Loosely-Stabilizing Phase Clocks and The Adaptive Majority Problem

Petra Berenbrink
Universität Hamburg, Germany

Felix Biermeier
Universität Hamburg, Germany

Christopher Hahn
Universität Hamburg, Germany

Dominik Kaaser
Universität Hamburg, Germany

Abstract

We present a loosely-stabilizing phase clock for population protocols. In the population model we are given a system of n identical agents which interact in a sequence of randomly chosen pairs. Our phase clock is leaderless and it requires $\mathrm{O}(\log n)$ states. It runs forever and is, at any point of time, in a synchronous state w.h.p. When started in an arbitrary configuration, it recovers rapidly and enters a synchronous configuration within $\mathrm{O}(n \log n)$ interactions w.h.p. Once the clock is synchronized, it stays in a synchronous configuration for at least $\mathrm{poly}(n)$ parallel time w.h.p.

We use our clock to design a loosely-stabilizing protocol that solves the adaptive variant of the majority problem. We assume that the agents have either opinion A or B or they are undecided and agents can change their opinion at a rate of $1/n$. The goal is to keep track which of the two opinions is (momentarily) the majority. We show that if the majority has a support of at least $\Omega(\log n)$ agents and a sufficiently large bias is present, then the protocol converges to a correct output within $\mathrm{O}(n \log n)$ interactions and stays in a correct configuration for $\mathrm{poly}(n)$ interactions, w.h.p.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Population Protocols, Phase Clocks, Loose Self-stabilization, Clock Synchronization, Majority, Adaptive

Digital Object Identifier 10.4230/LIPIcs.SAND.2022.7

Related Version *arXiv version*: `https://doi.org/10.48550/arXiv.2106.13002` [13]

1 Introduction

In this paper we introduce a loosely-stabilizing leaderless phase clock for the population model and demonstrate its usability by applying the clock to the comparison problem introduced in [3]. Population protocols have been introduced by Angluin et al. [5]. A population consists of n anonymous agents. A random scheduler selects in discrete time steps pairs of agents to interact. The interacting agents execute a state transition, as specified by the *algorithm* of the population protocol. Angluin et al. [5] gave a variety of motivating examples for the population model, including averaging in sensor networks, or modeling a disease monitoring system for a flock of birds. In [24] the authors introduce the notion of loose-stabilization. A population protocol is loosely-stabilizing if, from an arbitrary state, it reaches a state with correct output fast and remains in such a state for a polynomial number of interactions. In contrast, self-stabilizing protocols are required to converge to the correct output state from any possible initial configuration and stay in a correct configuration indefinitely. Many population protocols heavily rely on so-called *phase clocks* which divide the interactions into

1st Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2022).
Editors: James Aspnes and Othon Michail; Article No. 7; pp. 7:1–7:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

blocks of $O(n \log n)$ interactions each. The phase clocks are used to synchronize population protocols. For example, in [22, 16] they are used to efficiently solve leader election and in [15] they are used to solve the majority problem.

In the first part of this paper we present a loosely-stabilizing and leaderless phase clock with $O(\log n)$ many states per agent. We show that this clock can run forever and that, at any point of time, it is synchronized w.h.p.[1] In contrast to related work [1, 7, 15, 22], our clock protocol *recovers* rapidly in case of an error: from an arbitrary configuration it always enters a synchronous configuration within $O(n \log n)$ interactions w.h.p. Once synchronized it stays in a synchronous configuration for at least poly (n) interactions, w.h.p. Our phase clock can be used to synchronize population protocols into phases of $O(n \log n)$ interactions, guaranteeing that there is a big overlap between the phases of any pair of agents. Our clock protocol is simple, robust and easy to use.

In the second part of this paper we demonstrate how to apply our phase clock by solving an *adaptive majority* problem motivated by the work of [3, 4]. Our problem is defined as follows. Each agent has either opinion A, B, or U for being undecided. We say that agents change their input with rate r if in every time step an arbitrary agent can change its opinion with probability r. The goal is to output, at any time, the actual majority opinion. The idea of our approach is as follows. Our protocol simply starts, at the beginning of each phase, a *static* majority protocol as a black box. This protocol takes as an input the set of opinions at that time and calculates the majority opinion over these inputs. The outcome of the protocol is then used during the whole next phase as majority opinion. In order to highlight the simplicity of our phase clock, we first use the very natural protocol based solely on canceling opposing opinions introduced in [7]. Then we present a variant based on the undecided state dynamics from [8] which works as follows. The agents have one of two opinions A or B, or they are undecided. Whenever two agents with the same opinion interact, nothing happens. When two agents with an opposite opinion interact they will become undecided. Undecided agents interacting with an agent with either opinion A or opinion B adopt that opinion.

Without loss of generality we assume that A is the majority opinion in the following. When at least $\Omega(\log n)$ agents have opinion A, there is a constant factor bias between A and B, and the opinions change at most at rate $1/n$ per interaction, the system outputs A w.h.p. Our protocol requires only $O(\log n)$ many states. For the setting where all agents have either opinion A or B (none of the agent is in the undecided state U) and we have an additive bias of $n^{3/4+\varepsilon}$ for some constant $\varepsilon > 0$ is present, the system again converges to A w.h.p. In the latter setting we can tolerate a rate of order $r = \Omega(n^{-1/4+\varepsilon})$.

Related Work. Population protocols have been introduced by Angluin et al. [5]. Many of the early results focus on characterizing the class of problems which are solvable in the population model. For example, population protocols with a constant number of states can exactly compute predicates which are definable in Presburger arithmetic [5, 6, 9]. There are many results for majority and leader election, see [20] and [16] for the latest results. In [24] the authors introduce the notion of *loose*-stabilization to mitigate the fact that self-stabilizing protocols usually require some global knowledge on the population size (or a large amount of states). See [17] for an overview of self-stabilizing population protocols.

In [7] the authors present and analyze a phase clock which divides time into phases of $O(n \log n)$ interactions assuming that a unique leader exists. They also present a generalization using a junta of size n^{ε} (for constant ε) instead of a unique leader and analyze the

[1] The expression *with high probability* (w.h.p.) refers to a probability of $1 - n^{-\Omega(1)}$.

process empirically. In [22] the authors show that a junta can be elected using $O(\log \log n)$ many states and use the resulting clock to solve leader election. The protocol can easily be modified such that it requires only a constant number of states after the junta election [15]. In [1] the authors present a leaderless phase clock with $O(\log n)$ states. In contrast to our leaderless phase clock, the clock from [1] was not proven to be self-stabilizing. The analysis is based on the potential function analysis introduced in [25] for the greedy balls-into-bins strategy where each ball has to be allocated into one out of two randomly chosen bins. This analysis assumes an initially balanced configuration and it cannot be adopted to an arbitrary unbalanced state, which would be required to deal with unsynchronized clock configurations. In [10] the authors consider a variant of the population model, so-called clocked population protocols, where agents have an additional flag for clock ticks. The clock signal indicates when the agents have waited sufficiently long for a protocol to have converged. They show that a clocked population protocol running in less than ω^k time for fixed $k \geq 2$ is equivalent in power to nondeterministic Turing machines with logarithmic space.

Another line of related work considers the problem of *exact majority*, where one seeks to achieve (guaranteed) majority consensus, even if the additive bias is as small as one [21, 1, 14, 12]. The currently best protocol [20] solves exact majority with $O(\log n)$ states and $O(\log n)$ stabilization time, both in expectation and w.h.p. The authors of [8] solve the approximate majority problem. They introduce the undecided state dynamics in the population model for two opinions. They show that their 3-state protocol reaches consensus w.h.p. in $O(n \log n)$ interactions. If the bias is of order $\omega(\sqrt{n} \cdot \log n)$ the undecided state dynamics converges towards the initial majority w.h.p. In [18] the required bias is reduced to $\Omega(\sqrt{n \log n})$. For completeness [11] provides a survey about further protocols in the gossip model.

In [2] the authors define the **catalytic input model** (CI model). In this model the agents are divided into the two groups catalytic agents and non-catalytic agents. Non-catalytic agents perform pairwise interactions and change their state. Catalytic agents never change their state. Additionally to the normal state changes non-catalytic agents can perform spurious state changes; the so-called leak rate specifies the frequency of the spurious reactions. The goal of the non-catalytic agents is to compute a function over the states of the catalytic agents. The authors develop an algorithm for their model to detect whether there is a catalytic agent in a given state D or not. Note that, due to the leaky transactions non-catalysts can compute false-positives. In [4] the authors use the catalytic input model with n catalysts and m non-catalysts which they call worker agents ($N = n + m$). They solve the approximate majority problem for two opinions w.h.p. in $O(N \log N)$ interactions when the initial bias among the catalysts is $\Omega\big(\sqrt{N \log N}\big)$ and $m = \Theta(n)$. They show that the size of the initial bias is tight up to a $O\big(\sqrt{\log N}\big)$ factor. Additionally, they consider the approximate majority problem in the CI model and in the population model with leaks. Their protocols tolerate a leak rate of at most $\beta = O\big(\sqrt{N \log N}/N\big)$ in the CI model and a leak rate of at most $\beta = O\big(\sqrt{n \log n}/n\big)$ in the population model. They also show a separation between the computational power of the CI model and the population model.

In [3] the authors consider the CI model and introduce the *robust comparison problem*. The catalytic agents are either in state A or B and the goal of the worker agents is to decide which of the two states A and B has the larger support. In they dynamic version the number of agents in state A or B can change during the execution as long as the counts for A and B remain stable for a sufficiently long period allowing the algorithm to stabilize on an output. If at time t at least $\Omega(\log n)$ catalytic agents are in either A or B and the ratio between the numbers of agents supporting agents A and B is at least a constant, then most non-catalytic agents (up to $O(n/logn)$ agents) outputs w.h.p. the correct majority. The protocol needs with $O(\log n \cdot \log \log n)$ states per agent, assuming that the number of catalytic agents in A and B does not change in the meantime. They also mention that with standard population splitting $\mathrm{O}(\log n + \log\log n)$ states are sufficient under the constraint that only a constant fraction of the agents store the output. Additionally the authors show that their protocol is robust to leaky transitions at a rate of $\mathrm{O}(1/n)$. If the initial support of A and B states is $\Omega(\log^2 n)$ the authors can strengthen their results such that a ratio between the two base states of $1 + \mathrm{o}(1)$ is sufficient.

2 Population Model and Problem Definitions

In the population model we are given a set V of n anonymous *agents*. At each time step two agents are chosen independently and uniformly at random randomly to *interact*. We assume that interactions between two agents (u, v) are ordered and call u the *initiator* and v the *responder*. The interacting agents update their states according to a common transition function of their previous states. Formally, a *population protocol* is defined as a tuple of a finite set of *states* Q, a *transition function* $\delta : Q \times Q \to Q \times Q$, a finite set of *output symbols* Σ, and an *output function* $\omega : Q \to \Sigma$ which maps every state to an output. A *configuration* is a mapping $C : V \to Q$ which specifies the state of each agent. An execution of a protocol is an infinite sequence $C_0, C_1, \ldots$ such that for all C_i there exist two agents v_1, v_2 and a transition $(q_1, q_2) \to (q_1', q_2')$ such that $C_i(v_1) = q_1, C_i(v_2) = q_2, C_{i+1}(v_1) = q_1', C_{i+1}(v_2) = q_2'$ and $C_i(w) = C_{i+1}(w)$ for all $w \neq v_1, v_2$. The main quality criteria of a population protocol are the required number of states and the running time measured in interactions.

The goal of this paper is to develop protocols that are *loosely-stabilizing* according to the definitions of [24]. Let $\mathcal{C}$ denote an arbitrary subset of all possible configurations. Consider an infinite sequence of configurations $C_0, C_1, \ldots$. For an arbitrary configuration $C_i \notin \mathcal{C}$ the *convergence time* is defined as the smallest t such that $C_{i+t_1} \in \mathcal{C}$. Intuitively, the convergence time bounds the time it takes to reach a configuration in $\mathcal{C}$ when starting from a configuration not in $\mathcal{C}$. For an arbitrary configuration $C_i \in \mathcal{C}$ the *holding time* t_2 is defined as the largest t such that $C_{i+t_2} \in \mathcal{C}$. Intuitively, the holding time bounds the time during which the protocol remains in a configuration in $\mathcal{C}$ when starting from a configuration in $\mathcal{C}$.

▶ **Definition 1.** A protocol is loosely-stabilizing wrt. to a subset of configurations $\mathcal{C}$ if the maximum convergence time over all possible configurations is w.h.p. less than t_1 and the minimum holding time over all configurations in $\mathcal{C}$ is w.h.p. at least t_2.

Phase Clocks. Phase clocks are used to synchronize population protocols. We assume a phase clock is implemented by simple counters $\mathsf{clock}[u_1], \ldots, \mathsf{clock}[u_n]$ modulo $|Q|$ (see, e.g., [1, 7, 15, 19, 22]). Whenever $\mathsf{clock}[u]$ crosses zero, agent u receives a so-called *signal*. These signals will divide the time into *phases* of $\Theta(n \log n)$ interactions each. We say a (τ, w)-phase clock is synchronous in the time interval $[t_1, t_2]$ if every agent gets a signal every $\Theta(n \log n)$ interactions. More formally:

- Every agent receives a signal in the first $2 \cdot (w + 1) \cdot \tau \cdot n$ steps of the interval.
- Assume an agent u receives a signal at time $t \in [t_1, t_2]$.
  - For all $v \in V$, agent v receives a signal at time t_v with $|t - t_v| \leq \tau \cdot n$.
  - Agent u receives the next signal at time t' with $(w+1) \cdot \tau \cdot n \leq |t - t'| \leq 2 \cdot (w+1) \cdot \tau \cdot n$.

The above definition divides the time interval $[t_1, t_2]$ into a sequence of subintervals that alternates between so-called burst-intervals and overlap-intervals.

- A burst-interval has length at most $\tau \cdot n$ and every agent gets exactly one signal.
- An overlap-interval consists of those time steps between two burst-intervals where none of the agents gets a signal. It has length at least $w \cdot \tau \cdot n$.

A burst-interval together with the subsequent overlap-interval forms a *phase*.

To define loosely-stabilizing phase clocks, we need to define the set of *synchronous* configurations $\mathcal{C}$. Intuitively, we call a state C_t of a (τ, w)-phase clock at time t synchronous if the counters of all pairs of agents do not deviate much. More precisely, $\mathsf{clock}[u](t) - \mathsf{clock}[v](t) <_{|Q|} f(w, \tau)$ for all pairs of agents (u, v) (Here, "$\leq_{|Q|}$" denotes smaller w.r.t. the circular order modulo $|Q|$.) We define f and give the formal definition of a synchronous configuration in the next section.

3 Clock Algorithm

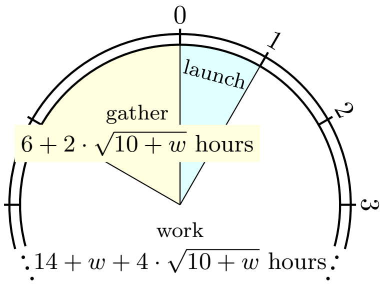

Figure 1 Schematic representation of the clock states.

In this section we introduce our phase clock protocol. For ease of notation, we assume in this section that the state space of an agent is Q Our (τ, w)-phase clock has a state space $Q = \{0, \ldots, \big(21 + w + 6 \cdot \sqrt{10 + w}\big) \cdot \tau - 1\}$. The clock states are divided into $\big(21 + w + 6 \cdot \sqrt{10 + w}\big)$ *hours*, and each hour consists of $\tau = \tau(c) = 36 \cdot (c + 4) \cdot \ln n$ *minutes*. The parameter $c \geq 6$ determines the error probability in each phase and thus the holding time (see Theorem 2). The parameter $w \geq 0$ can be chosen as needed by the application. As we will see, τ is a multiple of the running time of the *one-way epidemic* (see Lemma 4) and $w \cdot \tau \cdot n$ is the number of interactions in which our agents are synchronized. We divide the hours into three consecutive intervals (see Figure 1): the launching interval I_{launch} (first hour), the working interval I_{work} ($14 + w + 4 \cdot \sqrt{10 + w}$ hours) and the gathering interval I_{gather} (last $6 + 2 \cdot \sqrt{10 + w}$ hours). We say that agent u is in one of the intervals whenever its clock counter $\mathsf{clock}[u]$ is in that interval. If the agents are either all in I_{gather}, all in I_{work}, or all in I_{launch}, we say the configuration is *homogeneous*. For two agents u and v we define a *distance* $d(u, v) = \min\{|\mathsf{clock}[u] - \mathsf{clock}[v]|, |Q| - |\mathsf{clock}[u] - \mathsf{clock}[v]|\}$ that takes the cyclic nature of the clock into account. This allows us to define synchronous configurations as follows.

▶ **Definition** (Synchronous Configuration)**.** A configuration C is called *synchronous* if and only if for all pairs of agents (u, v) we have $d(u, v) < |I_{\text{launch}}| + |I_{\text{gather}}| = (7 + 2 \cdot \sqrt{10 + w}) \cdot \tau$.

Our clock works as follows. Assume agents (u, v) interact. With two exceptions, agent u increments its counter $\mathsf{clock}[u]$ by one minute modulo $|Q|$ (Rules 1 and 2). If, however, u is in I_{gather} and v is in I_{launch} then agent u adopts $\mathsf{clock}[v]$ (Rule 3): we say the agent *hops*. If u is in I_{gather} and v is in I_{work} then agent u returns to the beginning of I_{gather} (Rule 4): we say that the agent *resets*. We define that agent u receives a *signal* whenever its clock crosses the wrap-around from I_{gather} to I_{launch}. Formally, our clock uses the following transitions.

$$(q_1, q_2) \in (Q \setminus I_{\text{gather}}) \times Q\colon \quad (q_1, q_2) \to (q_1 + 1, q_2) \quad \text{(step forward)} \quad (1)$$
$$(q_1, q_2) \in I_{\text{gather}} \times I_{\text{gather}}\colon \quad (q_1, q_2) \to (q_1 + 1 \bmod |Q|, q_2) \quad \text{(step forward)} \quad (2)$$
$$(q_1, q_2) \in I_{\text{gather}} \times I_{\text{launch}}\colon \quad (q_1, q_2) \to (q_2, q_2) \quad \text{(hopping)} \quad (3)$$
$$(q_1, q_2) \in I_{\text{gather}} \times I_{\text{work}}\colon \quad (q_1, q_2) \to (|I_{\text{launch}}| + |I_{\text{work}}|, q_2) \quad \text{(reset)} \quad (4)$$

Note that $|I_{\text{work}}| = 2(|I_{\text{launch}}| + |I_{\text{gather}}|) + w \cdot \tau$ to have $w \cdot \tau \cdot n$ homogeneous working configurations between two signals. On the other hand, $|I_{\text{gather}}|/\tau = \Theta\big(\sqrt{|I_{\text{work}}|/\tau}\big)$ which is necessary to apply Chernoff bounds. We chose $|I_{\text{launch}}| = \tau$ for simplicity. On an intuitive level, the clock works as follows. Assume the clock is synchronized and all agents are in I_{launch}. Now consider the next $k = \Theta\big(n \cdot |Q|\big)$ interactions. All agents step forward according to Rule 1 until they reach I_{gather}. The maximum distance between any agents grows during the k interactions but can still bounded by $\mathrm{O}\big(\sqrt{k/n}\big) = \mathrm{O}\big(\sqrt{|w| \cdot \log n}\big)$, w.h.p. via Chernoff bounds. Hence, due to the choice of w there is no agent left behind in I_{launch} when the first agent reaches I_{gather}. Additionally, due to the size of I_{gather} when the last agent enters I_{gather} all of the other agents are still in $|I_{\text{gather}}|$. As soon as the first agent reaches I_{launch}, Rule 3 (agents hop onto agents in I_{launch}) ensures that all agents start the next phase without a large gap. Hence, there is an interaction after which all agents are in $|I_{\text{launch}}|$ which brings us back to our initial configuration (all agents in I_{launch}).

Now we consider an asynchronous configuration C_t where the agents can be arbitrarily distributed over the $|Q|$ states of the clock. The main idea for the recovery of our clock is as follows. We show that after $\mathrm{O}(n \log n)$ interactions there is a time t where I_{launch} is empty. After $\mathrm{O}(n \log n)$ additional steps most of the agents are in I_{gather}: agents cannot hop since I_{launch} is empty, and they reset as soon as they interact with an agent in I_{work}. They enter I_{launch} as soon as the first agent crosses 0 by increasing its clock counter.

We will show that the following two properties hold for our clock.

▶ **Theorem 2.** *Let $\tau = 36 \cdot (c + 4) \cdot \ln n$ and let w be a sufficiently large constant. Let t_1, t_2 with $t_1 \le t_2$ be two points in time and assume that the configuration C_{t_1} at time t_1 is a homogeneous launching configuration and $t_2 - t_1 \le n^c$. Then the clock counters of the agents implement a synchronous (τ, w)-phase clock in the time interval $[t_1, t_2]$ w.h.p.*

▶ **Theorem 3.** *The clock counters of the agents implement a $\big(\mathrm{O}\big(n \cdot \log n\big), \Omega\big(poly\ n\big)\big)$-loosely-stabilizing $\big(\Theta\big(\log n\big), w\big)$-phase clock.*

Note that our simulations suggest that the algorithm also works if τ is smaller by a constant fraction. We prove Theorem 2 in Section 4 and Theorem 3 in Section 5.

Auxiliary Results. The one-way-epidemic is a population protocol with state space $\{0, 1\}$ and transitions $(q_1, q_2) \to (\max(q_1, q_2), q_2)$. An agent in state 0 is called *susceptible* and an agent in state 1 is called *infected*. We say agent v infects agent u if v is infected and u initiates an interaction with v. The following result is folklore, see, e.g., [7]. Additional details can be found in the full version of this paper.

▶ **Lemma 4** (One-way-epidemic). *Assume an agent starts the one-way epidemic in step 1. All agents are infected after $t = \tau/4 \cdot n$ many steps with probability at least $1 - n^{-(7+2c)}$.*

The following lemma bounds the number of interactions initiated by some fixed agent u among a sequence of t interactions. It is used throughout Sections 4 and 5 and follows immediately from Chernoff bounds (see [23]).

▶ **Lemma 5.** *Consider an arbitrary sequence of t interactions and let X_u be the number of interactions initiated by agent u within this sequence. Then*

$$\Pr[X_u < (1+\delta)\cdot t/n] \geq 1 - n^{-\frac{12\cdot(c+4)t\cdot\delta^2}{n\cdot\tau}} \quad \textit{and} \quad \Pr[X_u > (1-\delta)\cdot t/n] \geq 1 - n^{-\frac{18\cdot(c+4)t\cdot\delta^2}{n\cdot\tau}}.$$

4 Maintenance: Proof of Theorem 2

In this section we first show the following main result. At the end of the section we show how Theorem 2 follows from this proposition.

▶ **Proposition 6** (Maintenance). *Consider our (τ, w)-phase clock for n agents with $\tau = 36\cdot(c+4)\cdot\ln n$ for any $c \geq 6$ and sufficiently large w. Let configuration C_{t_1} be a homogeneous launching configuration. Then, with probability at least $1 - n^{-(c+1)}$, there exists a $t_2 = \Theta\big(n\cdot w\cdot\log n\big)$ such that the following holds:*
1. *$C_{t_1+t_2}$ is a homogeneous launching configuration,*
2. *$\forall t \in [t_1, t_1+t_2]$: C_t is synchronous,*
3. *in the time interval $[t_1, t_1+t_2]$ there exists a contiguous sequence of homogeneous working configurations of length $w\cdot\tau\cdot n$.*

We split the proof of Proposition 6 into two parts, Lemmas 7 and 8. The formal proof follows.

Proof. Assume the configuration C_{t_1} at time t_1 is a homogeneous launching configuration. Statements 1 and 2 of Proposition 6 follow immediately from Lemmas 7 and 8:
- It follows from Lemma 7 that the agents transition via a sequence of synchronous configurations into a homogeneous gathering configuration within $\Theta\big(n\cdot w\cdot\log n\big)$ time w.h.p.
- It follow from Lemma 8 that the agents transition via a sequence of synchronous configurations back into a homogeneous launching configuration within $\Theta\big(n\cdot w\cdot\log n\big)$ further time w.h.p.

It remains to show Statement 3. Recall that in a synchronous configuration all pairs of agents have distance (w.r.t. the circular order modulo $|Q|$) at most $\Delta = (7 + 2\cdot\sqrt{10+w})\cdot\tau$. Since $|I_{\text{work}}| = w\cdot\tau + 2\Delta$ it immediately follows that there must be $w\cdot\tau\cdot n$ interactions where all agents are in I_{work}. This concludes the proof. ◀

The following lemma establishes that w.h.p. all agents transition from a homogeneous launching configuration into a homogeneous gathering configuration via a sequence of synchronous configurations.

▶ **Lemma 7.** *Let C_t be a homogeneous launching configuration. Let $t' = n\cdot\frac{|I_{\text{launch}}|+|I_{\text{work}}|}{1-\left(2\cdot\sqrt{|I_{\text{work}}|/\tau}\right)^{-1}}$. Then the following holds with probability at least $1 - n^{-(c+1)}/2$:*
1. *$C_{t+t'}$ is a homogeneous gathering configuration and*
2. *$\forall t'' \in [t, t+t'] : C_{t''}$ is synchronous.*

Proof. In the following we assume w.l.o.g. $t = 0$. We prove the two statements separately.

Statement 1. Our goal is to show that after t' interactions all agents are in I_{gather} when we start from a homogeneous launching configuration C_0 at time $t = 0$. We first show that there is no agent left in I_{launch} when the first agent enters I_{gather}. Let t_a be the first interaction in which an agent enters I_{gather}. Note that before t_a all agents are either in I_{launch} or in I_{work} and thus the agents increase their counter by one whenever they initiate an interaction.

First we show that w.h.p. $t_a \geq 2\cdot\tau\cdot n$. Let $X_u(2\cdot\tau\cdot n)$ denote the number of interactions agent u initiates before time $2\cdot\tau\cdot n$. From Lemma 5 it follows with $\delta = 1$ that $X_u(2\cdot\tau\cdot n) < 4\cdot\tau$ with probability at least $1 - n^{-24\cdot(c+4)}$. Since $4\cdot\tau < |I_{\text{work}}|$, it holds that

$\mathsf{clock}[u](2 \cdot \tau \cdot n) < |I_{\text{launch}}| + |I_{\text{work}}|$ in this case. Hence, agent u has not yet reached I_{gather} with probability at least $1 - n^{-24 \cdot (c+4)}$ at time t_a. It follows from a union bound over all agents that no agent has reached I_{gather} with probability at least $1 - n^{-24 \cdot (c+4)+1}$ at time t_a.

Next we show that w.h.p. at time $2 \cdot \tau \cdot n$ all agents have left I_{launch}. As before, let $X_u(2 \cdot \tau \cdot n)$ denote the number of interactions agent u initiates before time $2 \cdot \tau \cdot n$. From Lemma 5 it follows with $\delta = 1$ that $X_u(2 \cdot \tau \cdot n) > \tau$ with probability at least $1 - n^{-36 \cdot (c+4)}$. Since $\tau = |I_{\text{launch}}|$, it holds that $\mathsf{clock}[u](t + 2 \cdot \tau \cdot n) \geq |I_{\text{launch}}|$ in this case. Hence, agent u has left I_{launch} with probability at least $1 - n^{-36 \cdot (c+4)}$ at time t_a. Again, it follows from a union bound over all agents that all agents have left I_{launch} with probability at least $1 - n^{-36 \cdot (c+4)+1}$ at time t_a.

Let now t_b be the first interaction in which an agent enters the last minute of I_{gather} and observe that $t_b > t_a$. Then, w.h.p. no agent is in I_{launch} during the time interval $[t + t_a, t + t_b]$. Therefore, agents cannot hop. Thus, by definition of t_b, no agent can leave I_{gather} before time t_b. All initiators must therefore either increase their counter by one or reset.

First we show that w.h.p. $t_b > t'$. From Lemma 5 it follows with $\delta = \left(2 \cdot \sqrt{|I_{\text{work}}|/\tau}\right)^{-1}$ that $X_u(t') < |I_{\text{work}}| + |I_{\text{gather}}|$ with probability at least $1 - n^{-3(c+4)}$. (Note that we use $(1+\delta)/(1-\delta) < 1/(1 - 2 \cdot \delta)$ for $\delta < 0.5$ and $(|I_{\text{launch}} + |I_{\text{work}}|)/(1 - 2 \cdot \delta) = |I_{\text{work}}| + (w + 5 \cdot \sqrt{10 + w} + 16)/(\sqrt{10 + w} + 1)) \cdot \tau < |I_{\text{work}}| + |I_{\text{gather}}|$.) Thus, $\mathsf{clock}[u](t') \leq \mathsf{clock}[u](0) + X_u(t') < |I_{\text{launch}}| - 1 + |I_{\text{work}}| + |I_{\text{gather}}|$ (which is the last state of I_{gather}) with probability at least $1 - n^{-3(c+4)}$. By a union bound, this holds for all agents with probability at least $1 - n^{-(3c+11)}$.

Next we show that w.h.p. at time t' all agents have reached I_{gather}. From Lemma 5 it follows for our choice of δ that $X_u(t') > |I_{\text{launch}}| + |I_{\text{work}}|$ with probability at least $1 - n^{-9 \cdot (c+4)/2}$. Thus, $\mathsf{clock}[u](t') \geq \mathsf{clock}[u](0) + X_u(t') > 0 + |I_{\text{launch}}| + |I_{\text{work}}|$, with probability at least $1 - n^{-9 \cdot (c+4)/2}$. By a union bound, this holds for all agents with probability at least $1 - n^{-(17 + 9/2 \cdot c)}$.

Together it follows that at time t' no agent has left I_{gather} but all agents have entered it with probability at least $1 - n^{-(c+3)}$. Therefore, $C_{t'}$ is a homogeneous gathering configuration.

Statement 2. Recall that a synchronous configuration C is defined as a configuration where $\max_{(u,v)} \{ d(u,v) \} < |I_{\text{launch}}| + |I_{\text{gather}}|$. As before, let $X_u(i)$ denote the number of interactions agent u initiates before time i. Now fix a time $t \leq t'$ and a pair of agents (u, v) with $X_u(t) < X_v(t)$. We use Lemma 5 to bound the deviation of $X_u(t)$ and $X_v(t)$ at time t as follows: $\Pr[X_u(t) > t/n - |I_{\text{gather}}|/2] \geq 1 - n^{-6(c+4)}$ and $\Pr[X_v(t) < t/n + |I_{\text{gather}}|/2] \geq 1 - n^{-4(c+4)}$. Therefore, $|X_v(t) - X_u(t)| < |I_{\text{gather}}|$ with probability at least $1 - n^{-4(c+4)} - n^{-6(c+4)}$.

Note that Lemma 5 allows us to bound the deviation in the numbers of interactions initiated by agents u and v. However, this does not immediately give a bound on the difference of the clock counters $|\mathsf{clock}[v](t) - \mathsf{clock}[u](t)|$. To bound the deviation of clock counters (by $|I_{\text{launch}}| + |I_{\text{gather}}|$), we therefore distinguish three cases.

First, assume that neither u nor v have reached I_{gather} at time t. Then $\mathsf{clock}[u](t) = \mathsf{clock}[u](0) + X_u(t)$ and $\mathsf{clock}[v](t) = \mathsf{clock}[v](t) + X_v(t)$. Observe that by the assumption of the lemma, both u and v are in I_{launch} at time $t = 0$ and thus $|\mathsf{clock}[v](0) - \mathsf{clock}[u](0)| < |I_{\text{launch}}|$. Together with the above bound on $|X_v(t) - X_u(t)|$ we get $|\mathsf{clock}[v](t) - \mathsf{clock}[u](t)| < |I_{\text{launch}}| + |I_{\text{gather}}|$.

Secondly, assume that u has not reached I_{gather} but v has reached I_{gather} at time t. Then $\mathsf{clock}[u](t) = \mathsf{clock}[u](0) + X_u(t)$. For $\mathsf{clock}[v](t)$, however, it might have occurred that v has reset in some interactions before time t. Nevertheless, the clock counter of v is bounded by the number of initiated interactions such that $\mathsf{clock}[v](t) \leq \mathsf{clock}[v](t) + X_v(t)$. (Note that v can only increment its $\mathsf{clock}[v]$ counter or reset its value; hopping is not possible since we have shown in the proof of the first statement that I_{launch} is empty when the first agent enters I_{gather}.) Therefore, we get again $|\mathsf{clock}[v](t) - \mathsf{clock}[u](t)| < |I_{\text{launch}}| + |I_{\text{gather}}|$.

Finally, assume that both u and v are in I_{gather} at time t. Then $|\mathsf{clock}[v](t) - \mathsf{clock}[u](t)| \leq |I_{\text{gather}}| < |I_{\text{launch}}| + |I_{\text{gather}}|$ is trivially true.

There are no further cases: in the proof of the first statement we have shown that all agents transition from a homogeneous launching configuration to a homogeneous gathering configuration during the time interval $[0, t']$. The result now follows from a union bound over all $o(n^2)$ points in time $t \leq t'$ and all $n \cdot (n-1)$ pairs of agents. ◀

The following lemma is the main technical contribution of this section. It establishes that w.h.p. all agents transition from a homogeneous gathering configuration into a homogeneous launching configuration via a sequence of synchronous configurations. Consider a homogeneous gathering configuration and recall that whenever an agent hops from I_{gather} into I_{launch} it adopts the state of the responder. The main difficulty is to show that all agents hop into I_{launch} before the first agent leaves I_{launch}.

▶ **Lemma 8.** *Let C_t be a homogeneous gathering configuration. Then with probability at least $1 - n^{-(c+1)}/2$ the following holds:*
1. *there exists a $t_0 = \mathrm{O}\big(n \cdot \sqrt{w} \cdot \log n\big)$ such that the first agent enters I_{launch} at time $t + t_0$,*
2. *there exists a $t' \leq \tau/4 \cdot n$ such that $C_{t+t_0+t'}$ is a homogeneous launching configuration,*
3. *$\forall t'' \in [t, t+t'] : C_{t''}$ is synchronous.*

Proof. We prove first show Statement 1, Statement 2 and Statement 3 are shown together.

Statement 1. Let t_0 be defined such that the first agent u leaves I_{gather} at time $t + t_0$. Since C_t is a homogeneous gathering configuration, I_{launch} is empty at time t and hence agent u can only leave I_{gather} by increasing its counter. In every interaction before time $t + t_0$ some agent has to increase its state by one. Thus $t_0 \leq n \cdot |I_{\text{gather}}| = \mathrm{O}\big(n \cdot \sqrt{w} \cdot \log n\big)$.

Statement 2+3. We continue our analysis at time t_0 and again assume w.l.o.g. for the sake of brevity of notation that $t_0 = 0$. Note that at that time exactly one agent is in state 0 and all remaining agents are still in I_{gather}. We show the following: there exists a time $\tilde{t} = \tau/4 \cdot n$ such that at time $\tilde{t}$ all agents are in I_{launch} (Recall that $|I_{\text{launch}}| = \tau \cdot n$). To do so we first define a simplified process with the same state space Q, however, we refer to the last state of I_{launch} as stop. Agents in stop never change their state (which renders the states of I_{work} unreachable). The formal definition of the simplified process is as follows. Rule 2 and 3 are identical to the original process and Rule 1 and 4 are modified as follows.

$$(q_1, q_2) \in (I_{\text{launch}} \setminus \{\, \mathsf{stop} \,\}) \times Q: \qquad (q_1, q_2) \to (q_1 + 1, q_2) \qquad \text{(step forward)} \qquad (1)$$

$$(q_1, q_2) \in \{\, \mathsf{stop} \,\} \times Q: \qquad (q_1, q_2) \to (q_1, q_2) \qquad \text{(stopping)} \qquad (4)$$

For this simplified process we show a lower bound: after $\tilde{t} = \Theta\big(n \cdot \log n\big)$ interactions all agents are in I_{launch}. Then we show (for the simplified process) an upper bound: in $C_{\tilde{t}}$ none of the agents are in state stop. A simple coupling of the simplified process and the original process shows that under these circumstances none of the agents entered I_{work} for our original process. This finishes the proof with $t' = \tilde{t}$.

Lower Bound. In the simplified process agents can enter I_{launch} either via hopping or by making enough steps forward on their own. From Lemma 4 it follows that all agents enter I_{launch} after at most $\tilde{t} = \tau/4 \cdot n$ interactions with probability at least $1 - n^{-(5+c)}$. (For the upper bound, one can simply discard setting the clock counter to zero when an agent enters I_{launch} by increasing its counter.) Showing that none of the agents are in state stop is much harder. Due to the hopping the clock counters of agents in I_{launch} are highly correlated. Nevertheless, we can show that the clock counters of each agent can be majorized by independent binomially distributed random variables as follows.

Upper Bound. Let u_i be the i'th agent that enters I_{launch} and let t_i be the time when u_i enters I_{launch}. Let furthermore $X_i(t)$ be a random variable for the clock counter of agent u_i in I_{launch} in the time interval $[0, t]$. Formally, we define for a time step t that $X_i(t) = 0$ if u_i is in I_{gather} and $X_i(t) = \mathsf{clock}[u_i](t)$ if u_i is in I_{launch}. We show by induction on i that $X_i(t)$ is majorized by a random variable $Z_i(t)$ with binomial distribution $Z_i(t) \sim \text{Bin}\big(t, 1/n \cdot (1 + 1/(n-1))^{i-1}\big)$, i.e., $\Pr[X_i(t) > x] \leq \Pr[Z_i(t) > x]$ for all $x \geq 0$. Ultimately, our goal is to apply Chernoff bounds to $Z_i(\tilde{t})$ which shows that agent u_i does not reach stop w.h.p. The statement for the simplified process then follows from a union bound over all agents w.h.p.

Base Case. For the base case we consider all agents that enter I_{launch} on their own by incrementing their counters to 0 (modulo $|Q|$) in I_{gather}. Fix such an agent u_i. It holds that $X_i(t)$ for $t \geq t_i$ has binomial distribution $X_i(t) \sim \text{Bin}(t - t_i, 1/n)$. Therefore, $X_i(t) \prec Z_i(t)$ as claimed.[2] (Intuitively, this means that the clock counter of any other agent u_i with $i > 1$ that enters I_{launch} at time $t_i > 0$ is majorized by the clock counter of an agent which enters I_{launch} at time $t_1 = 0$ and increments its counter with probability $1/n$.)

Induction Step. For the induction step we now consider all agents that enter I_{launch} by hopping onto some other agent in I_{launch}. Fix such an agent u_i. Let S_i be the event that agent u_i is the i'th agent that enters I_{launch}. Let furthermore t_i be the time when u_i enters I_{launch}. We condition on S_i and observe that agent u_i enters I_{launch} by hopping onto some other agent $u_j \in \{ u_1, \ldots, u_{i-1} \}$. Intuitively, we would now like to exploit the fact that the counter of agent u_i is copied at time t_i from agent u_j such that $X_i(t_i) = X_j(t_i)$. Unfortunately, we must be extremely careful here: conditioning on S_i alters the probability space! (For example, under S_i the agent u_i with $i \geq 3$ cannot initiate an interaction with agent u_1 before agent u_2 does, since S_i rules out that u_i enters I_{launch} before agent u_2.) We account for the modified probability space as follows.

Let $\Omega_{S_i}(t)$ be the probability space of possible interactions conditioned on S_i at time $t \leq \tilde{t}$. Without the conditioning on S_i, the probability space $\Omega(t)$ at time t contains all (ordered) pairs of agents with $|\Omega(t)| = n \cdot (n-1)$. When conditioning on S_i, the event S_i rules out that agent u_i interacts with any other agent $u_j \in I_{\text{launch}}$ before time t_i. In particular, agent u_i cannot interact with another agent u_j with $j < i$ during the time interval $[t_j, t_i]$. In order to give a lower bound on $|\Omega_{S_i}(t)|$, we exclude all $(n-1)$ interactions (u_i, u_j) for $j \in [n]$ from $\Omega(t)$. Hence $|\Omega_{S_i}(t)| \geq n \cdot (n-1) - (n-1) = (n-1)^2$ for any time $t \leq t_i$. (The probability space after time t_i is not affected by conditioning on S_i, but the majorization holds nonetheless.) We now consider the event $E_{\hat{t}}$ for $\hat{t} \leq t_i$ that the interaction at time $\hat{t}$ increments $X_j(t)$ by 1 (recall that u_j is the agent onto which u_i hopped). It then holds for the reduced probability space Ω_{S_i} that $\Pr[E_{\hat{t}} \mid S_i] \leq \Pr[E_{\hat{t}}] \cdot |\Omega(t)|/|\Omega_{S_i}(t)|$. (Note that Ω_{S_i} is still a uniform probability space.) We calculate

$$\frac{|\Omega(t)|}{|\Omega_{S_i}(t)|} = \frac{n \cdot (n-1)}{(n-1)^2} = 1 + \frac{1}{n-1}$$

[2] The expression $X \prec Y$ means that the random variable X is majorized by the random variable Y.

and get $\Pr[E_{\hat{t}} \mid S_i] \le \Pr[E_{\hat{t}}] \cdot (1 + 1/(n-1))$ for $\hat{t} \le t_i$. Therefore, we use the induction hypothesis (that describes $X_j(t_i)$) and get $X_i(t_i) \prec Z_i(t_i)$, where $Z_i(t_i) \sim \mathrm{Bin}\left(t_i, 1/n \cdot (1 + 1/(n-1))^{i-1}\right)$. Similarly, we define $E_{\hat{t}}$ for $\hat{t} \ge t_i$ to be the event that u_i increments its counter in I_{launch}. Observe that $\Pr[E_{\hat{t}}] \le 1/n$ for $\hat{t} > t_i$. It follows that $X_i(t) \prec Z_i(t)$ with distribution $Z_i(t) \sim \mathrm{Bin}(\tilde{t}, 1/n \cdot (1 + 1/(n-1))^{i-1})$ for $t \le \tilde{t}$ as claimed. This concludes the induction.

Conclusions. From the induction it follows that for each agent u_i the clock counter $\mathsf{clock}[u_i](\tilde{t})$ at time $\tilde{t}$ is majorized by a random variable $Z(\tilde{t})$ with binomial distribution $Z(\tilde{t}) \sim \mathrm{Bin}(\tilde{t}, e/n)$. (Note that we used the inequality $(1 + 1/(n-1))^{(n-1)} < e$.) From Chernoff bounds (see [23]) it follows that $\Pr\left[Z(\tilde{t}) \ge \tau - 1\right] \le n^{-(c+4)}$. Finally, the proof for the simplified process follows from a union bound over all agents.

It is now straightforward to couple the actual phase clock process with the simplified process. Assume that we start both processes at time 0 when exactly one agent is in state 0. In the simplified process no agent reaches state τ in $\tau/4 \cdot n$ interactions with probability at least $1 - n^{-(c+3)}$. In this case, however, the simplified process and the actual phase clock process do not deviate and, in particular, no agent reaches the beginning of I_{work} in $\tau/4 \cdot n$ many interactions. Thus, the configuration $C_{t'}$ is a homogeneous launching configuration with probability at least $1 - n^{-(c+3)}$.

Since all agents started in I_{gather} and no agent reaches the beginning of I_{work}, the agents are in a synchronous configuration by definition during the whole time interval $[0, t']$. ◀

We are now ready to put everything together and prove our first theorem.

Proof of Theorem 2. The proof of Theorem 2 follows readily from the main result of this section, Proposition 6.

Assume the configuration at time t_1 is a homogeneous launching configuration. Then from Proposition 6 it follows w.h.p. that after $t_2 = \Theta\left(n \cdot w \cdot \log n\right)$ interactions the configuration C_{t_2} is again a homogeneous launching configuration, and all configurations in $[t_1, t_2]$ are synchronous. From Statement 3 it follows that no agent receives a signal in a contiguous subinterval $[t'_1, t'_2] \subset [t_1, t_2]$ of length $t'_2 - t'_1 = w \cdot \tau \cdot n$. This shows that we have w.h.p. the required *overlap* according to the definition of synchronous (τ, w)-phase clocks.

From Lemma 8 it follows w.h.p. that all agents transition from a homogeneous gathering configuration into a homogeneous launching configuration within $\tau/4 \cdot n$ interactions. Recall that whenever an agent crosses zero, it receives a signal. Therefore, when all agents transition from a homogeneous gathering configuration into a homogeneous launching configuration via a sequence of synchronous configurations, all agents receive exactly one signal, and the time between two signals of two agents (u, v) is w.h.p. at most $\tau/4 \cdot n$. This shows that we have w.h.p. the required *bursts* according to the definition of synchronous (τ, w)-phase clocks.

Together, the counters of our clock implement a synchronous (τ, w)-phase clock in $[t_1, t_2]$ with probability n^{-c}. It follows from an inductive argument that the clock counters implement a synchronous (τ, w)-phase clock during the n^c interactions that follow time t_1 w.h.p. ◀

5 Recovery: Proof of Theorem 3

In this section we first show the following main result. At the end of the section we show how Theorem 3 follows from this proposition.

▶ **Proposition 9** (Recovery). *Consider our (τ, w)-phase clock with n agents and sufficiently large c and w. Let C_{t_1} be an arbitrary configuration. Then with probability at least $1 - 1/n$, there exists a $t_2 = \mathrm{O}\big(n \cdot w \cdot \log n\big)$ such that $C_{t_1+t_2}$ is a homogeneous launching configuration.*

We say a configuration is an *almost homogeneous gathering configuration* if no agent is in I_{launch} and at least $0.9 \cdot n$ many agents are in I_{gather}. We start our analysis by showing that within $t = \mathrm{O}\big(n \cdot w \cdot \log n\big)$ interactions, we reach an almost homogeneous gathering configuration C_{t_1+t}.

▶ **Lemma 10.** *Let C_t be an arbitrary configuration. Then with probability at least $1 - 1/(3n)$, there exists a $t' = \mathrm{O}\big(n \cdot w \cdot \log n\big)$ such that $C_{t+t'}$ is an almost homogeneous gathering configuration.*

Proof Sketch. The main idea of the proof is as follows. If there are not too many agents in I_{gather}, the reset rule prevents agents from reaching the end of I_{gather}. Agents may still enter I_{launch} by hopping, but if no agent enters state 0, eventually there is no agent left in state 0 to hop on. Then the same argument applies to state 1, and so on. Eventually, there are no agents left in I_{launch} to hop onto. This means the agents are *trapped* in I_{gather} until a sufficiently large number of agents enters I_{gather} which renders resetting quite unlikely again. The resulting configuration is what we call an almost homogeneous gathering configuration. ◀

Next, we show that from an almost homogeneous gathering configuration we reach a homogeneous gathering configuration in $\mathrm{O}\big(n \cdot w \cdot \log n\big)$ interactions. From Lemma 8 in Section 4 it then follows that we reach a homogeneous launching configuration in an additional number of $\mathrm{O}(n \cdot \log n)$ interactions.

▶ **Lemma 11.** *Let C_t be an almost homogeneous gathering configuration. Then with probability at least $1 - 1/(3n)$, there exists a $t' = \Theta\big(n \cdot w \cdot \ln n\big)$ such that $C_{t+t'}$ is a homogeneous gathering configuration.*

Proof Sketch. If C_t is an almost homogeneous gathering configuration, then there are no agents in I_{launch} and at least $0.9 \cdot n$ many agents in I_{gather}. Thus, agents cannot hop until an agent enters I_{launch} on its own. Now there are two cases. If no agent enters I_{launch} on its own before the last agent enters I_{gather}, we are done: this is by definition of a homogeneous gathering configuration. Otherwise, we will show that a large fraction of agents leave I_{gather} together. This large fraction behaves similar as in the proof of the maintenance. The remaining agents have a small *head start* but then they are again *trapped* in I_{gather} until the bulk of agents arrives. Once the bulk of agents enters I_{gather} we have reached a homogeneous gathering configuration and all agents start to run through the clock synchronously. ◀

Proof of Theorem 3. The proof of Theorem 3 follows readily from the main result of this section, Proposition 9. Observe that $\tau = \Theta(\log n)$. According to Proposition 9, our clock recovers to a homogeneous launching configuration in $\mathrm{O}(n \cdot \log n)$ interactions. By Theorem 2, this marks the beginning of a time interval in which the agents implement a synchronous (τ, w)-phase clock. It follows immediately from Theorem 2 that this interval has length n^c. Together, this implies that our (τ, w)-phase clock is a $(\mathrm{O}(n \cdot logn), \Omega(\mathrm{poly}(n)))$-loosely-stabilizing $(\Theta(\log n), w)$-phase clock. ◀

6 Adaptive Majority Problem

In this section we consider the adaptive majority problem. At any time, every agent has as input either an *opinion* (A or B) or it has no input, in which case we say it is *undecided* (U). During the execution of the protocol, the opinions of the agents can change. In the adaptive majority problem, the goal is that all agents output (at all times) the opinion which is dominant among all inputs. In this setting we present a *loosely-stabilizing* protocol that solves the adaptive majority problem. We define a loosely-stabilizing adaptive majority protocol according to Definition 1 by defining $\mathcal{C}$ as all configurations where all agents output the correct majority opinion. Recall that the performance of a loosely-stabilizing protocol is measured in terms of the *convergence time* and the *holding time*. Note that the loose-stabilization comes from an application of our phase clock. The phase clocks guarantee synchronized phases for polynomial time. During this time we say a configuration C is *correct* w.r.t. the adaptive majority problem if the following conditions hold. Suppose there is a sufficiently large bias towards one opinion. Then every agent in a correct configuration outputs the majority opinion. Otherwise, if there is no sufficiently large bias, we consider any output of the agents as correct. In this setting, we show the following result: We show that a $(\mathrm{O}(\log n), \mathrm{poly}(n))$-loosely-stabilizing algorithm exists that solves adaptive majority, using $\mathrm{O}(\log n)$ states per agent.

6.1 Our Protocol

Our protocol is based on the (τ, w)-phase clock defined in Section 3 with $w = 566$. In addition to the states required by the clock, every agent v has three variables $\mathsf{input}[v]$, $\mathsf{opinion}[v]$, and output $\mathsf{output}[v]$. The variable $\mathsf{input}[v]$ always reflects the current input to the agent, $\mathsf{opinion}[v]$ holds the current opinion of agent v, and $\mathsf{output}[v]$ defines the current output value of agent v. All three variables take values in $\{ A, B, U \}$. A and B stand for the corresponding opinions and U stands for *undecided*. The state space of the protocol is $Q_c \times \{ A, B, U \}^3$ where Q_c is the state space of our clock for $\tau = 36 \cdot (c + 4) \ln n$ and $w = 566$.

We use the (τ, w)-phase clock to synchronize the agents. Then it follows from Proposition 6 that all configurations are synchronous w.h.p. Observe that in a synchronous configuration for our choice of parameters the clock counters of agents do not deviate by more than $\Delta = 55 \cdot \tau$. This allows us to define three *subphases* of I_{work}, where agents execute three different protocols, as follows. We split the working interval I_{work} into six contiguous subintervals of equal length. The clock counters $\mathsf{clock}[u]$ allows us to define a simple interface to the phase clock for each agent u as follows. The variable $\mathsf{subphase}[u]$ for each agent u is then defined as follows. We set $\mathsf{subphase}[u] = 1$ if $\mathsf{clock}[u]$ is in the first subinterval of I_{work}, $\mathsf{subphase}[u] = 2$ if $\mathsf{clock}[u]$ is in the third subinterval of I_{work}, and $\mathsf{subphase}[u] = 3$ if $\mathsf{clock}[u]$ is in the fifth subinterval of I_{work}. Otherwise, $\mathsf{subphase}[u] = \perp$. The clock now assures a clean separation into these subphases such that no two agents perform a different protocol at any time w.h.p. Additionally, we will show the overlap within each subphase is long enough such that the subprotocols for the corresponding subphases succeed w.h.p.

On an intuitive level, our protocol works as follows. At the beginning of the phase, the input is copied to the opinion variable. In the first protocol, the support of opinions A and B is amplified until no undecided agents are left. We call this the Pólya Subphase. In the second protocol, agents with opposite opinions cancel each other out, becoming undecided. We call this the Cancellation Subphase. Finally, in the third protocol the single remaining opinion is amplified again. We call this the Broadcasting Subphase. The resulting opinion is copied to the output variable after the working interval I_{work}. Formally, our protocol is specified in Algorithm 1.

In the remainder of this section, we let A_t and B_t denote the number of agents u with $\mathsf{opinion}[u] = A$ and $\mathsf{opinion}[u] = B$, respectively, at time t. Analogously, we let A_t^{IN} and B_t^{IN} denote the number of agents u with $\mathsf{input}[u] = A$ and $\mathsf{input}[u] = B$, respectively, at time t. We now state our main result for this section.

Algorithm 1 Interaction of agents (u, v) in the adaptive majority protocol.

update $\mathsf{clock}[u]$ according to Rules 1–4 with $w = 566$
if *Agent u receives a signal* **then** $\mathsf{opinion}[u] \leftarrow \mathsf{input}[u]$
if $\mathsf{subphase}[u] = 1 \wedge \mathsf{opinion}[u] = U$ **then** $\mathsf{opinion}[u] \leftarrow \mathsf{opinion}[v]$
if $\mathsf{subphase}[u] = 2 \wedge \mathsf{opinion}[u] \neq \mathsf{opinion}[v] \wedge \mathsf{opinion}[u], \mathsf{opinion}[v] \neq U$ **then**
    $\mathsf{opinion}[u], \mathsf{opinion}[v] \leftarrow U$
if $\mathsf{subphase}[u] = 3 \wedge \mathsf{opinion}[u] = U$ **then** $\mathsf{opinion}[u] \leftarrow \mathsf{opinion}[v]$
if $\mathsf{clock}[u] \geq |I_{\text{launch}}| + |I_{\text{work}}|$ **then** $\mathsf{output}[u] \leftarrow \mathsf{opinion}[u]$

▶ **Theorem 12.** *Algorithm 1 is a* $(\mathrm{O}(n \log n), \Omega(\mathrm{poly}\,(n)))$*-loosely stabilizing adaptive majority protocol.*

Note that we match the results of [3] for $r = 1/n$, a multiplicative bias of $1+\beta = 1+1/\log n$ and $\Omega(\log^2 n)$ many agents ($\alpha \geq \log n$). In contrast to their protocol, every agent outputs at any point of time the correct majority opinion, w.h.p. But, again in contrast to their work, we do not consider leaky transitions.

6.2 Analysis

In the following analysis, we consider an arbitrary but fixed phase. We condition on the event that the clock is synchronized according to Proposition 6. We show the following main result, and later in this section we describe how Theorem 12 follows from it this proposition. The proofs for the statements in this section can be found in the full version of this paper.

▶ **Proposition 13.** *Assume that at time t_1 the agents are in a homogeneous launching configuration and we have $A_{t_1} \geq \alpha \cdot \log n$ and $A_{t_1} \geq (1+\beta) \cdot B_{t_1}$. If α and β are large enough constants, then there exists a $t_2 = \Theta(n \cdot w \cdot \log n)$ such that all agents output A in configuration $C_{t_1+t_2}$ with probability $1 - n^{-c}$.*

The analysis is split into three parts, one for the *Pólya Subphase*, one for the *Cancellation Subphase*, and one for the *Broadcasting Subphase*. First, we assume that no changes in the input occur. Then we generalize our results: we adopt the undecided state dynamics introduced in [8], and show how we can tolerate input changes at various rates.

Observe that we get a separation between the subphases from the guarantees of the phase clock in Theorem 2: no two agents are more than 1/6 of I_{work} apart. We also know that every agent has copied its input at the beginning of the phase before the first agent enters the first subphase. The total time for the three subphases (including the separation time) is sufficiently large such that every agent has finished its work before the next phase starts.

When we refer to a distribution *before a subphase*, we mean the distribution at the time just before the first agent performs an interaction in that subphase. Analogously, when we refer to a distribution *after a subphase*, we mean the distribution at the time when the last agent has performed an interaction in that subphase. Recall that in the following analysis, we let A_t and B_t denote the number of agents u with $\mathsf{opinion}[u] = A$ and $\mathsf{opinion}[u] = B$, respectively, at time t. Furthermore, we let s_i and e_i (for *start* and *end*) be the first and the last time, respectively, when an agent performs an interaction in the i'th subphase.

Subphases. We first consider the Pólya Subphase, where we model the process by means of so-called Pólya urns. Pólya urns are defined as follows. Initially, the urn contains a red balls and b blue balls. In each step, a ball is drawn uniformly at random from the urn. The ball's color is observed, and it is returned into the urn along with an additional ball of the same color. The Pólya-Eggenberger distribution $\mathrm{PE}(a, b, m)$ describes the total number of red balls after m steps of this urn process.

This observation allows us to apply concentration bounds to the opinion distribution after the Pólya Subphase. Recall that s_1 and e_1 are the first and the last time steps, respectively, when an agent performs an interaction in the Pólya Subphase. We get the following lemma.

▶ **Lemma 14.** *Let $a = A_{s_1-1}$ and $b = B_{s_1-1}$. For any constant $\beta > 0$ there exists a constant α such that if $a > \alpha \cdot \log n$ and $a > (1+\beta) \cdot b$ then $A_{e_1} - B_{e_1} = \Omega(n)$ with probability at least $1 - n^{-(c+2)}$.*

Next we consider the Cancellation Subphase. The goal is to remove any occurrence of the minority opinion. Whenever an agent with opinion A interacts with another agent with opinion B, both agents become undecided. Formally, we show the following lemma.

▶ **Lemma 15.** *If $A_{s_2-1} - B_{s_2-1} = \Omega(n)$ then $A_{e_2} = \Omega(n)$ and $B_{e_2} = 0$ with probability at least $1 - n^{-(c+2)}$.*

Finally we consider the Broadcasting Subphase. The goal is to spread the (unique) remaining opinion to all other agents. Whenever an undecided agent u interacts with another agent v that has an opinion, agent u adopts the opinion of agent v. This leads to a configuration where every agent has the majority opinion w.h.p. Formally, we show the following lemma.

▶ **Lemma 16.** *If $A_{e_2} = \Omega(n)$ and $B_{e_2} = 0$, then $A_{e_3} = n$ and $B_{e_3} = 0$ with probability at least $1 - n^{-(c+2)}$.*

We have now everything we need to prove Proposition 13 and in turn Theorem 12.

Proof of Proposition 13. We assume the configuration at time t_1 is a homogeneous launching configuration. From Proposition 6 it follows that all configurations in the time interval $[t_1, t_1 + t_2]$ for some $t_2 = \Theta(n \cdot w \cdot \log n)$ are synchronous with probability at least $1 - n^{-(c+1)}$. This means that the three subphases are strictly separated as explained above. It therefore follows (each with probability at least $1 - n^{-(c+2)}$),from Lemma 14 that after the Pólya Subphase no agent is undecided, from Lemma 15 that after the Cancellation Subphase no agent has opinion B, and from Lemma 16 that after the Broadcasting Subphase all agents have opinion A. Once all agents have opinion A, this becomes the output when the agents enter I_{gather}. Together, this shows that all agents output the majority opinion after $\Theta(n \cdot w \cdot \log n)$ interactions with probability at least $1 - n^{-c}$. ◀

Proof of Theorem 12. Here we show the result without input changes. Fix a time t_1 and assume the agents are in an arbitrary configuration at time t_1. From Theorem 3 it follows the agents enter a synchronous configuration within $\mathrm{O}(n \log n)$ interactions and stay in synchronous configurations for $\mathrm{poly}(n)$ time w.h.p.

Now we consider a fixed synchronized phase $i < \mathrm{poly}(n)$ of our phase clock. It follows from Proposition 13 that all agents enter a correct configuration at the end of phase i with probability at least $1 - n^{-c}$. (Recall that in a correct configuration all agents have to output the majority opinion if there is a sufficiently large bias. Without a bias, any output constitutes a correct configuration.) From the guarantees of the phase clock it follows that the first synchronized phase starts within $\mathrm{O}(n \log n)$ time after time t_1 w.h.p. This shows a convergence time of $\mathrm{O}(n \log n)$. From a union bound over at most n^{c-1} phases it follows that the protocol is in a correct configuration for $\mathrm{poly}(n)$ interactions w.h.p. This shows a holding time of $\mathrm{poly}(n)$.

The proof with input changes can be found in the full version. The main idea is that we bound the number of input changes in $\Theta(n \log n)$ interactions by a simple application of Chernoff bounds. ◀

Improving the Bound. In order to show-case the simplicity of the application of our phase clock, we have presented a simplistic protocol, where we assumed a constant factor bias towards the majority opinion. If we replace the Cancellation Subphase and the Broadcasting Subphase (lines 6 to 9 in Algorithm 1) with the *undecided state dynamics* introduced in [8] we can show a tighter result.

Formally, we show the following statement, the proof can be found in the full version of this paper.

▶ Observation 17. If we use the undecided state dynamics, Proposition 13 also holds for $\alpha = \Omega\big(\beta^{-2}\big)$ provided that $\beta = \Omega\big(n^{-1/4+\varepsilon}\big)$.

This means that we can solve the adaptive majority problem with a multiplicative bias of $1 + \beta = 1 + 1/\log n = 1 + o\big(1\big)$ and asymptotically at least $\Omega\big(\log^2 n\big)$ many agents with opinion A or B (assuming sufficiently large constants). Hence we achieve similar results as in [3] for a model without leaky transitions.

References

1 Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2221–2239. SIAM, 2018. doi:10.1137/1.9781611975031.144.

2 Dan Alistarh, Bartlomiej Dudek, Adrian Kosowski, David Soloveichik, and Przemyslaw Uznanski. Robust detection in leak-prone population protocols. In *DNA Computing and Molecular Programming - 23rd International Conference, DNA*, volume 10467 of *Lecture Notes in Computer Science*, pages 155–171. Springer, 2017. doi:10.1007/978-3-319-66799-7_11.

3 Dan Alistarh, Martin Töpfer, and Przemyslaw Uznanski. Comparison dynamics in population protocols. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 55–65. ACM, 2021. doi:10.1145/3465084.3467915.

4 Talley Amir, James Aspnes, and John Lazarsfeld. Approximate majority with catalytic inputs. In Quentin Bramas, Rotem Oshman, and Paolo Romano, editors, *24th International Conference on Principles of Distributed Systems, OPODIS 2020, December 14-16, 2020, Strasbourg, France (Virtual Conference)*, volume 184 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.OPODIS.2020.19.

5 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Comput.*, 18(4):235–253, 2006. doi:10.1007/s00446-005-0138-3.

6 Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC*, pages 292–299. ACM, 2006. doi:10.1145/1146381.1146425.

7 Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Comput.*, 21(3):183–199, 2008. doi:10.1007/s00446-008-0067-z.

8 Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Comput.*, 21(2):87–102, 2008. doi:10.1007/s00446-008-0059-z.

9 Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Comput.*, 20(4):279–304, 2007. doi:10.1007/s00446-007-0040-2.

10 James Aspnes. Clocked population protocols. *J. Comput. Syst. Sci.*, 121:34–48, 2021. doi:10.1016/j.jcss.2021.05.001.

11 Luca Becchetti, Andrea E. F. Clementi, and Emanuele Natale. Consensus dynamics: An overview. *SIGACT News*, 51(1):58–104, 2020. doi:10.1145/3388392.3388403.

12 Stav Ben-Nun, Tsvi Kopelowitz, Matan Kraus, and Ely Porat. An $o(\log^{3/2} n)$ parallel time population protocol for majority with $o(\log n)$ states. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2020, Virtual Event, Italy, August 3-7, 2020*, page to appear, 2020.

13 Petra Berenbrink, Felix Biermeier, Christopher Hahn, and Dominik Kaaser. Self-stabilizing phase clocks and the adaptive majority problem. *CoRR*, abs/2106.13002, 2021. arXiv:2106.13002.

14 Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. Time-space trade-offs in population protocols for the majority problem. *Distributed Computing*, 2020.

15 Petra Berenbrink, Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Peter Kling, and Tomasz Radzik. Time-space trade-offs in population protocols for the majority problem. *Distributed Comput.*, 34(2):91–111, 2021. doi:10.1007/s00446-020-00385-0.

16 Petra Berenbrink, George Giakkoupis, and Peter Kling. Optimal time and space leader election in population protocols. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 119–129. ACM, 2020. doi:10.1145/3357713.3384312.

17 Janna Burman, Ho-Lin Chen, Hsueh-Ping Chen, David Doty, Thomas Nowak, Eric E. Severson, and Chuan Xu. Time-optimal self-stabilizing leader election in population protocols. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 33–44. ACM, 2021. doi:10.1145/3465084.3467898.

18 Anne Condon, Monir Hajiaghayi, David G. Kirkpatrick, and Ján Manuch. Approximate majority analyses using tri-molecular chemical reaction networks. *Nat. Comput.*, 19(1):249–270, 2020. doi:10.1007/s11047-019-09756-4.

19 Shlomi Dolev and Jennifer L. Welch. Self-stabilizing clock synchronization in the presence of byzantine faults. *J. ACM*, 51(5):780–799, September 2004. doi:10.1145/1017460.1017463.

20 David Doty, Mahsa Eftekhari, Leszek Gąsieniec, Eric Severson, Grzegorz Stachowiak, and Przemysław Uznański. A time and space optimal stable population protocol solving exact majority, 2022. FOCS 2022, to appear. arXiv:2106.10201.

21 David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *Distributed Computing*, 31(4):257–271, 2018. doi:10.1007/s00446-016-0281-z.

22 Leszek Gasieniec and Grzegorz Stachowiak. Fast space optimal leader election in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2653–2667. SIAM, 2018. doi:10.1137/1.9781611975031.169.

23 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005. doi:10.1017/CBO9780511813603.

24 Yuichi Sudo, Junya Nakamura, Yukiko Yamauchi, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Loosely-stabilizing leader election in a population protocol model. *Theor. Comput. Sci.*, 444:100–112, 2012. doi:10.1016/j.tcs.2012.01.007.

25 Kunal Talwar and Udi Wieder. Balanced allocations: A simple proof for the heavily loaded case. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP*, volume 8572 of *Lecture Notes in Computer Science*, pages 979–990. Springer, 2014. doi:10.1007/978-3-662-43948-7_81.