Enhanced Phase Clocks, Population Protocols, and Fast Space Optimal Leader Election

LESZEK GĄSIENIEC, University of Liverpool and Augusta University
GRZEGORZ STACHOWIAK, Uniwersytet Wrocławski

The model of *population protocols* refers to the growing in popularity theoretical framework suitable for studying *pairwise interactions* within a large collection of simple indistinguishable entities, frequently called *agents*. In this paper the emphasis is on the space complexity of fast *leader election* in population protocols governed by the *random scheduler*, which uniformly at random selects pairwise interactions between n agents.

One of the main results of this paper is the first fast space optimal *leader election protocol* which works with high probability. The new protocol operates in *parallel time* $O(\log^2 n)$ equivalent to $O(n \log^2 n)$ sequential *pairwise interactions* with each agent's memory space limited to $O(\log\log n)$ states. This double logarithmic space utilisation matches asymptotically the lower bound $\frac{1}{2}\log\log n$ on the number of states utilised by agents in any leader election algorithm with the running time $o\left(\frac{n}{\text{polylog } n}\right)$, see [7].

Our new solution expands also on the classical concept of phase clocks used to synchronise and to coordinate computations in distributed algorithms. In particular, we formalise the concept and provide a rigorous analysis of phase clocks operating in nested modes. Our arguments are also valid for phase clocks propelled by multiple leaders. The combination of the two results in the first time-space efficient leader election algorithm. We also provide a complete formal argumentation indicating that our solution is always correct, fast and it works with high probability.

CCS Concepts: • **Theory of computation → Design and analysis of algorithms**; **Randomness, geometry and discrete structures**.

Additional Key Words and Phrases: population protocols, leader election, randomised algorithm, distributed algorithm.

ACM Reference Format:
Leszek Gąsieniec and Grzegorz Stachowiak. 2020. Enhanced Phase Clocks, Population Protocols, and Fast Space Optimal Leader Election. *J. ACM* 1, 1 (September 2020), 21 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The model of *population protocols* adopted in this paper was introduced in the seminal paper of Angluin *et al.* [3]. Their model provides a universal theoretical framework for studying pairwise interactions within a large collection of indistinguishable entities, very often referred to as *agents* equipped with fairly limited storage, communication and computation capabilities. The agents are modelled as finite state machines. When two agents engage in a direct interaction their memory content is assessed and their states are modified according to the transition function that forms an integral part of the population protocol. In the *probabilistic variant* of population protocols,

Authors' addresses: Leszek Gąsieniec, Department of Computer Science, University of Liverpool, School of Computer and Cyber Sciences, Augusta University, l.a.gasieniec@liverpool.ac.uk; Grzegorz Stachowiak, Instytut Informatyki, Uniwersytet Wrocławski, gst@cs.uni.wroc.pl.

0004-5411/2020/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

considered in [3] and adopted in this paper, in each step the *random scheduler* selects a pair of agents uniformly at random. In this variant, in addition to space utilisation reflecting the maximum number of distinct *states* available at each agent, one is also interested in the *running time* of the considered algorithmic solutions. More recent studies on population protocols focus on the performance in terms of *parallel time* defined as the total number of pairwise interactions leading to stabilisation divided by the size (in our case n) of the population. Please note that the parallel time is also a good estimate of the number of interactions each agent was involved in.

A population protocol *terminates with success* if the whole population eventually stabilises, i.e., it arrives at and stays indefinitely in the final configuration of states reflecting the desired property of the solution. For example, in protocols targeting the majority in the population, the final configuration corresponds to each agent being in a state representing the colour of the majority, see, e.g., [4, 6, 39, 40, 49]. In *leader election*, however, in the final configuration a single agent is expected to conclude in a *leader* state and all other agents must stabilise in *follower* states. The leader election problem received in recent years greater attention in the context of population protocols thanks to a number of important developments in closely related challenges [27, 31]. In particular, the results from [27, 31] laid down the foundation for the proof that leader election cannot be solved in a sublinear time with agents utilising a fixed number of states [33]. In further work [9], Alistarh and Gelashvili studied the relevant upper bound, where they proposed a new leader election protocol stabilising in time $O(\log^3 n)$ assuming $O(\log^3 n)$ states per agent.

In a very recent work Alistarh *et al.* [7] consider a more general trade-off between the number of states used by agents and the time complexity of stabilisation. In particular, the authors provide a separation argument distinguishing between *slowly stabilising* population protocols which utilise $o(\log\log n)$ states and *rapidly stabilising* protocols with $O(\log n)$ states per agent. This result nicely coincides with another fundamental observation by Chatzigiannakis *et al.* [26] which states that population protocols utilizing $o(\log\log n)$ states are limited to semilinear predicates, while the availability of $O(\log n)$ states admits computation of symmetric predicates. More recent developments include also a protocol which elects the leader in time $O(\log^2 n)$ w.h.p. and in expectation utilizing $O(\log^2 n)$ states [21]. The number of states was later reduced to $O(\log n)$ by Alistarh et al. in [8] and by Berenbrink et al. in [19] through the application of two types of synthetic coins.

The most recent developments in population protocols solving the *majority problem* include [8], where Alistarh *et al.* show a lower bound $\Omega(\log n)$ on the number of states required by any protocol which stabilises in time $O(n^c)$, for any constant $c \leq 1$. They also match this bound from above by an algorithm which utilises $O(\log n)$ states at each agent, and stabilises in time $O(\log^2 n)$. This time performance has been recently improved to $O(\log^{5/3} n)$ by Berenbrink et al. in [17] and later to $O(\log^{3/2} n)$ by Ben Nun at al. in [16]. Two excellent surveys [10, 34] provide a more detailed discussion on recent advances in population protocols.

Our results. We show that the lower bound on the space complexity in the fast (polylogarithmic in n) leader election proved in [7] is asymptotically tight. The lower bound argument indicates that any leader election algorithm with the time complexity $o(\frac{n}{\text{polylog } n})$ requires $\frac{1}{2}\log\log n$ states per agent. We present a new fast *leader election* algorithm which stabilises in time $O(\log^2 n)$ in populations with agents utilising $c\log\log n$ states, for a sufficiently large constant c.

Our algorithm utilises a fast and small space reduction of potential leaders (candidates) in the population. The reduction process is intertwined with a novel robust initialisation and further utilisation of nested *phase clocks*. This synchronisation tool was developed and broadly applied in the self-stabilising literature [45]. Relevant work includes the seminal studies on clock synchronisation by Arora *et al.* [11], a further extension by Dolev and Welsh [30] to distributed systems prone to Byzantine faults, and a related research on pulse synchronisation by Daliot *et al.* [32]. Our variant of the phase clock refers directly to the work of Angluin *et al.* [5] in which the authors propose an efficient simulation of a *virtual register machine* supporting basic arithmetic operations. The simulation in [5] assumes availability of a single leader which coordinates the relevant exchange of information. In the same paper, the authors provide also some intuition behind the phase clock coordinated by a *junta* of n^ε leaders, for a small positive constant ε. In this work we prove that the phase clock based on a junta of cardinality n^ε, for any $\varepsilon < 1$, allows agents to count $\Theta(\log n)$ parallel time assuming a constant number of states per agent. We also consider an extension of the phase clock allowing to measure time $\Theta(\log^c n)$, for any integer constant c. Our main result is based on a rapid computation of a *junta* of leaders followed by a fast election of a single leader, all in time $O(\log^2 n)$ and $O(\log \log n)$ states available at each agent. Please note that the time complexity $O(\log^2 n)$ is secured with high probability. In a recent work Gąsieniec et al. [41] proposed a space optimal leader election protocol which stabilises in the expected parallel time $O(\log n \log \log n)$. And this result was later improved to the optimal time complexity $O(\log n)$ by Berenbrink et al. in [20].

Related work. Leader election is one of the fundamental problems in the field of Distributed Computing on par with *broadcasting*, *mutual-exclusion*, *consensus*, see, e.g., an excellent text book by Attiya and Welch [14]. The problem was originally studied in networks with nodes having distinct labels [47], where an early work focuses on the ring topology in synchronous [36, 46] as well as in asynchronous models [24, 51]. Also, in networks populated by mobile agents leader election was studied first in networks with labeled nodes [44]. However, very often leader election is also used as a powerful symmetry breaking mechanism enabling the feasibility and coordination of more complex protocols in systems based on uniform (indistinguishable) components. There is a large volume of work [2, 12, 13, 22, 23, 52, 53] on leader election in anonymous networks. In [52, 53] we find a characterisation of message-passing networks in which leader election is feasible when the nodes are anonymous. In [52], the authors study the problem of leader election in general networks under the assumption that the node labels are not unique. In [35], the authors study feasibility and message complexity of leader election in rings with possibly non-unique labels, while in [29] the authors provide solutions to generalised leader election in rings with arbitrary labels. The work in [38] focuses on space requirements for leader election in unlabelled networks. In [37], the authors investigate the running time of leader election in anonymous networks where the time complexity is expressed in terms of multiple network parameters. In [28], the authors study feasibility of leader election for anonymous agents that navigate in a network asynchronously. Also, an interesting study on trade-offs between the time complexity and knowledge available in anonymous trees can be found in recent work of Glacet *et al.* [43].

Another good example of the current studies on the exact space complexity in various models refers to plurality consensus. In particular, in [18] Berenbrink *et al.* proposed a plurality consensus protocol for C original opinions which converges in $O(\log C \cdot \log \log n)$ synchronous rounds using at most $\log C + O(\log \log C)$ bits of the local memory. They also show a slightly slower solution converging in $O(\log n \cdot \log \log n)$ rounds and using at most $\log C + 4$ bits of local memory. This disproved the conjecture by Becchetti et al. [15] implying that any protocol utilising $\log C + O(1)$ bits has the worst-case running time $\Omega(C)$. In [42] Ghaffari and Parter propose an alternative algorithm converging in time $O(\log C \log n)$ in which the messages and the local memory utilise $\log C + O(1)$ bits. In addition, some work on the application of the random walk in plurality consensus protocols can be found in [15, 39].

2 PRELIMINARIES AND OVERVIEW

We consider population protocols defined on the complete graph of interactions where the *random scheduler* picks uniformly at random pairs of agents drawn from the population of size n. The agents are anonymous, i.e., they don't have identifiers. The protocol assumes that all agents start in the same initial state. Our protocol utilises the classical model of population protocols [3, 5] where the consecutive interactions refer to ordered pairs of agents, namely (`initiator`, `responder`). On the conclusion of each interaction the two participating agents change their states (a, b) into (a', b') according to a fixed *deterministic transition function* denoted by $(a, b) \to (a', b')$.

Random coins. For the simplicity of presentation, in this paper we dispense fair random coins w.h.p. by observing actions of the random scheduler. It has been shown, however, that agents can generate *synthetic coins* which become almost uniform after a constant number of interactions [7]. More information about synthetic coins can be found in survey [10].

We focus here on two complexity measures: (1) the *space complexity* defined as the *number of states* utilised by each agent, and (2) the *time complexity* reflecting the total number of interactions required to stabilise the population protocol. In accordance to other recent work in the field, the emphasis here is on the *parallel time* of the solution defined as the total number of interactions divided by the population size. This time can be also seen as the local time observed by an agent, i.e., the number of pairwise interactions which the agent is involved in. In this work we aim at protocols based on $O(n \cdot \text{poly} \log n)$ interactions equivalent to the parallel time $O(\text{poly} \log n)$.

Our leader election algorithm is always correct and it stabilises rapidly *with high probability (w.h.p.)* which we define as follows. Let η be a universal constant referring to the reliability of our protocols. We say that an event occurs with *negligible* probability if it occurs with probability at most $n^{-\eta}$, and an event occurs with high probability (w.h.p.) if it occurs with probability at least $1 - n^{-\eta}$. This estimate is of an asymptotic nature, i.e., we assume n is large enough to validate the results. Similarly, we say that an algorithm succeeds with high probability if it succeeds with probability at least $1 - n^{-\eta}$. When we refer to the probability of failure p possibly different to $n^{-\eta}$, we say w.p. $1 - p$, which abbreviates *with probability at least* $1 - p$.

2.1 One-way epidemic

In our solution we adopt the concept of the *one-way epidemic* introduced in [5]. The one-way epidemic refers to the population protocol with state space $\{0, 1\}$ and transition rule $(x, y) \to (x, \max\{x, y\})$. One interprets 0's as *susceptible* (still healthy) agents and 1's as *infected* ones. This protocol corresponds to a simple epidemic process in which the transmission of the infection occurs if and only if the initiator is infected and the responder is susceptible. We will use the following theorem introduced in [5].

THEOREM 1 ([5]). *The stabilisation of the one-way epidemic (resulting in all agents being infected) starting with a single infected agent needs* $\Theta(n \log n)$ *pairwise interactions w.h.p.*

2.2 Overview

In [5] Angluin *et al.* defined and further analysed the concept of *phase clocks* capable of counting parallel time $\Theta(\log n)$ approximately, in which each agent participating in the population protocol utilises a constant number of states. The phase clocks studied in [5] work under the assumption of having already determined a unique leader in the population. In the same paper, the authors argue without giving any formal argument that phase clocks should also work with a *junta* of n^{ε} leaders, for some unspecified constant ε. Further on, the authors suggest that once the phase clock is in motion the junta of leaders can be reduced to a single leader with the help of coin tossing combined with propagation via the one-way epidemic. This process would lead to the election of a single leader in $\Theta(n \log^2 n)$ interactions assuming the relevant phase clock junta is already established. In this paper we adopt a similar approach in search for a single leader.

The protocol begins with the election of a junta of leaders of size $O(\sqrt{n \log n})$. We provide a brief explanation of how this is done in the next paragraph. Then using phase clocks based on this junta we gradually reduce it to a single leader. We implement this process in two *loops*, one nested inside the other. The *internal loop* operates in parallel time $\Theta(\log n)$ equivalent to $\Theta(n \log n)$ interactions, which in principle mimics actions of a standard finite state phase clock. This time interval allows each leader to choose at random value 0 or 1, and to distribute 1's throughout the population via the one-way epidemic. In turn, all leaders which drew 0s and received 1s get eliminated. The *external loop* is used to count $\Theta(\log n)$ executions to guarantee the computation of a single leader w.h.p. The *external loop* is controlled by a finite state phase clock too. However, this time the agents execute clock operations more seldom, i.e., only when they act for the first time as responders after each full execution of the internal loop. This way a single step of the external phase clock refers to time $\Theta(\log n)$ observed on the internal phase clock, and a full turn on the external phase clock is equivalent to the total time $\Theta(\log^2 n)$.

The process of electing a junta of size $O(\sqrt{n \log n})$ is done via the promotion of agents along subsequent levels. All agents start at level $l = 0$. Initially, each agent is eager to be promoted which is indicated by its dedicated variable $a = 1$. In due course, when the agent stops being promoted, $a = 0$ indefinitely. In particular, the first interactions of all agents retain permanently at least half of them at level $l = 0$, when the relevant variable a is altered to 0. The remaining agents are promoted to level $l = 1$ with $a = 1$. Later on, the agents interacting with those on the same or higher levels continue being promoted, while those interacting with the lower levels cease further promotions by adopting $a = 0$. We show that if a fraction A of n agents reaches level l, then the fraction that gets promoted to level $l + 1$ is of order A^2 w.h.p. In turn, the highest level reached by any agent is $\log \log n + O(1)$ w.h.p. The agents reaching this level form a junta of size $O(\sqrt{n \log n})$ w.h.p. We also show that junta election is concluded in time $\Theta(\log n)$.

For the convenience of the reader we provide the logical structure of the full argument in the form of a diagram, see Figure 1.

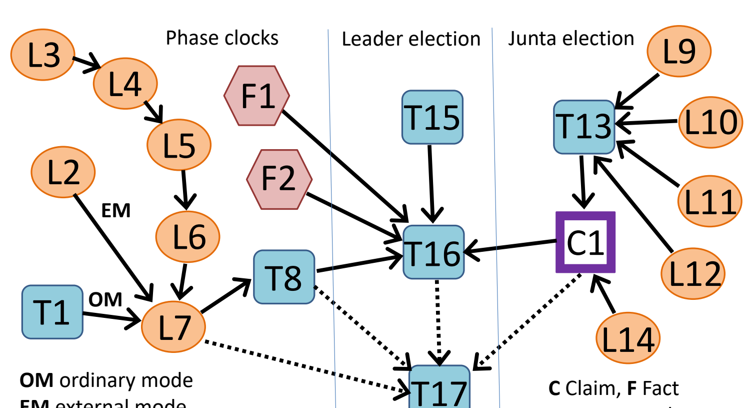

Fig. 1. The structure of the argument

3 PHASE CLOCK REVISITED

In this section we propose and analyse a modified version of phase clocks capable of approximate counting of time $\Theta(\log n)$, under the assumption that each agent utilises a constant number of states and the junta of leaders is of cardinality at most $n^{1-\varepsilon}$, for any constant ε, s.t., $0 < \varepsilon < 1$. For a technical reason and without loss of generality we adopt $\varepsilon = \varepsilon(k) = \frac{3}{3k+1}$, for some integer $k > 0$.

The states of agents controlling our phase clock protocol are structured as ordered pairs (x, b). The entry b is set to `leader` for leaders in the junta and to `follower` for all other agents in the population. The entry x represents the current clock *phase* of an agent, which is a number from the set $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$, where m is a constant positive integer. The clock phases can be interpreted as hours on the dial of an analogue clock. The observed clock phase increment is periodic and computed using the arithmetic modulo m denoted by $+_m$. We also define the maximum of two clock phases x, y in set $\mathbb{Z}_m$ as:

$$\max_m\{x, y\} = \begin{cases} \max\{x, y\} \text{ if } |x-y| \le m/2, \\ \min\{x, y\} \text{ if } |x-y| > m/2. \end{cases}$$

Finally we define a *cyclic order* on $\mathbb{Z}_m$, which is not partial, as $x \le_m y$ iff $\max_m\{x, y\} = y$.

S-sectors. On the clock dial we define m different *s-sectors* $sec_s(i)$, for $i = 0, \ldots, m-1$. Each sector $sec_s(i)$ is formed of s consecutive integers drawn from $\mathbb{Z}_m$, where i is the largest (in the cyclic order) integer in $sec_s(i)$. We refer to i as the *forefront* of $sec_s(i)$. We also distinguish the *primary sector* $sec_s(0) = \{m-s+1, \ldots, m, 0\}$.

The invariant. In our solutions we will use the following invariant. We say that the *population belongs to s-sector* $sec_s(i)$ if all agent phases are in this sector and the most advanced clock phase in the population is equal to i.

The population progresses through consecutive s-sectors in the clockwise direction. In particular, the population makes progress to sector $sec_s(i)$ when the most advanced clock phase present in the population gets incremented to the forefront i. Only the leaders contribute to this progress. The followers can only replicate the clock phases of other agents.

Rounds. The first round starts when the clocks of all agents are in phase 0, i.e., the whole population belongs to the primary s-sector. Any other *round* begins when the population progresses again to the primary sector.

The phase clock satisfies the following three properties w.h.p., for a polynomial period of time.

(1) The invariant stating that clock phases of all agents belong to some s-sector is maintained.
(2) For any constant $d > 0$, there exists m, s.t., the following statement holds. In each round there are at least $dn \log n$ interactions between the population makes progress to s-sector $sec_s(s)$ and the beginning of the next round.
(3) The parallel time of any *round* is $O(\log n)$.

The transition function governing actions of phase clocks is defined as follows:

$$((y, b), (x, \texttt{follower})) \rightarrow ((y, b), (\max_m\{x, y\}, \texttt{follower})),$$

and

$$((y, b), (x, \texttt{leader})) \rightarrow ((y, b), (\max_m\{x, y +_m 1\}, \texttt{leader})).$$

The loop structure. In this paper we carry out leader election with the help of repetitive drawing of 0/1-bits by the leaders, and further spreading of these values across the population. In our solution we utilise two loops nested in one another. The internal loop is used to force parallel time at least

$d \log n$. This parallel time allows leaders to draw all required 0/1-bits, to broadcast these bits via an epidemic process, and to terminate the epidemic process successfully. The external loop is used to execute the internal one at least $d \log n$ times.

In order to manage the two-tier loop structure with the help of constant size memory we utilise phase clocks to count approximately: (1) the number of interactions during each execution of the internal loop; and (2) the number of the internal loop executions. More precisely, we introduce hierarchical phase clocks operating in two modes nested in one another. The *ordinary mode* is used to count the interactions within the internal loop and the *external mode* to count the executions of the internal loop. These two modes of the phase clock share certain principles and tools. E.g., the same junta of leaders is used in both modes while each mode is using dissemination via the one-way epidemic independently. However, as these two modes count their phases independently, we will often interpret them as two different phase clocks. We say that the clock phase of an agent *passes through* 0 whenever its current phase $x <_m 0$ changes to a new phase $x' \geq_m 0$. As hinted earlier in this section the two modes differ in selecting pairwise interactions to the relevant phase clock actions.

- In the *ordinary mode* all interactions triggered by the random scheduler prompt actions of the phase clock. And once the ordinary mode clock phase passes through zero a *meaningful interaction* of the external mode occurs, where
- a *meaningful interaction* refers to the first interaction of an agent after its ordinary phase clock passes through 0 in which also the agent acts as the responder. Each agent passes through zero exactly once when the ordinary clock phases of all other agents are in some s-sector containing 0. Therefore all meaningful interactions can be split into consecutive *blocks* of length n, where each block is w.h.p. formed of an arbitrary permutation of all agents acting as responders matched with the corresponding initiators chosen at random (by the random scheduler).
- In the *external mode* only meaningful interactions are utilised in phase clock operations. All other interactions are ignored.

Observation. The use of two nested loops allows us to count parallel time $\Theta(\log^2 n)$. One can extend this approach to more than two loops nested one in another. In such a case, the most internal loop would be propelled by a phase clock running in the ordinary mode and all other (more external) loops would operate on clocks working in the external mode. And the use of k nested loops would allow to count parallel time $\Theta(\log^k n)$, for any integer constant k.

When proving facts about phase clocks operating in the external mode, for the convenience of presentation we introduce the notion of a *virtual external mode scheduler.* This scheduler generates a series of blocks of meaningful interactions. Each block is of length n. And in every block each agent acts as the responder exactly once with the initiators chosen at random.

Before we proceed with the full proof of Theorem 8, i.e., the main result of this section, we share with the reader several useful observations structured as lemmas. In the proofs referring to the ordinary mode we utilise Theorem 1 and we show that the one-way epidemic protocol concludes after $\Theta(n \log n)$ interactions w.h.p. We also need an analogue of this theorem for the external mode.

Lemma 2. *The one-way epidemic requires $O(n \log n)$ interactions from the external mode scheduler to stabilise w.h.p.*

Proof. Let v be the first infected agent. By the Chernoff bound, for any constant $c_1 > 0$ the number of blocks of n interactions agent v needs to infect directly $c_1 \log n$ agents is $O(\log n)$ w.p. $1 - n^{-\eta-1}$. Thus the number of infected agents after $O(n \log n)$ interactions is at least $c_1 \log n$ w.p. $1 - n^{-\eta-1}$. Also by the Chernoff bound, there exists a constant $c_1 > 0$, s.t., if the number of infected agents before a *block* of n interactions is A, where $c_1 \log n < A < n/2$, then on the conclusion of this block the number of other agents infected directly by these agents is at least $\frac{1}{4} \cdot A$ w.p. $1 - n^{-\eta-1}$. Thus thanks to the exponential growth, the number of infected agents reaches $n/2$ after $O(n \log n)$ interactions w.p. $1 - O(n^{-\eta-1} \log n)$. Furthermore, by taking an extra $c_2 \log n$ blocks of pairwise interactions each yet uninfected agent interacts $c_2 \log n$ times as the responder. One can choose a constant c_2, s.t., the probability of not getting infected during these interactions is at most $n^{-\eta-1}$ for a fixed uninfected agent. Finally, by the Union bound the probability of failure in any of these steps is at most $n^{-\eta-1} + O(n^{-\eta-1} \log n) + n^{-\eta-1} \cdot n/2 < n^{-\eta}$. □

For the simplicity of presentation in the next few lemmas we assume that the agents always start in phase 0. The lemmas can be easily modified to accommodate starting in any other phase. We also assume here that $\varepsilon = \frac{3}{3k+1}$ and $k < m/4$. The main purpose of these lemmas is to bound from above the sizes of sets of agents being in phases $1, 2, 3, \ldots$ on the conclusion of $O(n \log n)$ interactions. In what follows we formulate several lemmas which describe the behavior of phase clocks powered by interactions triggered by the random or the external mode scheduler. In particular, in Lemma 3 we consider sequences of interactions with the specifications reflecting the needs of Lemma 4.

Lemma 3. *Assume $j \leq k$ and interactions of the phase clock are triggered by either the random scheduler or the external mode scheduler. Assume also that at some point the number of agents in phases $x \geq_m i$ is at most $A \cdot n^{1-i\varepsilon}$, for all $i = 0, 1, \ldots, j$, and some value $A \in [1, n^{\varepsilon/3}]$. We consider either of the two:*

- *a sequence of at most $n/4$ consecutive interactions triggered by the random scheduler;*
- *a sequence of at most $n/4$ consecutive interactions in which no more than $n^{1-i\varepsilon}/4$ leaders act as responders, all drawn from a single block triggered by the external mode scheduler.*

After this sequence of interactions, the number of agents in phases $x \geq_m i$ is at most $3A \cdot n^{1-i\varepsilon}$, for all $i = 0, 1, \ldots, j$, w.p. at least $1 - 2j \cdot n^{-10}$.

Proof. We prove this lemma by induction on j. For $j = 0$ the thesis holds since the number of agents in phases $x \geq_m 0$ is at most $n < 3A \cdot n^{1-0\cdot\varepsilon}$ with probability 1. Assume now, the thesis is true for $j - 1$ and we extend it to value j. By the inductive assumption after the considered sequence of interactions the number of agents in phases $x \geq_m i$ is bounded from above by $3A \cdot n^{1-i\varepsilon}$, for all $i = 0, 1, \ldots, j - 1$, w.p. $1 - 2(j - 1)n^{-10}$.

Two types of agents can enter phases $x \geq_m j$ during these interactions. The first type refers to the leaders. A leader can enter phase $x \geq_m j$ if it acts as the responder in the interaction with some initiator in phase $y \geq_m j - 1$. The second type of new agents in phases $x \geq_m j$ refers to the followers. A follower enters phase $x \geq_m j$, if it interacts as the responder with an initiator being in phase $x \geq_m j$.

In both cases, the number of new leaders in phases $x \geq_m j$ can be bounded from above by examining a sequence σ of 0- and 1-bits corresponding to the relevant interactions. Initially the sequence σ is empty. If during an interaction a leader is moved to phase $x \geq_m j$ we extend σ by a 1-bit. During all other interactions the value of the extra bit varies. Assume the number of agents in phases $y \geq_m j - 1$ is at most $3A \cdot n^{1-(j-1)\varepsilon}$, which holds due to the inductive hypothesis w.p. $1 - 2(j - 1)n^{-10}$. We prove that σ contains at most $A \cdot n^{1-j\varepsilon}$ 1-bits w.p. $1 - n^{-10}$. This upperbounds the number of leaders entering phases $x \geq_m j$ by $A \cdot n^{1-j\varepsilon}$ w.p. $1 - n^{-10}$.

- First we give an estimation in the case with the random scheduler. A leader can enter phase $x \geq_m j$ if it interacts as the responder with an initiator being in phase $y \geq_m j - 1$. Thus the probability p_ι that a leader enters phase $x \geq_m j$ during an interaction ι belonging to the sequence is at most $3A \cdot n^{1-(j-1)\varepsilon} n^{1-\varepsilon}/n^2 = 3A \cdot n^{-j\varepsilon}$. Now we specify how σ is formed. If a leader enters phase $x \geq_m j$, we add a 1-bit to σ. If no leader enters phase $x \geq_m j$, a

1-bit is added to σ with probability $(3A \cdot n^{-j\varepsilon} - p_\iota)/(1 - p_\iota)$, and a 0-bit otherwise. This way the values of all bits in σ are independent, and each bit is equal to 1 with probability $3A \cdot n^{-j\varepsilon}$. The expected number of 1-bits in σ is at most $\frac{3}{4}A \cdot n^{1-j\varepsilon} \geq \frac{3}{4}n^{\varepsilon/3}$. By the Chernoff bound, the probability that this number is larger than $A \cdot n^{1-j\varepsilon}$ is negligible and smaller than $e^{-n^{\varepsilon/3}/36} < n^{-10}$, for sufficiently large n.

- Now we give the upper bound for the external mode scheduler considering only interactions of leaders that are in phases $x' <_m j$. Such a leader can enter phase $x \geq_m j$ if it interacts as the responder with an initiator being in phase $y \geq_m j - 1$. There are at most $n^{1-\varepsilon}/4$ interactions ι in the sequence in which leaders being in phases $x' <_m j$ act as the responders. The probability p_ι that such a leader enters a phase $x \geq_m j$ during interaction ι is at most $3A \cdot n^{-(j-1)\varepsilon}$. We construct the sequence σ as follows. If the leader enters phase $x \geq_m j$, we add a 1-bit to σ. If the leader does not enter phase $x \geq_m j$, a 1-bit is inserted to σ with probability $(3A \cdot n^{-(j-1)\varepsilon} - p_\iota)/(1 - p_\iota)$, and a 0-bit otherwise. This way the values of all bits in σ are independent, and each bit is equal to 1 with probability $3A \cdot n^{-(j-1)\varepsilon}$. If σ has less than $n^{1-\varepsilon}/4$ entries we add the missing entries by independent coin tosses each time obtaining 1 with probability $3A \cdot n^{-(j-1)\varepsilon}$, and 0 with the remaining probability. The expected number of 1-bits in σ is $\frac{3}{4}A \cdot n^{1-j\varepsilon} \geq \frac{3}{4}n^{\varepsilon/3}$. By the Chernoff bound, the probability that this number is greater than $A \cdot n^{1-j\varepsilon}$ is negligible and less than $e^{-n^{\varepsilon/3}/36} < n^{-10}$, for sufficiently large n.

The number of followers entering phases $x \geq_m j$ can be bounded from above for both schedulers in the same manner. We assume that the inequalities from the inductive hypothesis of the Lemma hold and the number of new leaders that entered phases $y \geq_m j$ is at most $An^{1-j\varepsilon}$. Due to the inductive hypothesis this assumption holds w.p. $1 - (2j - 1)n^{-10}$. To prove the upper bound we define a sequence ρ of length $n/4$ consisting of 0- and 1-bits which correspond to the relevant interactions. Initially ρ is empty. During each consecutive interaction ι, in which a follower being in phase $x' <_m j$ acts as the responder, the sequence ρ gets extended by a single bit. Such a follower enters phase $x \geq_m j$, if it is matched with an agent in phase $x \geq_m j$. Let p_ι be the probability of such an event. If $p_\iota > 3A \cdot n^{-j\varepsilon}$, then a 1-bit is inserted to ρ with probability $3A \cdot n^{-j\varepsilon}$, and a 0-bit otherwise. If $p_\iota \leq 3A \cdot n^{-j\varepsilon}$ and the follower enters phase $x \geq_m j$, a 1-bit is added to ρ. If $p_\iota \leq 3A \cdot n^{-j\varepsilon}$ and the follower does not enter phase $x \geq_m j$, then a 1-bit is added to ρ with probability $(3A \cdot n^{-j\varepsilon} - p_\iota)/(1 - p_\iota)$ and a 0-bit otherwise. Note that until more than $A \cdot n^{1-j\varepsilon}$ followers enter phase $x \geq_m j$, $p_\iota \leq 3A \cdot n^{1-j\varepsilon}/n = 3A \cdot n^{-j\varepsilon}$. If the resulting sequence ρ is shorter than $n/4$, then each missing entry is filled in with a 1-bit with probability $3A \cdot n^{-j\varepsilon}$, and a 0-bit otherwise. This way similarly to sequence σ the values of all bits in $n/4$-bit sequence ρ are independent, and each bit is equal to 1 with probability $3A \cdot n^{-j\varepsilon}$. The expected number of 1s in ρ is $\frac{3}{4}A \cdot n^{1-j\varepsilon} \geq \frac{3}{4}n^{\varepsilon/3}$. By the Chernoff bound the probability that this number is larger than $A \cdot n^{1-j\varepsilon}$ is negligible and smaller than $e^{-n^{\varepsilon/3}/36} < n^{-10}$, for sufficiently large n. Due to the initial assumptions at least first $A \cdot n^{1-j\varepsilon}$ followers entering phase $x \geq_m j$ are associated with 1-bits in ρ. Thus the number of followers entering phases $x \geq_m j$ is not greater than $A \cdot n^{1-j\varepsilon}$ w.p. $1 - n^{-10}$.

This concludes the proof that the number of agents in phases $x \geq_m i$ is at most $3A \cdot n^{1-i\varepsilon}$, for all $i = 0, 1, \ldots, j$, w.p. $1 - 2j \cdot n^{-10}$. □

Lemma 4. *Assume all agents start in the clock phase 0. The probability that after $\frac{1}{8(3k+1)} n \log_3 n$ interactions (either for the random or the external mode scheduler) there are at least $n^{2/(3k+1)}$ agents in phases $x \geq_m k$, is at most $2(\varepsilon/3)k \log_3 n \cdot n^{-10}$.*

Proof. In the beginning all agents are in phase 0, i.e., there are no agents in any other phase. Thus the number of agents in phases $x \geq_m i$ is at most $3A \cdot n^{1-i\varepsilon}$, for all $i = 0, 1, \ldots, k$, and $A = 1$.

We consider a sequence of $\frac{1}{8(3k+1)} n \log_3 n$ consecutive interactions. When the random scheduler is used, we split this sequence into consecutive subsequences of $n/8$ interactions. However, for the external mode scheduler, we first split this sequence into subsequent blocks of n interactions. Recall that in each block every agent acts as the responder exactly once. Each block is further split into eight subsequences, possibly of different lenghts, s.t., each subsequence contains at most $n/4$ interactions and in at most $n^{1-\varepsilon}/4$ interactions leaders act as responders. This way the series of $\frac{1}{8(3k+1)} n \log_3 n$ subsequent interactions is split into $\frac{1}{3k+1} \log_3 n$ subsequences. Now we apply to these subsequences Lemma 3, with increasing values of A equal to $1, 3, 9, \ldots, n^{1/(3k+1)}/3 = n^{\varepsilon/3}/3$. By Lemma 3 the number of agents in phases $x \geq_m k$ after all $\frac{1}{8(3k+1)} n \log_3 n$ interactions exceeds $n^{\varepsilon/3} n^{1-k\varepsilon} = n^{2/(3k+1)}$ with probability at most $2(\varepsilon/3)k \log_3 n \cdot n^{-10}$, which concludes the proof of this lemma. □

Lemma 5. *Assume all agents start in the clock phase 0. The probability that on the conclusion of $\frac{n \log_3 n}{8(3k+1)}$ interactions (either for the random or the external mode scheduler) there are any agents in phase $x \geq_m k+1$ is $O(n^{-\varepsilon/3} \log n)$.*

Proof. Recall that clock phase $x = k+1$ can be entered only by a leader which interacts as the responder with another agent being in phase $x = k$. By Lemma 4, the number of agents in clock phase $x = k$ is at most $n^{2\varepsilon/3}$ w.h.p. during the considered sequence of interactions.

For the random scheduler, the probability of incrementing clock to phase $x = k+1$ in a given interaction is at most $n^{1-\varepsilon} \cdot n^{2\varepsilon/3}/n^2 = n^{-1-\varepsilon/3}$. For the external mode scheduler in each block of n interactions, the relevant leaders act as responders $n^{1-\varepsilon}$ times. Thus the probability that any leader interacts as the responder with an agent in phase $x = k$ is at most $n^{1-\varepsilon} \cdot n^{2\varepsilon/3}/n = n^{-\varepsilon/3}$.

In conclusion, by the Union bound the probability of having such interactions during $\frac{n \log_3 n}{8(3k+1)}$ subsequent interactions in any mode is $O(n^{-\varepsilon/3} \log n)$. □

Lemma 6. *Assume all agents start in clock phase $x = 0$ and d is a positive constant. There exists an integer constant $K < m/2$, s.t., the probability that the first agent enters phase $x = K$ before interaction $dn \log n$ is negligible, for sufficiently large n.*

Proof. Assume that all relevant phases $x = 1, 2, \ldots K$ are grouped into κ consecutive chunks, for some integer $\kappa > 0$. Moreover, each chunk is formed of k phases, where k satisfies $\varepsilon = \frac{3}{3k+1}$ and $K = \kappa \cdot k$. Let t_i, for all $i = 0, 1, 2, \ldots, \kappa$, be the first interaction in which an agent enters phase $i \cdot k$, where $t_0 = 0$. Note that prior to interaction t_i all agents are in phases $x <_m ik$. Thus these agents progress to the subsequent clock phases not faster than if they already were in phase $i \cdot k$ just after interaction t_i. By Lemma 5 the probability that $t_i - t_{i-1} < \frac{n \log_3 n}{8(3k+1)}$ is smaller than $cn^{-\varepsilon/3} \log n$, for some constant $c > 0$. The probability, that for at least κ' different values i we have $t_i - t_{i-1} \leq \frac{n \log_3 n}{8(3k+1)}$ is by Union bound (for n large enough) smaller than

$$\binom{\kappa}{\kappa'} \left(cn^{-\varepsilon/3} \log n\right)^{\kappa'} \leq \kappa^{\kappa-\kappa'} \left(cn^{-\varepsilon/3} \log n\right)^{\kappa'}.$$

Now, for $\kappa' = 4\eta/\varepsilon$ and $\kappa - \kappa' = d \cdot 8(3k+1) \log 3$ and for sufficiently large n, we obtain $t_\kappa \leq (\kappa - \kappa') \cdot \frac{n \log_3 n}{8(3k+1)} = dn \log n$ with probability at most $\kappa^{\kappa-\kappa'} \left(cn^{-\varepsilon/3} \log n\right)^{4\eta/\varepsilon} < n^{-\eta}$. □

Lemma 7. *For any constant $d > 0$ there is another constant $K > 0$, s.t., if $m > 6K$ and after interaction t there is an agent in phase i and all other agents are in phases $x : i - 2K \leq_m x \leq_m i$, then w.h.p.*

- *the first interaction t' during which an agent enters phase $i + K$ satisfies $t' > t + dn \log n$, and*

- *on the conclusion of interaction t' all agents are in phases x, where $i \leq_m x \leq_m i + K$.*

Proof. By Theorem 1 and Lemma 2 there exists a positive constant d', s.t., the one-way epidemic succeeds within $d' \cdot n \log n$ interactions w.h.p. On the other hand by Lemma 6, for a constant $D = \max\{d, d'\}$ there exists a constant K, s.t., after $D \cdot n \log n$ interactions all agents starting in phase i move to phase smaller or equal to $i + K - 1$ w.h.p. One can observe that if agents start in phases $x : i - 2K \leq_m x \leq_m i$, the phases on their clocks after $D \cdot n \log n$ interactions do not exceed (in the sense of $\leq_m$) those reached in the case when all agents start in phase i. Thus $t' > Dn \log n \geq dn \log n$ w.h.p. Since the one-way epidemic initiated by an agent in phase i during interaction t succeeds w.h.p., after interaction t' all agents are in phases $x \geq_m i$ w.h.p. □

Now we are ready to prove the main theorem of this section

Theorem 8. *Assume all agents start executing the phase clock protocol from phase* 0 *when at least one but not more than $n^{1-\varepsilon}$ leaders are available. Assume $\varepsilon, \eta, d > 0$ are fixed and n is large enough. There exists a constant m, s.t., the finite-state phase clock with parameter m completes n^{η} rounds, and the following conditions are satisfied w.h.p. (w.p. $1 - n^{-\eta}$).*

(1) At any time phases of all agents belong to some $m/5$-sector.
(2) In each round the parallel time separating progress of the population to $m/5$-sector $sec_{m/5}(m/5)$ and the beginning of the next round is at least $d \log n$.
(3) The parallel time of any round *is $O(\log n)$.*

Proof. By Lemma 7 there exists K, s.t., for $m = 10K$ the thesis of this lemma holds also for $d > 0$ from the thesis of this theorem. We consider ten $(K + 1)$-sectors

$$sec_{K+1}(0), sec_{K+1}(K), sec_{K+1}(2K), \ldots, sec_{K+1}(9K)$$

of $\mathbb{Z}_m$. By Lemma 7 there is m, s.t., if the whole population belongs to $sec_{K+1}(iK)$ it will need at least $dn \log n$ interactions w.h.p. to make progress to $sec_{K+1}(iK +_{10} K)$. Note that in the meantime the population belongs to a longer sector $sec_{2K}(iK + K - 1) = sec_{K+1}(iK) \cup sec_{K+1}(iK +_{10} K) \setminus \{iK +_{10} K\}$ and $m/5 = 2K$. This guarantees that at any time the population belongs to some $m/5$-sector, and the parallel time between progresses to $sec_{m/5}(m/5)$ and to $sec_{m/5}(0)$ is at least $d \log n$. Since the one-way epidemic operates in $O(n \log n)$ interactions w.h.p. each agent increments its phase during $O(n \log n)$ interactions. Thus the total number of interactions in each round is $O(n \log n)$ w.h.p. □

In conclusion, we formulate two useful facts related to phase clocks. Fact 1 states that if some leaders become followers during the phase clock protocol, then the phase clock can only slow down, but the upper bound on the number of interactions remains $O(n \log n)$. Fact 2 states that any unsuccessful interactions can only slow down the phase clock.

Fact 1. *The reduction of leaders during execution of the phase clock protocol can only slow down phase progression on agents' phase clocks. And if at least one agent remains as leader the number of interactions in each round is still $O(n \log n)$ w.h.p.*

Fact 2. *If some interactions of the phase clock for a period of $O(n \log n)$ interactions are faulty, i.e., they do not contribute to phase progression, the resulting final phases on the clocks of all participating agents are not greater than in the protocol without faults.*

4 FORMING A JUNTA

In this section we describe `Forming_junta` protocol. The purpose of this protocol is to rapidly elect from n agents a junta of $O(\sqrt{n \log n})$ leaders assuming each agent utilises $O(\log \log n)$ states. This junta of leaders will be used to support phase clocks and eventual election of a unique leader.

The states of agents are represented as pairs (l, a) where $a \in \{0, 1\}$. The value l is a non-negative integer which we refer to as a *level*. During the execution of the protocol agents with $a = 0$ do not update their states. However, any agent v with value $a = 1$ increments its level l by 1 or changes its value a to 0 during all interactions v participates in. The protocol stabilises when all agents conclude with $a = 0$. The transition function is defined, s.t., on the conclusion of this protocol there are $O(\sqrt{n \log n})$ agents equipped with the highest computed value l w.h.p. These agents form the desired *junta of leaders*.

All agents start in the same state $(l, a) = (0, 1)$. As agents in states $(l, 0)$ do not get updated, we only need to specify how to update agents in states $(l, 1)$ during pairwise interactions. The transition function at level $l = 0$ differs from levels $l > 0$. When an agent in state $(0, 1)$ interacts with any agent in state $(0, 1)$, the final state of the initiator is $(1, 1)$ and $(0, 0)$ of the responder, i.e.,

$$((0, 1), (0, 1)) \longrightarrow ((1, 1), (0, 0)).$$

When an agent v in state $(0, 1)$ interacts with any agent in state (l, a), for levels $l > 0$, or with an agent in state $(0, 0)$, the resulting state of v is $(0, 0)$. If for any $l > 0$ an agent v in state $(l, 1)$, participates in an interaction, its state changes only if v acts as the responder. If the initiator is in state (l', a) such that $l \leq l'$, the responder's state becomes $(l + 1, 1)$. If the initiator is in state (l', a) such that $l > l'$, the responder's state becomes $(l, 0)$.

Let B_l be the number of agents which reach level l during the execution of `Forming_junta`. The value of B_l depends on the execution thread of the protocol. We first prove an upper bound on B_1.

Lemma 9. *For any integer $n > 1$ we have $1 \leq B_1 \leq n/2$.*

Proof. During interactions in which both agents are in state $(0, 1)$ exactly half of the participating agents increase their level l to 1. The remaining half end up in state $(0, 0)$ which becomes their final state. During any other interaction in which an agent v in state $(0, 1)$ participates, v changes its state to $(0, 0)$. So at least half of the agents end up in state $(0, 0)$. Finally, since the first interaction of the protocol is between two agents in states $(0, 1)$, at least one agent results in a state with $l > 0$. □

Due to the definition of B_l the following holds $B_1 \geq B_2 \geq B_3 \geq B_4 \geq \ldots$. We show that this sequence is finite, and w.h.p. the last nonzero element $B_L = O(\sqrt{n \log n})$, where $L = O(\log \log n)$. We obtain this by limiting values of B_l, for all $1 < l \leq L$, see below.

Lemma 10. *Assume $n^{-1/3} \leq A < 1$ and $B_l \leq A \cdot n$, then $B_{l+1} \leq \frac{11}{10} A^2 \cdot n$ w.p. $1 - e^{-n^{1/3}/300}$.*

Proof. An agent v contributing to the value of B_l results in state $(l, 1)$ as soon as it reaches level l during the relevant interaction t_v. Consider the first interaction ι_v succeeding t_v in which v acts as the responder. During this interaction the initiator is on level $l' \geq l$ with probability $p(\iota_v) \leq B_l/n \leq A$. Thus v moves to level $l + 1$ with probability at most A as otherwise the responder would end up in state $(l, 0)$ and would not contribute to B_{l+1}. Consider now the sequence of B_l interactions ι_v for all agents v contributing to B_l. We can attribute to these interactions a binary 0-1 sequence σ of length B_l, s.t., if during interaction ι_v agent v ends up in state $(l + 1, 1)$, the respective entry in σ becomes 1. Otherwise, this entry becomes 1 with probability $(A - p(\iota_v))/(1 - p(\iota_v))$ and 0 with probability $(1 - A)/(1 - p(\iota_v))$. Thus the probability of each entry being 1 is independently equal to A and the number of 1s in σ is at least B_{l+1}. The expected number of these 1s is $A \cdot B_l \leq A^2 n$. By the Chernoff bound $B_{l+1} > \frac{11}{10} A^2 \cdot n$ with probability at most $e^{-A^2 n/300} < e^{-n^{1/3}/300}$. □

Lemma 11. *If $B_l \leq n^{1/3}$ we obtain $B_{l+1} = 0$ w.p. $1 - n^{-1/3}$.*

Proof. If $B_l \leq n^{1/3}$, the probability that any agent on level l gets to level $l + 1$ is at most $n^{-2/3}$. Thus by the Union bound the probability of some agent getting to level $l + 1$ is at most $n^{-1/3}$. □

Lemma 12. *There is a constant $c > 0$, s.t., if $B_l \geq c\sqrt{n \log n}$, then the event $B_{l+1} > 0$ occurs w.h.p.*

Proof. Consider a group of $c\sqrt{n \log n}/2$ agents which move to level l when this level is already populated by $c\sqrt{n \log n}/2$ other agents. Any agent in this group moves to level $l+1$ with probability at least $c\sqrt{\log n/4n}$. Since all these agents advance to level $l+1$ independently, the probability that $B_{l+1} = 0$ is at most

$$\left(1 - c\sqrt{\log n/4n}\right)^{c\sqrt{n \log n}/2} < e^{-c^2 \log n/4} < n^{-c^2/4}.$$

The latter value is smaller than $n^{-\eta}$, for c large enough. □

Theorem 13. *In protocol* `Forming_junta` *the largest level L for which $B_L > 0$ satisfies $L = \log \log n + c$ for some constant c and $B_L = O(\sqrt{n \log n})$ w.h.p.*

Proof. By Lemma 9 we have $B_1 \leq n/2$. By Lemma 10 we conclude $B_2 \leq \frac{11}{10} \cdot \frac{n}{4}$ w.p. $1 - e^{-n^{1/3}/300}$. Furthermore, $B_3 \leq (\frac{11}{10})^3 \cdot \frac{n}{2^4}$ w.p. $1 - 2e^{-n^{1/3}/300}$. And in general $B_l \leq (\frac{11}{10})^{2^l - 1} \cdot n/2^{2^l}$ w.p. $1 - le^{-n^{1/3}/300}$. Thus for $L' = \log \log n + 1$ we get $B_{L'} \leq n^{1/3}$ w.h.p. Further, by Lemma 11 the value of $B_{L''}$ is 0 w.h.p., where $L'' = L' + c$, for some constant c. We also have $L < L''$. By Lemma 12 on the last level L for which $B_L > 0$ we have $B_L = O(\sqrt{n \log n})$ w.h.p. Thus the thesis of the theorem follows. □

The following lemma bounds from above the running time of protocol `Forming_junta`.

Lemma 14. *The protocol* `Forming_junta` *stabilises in $O(n \log n)$ interactions w.h.p.*

Proof. Recall from Lemma 9 that $B_1 \leq n/2$ and the number of agents with the final state $(0, 0)$ is at least $n/2$. Each agent in this group ends up in this state during its first interaction. Since every agent interacts at least once during the first $O(n \log n)$ interactions of the protocol w.h.p., all agents resulting in state $(0, 0)$ do so during this time w.h.p. Indeed, one can show that an agent does not experience any interactions during the first $cn \ln n$ interactions, for a constant c, with probability

$$\left(1 - \frac{2}{n}\right)^{cn \ln n} \leq n^{-2c}.$$

Thus by the Union bound there exists a positive constant c, s.t., after $cn \ln n$ interactions each agent experiences its first interaction w.p. $1 - n^{-\eta-1}$. Any agent that interacts as the responder with an agent in state $(0, 0)$ sets its value a to 0 which concludes the transition process. And after at least $n/2$ agents are in state $(0, 0)$, the probability that the current interaction is of this type is at least $\frac{1}{2n}$. Thus the probability that a given agent does not have $a = 0$ after $c'n \ln n$ iterations is

$$\left(1 - \frac{1}{2n}\right)^{c'n \ln n} \leq n^{-c'/2},$$

and for the constant c' big enough $n^{-c'/2} < n^{-\eta-1}$. Thus the number of interactions needed to obtain $a = 0$ in all agents is $O(n \log n)$ w.h.p. □

Finally, we prove a corollary stating that "spoiling" (for the definition check below) protocol `Forming_junta` does not affect validity of statements from Theorem 13 and Lemma 14. Using the notion of a spoiled protocol instead of the flawless one is needed to bound the total number of states in the leader election protocol to $O(\log \log n)$. Let *spoiled* `Forming_junta` protocol be any protocol obtained by changing spontaneously some states from (l, a) to $(0, 0)$, where l is not the highest level reached so far in the population. For every level l, we denote by B_l^* the total number of agents that reach this level in the spoiled protocol, and the highest level for which $B_l^* > 0$ we denote by L^*. Observe that in the spoiled protocol no agent reaching level L^* arrives in state $(0, 0)$.

Corollary 1. *Level L^* satisfies the condition $L^* = O(\log\log n)$ and $B^*_{L^*} = O(\sqrt{n\log n})$ w.h.p. Moreover, spoiled* `Forming_junta` *protocol stabilises after $O(n\log n)$ interactions w.h.p.*

Proof. The numbers B^*_l of agents reaching level l in the spoiled protocol are not larger than the numbers B_l from the flawless protocol respectively, thus $L^* = O(\log\log n)$. Moreover, the reasoning used in the proof of Lemma 12 bounds also $B^*_{L^*}$ by $O(\sqrt{n\log n})$ w.h.p. And in turn the running time of the spoiled protocol is not larger than the flawless one. □

5 LEADER ELECTION

In this section we describe how to combine the protocols described in the two previous sections to obtain a new fast leader election population protocol. This new leader election protocol operates in (parallel) time $\Theta(\log^2 n)$ on populations of agents equipped with $\Theta(\log\log n)$ states.

The new leader election protocol assumes that at the beginning there is a non-empty subset (possibly the whole population) of agents which are the leader candidates, and this subset is gradually reduced to a singleton. The protocol consists of $\Theta(\log n)$ executions of a *reduction procedure* formed of $\Theta(n\log n)$ interactions controlled by the ordinary mode of the phase clock. In this reduction procedure every remaining candidate picks independently at random either a 0-bit or a 1-bit by tossing a fair coin. In real terms, the coin tossing process relies on the initiator versus responder selection performed by the random scheduler. The candidates which pick a 1-bit broadcast message "1" to all other agents. And when a candidate which picked a 0-bit receives message "1", it stops being a candidate for the leader indefinitely.

Theorem 15. *The proposed leader election scheme selects a unique leader w.h.p. via $\Theta(\log n)$ executions of the reduction procedure.*

Proof. If the number of candidates is at least 2, the probability that the relevant execution of the reduction procedure deselects a half or more of the candidates is at least $3/8$. And this happens when at least half of the candidates draw 0-bits, but not all of them. Consider a series of $c\log n$ consecutive executions of the reduction procedure and form a binary 0-1 sequence σ of length $c\log n$, in which the entries correspond to the outcome of these executions. If prior to an execution only one candidate remains, the entry in σ is chosen uniformly at random by drawing a 1-bit with probability $5/8$, and a 0-bit with the remaining probability. If there are more candidates drawing 1-bits than 0-bits, or all candidates draw 0-bits, the relevant entry becomes 1. If there are two candidates or more remaining and at least half but not all of them draw 0-bits, an extra random selection is triggered, where the probability of choosing 0 for σ is exactly $3/8$. The worst case applies to exactly three remaining candidates when the probability of deselecting at least half of the candidates is exactly $3/8$.

Note, that if sequence σ has at least $\log n$ 0s, then exactly one leader remains. By the Chernoff bound the probability that σ contains less than $\log n$ 0s is smaller than $e^{-(1-8/3c)^2 3c\log n/16}$, and in turn smaller than $n^{-\eta}$, for a constant c large enough. □

The multi-broadcast required by this scheme can be implemented via the one-way epidemic described in Section 2. This process is controlled by the phase clock run in the ordinary mode, using a constant number of states. The total number of $\Theta(\log n)$ repetitions of this process is counted by the phase clock run in the external mode. This is conditioned by forming a junta of at most $n^{1-\varepsilon}$ leaders. In Section 4 we described the relevant `Forming_junta` protocol which reduces the number of leaders to $O(\sqrt{n\log n})$ and which utilises $\Theta(\log\log n)$ states at each agent. Our leader election protocol starts with a single execution of protocol `Forming_junta` which is followed by the leader reduction mechanism allowing to deplete the original junta to a single leader.

All agents enter the leader election protocol in the same state. The current state of an agent is represented by a vector $(l, a, b, x, y, \mathbf{z})$ where all entries, with the exception of l, have constant size descriptions. A non-negative integer l refers to the number of levels bounded by $O(\log\log n)$. Other positions contain small integer constants $a \in \{0, 1\}$, $b \in \{\texttt{leader}, \texttt{follower}\}$, which refer to the leadership status, and $x, y \in \mathbb{Z}_m$ are utilised in phase clocks in the ordinary and the external modes respectively. The remaining state overheads imposed by our protocol are encoded in $\mathbf{z} = (z_0, z_1, z_2)$ which is limited to a constant number of values used to steer the protocol of leader elimination. Here $z_0 \in \{\texttt{draw}, \texttt{spread}\}$ stands for drawing a 0-bit or a 1-bit by a leader, and spreading value 1 if a 1-bit was drawn. Moreover, $z_1 \in \{\emptyset, 0, 1\}$ is the value recently drawn by an agent being a leader, and value $z_2 \in \{0, 1\}$ indicates whether the agent is aware ($z_2 = 1$) of any leader with $z_1 = 1$. Since l assumes $O(\log\log n)$ values and all other variables can have only a constant number of values, the total number of states in the protocol is $O(\log\log n)$. This number of states can be estimated more accurately since l can be upperbounded by $\log\log n + c$, where c is a constant depending on η. Indeed a more careful estimation based on Theorem 13 gives a bound $B_l \leq n^{1/3}$ for $l = \log\log n + 1$ w.h.p. And then we get $B_{l+c} = 0$ w.p. $1 - n^{-\eta}$ for some constant c depending on η. Assuming all possible valuations of variables $a, b, x, y, \mathbf{z}$ we get the total number of states bounded by $48m^2(\log\log n + c)$ for m, c depending on η.

Protocol 1 `Leader_election`(A:$(l, a, b, x, y, \mathbf{z})$)

execute `Forming_junta` ▷ concludes with $a = 0$
$(x, y, z_0, z_1, z_2) \longleftarrow (0, 0, \texttt{draw}, \emptyset, 0)$ ▷ set entries
loop through all interactions ▷ main loop
  $\{A$ meets an agent in state $(l', 0, b', x', y', \mathbf{z}')\}$
  if $(l < l')$ **then**
    $(l, b, x, y, z_0, z_1, z_2) \longleftarrow (l', \texttt{follower}, 0, 0, \texttt{draw}, \emptyset, 0)$ ▷ updates clock level
  if (A is responder) **then**
    perform relevant operations of phase clocks ▷ updates y only just after x passes 0
  if (phase x just passed through 0) **then**
    if $(z_0 = \texttt{draw})$ **then**
      $(z_0, z_2) \longleftarrow (\texttt{spread}, 0)$ ▷ initiates spreading mode
    else $(z_0, z_1) \longleftarrow (\texttt{draw}, \emptyset)$ ▷ initiates drawing mode
  if $(z_0 = \texttt{draw}\ \&\ z_1 = \emptyset\ \&\ b = \texttt{leader}\ \&\ b' = \texttt{follower})$ **then**
    if (A is responder) **then** ▷ draws 0 or 1 for a leader A
      $z_1 \longleftarrow 0$
    else $z_1 \longleftarrow 1$
  if $(z_0 = \texttt{spread}\ \&\ A$ is responder$)$ **then**
    $z_2 \longleftarrow \max\{z_2, z_1', z_2'\}$ ▷ spreads $z_2 = 1$ if some leader draws 1
  if $(z_0 = \texttt{spread}\ \&\ l = \texttt{leader}\ \&\ z_1 = 0\ \&\ z_2 = 1)$ **then**
    $l \longleftarrow \texttt{follower}$ ▷ leader moves to the `follower` state after loosing a round
end loop

Spoiled `Forming_junta` *protocol.* At the start of the leader election protocol all agents initiate their states to $(l, a, b, x, y) = (0, 1, \texttt{leader}, 0, 0)$, with other variables initiated arbitrarily. For as long as $a = 1$, the value of b remains `leader`. And as soon as in spoiled `Forming_junta` protocol, in any agent pair (l, a) becomes $(0, 0)$, variable b is set to `follower` and this change is permanent. This releases variable l which can be used during subsequent leader elimination to denote the relevant levels of phase clocks. Nevertheless, in the remaining stages of `Forming_junta` if an agent has

$b = \texttt{follower}$, its pair (l, a) is interpreted as $(0, 0)$. According to Corollary 1 each agent concludes spoiled `Forming_junta` protocol within $\Theta(n \log n)$ interactions w.h.p.

Phase clocks on different levels. Once variable a of an agent becomes 0, the agent starts its phase clocks on level l as the leader with phases $x = y = 0$. When two interacting agents have their phase clocks operating at two different levels $l < l'$ respectively, the state of the one at level l is rewritten $(l, b, x, y, z_0, z_1, z_2) \leftarrow (l', \texttt{follower}, 0, 0, \texttt{draw}, \emptyset, 0)$. In other words, this agent becomes a follower, adopts level l', and resets its both phase clocks to 0s. Since this agent is now a follower, its pair (l, a) is interpreted as $(0, 0)$ w.r.t. the spoiled `Forming_junta`protocol. The level of the phase clock can be incremented this way many times until it attains the maximum level L^* ever reached by the population. And eventually all agents collectively run both phase clock protocols on level L^*. All agents which advanced to level L^* during the spoiled `Forming_junta` protocol become the leaders of the phase clocks, and the others act as followers. Similarly, while for some time leader election is also executed at levels lower than L^*, this process is gradually phased out. I.e., eventually all agents collectively run the leader election protocol on level L^*, and only this process has a direct influence on the final outcome.

We run the phase clock in the ordinary and in the external mode simultaneously to implement the two loops described in the beginning of Section 3. In the ordinary mode the phase clock is driven by all interactions in which the responder has variable $a = 0$. If the responder interacts with the relevant initiator on a higher level it advances its clock level as described above. If the responder has the same clock level as the initiator, they both perform one interaction in the ordinary as well as the external mode when this interaction is meaningful. If the responder interacts with the initiator on a lower level or having $a = 1$, then this interaction is void in both modes. The phase clock operates in the ordinary mode until it passes through 0 for the first time. And it counts for each agent the first $\Theta(n \log n)$ interactions by Fact 2.

Random coin tosses. Each of the remaining leaders v picks randomly 0 or 1 during the first interaction with a non-leader after the phase of v (in the ordinary mode of the clock) passes through 0. If the non-leader is the initiator, then v chooses 1, otherwise v picks 0. This gives a truly random value to each leader, and since there are $O(\sqrt{n \log n})$ leaders, this process is completed w.h.p. during $O(n \log n)$ interactions.

Leader candidate elimination on the highest level. After choosing value 0 or 1 at random, the leaders multi-broadcast 1s to the whole population via the one-way epidemic. The required $\Theta(n \log n)$ interactions are counted with the help of the phase clock in the ordinary mode. In order to obtain a unique leader w.h.p., this process is iterated $\Theta(\log n)$ times by the external loop and controlled by the phase clock in the external mode. Since in the algorithm we need only one round of the external phase clock, we can replace operation $\max_m$ of this clock by the standard integer operation max. The protocol concludes at each agent when its external clock attains phase $m - 1$. The following theorem holds.

THEOREM 16. *The protocol described above utilises $O(\log \log n)$ states per agent and finds a unique leader in parallel time $O(\log^2 n)$ w.h.p.*

Finally we formulate a corresponding Las Vegas variant of our algorithm to more accurately match the existing lower bound $\Omega(\log \log n)$ on the number of states in fast leader election [7]. This version of the Las Vegas algorithm concludes in a stable configuration of states in which there is only one leader in the expected parallel time $O(\log^2 n)$.

THEOREM 17. *There exists a Las Vegas type leader election protocol which utilises $O(\log \log n)$ states per agent and always elects a single leader in the expected parallel time $O(\log^2 n)$.*

Proof. The state vector in this protocol remains unchanged with the exception of variable b which can now attain three values {`leader`, `passive`, `follower`}. We modify the transition functions of both phase clocks to preserve the desired properties and performance guarantees w.h.p., and to deal with clocks' desynchronisation. The external clock is based on the same transitions, however, the periodic $\max_m$ function is replaced by the standard maximum operation max. Moreover, when eventually the external clock of some agent is in phase $m-2$ or $m-1$, the agent no longer updates the ordinary clock. In addition, the relevant external clock phase is propagated to all other agents via the one-way epidemic.

As before, during the elimination process all leader candidates must belong to the junta elected by `Forming_junta` protocol. However, the leader candidates which are chosen for elimination (they drew 0s while someone drew 1 before their external clocks enter phase $m-2$) change their state to `passive` rather than to `follower`. During further leader elimination interactions all `passive` leaders behave as they were in `follower` state. However, the holders of `passive` state retain some functionality of leader candidates. In particular, these agents form a reserve pool of leader candidates which can be used in the unlikely event of elimination of all agents in `leader` state. They also contribute to updates of both phase clocks until their phases reach $m-2$. Thus the transition from phase $m-2$ to $m-1$ in the external clock is triggered only by the remaining holders of `leader` state. Once a holder of `passive` state enters phase $m-1$ (via the one-way epidemic) it transitions to `follower` state. This guarantees that if the leader election protocol elects a single leader among the remaining `leader` candidates, all other agents result in `follower` state. And for m large enough, this happens in time $O(\log^2 n)$ w.p. $1-n^{-3}$.

There exist some unlikely scenarios in which a single leader is not elected within the time limit $O(\log^2 n)$. These include the cases where (1) the election process is correct but slower, (2) there are at least two `leader` candidates on the conclusion of the election process, and (3) the ordinary clock becomes desynchronised. In order to accommodate for these deficiencies we propose two extra mechanisms which guarantee successful stabilisation with a single leader elected in the expected parallel time $O(\log^2 n)$. The *first mechanism* guarantees a successful stabilisation in cases (1) and (2), however to reach a successful stabilisation in case (3) we also need the *second mechanism*.

The *first mechanism* refers to the slow leader election protocol which governs interactions between leader candidates as follows. If both agents are in `leader` state, then the one with an earlier external clock phase transitions to `follower` state. If one of them is in `leader` and the other in `passive` state then the latter is changed to `follower`. In all other cases including ties, the responder transitions to `follower` state. This assures the election of a single leader in the unlikely cases in the expected parallel time $O(n \log n)$, as the winner may have to interact with all agents.

The *second mechanism* is triggered if the ordinary clock phases of interacting agents differ by more than $m/5$ phases. This may happen due to some unlikely events such as selection of too large junta or isolation of an agent for a prolonged time. When two agents with desynchronised ordinary phase clocks interact, they set their external clock phases to $m-2$. This external clock phase is then propagated to all other agents in the expected parallel time $O(\log n)$. If all leader candidates end up in `passive` state, all agents conclude in phase $m-2$. If at least one leader candidate remains in the `leader` state, it increments the external clock phase to value $m-1$ which is then propagated to all agents.

The correctness of this protocol can be proved as follows. If some agents in `leader` state reach the external clock phase $m-2$, they advance to phase $m-1$ and elect the unique leader amongst themselves. Otherwise, agents in `passive` state use the slow protocol to finally elect the single leader. Thus the protocol always concludes with a single leader.

It remains to show that the protocol always stabilises in the expected time $O(\log^2 n)$. We define a *progressive* sequence of interactions which is a sequence of subsequent interactions generated by the random scheduler and assures a measurable progress of the ordinary phase clock. Formally this is a sequence of interactions generated by the random scheduler. This sequence assures a success of the one-way epidemic which starts from any agent in the population and it is followed by an interaction in which some leader acts as the responder. A configuration is *synchronised* if all agents with the external clock phase smaller than $m-2$ have their ordinary clock phase inside a $m/5$-sector. Any other configuration is deemed *desynchronised.* We will use the following fact about phase clocks, which follows from Theorem 8

Fact 3. *Let m be the number of phases on both phases clocks. There exists $d > 0$, s.t., when the ordinary phase clock remains synchronised for $d\log n$ rounds, the external clock phase gets to $m-2$ w.p. $1-n^{-3}$.*

This fact follows from the observation that $d\log n$ rounds of the ordinary phase clock, for d large enough, generate a sequence of meaningful interactions which guarantee progress to phase $m-2$ of the external clock w.p. $1-n^{-3}$. Without a proof we formulate next two easy facts.

Fact 4. *There exists a constant $c > 0$ for which any $cn\log n$ subsequent interactions form a progressive sequence w.p. $1-n^{-3}$.*

Fact 5. *For as long as the configuration is synchronised, during any progressive sequence of interactions the maximum phase of the ordinary clock is incremented at least once in the cyclic order.*

The `Forming_junta` subprotocol terminates in the expected time $O(\log n)$. Let's denote by $M(n)$ the maximum expected parallel stabilisation time of the protocol over all starting configurations reachable on the conclusion of `Forming_junta`.

Claim 1. *We have $M(n) = O(n^2\log^2 n)$.*

A configuration is *strongly synchronised* if it is synchronised and within parallel time $cdm\log^2 n$ it transitions to a desynchronised configuration with probability smaller than $1/2$. We have two cases. In the first case the average time $M(n)$ is guaranteed for some strongly synchronised configuration. In such a case w.p. $1-dmn^{-3}\log n$ the sequence of the first $cdmn\log^2 n$ interactions generated by the random scheduler is a series of $dm\log n$ progressive sequences. Thus the conditional probability that this is a series of $dm\log n$ progressive sequences, assuming the population stays in a synchronised configuration within parallel time $cdm\log^2 n$, is at least $1-n^{-2}$. Such a sequence guarantees that the ordinary phase clock executes at least $d\log n$ rounds unless the external phase clock reaches phase $m-2$. Also, during $d\log n$ rounds of the ordinary phase clock, the external clock reaches $m-2$, w.p. $1-n^{-3}$. And in turn the external clock phase reaches $m-2$ w.p. $1/2-n^{-1}$ in this case. After an extra parallel time $O(\log n)$ the external phases of all agents stabilise in either $m-2$ or $m-1$. This final phase depends on whether on the conclusion there are agents in `leader` state. This concludes phase clocks transitions. The final stabilisation is done via the slow leader election protocol in parallel time $O(n\log n)$. Thus

$$M(n) \le (1/2-n^{-1})(O(\log^2 n)+O(n\log n)) + (1/2+n^{-1})(O(\log^2 n)+M(n))$$

and $M(n) = O(n\log n)$.

The remaining case refers to time $M(n)$ obtained from an initial configuration which is not strongly sychronised. In such a case with probability at least $1/2$ and within at most $cdm\log^2 n$ interactions there is a configuration in which two agents operate in the ordinary clock phases at distance larger than $m/5$. These agents interact with probability larger than $2n^{-2}$. And if this happens they both adopt phase $m-2$ on their clocks and propagate this phase to all other agents in parallel time $O(\log n)$. Moreover, if there are some agents in `leader` state their external clocks reach phase $m-1$ and they propagate this phase to all other agents in parallel time $O(\log n)$. Thus on the conclusion of this process all agents are either in phase $m-2$ or $m-1$ no later than in parallel time $O(\log n)$. This ends the phase clocks operations. The final stabilisation by the election of a single leader utilising the slow leader election protocol is done in parallel time $O(n \log n)$. Thus

$$M(n) \leq 2n^{-2}(O(\log^2 n) + O(n \log n)) + (1 - 2n^{-2})(O(\log^2 n) + M(n))$$

or $2n^{-2}M(n) \leq n^{-2}O(n \log n) + O(\log^2 n)$, so $M(n) = O(n^2 \log^2 n)$. This ends the proof of the claim.

Now by the claim and since the algorithm stabilises in time $O(\log^2 n)$ w.p. $1 - n^{-3}$, the expected stabilisation time is

$$O(\log^2 n) + n^{-3}M(n) = O(\log^2 n).$$

□

6 CONCLUSION

In this paper we studied fast and space efficient leader election in population protocols. Our new protocol stabilises in parallel time $O(\log^2 n)$ when each agent is equipped with $O(\log \log n)$ states. This double logarithmic space utilisation matches asymptotically the lower bound $\frac{1}{2} \log \log n$ on the minimal number of states required by agents in any leader election algorithm with the running time $o\left(\frac{n}{\text{polylog } n}\right)$, see [7].

Open problems. There are several open problems left for further consideration. The most immediate one refers to the question whether it is possible to select a unique leader in time $o(\log^2 n)$ utilizing $O(\log \log n)$ states at each agent. Only partial answers to this question are given in [41], where one can find a leader election algorithm operating in the expected parallel time $O(\log n \log \log n)$, and in a very recent work of Berenbrink *et al.* [20], in the optimal expected parallel time $O(\log n)$. However, we still do not know whether a substantial improvement in comparison to $O(\log^2 n)$-time leader election is possible w.h.p. For a more comprehensive list of open problems we refer the reader to the two recent surveys on advances in population protocols [10, 34].

Acknowledgements. We would like to thank Dave Doty, Rati Gelashvili and Dominik Kaaser and several anonymous reviewers for their valuable comments on subsequent versions of this paper. Special thanks go to Przemek Uznański whose suggestions helped us to clarify several arguments and led to a simple Las Vegas type algorithm. This work is sponsored in part by the University of Liverpool initiative Networks Sciences and Technologies (NeST) and by the Polish National Science Centre grant DEC-2012/06/M/ST6/00459.

REFERENCES

[1] A. Andoni and I.P. Razenshteyn, Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing. Proc. Symposium on Computational Geometry, 2016, 9:1–9:11.

[2] D. Angluin, Local and global properties in networks of processors, Proc. *12th Annual ACM Symposium on Theory of Computing*, STOC 1980, 82–93.

[3] D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta, Computation in networks of passively mobile finite-state sensors. Proc. *23rd Annual ACM Symposium on Principles of Distributed Computing*, PODC 2004, 290–299.

[4] D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta, Computation in networks of passively mobile finite-state sensors. Distributed Computing, 18(4), 2006, 235–253.

[5] D. Angluin, J. Aspnes, D. Eisenstat. Fast computation by population protocols with a leader, Distributed Computing 21(3), 2008, 183–199.

[6] D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority, Distributed Computing, 21(2), 2008, 87–102.

[7] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili and R.L. Rivest, Time-Space Trade-offs in Population Protocols, Proc. *28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2017, 2560–2579.

[8] D. Alistarh, J. Aspnes, and R. Gelashvili, Space-Optimal Majority in Population Protocols, Proc. *29th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2018, 2221–2239.

[9] D. Alistarh and R. Gelashvili, Polylogarithmic-time leader election in population protocols, Proc. *42nd International Colloquium on Automata, Languages, and Programming*, ICALP 2015, 479–491.

[10] D. Alistarh and R. Gelashvili, Recent Algorithmic Advances in Population Protocols ACM SIGACT News 49(3), 2018, 63–73.

[11] A. Arora, S. Dolev, and M.G. Gouda, Maintaining digital clocks in step, Proc. *5th International Workshop on Distributed Algorithms*, (LNCS 579) 1991, 71–79.

[12] H. Attiya and M. Snir, Better Computing on the Anonymous Ring, Journal of Algorithms 12, 1991, 204–238.

[13] H. Attiya, M. Snir, and M. Warmuth, Computing on an Anonymous Ring, Journal of the ACM 35, 1988, 845–875.

[14] H. Attiya and J. Welch, Distributed Computing: Fundamentals, Simulations, and Advanced Topics, 2nd Edition, Willey, April 2004.

[15] L. Becchetti, A.E.F. Clementi, E. Natale, F. Pasquale, and R. Silvestri, Plurality consensus in the gossip model, Proc. *26th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2015,371–390.

[16] S. Ben Nun, T. Kopelowitz, M. Kraus, E. Porat, An $O(\log^{3/2} n)$ Parallel Time Population Protocol for Majority with $O(\log n)$ States. Proc. *39th ACM Symp. on Principles of Distributed Computing*, PODC 2020: 191–199.

[17] P. Berenbrink, R. Elsässer, T. Friedetzky, D. Kaaser, P. Kling, and T. Radzik, A Population Protocol for Exact Majority with $O(\log^{5/3} n)$ Stabilization Time and $\Theta(\log n)$ States. *32nd International Symposium on Distributed Computing*, DISC 2018, 10:1–10:18.

[18] P. Berenbrink, T. Friedetzky, G. Giakkoupis, and P. Kling. Efficient plurality consensus, or: The benefits of cleaning up from time to time. Proc. *43rd International Colloquium on Automata, Languages, and Programming*, ICALP 2016, 1–14.

[19] P. Berenbrink, D. Kaaser, P. Kling, and L. Otterbach, Simple and efficient leader election. Proc. *Symposium on Simplicity in Algorithms*, SOSA 2018, 9:1–9:11.

[20] P. Berenbrink, G. Giakkoupis, P. Kling, Optimal Time and Space Leader Election in Population Protocols Proc. *52nd Annual ACM Symposium on Theory of Computing*, STOC 2020, 119–129.

[21] A. Bilke, C. Cooper, R. Elsässer, and T. Radzik, Brief announcement: Population protocols for leader election and exact majority with O(log2 n) states and O(log2 n) convergence time. Proc. *36th ACM Symp. on Principles of Distributed Computing*, PODC 2017, 451–453.

[22] P. Boldi, S. Shammah, S. Vigna, B. Codenotti, P. Gemmell, and J. Simon, Symmetry Breaking in Anonymous Networks: Characterizations, Proc. *4th Israel Symposium on Theory of Computing and Systems*, ISTCS 1996, 16–26.

[23] P. Boldi and S. Vigna, Computing Anonymously with Arbitrary Knowledge, Proc. *18th ACM Symp. on Principles of Distributed Computing*, PODC 1999, 181–188.

[24] J.E. Burns, A Formal Model for Message Passing Systems, Tech. Report TR-91, Computer Science Department, Indiana University, Bloomington, September 1980.

[25] L. Cardelli, M. Kwiatkowska, and L. Laurenti, Programming Discrete Distributions with Chemical Reaction Networks. Proc. *22nd International Conference DNA Computing and Molecular Programming*, DNA 2016, 35–51.

[26] I. Chatzigiannakis, O. Michail, S. Nikolaou, A. Pavlogiannis, and P.G. Spirakis, Passively mobile communicating machines that use restricted space. Proc. *7th ACM ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing*, 2011, 6–15.

[27] H.-L. Chen, R. Cummings, D. Doty, and D. Soloveichik, Speed faults in computation by chemical reaction networks, Distributed Computing, Springer 2014, 16–30.

[28] D. Dereniowski and A. Pelc, Leader election for anonymous asynchronous agents in arbitrary networks, Distributed Computing 27, 2014, 21–38.

[29] S. Dobrev and A. Pelc, Leader Election in Rings with Nonunique Labels, Fundamenta Informaticae 59, 2004, 333–347.

[30] S. Dolev and J.L. Welch, Self-stabilizing clock synchronization in the presence of Byzantine faults, Journal of the ACM, 51(5), 2004, 780–799.

[31] D. Doty, Timing in chemical reaction networks. Proc. *25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2014, 772–784. SIAM.

[32] A. Daliot, D. Dolev, and H. Parnas, Self-stabilizing pulse synchronization inspired by biological pacemaker networks, Proc. *Self-Stabilizing Systems*, SSS 2003, 32–48.

[33] D. Doty and D. Soloveichik, Stable leader election in population protocols requires linear time, Proc. *29th International Symposium on Distributed Computing*, DISC 2015, 602–616.

[34] R. Elsässer, and T. Radzik, Recent Results in Population Protocols for Exact Majority and Leader Election, EATCS Bulletin 126, October 2018.

[35] P. Flocchini, E. Kranakis, D. Krizanc, F.L. Luccio and N. Santoro, Sorting and Election in Anonymous Asynchronous Rings, Journal of Parallel and Distributed Computing 64, 2004, 254–265.

[36] G.N. Fredrickson and N.A. Lynch, Electing a Leader in a Synchronous Ring, Journal of the ACM 34, 1987, 98–115.

[37] E. Fusco and A. Pelc, Knowledge, level of symmetry, and time of leader election, Proc. *20th Annual European Symposium on Algorithms*, ESA 2012, 479–490.

[38] E. Fusco, A. Pelc, Trade-offs between the size of advice and broadcasting time in trees, Algorithmica 60, 2011, 719–734.

[39] L. Gąsieniec, D.D. Hamilton, R. Martin, and P.G. Spirakis, The match-maker: Constant space distributed majority via random walks, Proc. *17th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, SSS 2015, 67–80.

[40] L. Gąsieniec, D. Hamilton, R. Martin1, P.G. Spirakis, and G. Stachowiak, Deterministic Population Protocols for Exact Majority and Plurality, OPODIS'16, 14:1–14:14.

[41] L. Gąsieniec, G. Stachowiak, and Przemysław Uznański, Almost logarithmic-time space optimal leader election in population protocols, `arXiv:1802.06867 [cs.DC]`

[42] M. Ghaffari and M. Parter, A polylogarithmic gossip algorithm for plurality consensus. Proc. *34th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC 2016, 117–126.

[43] Ch. Glacet, A. Miller, and A. Pelc, Time vs. Information Tradeoffs for Leader Election in Anonymous Trees, Proc. *27th annual ACM-SIAM symposium on Discrete algorithms*, SODA 2016, 600–609

[44] M.A. Haddar, A.H. Kacem, Y. Métivier, M. Mosbah, and M. Jmaiel, Electing a Leader in the Local Computation Model using Mobile Agents, Proc. *6th ACS/IEEE International Conference on Computer Systems and Applications*, AICCSA 2008, 473–480.

[45] T. Herman, Phase clocks for transient fault repairs, IEEE Transactions on Parallel and Distributed Systems 11(10), 2000, 1048–1057.

[46] D.S. Hirschberg, and J.B. Sinclair, Decentralized Extrema-Finding in Circular Configurations of Processes, Communications of the ACM 23, 1980, 627–628.

[47] G. Le Lann, Distributed Systems - Towards a Formal Approach, Proc. *IFIP Congress*, 1977, 155–160.

[48] Y. Mocquard, E. Anceaume, J. Aspnes, Y. Busnel, and B. Sericola, Counting with population protocols, Proc. 2015 *IEEE 14th International Symposium on Network Computing and Applications*, NCA 2015, 35–42.

[49] G.B. Mertzios, S.E. Nikoletseas, C. Raptopoulos, and P.G. Spirakis, Determining majority in networks with local interactions and very small local memory. Proc. *41st International Colloquium on Automata, Languages, and Programming*, ICALP 2014, 871–882.

[50] O. Michail and P.G. Spirakis, Simple and efficient local codes for distributed stable network construction, Proc. *32nd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC 2014, 76–85.

[51] G.L. Peterson, An $O(nlogn)$ Unidirectional Distributed Algorithm for the Circular Extrema Problem, ACM Transactions on Programming Languages and Systems 4, 1982, 758–762.

[52] M. Yamashita and T. Kameda, Electing a Leader when Procesor Identity Numbers are not Distinct, Proc. *3rd Workshop on Distributed Algorithms*, WDAG 1989, 303–314.

[53] M. Yamashita and T. Kameda, Computing on Anonymous Networks: Part I - Characterizing the Solvable Cases, IEEE Trans. Parallel and Distributed Systems 7, 1996, 69–89.