhead. Thus, we get a dynamics in which each node has at most $O(k)$ states—i.e., within constant factor of the trivially optimal—instead of the $O(k \log k)$ states of the previous algorithm.

Intuitive Discussions: If we are to reduce the memory from $\log k + \log \log k + O(1)$ to $\log k + O(1)$, we cannot require each node to know both an opinion and the round number modulo R . Nodes not knowing the time create the risk that nodes might not be “in sync” in regards to whether at a given point of time, they should be performing *gap amplification* or *healing*. We try to minimize this effect and keep the vast majority of the nodes essentially “in sync”.

Notice that when we allow memory size of $\log k + O(1)$, we have (in fact far more than) enough space for keeping the time modulo R ; we just cannot ask each node to keep both time and an opinion. To overcome this, we simply *split the responsibilities among the nodes*: particularly, at the start, each node tosses a fair random coin and decides to be a *clock-node* or a *game-player*, each with probability $1/2$. Clock-nodes forget their initial opinion and instead keep the time (say modulo $4R$, thus needing at most $\log k + O(1)$ bits). On the other hand, the game-players will try to follow the approach of the take 1 algorithm, using the help of the clock-nodes.

There are three subtleties this intuitive idea: First, note that now the *game-players*, who actually play the main role in getting to plurality consensus, do not know the time and

Algorithm 1 The algorithm run at each game-player node v

```

In each round, node  $v$  does as follows (assuming  $u$  is the
node that  $v$  contacted):
1: if  $u.isClock = true$  then
2:   if  $phase_v \neq \text{end-game}$  or ( $phase_v = \text{end-game}$  and
 $phase_u = 0$ ) then
3:      $phase_v \leftarrow phase_u$ ;
4: else
5:   switch  $phase_v$  do
6:     case 0: - Time Buffer 1-
7:        $sampld \leftarrow false$ ;  $forget \leftarrow false$ ;
8:     case 1: - Gap Amplification (Sampling) -
9:       if  $sampld = false$  and  $v.opinion \neq$ 
 $u.opinion$  then  $forget \leftarrow true$ ;
10:       $sampld \leftarrow true$ ;
11:     case 2: - Time Buffer 2-
12:      if  $forget = true$  then  $v.opinion \leftarrow 0$ ;
 $forget \leftarrow false$ ;
13:     case 3: - Healing -
14:      if  $v.opinion = 0$  then  $v.opinion \leftarrow$ 
 $u.opinion$ ;
15:       $sampld \leftarrow false$ ;  $forget \leftarrow false$ ;
16:     case end-game: - End Game -
17:      if  $v.opinion \neq 0$  and  $v.opinion \neq u.opinion$ 
then  $v.opinion \leftarrow 0$ ;
18:      if  $v.opinion = 0$  then  $v.opinion \leftarrow$ 
 $u.opinion$ ;

```

thus cannot perform the take 1 algorithm *per se*. Furthermore, they cannot really expect to receive the precise time from the clock-nodes and remember it, as that would again require $\log k + \log \log k + O(1)$ memory bits. We will make the clock nodes report the time only in (integer) factors of $R = O(\log k)$, and modulo say $4R$. That is, the reported phase number is always in $\{0, 1, 2, 3\}$. Reporting these numbers and remembering them will thus require only $O(1)$ bits. Second, now the *game-players* do not really know the precise time, and thus they cannot update the phase numbers on their own. They will have to instead rely completely on receiving phase-updates from *clock-nodes*. Third, we want that eventually all the nodes to hold the plurality opinion, including the *clock-nodes*. Hence, each clock-node needs to at some point forget it's *time-keeping responsibilities* and try to learn the plurality opinion. This has to be done in a judicious manner so that the system as a whole has enough time-keepers, so long as they are needed. We next present the algorithm that deals with these challenges.

The Algorithm Description: The algorithm performed by the game-player nodes is described in [Algorithm 1](#) and the algorithm performed by the clock-nodes is presented in [Algorithm 2](#).

We next provide some intuitive explanation for what these algorithms are doing. The algorithm run the by game-player nodes can be viewed as *interpolating* between two algorithms: the protocol presented in the previous section, which we will call hereafter the Gap Amplification (GA) protocol, and the Undecided State (Undecided) protocol⁶ as used in [\[BCN⁺15a\]](#). We have the Undecided protocol for the time that the number of clock-nodes that are following their time-keeping responsibilities goes down, and thus we

⁶We note that in fact any other protocol with a minimal plurality amplification behavior would suffice for this part; we choose the Undecided protocols just for convenience.

cannot follow the GA protocol. In reality, as long as p_1 is far from 1, there will be a large population of clock-nodes that are still performing their time-keeping responsibilities, and thus the GA protocol will be the dominant part of the system (formalizing this will take some care). Hence, during that time, the system will enjoy the fast gap amplification behavior of GA. Once p_1 gets close to 1, the number of active clock-nodes drops and the game-players gradually move to the Undecided protocol. At this time, we no longer need the power of the GA protocol; regardless of how game-player nodes are split between the two protocol, as we will show, the system evolves almost as good as the Undecided protocol and gets to plurality consensus fast.

Let us now discuss the GA protocol, which is the core of the algorithm of the game-players, and it is essentially a simulation of the take 1 algorithm. We have a few small changes, in comparison to the take 1 algorithm, which are added to deal with the subtleties described above: Particularly, since the game-player nodes will not be able to know the precise time, to mitigate the effects of asynchrony, we extend the *gap amplification* to a full phase (instead of one round) and we also add some time buffer between the *gap amplification* period and the *healing* period. Thus, now the execution of the algorithm can be broken to intervals made of 4 consecutive phases, which we call a “long-phase”. Particularly, a long-phase is made of 4 phases, the two main ones being gap amplification and healing phases, interspersed with two (essentially) time buffer phases. Although the gap amplification period is now one full phase, we still make each node act based on a single sample: that is, in the first round in this phase that the game-player node v contacts another game-player u , node v decides whether to keep its opinion or not, and it remains with this decision. If v decided to forget its opinion, then it remembers this decision (via a Boolean variable called *forget*) and once v sees the next phase (by meeting a clock-node of phase 2), it becomes undecided. With a slight informality, from now on, we will refer to the protocol described by phases $\{0, 1, 2, 3\}$ as the GA (gap amplification) protocol. A game-player node goes out of the GA protocol only if it meets a clock-node that has moved to the end-game (to be explained while discussing the algorithm of the clock-nodes). Game-players that move out of the GA protocol go to the Undecided protocol, as described the pseudocode in the **end-game** switch case. However, each such game-player node might come back to the GA protocol later on, if it meets a clock-node that reports phase 0.

Let us now discuss the algorithm for the clock-nodes. As mentioned above, an important challenge is for the clock-nodes to recognize that the game-players have reached consensus, so that they can switch to adopt an opinion then. Since the GA protocol might take $O(\log n \log k)$ rounds to reach plurality consensus, we cannot require the clock-nodes to keep the precise time until this period has passed, as that would require $\Omega(\log \log n)$ memory bits. Instead, we make the clock-nodes use the existence of undecided nodes as a signal to detect that the game-players still have not reached consensus. Although this is not a perfect check, it will be good enough for our purposes, i.e., so long as p_1 is far from 1. Particularly, note that in the GA protocol, so long as we are far from reaching consensus (while p_1 is far from 1), there will be a considerable fraction of undecided nodes (at some point during at least one of the phases). We make the clock-nodes remain active—i.e., keeping time—as long

Algorithm 2 The algorithm run at each clock-node v

In each round, node v does as follows (assuming u is the node that v contacted):

```
1: switch  $v.status$  do
2:   case counting:                - Time Keeping -
3:      $v.opinion \leftarrow \text{null}$ ;
4:      $v.time \leftarrow v.time + 1 \bmod 4R$ ;
5:      $v.phase \leftarrow \lceil v.time/R \rceil \bmod 4$ ;
6:     if  $u.isClock = \text{false}$  and  $u.opinion = 0$  then
        $v.consensus \leftarrow \text{false}$ ;
7:     if  $u.isClock = \text{true}$  and  $u.consensus = \text{false}$ 
       then  $v.consensus \leftarrow \text{false}$ ;
8:     if  $v.time \bmod 4R = 0$  then
9:       if  $v.consensus = \text{true}$  then  $v.status \leftarrow$ 
       end-game;
10:       $v.consensus \leftarrow \text{true}$ ;
11:     case end-game:                - End Game -
12:       $v.time \leftarrow \text{null}$ ;
13:       $v.phase \leftarrow \text{end-game}$ ;
14:      if  $u.isClock = \text{false}$  then  $v.opinion \leftarrow$ 
        $u.opinion$ 
15:      else if  $u.status = 0$  and  $u.consensus = \text{false}$ 
       then
16:         $v.status \leftarrow \text{counting}$ ;
17:         $v.opinion \leftarrow \text{null}$ ;
18:         $v.time \leftarrow u.time$ ;  $v.phase \leftarrow u.phase$ ;
        $v.consensus \leftarrow u.consensus$ ;
```

as they meet an undecided node directly or indirectly. Here, indirectly means by meeting a clock-node that is aware of the existence of undecided nodes. This information spreading essentially in the usual gossip spreading style and via the boolean variable *consensus*, for which *consensus* = *false* indicates that an undecided node still exists, i.e., an indication for that the system has not reached plurality consensus yet. If during a long-phase, a clock-node receives no information about the existence of undecided nodes, it neglects its time keeping task, by forgetting time, and attempts to move to the end-game reporting phase (possibly temporarily). Here, the phase number it reports is a special symbol “end-game”. Also, at this time, this clock-node that now has forgotten time adopts an opinion, always remembering the opinion of the last game-player that it met. However, this clock-node that has become “deactivated” from time-keeping by switching to the end-game mode might get reactivated again: particularly, if this “deactivated” clock-node v meets an active clock-node u who is aware of the existence of undecided nodes (has $u.consensus = \text{false}$), then v gets reactivated again, keeping track of time, using the clock that it gets from u , and reporting the phase number in $\{0, 1, 2, 3\}$.

The analysis of the algorithm is provided in the full version. We next provide some intuition for the analysis, for the cases in which $k \geq n^{0.1}$. The analysis for the complementary case of $k < n^{0.1}$ also follows a similar outline, roughly speaking, but typically requires more care.

Analysis Intuition, for when $k \geq n^{0.1}$: Here the length of each phase is $R = \Theta(\log k) = \Theta(\log n)$, which allows us to say that some good events happen w.h.p. without needing to keep track of their precise probability, e.g., this is sufficient for information to spread to the whole system.

Specifically, the first analysis step is to establish that as long as $p_1 \leq 1 - \Theta(\log n/n)$, all clock-nodes will continue to perform their time-keeping responsibilities. Thus, during this period, all game-player nodes will be performing the GA

protocol without going to the end-game of the **Undecided** protocol. The (formally inductive) argument for why all clock-nodes will remain is roughly as follows: Consider one long-phase (4 consecutive phases). Because game-player nodes are following the GA protocol and there are $\Omega(\log n)$ non-plurality nodes, during phase 2 of this long-phase, there will be $\Theta(\log n)$ undecided nodes w.h.p. Thus, w.h.p., there will be clock-nodes that notice this undecided population. This information about the existence of undecided nodes will spread among the clock-nodes following essentially the pattern of a usual rumor spreading. Particularly, w.h.p. within $O(\log n)$ rounds all clock-nodes will be aware of the existence of undecideds and thus, at the end of the long-phase, none of the clock-nodes switches to the end-game mode. Given this, during this time of $p_1 \leq 1 - \Theta(\log n/n)$, all game-players are acting according to the GA protocol. This allows us to use the analysis of take 1, modulo minor details, and argue that in $O(\log n)$ phases, w.h.p., we reach $p_1 \geq 1 - \Theta(\log n/n)$.

Once p_1 reaches $1 - \Theta(\log n/n)$, some clock-nodes might start to switch to the end-game mode and thus some game-player nodes might act based on the **Undecided** protocol. However, now that p_1 is very close to 1, we do not need the power of GA. Here, even the **Undecided** protocol would be pushing us towards consensus on plurality quickly enough, and we can actually prove that the system (which is an uncontrolled mix of GA and **Undecided**) does not behave worse than the worst of them. Particularly, we can first see that w.h.p. the system remains in this regime of p_1 very close to 1, and then use this to establish that non-plurality opinions get filtered out within $O(1)$ phases and that plurality reaches totality among game-players within $O(1)$ phases. By the end of that time, all clock-nodes switch to the end-game, as they will not see undecided game-players anymore. Since clock-nodes in the end-game adopt the opinion of the last game-player they have seen, within $O(1)$ phases, all nodes will hold the plurality opinion.

4. REFERENCES

- [AAD⁺06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.
- [AAE08] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008.
- [AFJ06] Dana Angluin, Michael J Fischer, and Hong Jiang. Stabilizing consensus in mobile networks. In *Distributed Computing in Sensor Systems*, pages 37–50. Springer, 2006.
- [AGV15] Dan Alistarh, Rati Gelashvili, and Milan Vojnovic. Fast and exact majority in population protocols. In *the Proc. of the Int’l Symp. on Princ. of Dist. Comp. (PODC)*, pages 47–56, 2015.
- [BCN⁺14] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, Riccardo Silvestri, and Luca Trevisan. Simple dynamics for plurality consensus. In *Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures*, pages 247–256. ACM, 2014.

- [BCN⁺15a] L Becchetti, A Clementi, E Natale, F Pasquale, and R Silvestri. Plurality consensus in the gossip model. In *Pro. of ACM-SIAM Symp. on Disc. Alg. (SODA)*, pages 371–390, 2015.
- [BCN⁺15b] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Stabilizing consensus with many opinions. *to appear at SODA'16, arXiv preprint arXiv:1508.06782*, 2015.
- [BFGK16] Petra Berenbrink, Tom Friedetzky, George Giakkoupis, and Peter Kling. Efficient plurality consensus, or: The benefits of cleaning up from time to time. In *the Pro. of the Int'l Colloquium on Automata, Languages and Programming (ICALP)*, page to appear, 2016.
- [BICS10] Radu Berinde, Piotr Indyk, Graham Cormode, and Martin J Strauss. Space-optimal heavy hitters with strong error bounds. *ACM Transactions on Database Systems (TODS)*, 35(4):26, 2010.
- [BSDDS14] Ohad Ben-Shahar, Shlomi Dolev, Andrey Dolgin, and Michael Segal. Direction election in flocking swarms. *Ad Hoc Networks*, 12:250–258, 2014.
- [CCN12] Luca Cardelli and Attila Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific reports*, 2, 2012.
- [CDS⁺13] Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from DNA. *Nature nanotechnology*, 8(10):755–762, 2013.
- [CKFL05] Iain D Couzin, Jens Krause, Nigel R Franks, and Simon A Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.
- [CSWB09] Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. Programmability of chemical reaction networks. In *Algorithmic Bioprocesses*, pages 543–584. Springer, 2009.
- [DGM⁺11] Benjamin Doerr, Leslie Ann Goldberg, Lorenz Minder, Thomas Sauerwald, and Christian Scheideler. Stabilizing consensus with the power of two choices. In *Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures*, pages 149–158. ACM, 2011.
- [Dot14] David Doty. Timing in chemical reaction networks. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–784. SIAM, 2014.
- [DV12] Moez Draief and Milan Vojnovic. Convergence speed of binary interval consensus. *SIAM Journal on Control and Optimization*, 50(3):1087–1109, 2012.
- [DW83] Peter Donnelly and Dominic Welsh. Finite particle systems and infection models. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 94, pages 167–182. Cambridge Univ Press, 1983.
- [Hop71] John Hopcroft. An $n \log n$ algorithm for minimizing states in a finite automaton. *Theory of Machines and Computations*, pages 189–196, 1971.
- [HP01] Yehuda Hassin and David Peleg. Distributed probabilistic polling and applications to proportionate agreement. *Information and Computation*, 171(2):248–268, 2001.
- [KDG03] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *Proc. of the Symp. on Found. of Comp. Sci. (FOCS)*, pages 482–491, 2003.
- [Lig12] Thomas Liggett. *Interacting particle systems*, volume 276. Springer Science & Business Media, 2012.
- [Lyn96] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.
- [MNRS14] George B Mertzios, Sotiris E Nikolettseas, Christoforos L Raptopoulos, and Paul G Spirakis. Determining majority in networks with local interactions and very small local memory. In *Automata, Languages, and Programming*, pages 871–882. Springer, 2014.
- [MS] Elchanan Mossel and Grant Schoenebeck. Reaching consensus on social networks.
- [PVV09] Etienne Perron, Dinkar Vasudevan, and Milan Vojnovic. Using three states for binary consensus on complete graphs. In *the Proc. of IEEE INFOCOM*, pages 2527–2535, 2009.
- [SKJ⁺08] David JT Sumpter, Jens Krause, Richard James, Iain D Couzin, and Ashley JW Ward. Consensus decision making by fish. *Current Biology*, 18(22):1773–1777, 2008.
- [Spe64] H. Spencer. *The Principles of Biology*. Number v. 1 in Spencer, Herbert: A system of synthetic philosophy. Williams and Norgate, 1864.
- [TBA86] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
- [Tsi84] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, DTIC Document, 1984.
- [WH04] Fang Wu and Bernardo A Huberman. Social structure and opinion formation. *arXiv preprint cond-mat/0407252*, 2004.
- [Yu05] Sheng Yu. State complexity: Recent results and open problems. *Fundamenta Informaticae*, 64(1-4):471–480, 2005.