Fast Approximate Counting and Leader Election in Populations *

Othon Michail[1], Paul G. Spirakis[1,2], and Michail Theofilatos[1]

[1] Department of Computer Science, University of Liverpool, UK
[2] Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece
Email: Othon.Michail@liverpool.ac.uk, P.Spirakis@liverpool.ac.uk,
Michail.Theofilatos@liverpool.ac.uk

Abstract. We study the problems of leader election and population size counting for *population protocols*: networks of finite-state anonymous agents that interact randomly under a uniform random scheduler. We first give an approximate counting protocol which provides an upper bound $\hat{n}$ of the size n of the population, where $\hat{n}$ is at most n^a for some $a > 1$. This protocol assumes the existence of a unique leader in the population and stabilizes in $\Theta(\log n)$ parallel time, using constant number of states in every node, except the unique leader which is required to use $\Theta(\log n)$ states. Finally, we propose a protocol for leader election that terminates in $O(\log_m(n) \cdot \log_2 n)$ parallel time, where m is a parameter, using $O(\max\{m, \log n\})$ states. By adjusting the parameter m between a constant and n, we obtain a leader election protocol whose time and space can be smoothly traded off between $O(\log^2 n)$ to $O(\log n)$ parallel time and $O(\log n)$ to $O(n)$ states.

Keywords: population protocol, epidemic, leader election, counting, approximate counting, polylogarithmic time protocol

Best student paper award. This paper is eligible for the best student paper award as Michail Theofilatos was a full-time student at the time of submission and has made a significant contribution to the paper.

All omitted proofs are included in a clearly marked Appendix, to be read at the discretion of the Program Committee.

1 Introduction

Population protocols [AAD+06] are networks that consist of very weak computational entities (also called *nodes* or *agents*), regarding their individual capabilities. These networks have been shown that are able to construct complex shapes [MS16] and perform complex computational tasks when they work collectively. Leader Election, which is a fundamental problem in distributed computing, is the process of designating a single agent as the coordinator of some task distributed among several nodes. The nodes communicate among themselves in order to decide which of them will get into the *leader* state. *Counting* is also a fundamental problem in distributed computing, where nodes must determine the size n of the population. Finally, we call *Approximate Counting* the problem in which nodes must determine an estimation k of the population size n. Counting can be then considered as a special case of approximate counting, where $k = n$.

Many distributed tasks require the existence of a leader prior to the execution of the protocol and, furthermore, some knowledge about the system (for instance the size of the population) can also help to solve these tasks more efficiently with respect both to time and space.

* All authors were supported by the EEE/CS initiative NeST. The last author was also supported by the Leverhulme Research Centre for Functional Materials Design.

Consider the setting in which an agent is in an initial state a, the rest $n-1$ agents are in state b and the only existing transition is $(a,b) \to (a,a)$. This is the *one-way epidemic* process and it can be shown that the expected time to convergence under the uniform random scheduler is $\Theta(n \log n)$ (e.g., [AAE08]), thus *parallel time* $\Theta(\log n)$. In this work, we make an extensive use of epidemics, which means that information is being spread throughout the population, thus all nodes will obtain this information in $O(\log n)$ expected parallel time. We use this property to construct an algorithm that solves the *Leader Election* problem. In addition, by observing the rate of the epidemic spreading under the uniform random scheduler, we can extract valuable information about the population. This is the key idea of our *Approximate Counting* algorithm.

1.1 Related Work

The framework of population protocols was first introduced by Angluin et al. [AAD+06] in order to model the interactions in networks between small resource-limited mobile agents. When operating under a uniform random scheduler, population protocols are formally equivalent to a restricted version of stochastic Chemical Reaction Networks (CRNs), which model chemistry in a well-mixed solution [SCWB08]. "CRNs are widely used to describe information processing occurring in natural cellular regulatory networks, and with upcoming advances in synthetic biology, CRNs are a promising programming language for the design of artificial molecular control circuitry" [CDS14, Dot14]. Due to a formal equivalence between CRNs and population protocols, results transfer readily between the two models.

Most computability issues in the area of population protocols have now been resolved. Finite-state processes on a complete interaction network, i.e., one in which every pair of processes may interact, (and several variations) compute the *semilinear predicates* [AAER07]. Semilinearity persists up to $o(\log \log n)$ local space but not more than this [CMN+11]. If, additionally, the connections between processes can hold a state from a finite domain then the computational power dramatically increases to the commutative subclass of $\mathbf{NSPACE}(n^2)$ [MCS11]. Equally powerful variants of population protocols may also be obtained by equipping the nodes with unique identifiers, as, e.g., in [GR09]. For introductory texts to population protocols the interested reader is encouraged to consult [AR09, MCS11] and [MS18] (the latter discusses population protocols and related developments as part of a more general overview of the emerging theory of dynamic networks).

Optimal algorithms, regarding the time complexity of fundamental tasks in distributed networks, for example leader election and majority, is the key for many distributed problems. For instance, the help of a central coordinator can lead to simpler and more efficient protocols [AAE08]. There are many solutions to the problem of leader election, such as in networks with nodes having distinct labels or anonymous networks [Ang80, ASW85, AG15, GS18, FJ06].

Although the availability of an initial leader does not increase the computational power of standard population protocols (in contrast, it does in some settings where faults can occur [DLFI+17]), still it may allow faster computation. Specifically, the fastest known population protocols for semilinear predicates without a leader take as long as linear parallel time to converge ($\Theta(n)$). On the other hand, when the process is coordinated by a unique leader, it is known that any semilinear predicate can be stably computed with polylogarithmic expected convergence time ($O(\log^5 n)$) [AAE06].

For several years, the best known algorithm for leader election in population protocols was the pairwise-elimination protocol of Angluin et al. [AAD+06], in which all nodes are leaders in state l initially and the only effective transition is $(l,l) \to (l,f)$. This protocol always stabilizes to a configuration with unique leader, but this takes on average linear time. Recently, Doty and Soloveichik [DS15] proved that not only this, but any standard population protocol requires linear

time to solve leader election. This immediately led the research community to look into ways of strengthening the population protocol model in order to enable the development of sub-linear time protocols for leader election and other problems (note that Belleville, Doty, and Soloveichik [BDS17] recently showed that such linear time lower bounds hold for a larger family of problems and not just for leader election). Fortunately, in the same way that increasing the local space of agents led to a substantial increase of the class of computable predicates [CMN+11], it has started to become evident that it can also be exploited to substantially speed-up computations. Alistarh and Gelashvili [AG15] proposed the first sub-linear leader election protocol, which stabilizes in $O(\log^3 n)$ parallel time, assuming $O(\log^3 n)$ states at each agent. In another recent work, Gasieniec and Stachowiak [GS18] designed a space optimal ($O(\log \log n)$ states) leader election protocol, which stabilises in $O(\log^2 n)$ parallel time. They use the concept of phase clocks (introduced in [AAE08] for population protocols), which is a synchronization and coordination tool in distributed computing. General characterizations, including upper and lower bounds, of the trade-offs between time and space in population protocols were recently achieved in [AAE+17]. Moreover, some papers [MOKY12, DDLF+17] have studied leader election in the mediated population protocol model.

For counting, the most studied case is that of *self-stabilization*, which makes the strong adversarial assumption that arbitrary corruption of memory is possible in any agent at any time, and promises only that eventually it will stop. Thus, the protocol must be designed to work from any possible configuration of the memory of each agent. It can be shown that counting is *impossible* without having one agent (the "base station") that is protected from corruption [BCM+07]. In this scenario $\Theta(n \log n)$ time is sufficient [BBCS15] and necessary [ABBS17] for self-stabilizing counting.

In the less restrictive setting in which all nodes start from the same state (apart possibly from a unique leader and/or unique ids), not much is known. In a recent work, Michail [Mic15] proposed a terminating protocol in which a pre-elected leader equipped with two n-counters computes an approximate count between $n/2$ and n in $O(n \log n)$ parallel time with high probability. The idea is to have the leader implement two competing processes, running in parallel. The first process counts the number of nodes that have been encountered once, the second process counts the number of nodes that have been encountered twice, and the leader terminates when the second counter catches up the first. In the same paper, also a version assuming unique ids instead of a leader was given.

The task of counting has also been studied in the related context of worst-case dynamic networks [IKIW14, KLO10, MCS13, LBBC14, CFQS12].

1.2 Contribution

In this work we employ the use of simple epidemics in order to provide efficient solutions to approximate counting the size of a population of agents and also to leader election in populations. Our model is that of population protocols. Our goal for both problems is to get polylogarithmic parallel time and to also use small memory per agent. First, we show how to approximately count a population fast (with a leader) and then we show how to elect a leader (very fast) if we have a crude population estimate.

(a) We start by providing a protocol which provides an upper bound $\hat{n}$ of the size n of the population, where $\hat{n}$ is at most n^a for some $a > 1$. This protocol assumes the existence of a unique leader in the population. The runtime of the protocol until stabilization is $\Theta(\log n)$ parallel time. Each node except the unique leader uses only a constant number of states. However, the leader is required to use $\Theta(\log n)$ states.

(b) We then look into the problem of electing a leader. We assume an approximate knowledge of the size of the population (i.e., an estimate $\hat{n}$ of at most n^a, where n is the population size) and provide a protocol (parameterized by the size m of a counter for drawing local random numbers) that elects

a unique leader w.h.p. in $O(\frac{\log^2 n}{\log m})$ parallel time, with number of states $O(\max\{m, \log n\})$ per node. Therefore, by adjusting the parameter m, our protocol can improve the state of the art in leader election in population protocols. When m is a constant number, the stabilization time is $O(\log^2 n)$, using $O(\log n)$ states and when $m = n$, the stabilization time drops to $O(\log n)$, using $O(n)$ states.

2 The model

In this work, the system consists of a population V of n distributed and anonymous (i.e., do not have unique IDs) *processes*, also called *nodes* or *agents*, that are capable to perform local computations. Each of them is executing as a deterministic state machine from a finite set of states Q according to a transition function $\delta : Q \times Q \to Q \times Q$. Their interaction is based on the probabilistic (uniform random) scheduler, which picks in every discrete step a random edge from the complete graph G on n vertices. When two agents interact, they mutually access their local states, updating them according to the transition function δ. The transition function is a part of the population protocol which all nodes store and execute locally.

The time is measured as the number of steps until stabilization, divided by n (parallel time). The protocols that we propose do not enable or disable connections between nodes, in contrast with [MS16], where Michail and Spirakis considered a model where a (virtual or physical) connection between two processes can be in one of a finite number of possible states. The transition function that we present throughout this paper, follows the notation $(x, y) \to (z, w)$, which refers to the process states before *(x and y)* and after *(z and w)* the interaction, that is, the transition function maps pairs of states to pairs of states.

The Leader Election Problem. The problem of leader election in distributed computing is for each node eventually to decide that whether it is a leader or not subject to only one node decides that it is the leader. An algorithm A solves the leader election problem if eventually the states of agents are divided into *leader* and *follower*, a leader remains elected and a follower can never become a leader. In every execution, exactly one agent becomes leader and the rest determine that they are not leaders. All agents start in the same initial state q and the output is $O = \{leader, follower\}$. A randomized algorithm R solves the leader election problem if eventually only one leader remains in the system w.h.p.

The Approximate Counting Problem. We define as *Approximate Counting* the problem in which a leader must determine an estimation $\hat{n}$ of the population size, where $\frac{\hat{n}}{a} < n < \hat{n}$. We call a the estimation parameter.

3 Fast Counting with a unique leader

In this section we present our *Approximate Counting* protocol. The protocol is presented in Section 3.1. In Section 3.2 we prove the correctness of our protocol and finally, in Section 5, experiments that support our analysis can be found.

3.1 Abstract description and protocol

In this section, we construct a protocol which solves the problem of approximate counting. Our probabilistic algorithm for solving the approximate counting problem requires a unique leader who is responsible to give an estimation on the number of nodes. It uses the epidemic spreading technique

and it stabilizes in $O(\log n)$ parallel time. There is initially a unique leader l and all other nodes are in state q. The leader l stores two counters in its local memory, initially both set to 0. We use the notation $l_{(c_0,c_1)}$, where c_0 is the value of the first counter and c_1 is the value of the second one. The leader, after the first interaction starts an epidemic by turning a q node into an a node. Whenever a q node interacts with an a node, its state becomes a ($(a,q) \to (a,a)$). The first counter c_0 is being used for counting the q nodes and the second counter c_1 for the a nodes, that is, whenever the leader l interacts with a q node, the value of the counter c_0 is increased by one and whenever l interacts with an a node, c_1 is increased by one. The termination condition is $c_0 = c_1$ and then the leader gives an estimation on the number of nodes in the population, which we prove that with high probability is $2^{c_0+1} = 2^{c_1+1}$.
We first describe a simple terminating protocol that guarantee with high probability $n^{-a} \leq n_e \leq n^a$, for a constant a, i.e., the population size estimation is polynomially close to the actual size. Chernoff bounds then imply that repeating this protocol a constant number of times suffices to obtain $n/2 \leq n_e \leq 2n$ with high probability. We proceed by presenting our *Approximate Counting* protocol.

Approximate Counting (APC): $(l_{0,0}, q) \to (l_{1,0}, a)$, $(a,q) \to (a,a)$, $(l_{c_0,c_1}, q) \to (l_{c_0+1,c_1}, q)$ *if* $c_0 > c_1$, $(l_{c_0,c_1}, a) \to (l_{c_0,c_1+1}, a)$ *if* $c_0 > c_1$, $(l_{c_0,c_1}, \cdot) \to (halt, \cdot)$ *if* $c_0 = c_1$

3.2 Analysis

Lemma 1. *When half or less of the population has been infected, with high probability $c_0 > c_1$. In fact, $c_o - c_1 \approx \ln(n/2) - \sqrt{\log n} > 0$.*

Proof. We divide the process of the epidemic elimination into rounds i, where round i means that there exist i infected nodes in the population. Call an interaction a success if an effective rule applies and a new a appears on some node. Let the random variable X be the total number of interactions between the leader l and non-infected nodes q, the random variable Y be the total number of interactions between l and infected nodes a and the r.v. I be the total number of interactions in the population until all nodes become infected. We also define the r.v. X_i, Y_i and I_i to be the corresponding numbers in round i. Then, it holds that $X = \sum_{i=1}^{n} X_i$, $Y = \sum_{i=1}^{n} Y_i$ and $I = \sum_{i=1}^{n} I_i$. Finally, let the r.v. X_{ij} and Y_{ij} be independent Bernoulli trials such that for $1 \leq j \leq I_i$, $Pr[X_{ij} = 1] = p_{Xi}$, $Pr[X_{ij} = 0] = 1 - p_{Xi}$, $Pr[Y_{ij} = 1] = p_{Yi}$ and $Pr[Y_{ij} = 0] = 1 - p_{Yi}$. This means that in every interaction in round i, the leader, if chosen, interacts with a q node with probability p_{Xi} and with an a node with probability p_{Yi}. Then, it holds that $X_i = \sum_{i=1}^{I_i} X_{ij}$ and $Y_i = \sum_{i=1}^{I_i} Y_{ij}$, where I_i is the number of interactions until a success in round i.

$$p_{Xi} = \frac{2(n-i)}{n(n-1)},\ p_{Yi} = \frac{2i}{n(n-1)} \text{ and } p_{Ii} = \frac{2i(n-i)}{n(n-1)}$$

We also divide the whole process into two phases; the first phase ends when half of the population has been infected, that is $1 \leq i \leq \frac{n}{2}$ and for the second phase it holds that $\frac{n}{2} + 1 \leq i \leq n$. We shall argue that if the counter c_0 reaches a value which is a function of n, before the second counter c_1 reach c_0, the leader gives a good estimation. We use X^a and Y^a to indicate the r.v. X and Y during the first phase and X^b, Y^b for the second phase.
For $1 \leq i \leq \frac{n}{2}$ and by linearity of expectation we have:

$$E[X^a] = E[\sum_{i=1}^{n/2} X_i] = E[\sum_{i=1}^{n/2} \sum_{j=1}^{I_i} X_{ij}] = \sum_{i=1}^{n/2} \sum_{j=1}^{I_i} E[X_{ij}]$$

and by Wald's equation, we have that $E[\sum_{i=1}^{I_i} X_{ij}] = E[I_i]E[X_{ij}]$.

$$E[X^a] = \sum_{i=1}^{n/2} \frac{n(n-1)}{2i(n-i)} \frac{2(n-i)}{n(n-1)} = \sum_{i=1}^{n/2} \frac{1}{i} = H_{n/2} = \ln \frac{n}{2} + a_{n/2} \geq \ln \frac{n}{2}$$

where $H_{n/2}$ denotes the $(\frac{n}{2})$th Harmonic number and $0 < a_n < 1$ for all $n \in \mathbb{N}$ (Euler-Mascheroni constant).

$$E[Y^a] = E[\sum_{i=1}^{n/2} Y_i] = E[\sum_{i=1}^{n/2} \sum_{j=1}^{I_i} Y_{ij}] = \sum_{i=1}^{n/2} \sum_{j=1}^{I_i} E[Y_{ij}]$$

and by Wald's equation, we have that $E[\sum_{i=1}^{I_i} Y_{ij}] = E[I_i]E[Y_{ij}]$.

$$E[Y^a] = \sum_{i=1}^{n/2} \frac{n(n-1)}{2i(n-i)} \frac{2i}{n(n-1)} = \sum_{i=1}^{n/2} \frac{1}{n-i} = \sum_{i=1}^{n-1} \frac{1}{i} - \sum_{i=1}^{n/2-1} \frac{1}{i} = H_{n-1} - H_{n/2-1} \approx \ln 2$$

By Chernoff Bound, the probabilities that the r.v. X^a is less than $(1-\delta)E(X^a)$ and more than $(1+\delta)E(X^a)$ are

$$Pr[X^a \leq (1-\delta)E(X^a)] \leq e^{-\frac{\ln (n/2)\delta^2}{2}} = \frac{1}{(\frac{n}{2})^{\delta^2/2}}$$

$$Pr[X^a \geq (1+\delta)E(X^a)] \leq e^{-\frac{\ln (n/2)\delta^2}{3}} = \frac{1}{(\frac{n}{2})^{\delta^2/3}}$$

that is, X^a does not deviate far from its expectation. The probability that the r.v. Y^a is more than $(1+\delta)E(Y^a)$, for $\delta = \frac{2\sqrt{\log n}}{\ln 2}$ is

$$Pr[Y^a \geq (1+\delta)E(Y^a)] \leq e^{-\frac{\ln 2 \frac{2\sqrt{\log n}}{\ln 2}}{2}} = \frac{1}{n^{1/2}}$$

Thus, the leader interacts constant number of times with a nodes and less that $(1+\delta)E[Y^a]$ times w.h.p., until the epidemic infects half of the population and $O(\log n)$ times with non-infected nodes w.h.p.. In section 5, we have tested our results and the Figure 3 confirms this behavior. During the second phase, the infected nodes are more than the non-infected nodes, thus, eventually, the second counter c_1 will reach c_0 and the leader terminates. By that time, the first counter will already hold a function of n w.h.p. ($c_o - c_1 \approx \ln (n/2) - \sqrt{\log n} > 0$).

Corollary 1. *APC does not terminate w.h.p. until more than half of the population has been infected.*

□

Lemma 2. *Our Approximate Counting protocol terminates after $\Theta(\log n)$ parallel time w.h.p..*

Proof. After half of the population has been infected, it holds that $|c_1 - c_0| = \Theta(\log n)$. When this difference reaches zero, the unique leader terminates. We focus only on the effective interactions, which are always interactions between the leader l and a or q nodes. The probability that an interaction is (l, a) is $p_i = i/n > 1/2$, as more than half of the population is infected. Thus, the probability that an interaction is (l, q) is $q_i = 1 - p_i = (n-i)/n < 1/2$. In fact, the probability p_i

is constantly decreasing as the epidemic spreads throughout the population. This process may be viewed as a random walk on a line with positions $[0, \infty)$. The particle starts from position $a \log n$ and there is an absorbing barrier at 0. The position of the particle corresponds to the difference $|c_1 - c_0|$ of the two counters and it moves towards zero with probability $p_i > 1/2$. By the basic properties of random walks, after $\Theta(\log n)$ steps, the particle will be absorbed at 0. Thus, the total parallel time to termination is $\Theta(\log n)$.

Corollary 2. *When $c_0 = c_1$, w.h.p. 2^{c_0+1} is an upper bound on n.*

□

4 Leader Election with approximate knowledge of n

The existence of a *unique leader agent* is a key requirement for many population protocols [AAE08] and generally in distributed computing, thus, having a fast protocol that elects a unique leader is of high significance. In this section, we present our *Leader Election* protocol, giving, at first, an abstract description 4.1, the algorithm 4.2 and then, we present the analysis of it 4.3. Finally, we have measured the stabilization time of this protocol for different population sizes and the results can be found in section 5.

4.1 Abstract description

We assume that the nodes know *an upper bound on the population size n^b, where n is the number of nodes and b is any big constant number.*
All nodes store three variables; the round e, a random number r and a counter c and they are able to compute random numbers within a predefined range $[1, m]$. We define two types of states; the leaders (l) and the followers (f). Initially, all nodes are in state l, indicating that they are all potential leaders. The protocol operates in rounds and in every round, the leaders compete with each other trying to survive (i.e., do not become followers). The followers just copy the *tuple* (r, e) from the leaders and try to spread it throughout the population. During the first interaction of two l nodes, one of them becomes follower, a random number between 1 and m is being generated, the leader enters the first round and the follower copies the round e and the random number r from the leader to its local memory. The followers are only being used for information spreading purposes among the potential leaders and they cannot become leaders again. Throughout this paper, n denotes the *population size* and m *the maximum number that nodes can generate.*

Information spreading. It has been shown that the epidemic spreading of information can accelerate the convergence time of a population protocol. In this work, we adopt this notion and we use the followers as the means of competition and communication among the potential leaders. All leaders try to spread their information (i.e., their round and random number) throughout the population, but w.h.p. all of them except one eventually become followers. We say that a node x wins during an interaction if one of the following holds:

- Node x is in a bigger round e.
- If they are both in the same round, node x has bigger random number r.

One or more leaders L are in the *dominant state* if their tuple (r_1, e_1) wins every other tuple in the population. Then, the tuple (r_1, e_1) is being spread as an epidemic throughout the population, independently of the other leaders' tuples (all leaders or followers with the tuple (r_1, e_1) always

win their competitors). We also call leaders L the *dominant leaders*.

Transition to next round. After the first interaction, a leader l enters the first round. We can group all the other nodes that l can interact with into three independent sets.

- The first group contains the nodes that are in a bigger round or have a bigger random number, being in the same round as l. If the leader l interacts with such a node, it becomes follower.
- The second group contains the nodes that are in a smaller round or have a smaller random number, being in the same round as l. After an interaction with a node in this group, the other node becomes a follower and the leader increases its counter c by one.
- The third group contains the followers that have the same tuple (r, e) as l. After an interaction with a node in this group, l increases its counter c by one.

As long as the leader l survives (i.e., does not become a follower), it increases or resets its counter c, according to the transition function δ. When the counter c reaches $b \log n$, where n^b is the upper bound on the population size, it resets it and round r is increased by one. The followers can never increase their round or generate random numbers.

Stabilization. The protocol that we present stabilizes, as the whole population will eventually reach in a final configuration of states. To achieve this, when the round of a leader l reaches $\lceil \frac{2b \log n - \log(b \log^2 n)}{\log m} \rceil$, l stops increasing its round r, unless it interacts with another leader. This rule guarantees the stabilization of our protocol.

4.2 The protocol

In this section, we present our *Leader Election* protocol. We use the notation $p_{r,e}$ to indicate that node p has the random number r and is in the round e. Also, we say that $(r_1, e_1) > (r_2, e_2)$ if the tuple (r_1, e_1) wins the tuple (r_2, e_2). A tuple (r_1, e_1) wins the tuple (r_2, e_2) if $e_1 > e_2$ or if they are in the same round $(e_1 = e_2)$, it holds that $r_1 > r_2$.

4.3 Analysis

The leader election algorithm that we propose, elects a unique leader after $O(\frac{\log^2 n}{\log m})$ parallel time w.h.p.. To achieve this, the algorithm works in stages, called *epochs* throughout this paper and the number of potential leaders decreases exponentially between the epochs. An epoch i starts when any leader enters the ith round $(r = i)$ and ends when any leader enters the $(i + 1)$th round $(r = i + 1)$.
All omitted proofs in the following lemmas and theorems can be found in the Appendix.

Lemma 3. *During the execution of the protocol, at least one leader will always exist in the population.*

Lemma 4. *Assume an epoch e and k leaders with the dominant tuple (r, e) in this epoch. The expected parallel time to convergence of their epidemic in epoch e is $\Theta(logn)$.*

Lemma 5. *If a counter c of a leader l reaches $b \log n$, its epidemic will have already been spread throughout the population w.h.p..*

Theorem 1. *After $O(\frac{\log n}{\log m})$ epochs, there is a unique leader in the population w.h.p..*

Corollary 3. *After $t = \frac{\log n - \log(a \log^2 n)}{\log \log n}$ epochs, the remaining leaders are at most $a \log^2 n$ w.h.p..*

Protocol 1 Leader Election

$Q = \{l, f_{r,e}, l_{r,e}\} : r \in [1, m]$
δ :

#First interaction between two nodes. One of them becomes follower and the other remains leader. The leader generates a random number r and enters the first round ($e = 1$).
$(l, l) \rightarrow (l_{r,1}, f_{r,1})$

#A leader in round 0 always loses (i.e., becomes a follower) against a node in a higher round.
$(f_{r,e}, l) \rightarrow (f_{r,e}, f_{r,e})$
$(l_{r,e}, l) \rightarrow (l_{r,e}, f_{r,e}),\ l_{counter} = l_{counter} + 1$

#The winning node propagates its tuple. If a leader loses, it becomes follower.
$(f_{r,i}, f_{s,j}) \rightarrow (f_{k,l}, f_{k,l})$, **if** $(r, i) > (s, j)$ **then** $(k, l) = (r, i)$ **else** $(k, l) = (s, j)$
$(l_{r,i}, l_{s,j}) \rightarrow (l_{k,l}, f_{k,l}),\ l_{counter} = l_{counter} + 1$, **if** $(r, i) \geq (s, j)$ **then** $(k, l) = (r, i)$ **else** $(k, l) = (s, j)$
$(l_{r,i}, f_{s,j}) \rightarrow (f_{s,j}, f_{s,j})$, **if** $(s, j) > (r, i)$
$(l_{r,i}, f_{s,j}) \rightarrow (l_{r,i}, f_{r,i}),\ l_{counter} = l_{counter} + 1$, **if** $(r, i) > (s, j)$
$(l_{r,e}, f_{r,e}) \rightarrow (l_{k,j}, f_{k,j}),\ l_{counter} = l_{counter} + 1$

#When a leader increases its counter, the following code is being executed. It checks whether it has reached $c \log n$. If yes, it moves to the next round, generates a new random number and checks if it has reached the final round in order to terminate.
if $(l_{counter} = b \log n)$ **then**{
    Increase round;
    Generate a new random number between 1 and m;
    Reset counter to zero;
    if $(Round = \lceil \frac{2b \log n - \log(b \log^2 n)}{\log m} \rceil)$ **Stop increasing the round, unless you interact with a leader;**
}

Theorem 2. *Our Leader Election protocol elects a unique leader in* $O(\frac{\log^2 n}{\log \log n})$ *parallel time w.h.p..*

Proof. There are initially n leaders in the population. During an epoch e, by Lemma 4 the dominant tuple spreads throughout the population in $\Theta(\log n)$ parallel time, by Lemma 5 no (dominant) leader can enter to the next epoch if their epidemic has not been spread throughout the whole population before and by Theorem 1, there will exist a unique leader after $O(\frac{\log n}{\log m})$ epochs w.h.p., thus, for $m = b \log n$ the overall parallel time is $O(\frac{\log^2 n}{\log \log n})$. Finally, by Lemma 3, this unique leader can never become follower and according to the transition function in Protocol 1, a follower can never become leader again. The rule which says the leaders stop increasing their rounds if $r >= \frac{2b \log n - \log(b \log^2 n)}{\log m}$, unless they interact with another leader, implies that the population stabilizes in $O(\frac{\log^2 n}{\log \log n})$ parallel time w.h.p. and when this happens, there will exist only one leader in the population and eventually, our protocol always elects a unique leader.

Remark 1. By adjusting m to be any number between a constant and n and conducting a very similar analysis we may obtain a single leader election protocol whose time and space can be smoothly traded off between $O(\log^2 n)$ to $O(\log n)$ time and $O(\log n)$ to $O(n)$ space.

5 Experiments

We have also measured the stabilization time of our *Leader Election* and *Approximate Counting using a unique leader* algorithms for different network sizes. We have executed our protocols 100 times for each population size n, where $n = 2^i$ and $i = [3, 14]$. Regarding the *Leader Election*

algorithm which assumes some knowledge on the population size, the results (Figure. 1) support our analysis and confirm its logarithmic behavior. In these experiments, the maximum number that the nodes could generate was $m = 10$. Finally, all executions elected a unique leader except one which elected two leaders after $\frac{\log^2 n}{\log \log n}$ parallel time.
The stabilization time of our *Approximate Counting with a unique leader* algorithm is shown in Figure 2a. The algorithm always gives very close estimations to the actual size of the population (Figure 2b). Moreover, in Figure 3, we show the values of the counters c_0 and c_1, when half of the population has been infected by the epidemic. These experiments support our analysis, while the counter of infected nodes reaches a constant number and the counter of non-infected nodes reaches a value related to $\log n$. The figures can be found in section A (Figures).

6 Open Problems

Call a population protocol *size-oblivious* if its transition function does not depend on the population size. Is there a polylogarithmic time population protocol, correct with high probability, for the problem of:

1. leader election, which is *terminating* and *size-oblivious*?
2. leader election, which is *polylog state-bounded* and *size-oblivious*?
3. exact population size computation, which is *terminating*?
4. exact population size computation, which is $O(n)$ *state-bounded*?
5. leader election and counting on the model of network constructors [MS16]?
6. leader election and counting, if the agents are allowed to communicate only constant amount of data during an interaction?

Our leader election protocol requires a rough estimate on the size of the population in order to elect a leader in polylogarithmic time. Is it possible to completely drop this assumption by composing our protocol with a counting protocol whose goal will be to provide the required estimate?

Acknowledgments We would like to thank David Doty and Mahsa Eftekhari for their valuable comments and suggestions during the development of this research work.

References

[AAD+06] 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]:235–253, March 2006.

[AAE06] D. Angluin, J. Aspnes, and D. Eisenstat. Stably computable predicates are semilinear. In *25th annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 292–299, New York, NY, USA, 2006. ACM Press.

[AAE08] D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21[3]:183–199, September 2008.

[AAE+17] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, and R. L. Rivest. Time-space trade-offs in population protocols. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2560–2579. SIAM, 2017.

[AAER07] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distributed Computing*, 20[4]:279–304, November 2007.

[ABBS17] J. Aspnes, J. Beauquier, J. Burman, and D. Sohier. Time and Space Optimal Counting in Population Protocols. In *20th International Conference on Principles of Distributed Systems (OPODIS 2016)*, volume 70, pages 13:1–13:17, 2017.

[AG15] D. Alistarh and R. Gelashvili. Polylogarithmic-time leader election in population protocols. In *42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 9135 of *Lecture Notes in Computer Science*, pages 479 – 491. Springer, Berlin, Heidelberg, 2015.

[Ang80] D. Angluin. Local and global properties in networks of processors. In *Proceedings of the 12th annual ACM symposium on Theory of computing (STOC)*, pages 82–93. ACM, 1980.

[AR09] J. Aspnes and E. Ruppert. An introduction to population protocols. In B. Garbinato, H. Miranda, and L. Rodrigues, editors, *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer-Verlag, 2009.

[ASW85] C. Attiya, M. Snir, and M. Warmuth. Computing on an anonymous ring. In *Proceedings of the fourth annual ACM symposium on Principles of distributed computing*, PODC '85, pages 196–203. ACM, 1985.

[BBCS15] J. Beauquier, J. Burman, S. Claviere, and D. Sohier. Space-optimal counting in population protocols. In *DISC 2015: International Symposium on Distributed Computing*, pages 631–646. Springer, 2015.

[BCM+07] J. Beauquier, J. Clement, S. Messika, L. Rosaz, and B. Rozoy. Self-stabilizing counting in mobile sensor networks with a base station. In *Distributed Computing*, pages 63–76. Springer Berlin Heidelberg, 2007.

[BDS17] A. Belleville, D. Doty, and D. Soloveichik. Hardness of Computing and Approximating Predicates and Functions with Leaderless Population Protocols. In I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 141:1–141:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CDS14] H.-L. Chen, D. Doty, and D. Soloveichik. Deterministic function computation with chemical reaction networks. *Nat. Comput. 7*, pages 517 – 534, 2014.

[CFQS12] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27[5]:387–408, 2012.

[CMN+11] I. Chatzigiannakis, O. Michail, S. Nikolaou, A. Pavlogiannis, and P. G. Spirakis. Passively mobile communicating machines that use restricted space. *Theoretical Computer Science*, 412[46]:6469–6483, October 2011.

[DDLF+17] S. Das, G. A. Di Luna, P. Flocchini, N. Santoro, and G. Viglietta. Mediated population protocols: Leader election and applications. In *International Conference on Theory and Applications of Models of Computation*, pages 172–186. Springer, 2017.

[DLFI+17] G. A. Di Luna, P. Flocchini, T. Izumi, T. Izumi, N. Santoro, and G. Viglietta. Population protocols with faulty interactions: the impact of a leader. In *International Conference on Algorithms and Complexity (CIAC)*, pages 454–466. Springer, 2017.

[Dot14] D. Doty. Timing in chemical reaction networks. In *Proc. of the 25th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 772–784, 2014.

[DS15] D. Doty and D. Soloveichik. Stable leader election in population protocols requires linear time. In *International Symposium on Distributed Computing (DISC)*, pages 602–616. Springer, 2015.

[FJ06] M. Fischer and H. Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. *OPODIS*, vol 4305, 2006.

[GR09] R. Guerraoui and E. Ruppert. Names trump malice: Tiny mobile agents can tolerate byzantine failures. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 484–495. Springer, 2009.

[GS18] L. Gasieniec and G. Stachowiak. Fast space optimal leader election in population protocols. In *SODA 2018: ACM-SIAM Symposium on Discrete Algorithms*, 2018. to appear.

[IKIW14] T. Izumi, K. Kinpara, T. Izumi, and K. Wada. Space-efficient self-stabilizing counting population protocols on mobile sensor networks. *Theor. Comput. Sci.*, 552:99–108, 2014.

[KLO10] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the 42nd ACM symposium on Theory of computing (STOC)*, pages 513–522. ACM, 2010.

[LBBC14] G. A. D. Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Counting in anonymous dynamic networks under worst-case adversary. *IEEE 34th International Conference on Distributed Computing Systems (ICDCS)*, 2014.

[MCS11] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. *New Models for Population Protocols*. N. A. Lynch (Ed), Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2011.

[MCS13] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. In *15th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 281–295. Springer, 2013.

[Mic15] O. Michail. Terminating distributed construction of shapes and patterns in a fair solution of automata. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 37–46, 2015. Also in *Distributed Computing*, 2017.

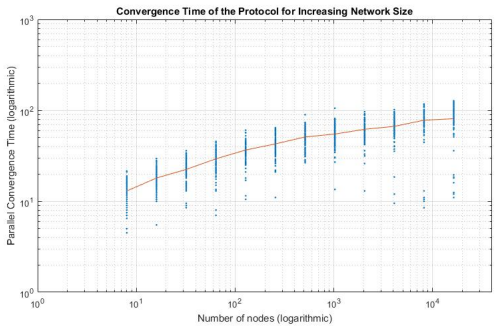
[MOKY12] R. Mizoguchi, H. Ono, S. Kijima, and M. Yamashita. On space complexity of self-stabilizing leader election in mediated population protocol. *Distributed Computing*, 25[6]:451–460, 2012.

[MS16] O. Michail and P. G. Spirakis. Simple and efficient local codes for distributed stable network construction. *Distributed Computing*, 29[3]:207–237, 2016.

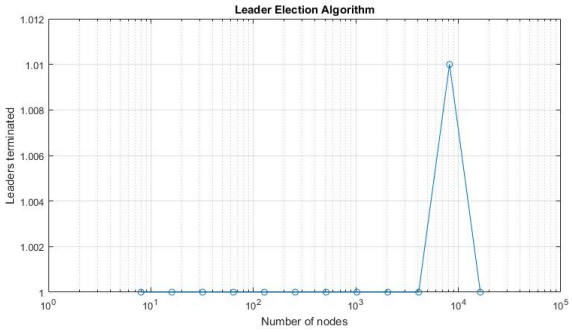
[MS18] O. Michail and P. G. Spirakis. Elements of the theory of dynamic networks. *Communications of the ACM*, 61[2], 2018.

[SCWB08] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. Computation with finite stochastic chemical reaction networks. *Nat. Comput. 7*, pages 615 – 633, 2008.

A Figures

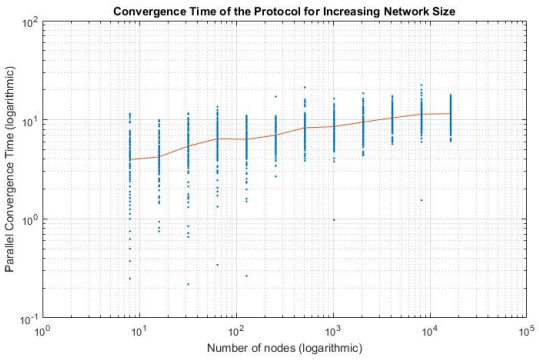


(a) Convergence time.

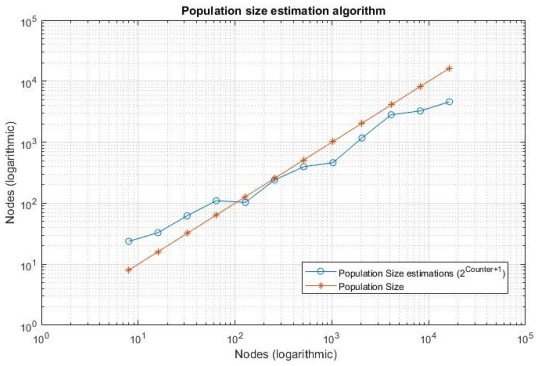


(b) Number of leaders after $\frac{\log^2 n}{\log \log n}$ parallel time.

Fig. 1: Leader Election with approximate knowing of n. Both axes are logarithmic. In (a) the dots represent the results of individual experiments and the line represents the average values for each network size.



(a) Convergence time.



(b) Estimations and actual sizes of the population.

Fig. 2: Approximate Counting with a unique leader. Both axes are logarithmic. In (a) the dots represent the results of individual experiments and the line represents the average values for each network size.

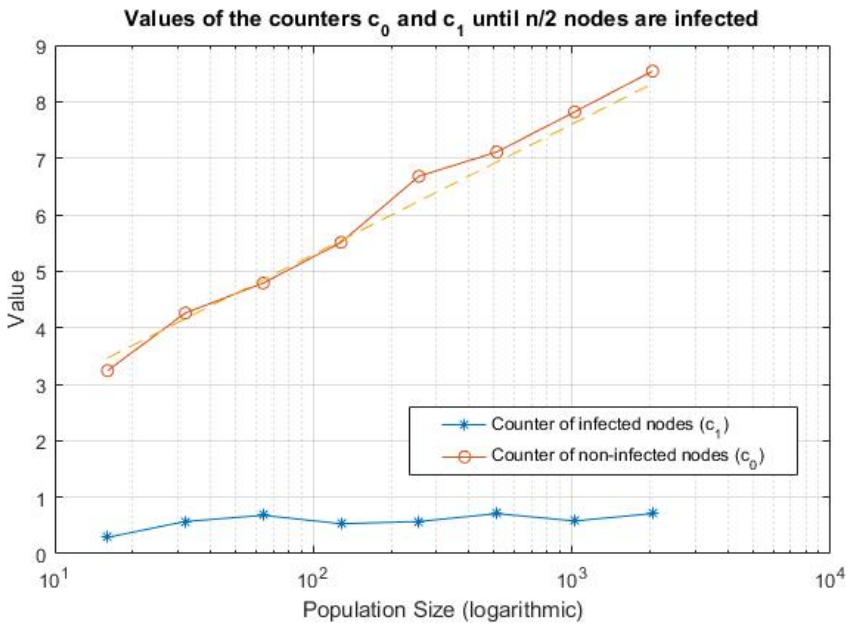


Fig. 3: Counters c_0 and c_1 when half of the population has been infected by the epidemic.

APPENDIX

Omitted analysis of the Leader Election algorithm

Lemma 3. *During the execution of the protocol, at least one leader will always exist in the population.*

Proof. Assume an epoch e, in which only one leader l_1 with the tuple (r_1, e_1) exists in the population and the rest of the nodes have become followers. In order for l_1 to become follower, there should be a follower with a tuple (r_2, e_2), where $(r_2, e_2) > (r_1, e_1)$. But, while the followers can never increase their epoch or generate a new random number, that would imply that there exists another leader l_2 with the tuple (r_2, e_2). □

Lemma 4. *Assume an epoch e and k leaders with the dominant tuple (r, e) in this epoch. The expected parallel time to convergence of their epidemic in epoch e is $\Theta(logn)$.*

Proof. Let the random variable X be the total number of interactions until all nodes have the dominant tuple (r, e). We divide the interactions of the protocol into rounds, where round i means that the epidemic has been spread to i nodes. Initially, $i = k$, that is, the k leaders are already infected by the epidemic, but we study the worst case where $k = 1$. Call an interaction a success if the epidemic spreads to a new node. Let also the random variables $X_i, 1 \leq i \leq n-1$, be the number of interactions in the ith round. Then, $X = \sum_{i=1}^{n-1} X_i$. The probability p_i of success at any interaction during the ith round is:

$$p_i = \frac{2i(n-i)}{n(n-1)}$$

where $i(n-i)$ are the effective interactions and $\frac{n(n-1)}{2}$ are all the possible interactions. By linearity of expectation we have:

$$\begin{aligned}
E[X] = E[\sum_{i=1}^{n-1} X_i] &= \sum_{i=1}^{n-1} E[X_i] = \sum_{i=1}^{n-1} \frac{1}{p_i} = \sum_{i=1}^{n-1} \frac{n(n-1)}{2i(n-i)} \\
&= \frac{n(n-1)}{2} \sum_{i=1}^{n-1} \frac{1}{i(n-i)} \\
&= \frac{n(n-1)}{2} \sum_{i=1}^{n-1} \frac{1}{n} (\frac{1}{i} + \frac{1}{n-i}) \\
&= \frac{(n-1)}{2} [\sum_{i=1}^{n-1} \frac{1}{i} + \sum_{i=1}^{n-1} \frac{1}{n-i}] \\
&= \frac{(n-1)}{2} 2H_{n-1} \\
&= (n-1)[ln(n-1) + a_{n-1}] = \Theta(n \log n)
\end{aligned}$$

where H_n denotes the nth Harmonic number and $a_n := H_n - \log n, (n \in \mathbb{N})$ is a decreasing sequence and $0 < a_n < 1$ for all $n \in \mathbb{N}$ (*Euler-Mascheroni constant*). It terms of parallel time, it holds that $E[\frac{X}{n}] = \frac{E[X]}{n} = \Theta(\log n)$. □

Lemma 5. *If a counter c of a leader l reaches $b \log n$, its epidemic will have already been spread throughout the population w.h.p..*

Proof. Let the r.v. X be the total number of interactions until all nodes have been infected by the dominant tuple. By Lemma 4, the expected interactions until the epidemic spreads throughout the whole population is $(n-1)\ln(n-1)+\Theta(1)$. By Chernoff Bound and for $\delta = 1/2$, it holds that

$$Pr[X \leq (1-\delta)\mu] \leq e^{\frac{-\delta^2\mu}{2}} \leq e^{-\frac{(n-1)\ln(n-1)}{8}} \leq \left(\frac{1}{n-1}\right)^{(n-1)/8}$$

Thus, the interactions per node under the uniform random scheduler until all nodes become infected are w.h.p. $\frac{(n-1)\ln(n-1)}{n} < \frac{n\ln n}{n} = \ln n$. Thus, after $b\log n$ interactions, where n^b is the population size estimation and b a large constant, there are no non-infected nodes w.h.p..

□

Theorem 1. *After $O(\frac{\log n}{\log m})$ epochs, there is a unique leader in the population w.h.p..*

Proof. Assume an epoch e, in which there are k leaders with the dominant tuple (r, e) and m is the biggest number that the leaders can generate. We shall argue that by the end of the next epoch $e+1$, approximately $\frac{k(m-1)}{m}$ leaders will have become followers and approximately $\frac{k}{m}$ leaders will have a new dominant tuple (r_2, e_2). Whenever the k leaders enter to the next epoch $e+1$, they generate a new random number between 1 and m. Let the random variable X_e be the number of leaders that have randomly generated the biggest number in epoch e. We view the possible values of the random choices as m bins and we investigate how many leaders shall go to each bin. Assume the sequence of the random numbers $C_i^e, 1 \leq i \leq k$ that the leaders generate in epoch e. Let the random variables X_i^e be independent Bernoulli trials such that, for $1 \leq i \leq k$, $Pr[X_i^e = 1] = p_i$ and $Pr[X_i^e = 0] = 1 - p_i$ and $X_e = \sum_{i=1}^{k} X_i^e$. The probability that a leader chooses randomly a number is

$$p_i = \frac{1}{m}$$

Then, the expected number of balls in each bin, thus in the biggest bin also (X_e) is

$$\mu = E(X_e) = E(\sum_{i=1}^{k} X_i^e) = \sum_{i=1}^{k} E(X_i^e) = \sum_{i=1}^{k} p_i = \sum_{i=1}^{k} \frac{1}{m} = \frac{k}{m}$$

Assume now inductively that $X_e \geq a\log^2 n$, where $a > 0$ and $m = \log n$. By the Chernoff bound and observing that $k \geq ma\log n \Rightarrow \frac{k}{m} \geq a\log n \Rightarrow \mu \geq a\log n$, we prove that the number of the new dominant leaders will be more than or equal to $\frac{k}{m}(1+\delta)$ with a negligible probability.

$$Pr[X_e \geq (1+\delta)\mu] \leq e^{-\frac{\mu\delta^2}{3}} \leq e^{-\frac{a\log n\delta^2}{3}} = n^{-\frac{a\delta^2}{3}} = n^{-\phi}$$

For $a \geq \frac{9}{\delta^2}$ it holds that $Pr[X_e \geq (1+\delta)\mu] \leq n^{-3}$. Consequently, if we had X_e leaders in epoch e, we now shall have no more than $X_{e+1} \leq (1+\delta)\frac{X_e}{m}$ leaders in epoch $e+1$ with probability $Pr[X_{e+1} \leq (1+\delta)\frac{X_e}{m}] \geq 1 - \frac{1}{n^3}$.
We can now assume that the expected number of leaders between the epochs can be described by the following recursive function.

$$G_e = \begin{cases} \frac{G_{e-1}}{m}, & i \geq 1 \\ n, & i = 0 \end{cases} \tag{1}$$

where $G_e = (1+\delta)X_e$. Then,

$$G_e = \frac{G_{e-1}}{m} = \frac{G_{e-2}}{m^2} = ... = \frac{n}{m^e}$$

The number of the expected epochs until at most $a\log^2 n$ leaders remain in the population is

$G_t = a\log^2 n \Rightarrow \frac{G_{t-1}}{m} = a\log^2 n \Rightarrow \frac{G_{t-2}}{m^2} = a\log^2 n \Rightarrow ... \Rightarrow \frac{n}{m^t} = a\log^2 n \Rightarrow$

$m^t = \frac{n}{a\log^2 n} \Rightarrow log_m(m^t) = log_m(\frac{n}{a\log^2 n}) \Rightarrow t = \log_m n - \log_m(a\log^2 n) \Rightarrow t = \frac{\log n - \log(a\log^2 n)}{\log m} \Rightarrow$

$t = \frac{\log n - \log(a\log^2 n)}{\log\log n}$

Let $E(e)$, be the event that in epoch e, there are at most G_e dominant leaders. We consider a success when $(E(e) \mid E(1) \cap E(2) \cap \ldots \cap E(e-1))$ occurs until we have at most $\log n$ leaders. By taking the union bound, the probability to fail after $t = \frac{\log n - \log(a\log^2 n)}{\log\log n}$ epochs is given by

$$Pr(\text{fail after } t \text{ epochs}) \leq \sum_{i=0}^{t} Pr[\text{fail in epoch i} \mid \text{success until (i-1)th epoch}]$$
$$\leq \sum_{i=0}^{t} \frac{1}{n^\phi} = \frac{\frac{\log n - \log(a\log^2 n)}{\log\log n}}{n^\phi} \leq \frac{1}{n^{\phi-1}} \leq \frac{1}{n^2}$$

Corollary 3. *After $t = \frac{\log n - \log(a\log^2 n)}{\log\log n}$ epochs, the remaining leaders are at most $a\log^2 n$ w.h.p..*

We argue that the number of leaders can be reduced from $a\log^2 n$ to $a\log n$ in one round w.h.p.. The expected value of dominant leaders is now $E[X_{t+1}] = a\log n$, thus, by the Chernoff Bound it holds that $Pr[X_{t+1} \geq (1+\delta)\mu] \leq e^{-\frac{a\log n\delta^2}{3}}$, and for $a \geq \frac{9}{\delta^2}$, $Pr[X_{t+1} \geq (1+\delta)\mu] \leq n^{-3}$.

Assume w.l.o.g. that $m = a\log n$ and according to the previous analysis, there exist $k = a\log n$ leaders after $t' = \frac{\log n - \log(a\log^2 n)}{\log\log n} + 1$ epochs. The expected value of $X_{t'+1}$ is now $\mu = E[X_{t'+1}] = 1$. Thus, by the Markov Inequality, the probability that the number of the dominant leaders in the next epoch are at least 2 is

$$P(X_{t'+1} \geq 2) \leq \frac{E[X_{t'+1}]}{2} = \frac{1}{2}$$

The probability that after $\log_m n$ epochs, there is no unique leader in the population is

$$P[\text{at least 2 leaders exist after } \log_m n \text{ epochs}] \leq (\frac{1}{2})^{\log_m n} = \frac{1}{2^{log_m n}}$$

The total number of epochs until there exists a unique leader in the population is w.h.p. $\frac{2\log n - \log(a\log^2 n)}{\log m} + 1 = O(\frac{\log n}{\log m})$.

□