Simple and Efficient Leader Election

Petra Berenbrink[1], Dominik Kaaser[2], Peter Kling[3], and Lena Otterbach[4]

1 **Universität Hamburg, Germany**
`berenbrink@informatik.uni-hamburg.de`
2 **Universität Hamburg, Germany**
`dominik.kaaser@uni-hamburg.de`
3 **Universität Hamburg, Germany**
`peter.kling@uni-hamburg.de`
4 **Universität Hamburg, Germany**
`otterbach@informatik.uni-hamburg.de`

Abstract

We provide a simple and efficient population protocol for leader election that uses $O(\log n)$ states and elects exactly one leader in $O\big(n \cdot (\log n)^2\big)$ interactions with high probability and in expectation. Our analysis is simple and based on fundamental stochastic arguments. Our protocol combines the tournament based leader elimination by Alistarh and Gelashvili, ICALP'15, with the synthetic coin introduced by Alistarh et al., SODA'17.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, G.3.16 Stochastic Processes

Keywords and phrases population protocols, leader election, distributed, randomized

Digital Object Identifier 10.4230/OASIcs.SOSA.2018.9

1 Introduction

We consider the *leader election* problem for *population protocols* introduced by [4], where one seeks a simple, distributed protocol that establishes a leader in a system of n initially identical agents. In this problem, in each round a pair of randomly chosen agents interact. The interacting agents observe each other's state and update their own state according to a simple deterministic rule, which is identical for each agent. A protocol's quality is measured by the number of interactions until a unique leader is found and by the number of states per agent required by the protocol. A key aspect of this model is that a unique leader *must* be found eventually. In particular, the protocol may not fail even with negligible probability.

Related Work. We give an overview of recent results in population protocols, with a focus on the leader election problem. We refer to [6] or the more recent [1] for a general survey on population protocols.

[4] introduce the population protocol model. They present protocols that stably compute any predicate definable via Pressburger arithmetic, which includes fundamental distributed tasks like leader election or consensus. [5, 6] show that predicates stably computable by population protocols are semi-linear. These early results restrict the number of states per agent to a constant and focus on what can and cannot be computed (in contrast to what can be computed efficiently). [8] prove that any population protocol that elects a leader with a constant number of states requires an expected number of $\Omega(n^2)$ interactions. Under some natural protocol assumptions (met, as far as we know, by all known population protocols), [1]

1st Symposium on Simplicity in Algorithms (SOSA 2018).
Editor: Raimund Seidel; Article No. 9; pp. 9:1–9:11
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

strengthen this lower bound by showing that population protocols using less than $1/2 \cdot \log\log n$ states need an expected number of $\Omega(n^2/\operatorname{polylog} n)$ interactions to elect a leader (their lower bound holds also for a broader class of problems).

To beat this polynomial lower bound on the time to elect a leader, recent results consider population protocols with polylogarithmically many states. [3] present a tournament based protocol that elects a leader in $O(n \cdot (\log n)^3)$ expected interactions using $O((\log n)^3)$ states. This protocol is quite intuitive and simple: Each leader candidate has a counter that is increased whenever it interacts with another agent. When a leader candidate meets an agent with a larger counter, it becomes a *minion*. Minions copy the largest counter seen so far. The main idea of the analysis is to show that, after $O(n \cdot (\log n)^3)$ interactions, one of the remaining leaders, say v, has a counter that exceeds any other leader's counter by $\Theta(\log n)$. This head start allows v to broadcast its counter to all other remaining leaders before their counters catch up.

[1] decrease the number of states to $O((\log n)^2)$ at the cost of an increased number of $O(n \cdot (\log n)^{5.3} \cdot \log\log n)$ expected interactions and a much more involved protocol. A key part of their protocol is the use of *synthetic coins*, which allow agents to access a random bit. More precisely, each agent has a bit that is flipped at the end of each interaction. One can show that, after roughly a linear number of interactions, about half of the agents have their bit set. Thus, by accessing the bit of its interaction partner (which is chosen uniformly at random), an agent has access to an almost uniformly random bit.

Three very recent, yet unpublished results [7, 2, 9] further improve upon these bounds. [7] present a protocol that requires $O(n \cdot (\log n)^2)$ interactions in expectation and $O((\log n)^2)$ states. [2] reduce the number of states to $O(\log n)$ while maintaining the number of required interactions. Finally, [9] further reduce the number of states to $O(\log\log n)$, matching the lower bound mentioned above. All these protocols and analysis are rather involved. In particular, [2, 9] are based on a phase clock to actively synchronize the behavior of the agents.

Our Contribution. We introduce a natural and simple leader election protocol that elects a single leader in $O(n \cdot (\log n)^2)$ expected interactions and uses $O(\log n)$ states. Our analysis is simple and based on fundamental stochastic arguments. It combines the tournament based leader elimination from [3] with the synthetic coin introduced in [1]. Using the synthetic coin, we initially mark $n/\log n$ agents. Since an agent's interaction partners are chosen uniformly at random, this effectively gives each agent access to a $(1/\log n)$-coin. This allows agents participating in the tournament to increase their counter only with a probability of $1/\log n$. As a result, we can show that an agent only needs a constant head start to broadcast its counter to all other remaining leaders before their counters can catch up. Our analysis relies on a simplified and slightly stronger analysis of the synthetic coin from [1].

Formally, we show the following theorem.

▶ **Theorem 1.** *With high probability*[1] *and in expectation, the protocol defined in Algorithm 1 elects exactly one leader in $O(n \cdot (\log n)^2)$ interactions. Furthermore, the protocol eventually reaches and stays in a configuration where exactly one leader contender is left with probability* 1.

2 Model and Protocol

We consider a population of n agents[2], also referred to as nodes. A *population protocol* specifies a set of possible *states*, the initial state of each agent, and an *update rule*. The

[1] The expression *with high probability* refers to a probability of $1 - n^{-\Omega(1)}$.
[2] All our results assume n to be larger than a suitable constant.

(deterministic) update rule is defined from the perspective of a single agent that knows its own state and the state of its communication partner. All agents start in the same initial state and use the same update rule. In every round a pair of agents is *activated* uniformly at random. In such an *interaction* the two activated nodes observe each other's state and apply the update rule. The goal is to reach a configuration where exactly one agent's state labels the agent as a (potential) leader and all other agents know that they are not a leader. Additionally, we require that every following configuration also have exactly one leader.

Protocol

We start with an informal description of our protocol. Every node has a *counter* and uses it to compete with other nodes. In the beginning some nodes will be marked. Leader candidates only increment their counter if they interact with a marked node. To initially mark a small fraction of nodes the protocol is split into two phases: the *marking phase* and the *tournament phase*.

Marking Phase. In the first phase, $\Theta(n/\log n)$ nodes get marked, see Section 3. To derive this, each node is equipped with an additional bit, the *flip bit*, that is flipped at the end of each interaction. After its first $3 \log\log n$ activations, a node starts to study the flip bits of its interaction partners. It marks itself if and only if all of its next $\log\log n$ interaction partners have their flip bits set. We refer to these at most $\log\log n$ crucial interactions as a node's *marking trials*. After a node's marking trials, it enters the tournament.

Tournament Phase. The second phase is responsible for electing a unique leader, see Section 4. At the beginning of its tournament phase every node is a possible leader and regards itself as a *contender*. Contenders count the number of their interactions with marked nodes. Whenever a contender interacts with another agent having a larger counter, it sets its role to *minion*. Minions carry the largest counter seen so far. Since the counter values never decrease we always have at least one contender left (see Lemma 6). The single remaining contender will be the unique leader.

Below, we summarize the parameters that constitute the state of a node v.

- **role** $r(v) \in \{\,\texttt{contender},\ \texttt{minion}\,\}$. Each node starts as a contender.
- **flip bit** $f(v) \in \{\,0, 1\,\}$. Initially the flip bit is set to 0. The flip bit will be used to approximate a random coin which is zero or one with probability $1/2$.
- **marker** $m(v) \in \{\,0, 1\,\}$. Initially the marker is set to 0 for *unmarked*. Nodes that mark themselves after their marking trials set this marker to 1. The marked nodes will be used in the tournament phase to approximate a random coin with a probability $1/\log n$ to be one.
- **phase** $p(v) \in \{\,\texttt{marking},\ \texttt{tournament}\,\}$. Each node starts in the marking phase.
- **counter** $c(v) \in \{\,0, \ldots, \mathrm{O}(\log n)\,\}$. The counter, initialized to 0, is used in both phases: In the marking phase to skip the first $3 \log\log n$ activations and then count the marking trials. In the tournament phase, the counter is used to determine the *winner* of an encounter.

In the following we assume $c(v)$ to be a variable that may count up to $\mathrm{O}(\log n)$. Hence, $c(v)$ can assume $\mathrm{O}(\log n)$ different values. Each of the parameters role, flip bit, marker, and phase only doubles the state space. Therefore, the total number of states per node is $\mathrm{O}(\log n)$.

For the case that all nodes reach the maximal possible counter value before all but one contender are eliminated, we let contenders with equal counter compete via their flip bits.

```
Algorithm leader-election(node v, node u)
  ──────────── marking phase ────────────
  if phase p(v) = marking then
    if counter c(v) ≥ 3 log log n and flip bit f(u) = 0 then
      phase p(v) ← tournament;                 /* leave phase unmarked */
    else
      increment counter c(v) ← c(v) + 1;
    if counter c(v) = 4 log log n then
      marker m(v) ← 1;
      phase p(v) ← tournament;                   /* leave phase marked */
  ──────────── tournament phase ────────────
  if phase p(v) = tournament then
    if role r(v) = contender then
      if marker m(u) = 1 and counter c(v) ≤ U log n then
        increment counter c(v) ← c(v) + 1;
      if c(v) < c(u) then
        role r(v) ← minion;          /* lose the duel due to the counter */
      if r(u) = contender and c(v) = c(u) and f(v) < f(u) then
        role r(v) ← minion;         /* lose the duel due to the flip bit */
    update counter c(v) ← max { c(u), c(v) };  /* adopt the maximum counter */
  ─────────────────────────────────────────
  flip the flip bit f(v) = 1 − f(v);
```

Algorithm 1 The leader election algorithm from the perspective of a single node v upon an interaction with communication partner u. Here, U is a large enough constant.

The complete update rule from the viewpoint of a node v interacting with a node u is formally defined in Algorithm 1. To avoid concurrency issues, we assume that node v operates on values of node u as they were *before* the interaction.

3 Analysis of the Marking Phase

In the first part of the analysis, our goal is to prove the following proposition, which states that after the marking phase, roughly $n/\log n$ nodes are marked. We use this in Section 4 to prove our main result.

▶ **Proposition 2.** *With high probability, after $4 \ln n$ interactions $\Theta(n/\log n)$ nodes are marked and all nodes are in the tournament phase.*

The proof of Proposition 2 works as follows: A node marks itself if and only if all communication partners of its $\log\log n$ marking trials have their bit set. If the bit of an interaction partner were set with probability $1/2$, this would imply that v marks itself with probability $1/2^{\log\log n} = 1/\log n$ and the desired result would follow via Chernoff bounds. The major difficulty is to show that, when a node starts its marking trials, the probability that a flip bit is set is close to $1/2$. We prove this in Section 3.1. Additionally, we have to show that not too many nodes start their marking trials before the balancing of the flip bits has finished. This is done in Section 3.2. The proof of Proposition 2 is finally given in Section 3.3.

3.1 Concentration of the $1/2$-Coin

As described in the overview, the major technical tool to prove Proposition 2 is the following concentration result for the number of flip bits set:

▶ **Lemma 3.** *Let $a > 0$ and consider an interaction t with $n \cdot \ln(\log\log n)/2 \leq t \leq n^a$. The number of flip bits that equal zero at the beginning of interaction t lies with probability at least $1 - n^{-a}$ in $(1 \pm 1/\log\log n) \cdot n/2$.*

The flip bit of a node is set if and only if the node attended an odd number of interactions. Observe that the number of interactions a node attended can be modeled by a balls into bins game: Nodes correspond to bins and activations to balls. For each interaction, we throw two balls into two random bins. Nodes with flip bit equal zero correspond to bins with an even load. Analyzing the number of such bins directly is difficult, since the bins' loads are correlated. However, we can use the Poisson approximation technique (see, e.g., the textbook [11, Chap. 5.4]).

More formally, assume we throw m balls independently and uniformly at random into n bins. Let X_i be the resulting load of bin i for $i \in \{1, 2, \ldots, n\}$. Additionally, let Y_i for $i \in \{1, 2, \ldots, n\}$ denote independent Poisson random variables with parameter m/n. We call $(X_1, \ldots, X_n)$ the *exact case* and $(Y_1, \ldots, Y_n)$ the *Poisson case*. The following well-known result relates these processes:

▶ **Known Result 1** (Corollary 5.9, [11])**.** *Any event that takes place with probability p in the Poisson case takes place with probability at most $pe\sqrt{m}$ in the exact case.*

Recall that we are interested in the number of bins with even load. This can be easily bounded in the Poisson case:

▶ **Lemma 4.** *Let $a > 0$ and $Y_1, \ldots, Y_n$ be independent Poisson random variables, each with parameter $\lambda \geq 2$. Define $\alpha := \min\{\lambda, \ln n/8\}$. The number of variables that are even lies with probability at least $1 - n^{-a}$ in $(1 \pm e^{-\alpha}) \cdot n/2$.*

Proof. One easily verifies that the probability for Y_i to be even is

$$\Pr[Y_i \text{ is even}] = \frac{1}{2} \cdot \left(1 + e^{-2\lambda}\right), \tag{1}$$

see Appendix A. Let the indicator random variable Z_i be 1 if and only if Y_i is even and 0 otherwise. By construction $Z_1, \ldots Z_n$ are independent 0-1 random variables and $Z = \sum_{i=1}^{n} Z_i$ is the number of variables that are even. By Equation (1), $\mathbb{E}[Z] = n \cdot \left(1 + e^{-2\lambda}\right)/2$. Set $\delta := e^{-2\alpha}$ and note that, since $\lambda \geq \max\{2, \alpha\}$, we have $\left(1 + e^{-2\lambda}\right) \cdot (1 + \delta) \leq (1 + e^{-\alpha})$ and $\left(1 + e^{-2\lambda}\right) \cdot (1 - \delta) \geq (1 - e^{-\alpha})$. Thus, standard Chernoff bounds (Lemma 11) yield

$$\begin{aligned} &\Pr\Big[Z \geq \frac{n}{2}(1 + e^{-\alpha})\Big] \leq \Pr[Z \geq (1 + \delta) \cdot \mathbb{E}[Z]] \leq e^{-\mathbb{E}[Z]\delta^2/3} \leq \frac{n^{-a}}{2} \quad \text{and} \\ &\Pr\Big[Z \leq \frac{n}{2}(1 - e^{-\alpha})\Big] \leq \Pr[Z \leq (1 - \delta) \cdot \mathbb{E}[Z]] \leq e^{-\mathbb{E}[Z]\delta^2/2} \leq \frac{n^{-a}}{2}\ . \end{aligned} \tag{2}$$

Combining both bounds gives the desired statement. ◀

Proof of Lemma 3. Fix an interaction t with $n \cdot \ln(\log\log n)/2 \leq t \leq n^a$ and set $\alpha := \min\{2t/n, \ln n/8\}$. Note that $\alpha \geq \ln(\log\log n)$. Let X denote the number of nodes that have their flip bit equal 0. As mentioned above, X also equals the number of bins with an even load when we throw $2t$ balls into n bins chosen independently and uniformly at random in the exact case. Let Y be the number of bins with an even load in the Poisson case (that is, each of the independent n Poisson random variables has parameter $\lambda = 2t/n$). By Known Result 1, we know that for any set $\mathcal{A} \subseteq \{0, 1, \ldots, n\}$ $\Pr[X \in \mathcal{A}] \leq e\sqrt{2t} \cdot \Pr[Y \in \mathcal{A}]$. Let $\mathcal{A} := [0, (1-e^{-\alpha}) \cdot n/2) \cup ((1+e^{-\alpha}) \cdot n/2, n]$. By Lemma 4, $e\sqrt{2t} \cdot \Pr[Y \in \mathcal{A}] \leq n^{-a}$. Using that $e^{-\alpha} \leq 1/\log\log n$, we get the desired statement. ◀

3.2 Bounding the Number of Early Marking Trials

By Lemma 3 we know that if a node starts its marking trials after (global) interaction $n \cdot \ln(\log\log n)/2$, the fraction of flip bits equal zero in the system is very close to 1/2. In the following, we bound the number of nodes that start their marking trials earlier.

▶ **Lemma 5.** *Let $a > 0$. With probability $1 - n^{-a}$, at most $n/\log n$ nodes start their marking trials before the $\big(n \cdot \ln(\log\log n)/2\big)$-th (global) interaction.*

Proof. Fix interaction $T_0 := n \cdot \ln(\log\log n)/2$ and consider the number of nodes that start their marking trials before T_0. We analyze this number using the Poisson approximation technique.

Performing T_0 global interactions corresponds to throwing $2T_0$ balls. Hence, we consider independent Poisson random variables $Y_1, \ldots, Y_n$, each with parameter $\lambda := 2T_0/n = \ln(\log\log n)$. A node starts its marking trials once it got activated $t := 3\log\log n$ times. The Chernoff bound for Poisson random variables (Lemma 12) gives

$$\begin{aligned}\Pr[Y_i \geq t] &\leq \frac{e^{-\lambda}(e\lambda)^t}{t^t} = \frac{1}{\log\log n}\left(\frac{e\ln(\log\log n)}{3\log\log n}\right)^{3\log\log n} \\ &\leq \frac{1}{\log\log n}\left(\frac{1}{2}\right)^{3\log\log n} \leq \frac{1}{(\log n)^3},\end{aligned} \tag{3}$$

where the second inequality follows from $\ln(x)/x \leq 1/2$ for any $x > 0$. Now consider binary random variables Z_i that are 1 if and only if $Y_i \geq t$ and let $Z := \sum_{i=1}^{n} Z_i$. It is $\mathbb{E}[Z] \leq n/(\log n)^3 \leq n/(2\log n)$. Lemma 11 implies for $\delta := 1$

$$\Pr\left[Z \geq \frac{n}{\log n}\right] \leq \Pr[Z \geq (1+\delta) \cdot \mathbb{E}[Z]] \leq e^{-n/(3(\log n)^3)} \leq e^{-(a+2)\ln n} = n^{-(a+2)}. \tag{4}$$

As in the proof of Lemma 3, we can now apply Known Result 1 to get the same guarantee for the exact case with probability n^{-a}. Therefore, with probability $1 - n^{-a}$, at most $n/\log n$ nodes have been activated more than $3\log\log n$ times before the T_0-th interaction, finishing the proof. ◀

3.3 Proof of Proposition 2

We show that, with high probability, after $T := 4n\ln n$ interactions all nodes are in their tournament phase and at least $n \cdot (1 - 1/\log n)$ of them have marking probability $\Theta(1/\log n)$. We conclude that the expected number of marked nodes is $\Theta(n/\log n)$ and use Chernoff to get the same result with high probability.

A node v enters the tournament phase at the latest when it was activated $4\log\log n$ times. Let N denote the number of interactions in which v was activated at the end of interaction T. Then $\mathbb{E}[N] = 2T/n = 8\ln n$. Set $\delta := 1/\sqrt{2}$ and note that $4\log\log n \leq 2\ln n \leq (1 - \frac{1}{\sqrt{2}})8\ln n = (1-\delta) \cdot \mathbb{E}[N]$. Hence, Lemma 11 implies

$$\Pr[N \leq 4\log\log n] \leq e^{-\mathbb{E}[N]\cdot\delta^2/2} = e^{-2\ln n} = n^{-2}. \tag{5}$$

A union bound over all nodes yields that, with high probability, all nodes are in the tournament phase after interaction T.

Lemma 3 implies that at the end of any interaction t with $T \geq t \geq T_0 := n \cdot \ln(\log\log n)/2$ the number of flip bits set lies in $n - (1 \pm 1/\log\log n) \cdot n/2$ with probability at least $1 - n^{-3}$. Thus, via a union bound over the $T - T_0 < 4n \ln n$ many interactions in the interval $[T_0, T]$, with high probability the number of flip bits set lies in $n - (1 \pm 1/\log\log n) \cdot n/2$ during the whole interval $[T_0, T]$. We use $\mathcal{E}$ to denote this event. Now consider a node v that has all of its marking trials during $[T_0, T]$ and let M_v be the event that v leaves its marking phase marked. Using that $(1 - 1/\log\log n)^{\log\log n} \geq 1/(2e)$ and $(1 + 1/\log\log n)^{\log\log n} \leq e$ we get

$$\frac{1}{2e\log n} \leq \Pr[M_v \mid \mathcal{E}] \leq \frac{e}{\log n}. \tag{6}$$

By Lemma 5, with high probability there are no more than $n/\log n$ nodes that start their marking trials before interaction T_0. Denote this high probability event with $\mathcal{E}'$. The two worst case scenarios are that all or non of the early nodes get marked. Let M be the number of marked nodes after T interactions. With the above argumentation we obtain

$$\frac{n}{6\log n} \leq \left(n - \frac{n}{\log n}\right) \cdot \frac{1}{2e\log n} \leq \mathbb{E}[M \mid \mathcal{E}, \mathcal{E}'] \leq n \cdot \frac{e}{\log n} + \frac{n}{\log n} \leq \frac{4n}{\log n}. \tag{7}$$

Applying Lemma 11 with $\delta = 1/4$ and $\delta = 5/6$, respectively, yields

$$\Pr\left[M \geq \frac{5n}{\log n} \;\middle|\; \mathcal{E}, \mathcal{E}'\right] \leq e^{-\frac{n}{288\log n}} \qquad \text{and} \qquad \Pr\left[M \leq \frac{n}{\log n} \;\middle|\; \mathcal{E}, \mathcal{E}'\right] \leq e^{-\frac{25n}{432\log n}}. \tag{8}$$

Thus, using the law of total probability and the fact that the events $\mathcal{E}$, $\mathcal{E}'$ happen with high probability, we get that $M \in \Theta(n/\log n)$ with high probability, finishing the proof. ◀

4 Analysis of the Main Algorithm

In the following section we analyze the tournament phase of our protocol. First we show that, at the beginning of any interaction, at least one node will have the contender role.

▶ **Lemma 6.** *At the beginning of any interaction, there will be at least one contender.*

Proof. Let $M = \max\{c(v)\}$ be the maximum counter value of all nodes. We observe that from the definition of the protocol in Algorithm 1 it follows that a node's counter cannot decrease. We show by an induction over the number of interactions that there always exists at least one contender which has the largest counter.

Initially, all nodes have the role contender and counter value 0. Therefore the base of the induction holds. For the induction step consider an arbitrary but fixed interaction between v and u. W.l.o.g. assume that $c(v) \geq c(u)$. We distinguish the following two cases, depending on the status of v and u before the interaction.

Case 1: $c(v) = M$**.**
If node v is a contender, it can only become a minion upon interaction with another contender u with $c(u) = M$ and $f(u) = 1$ while $f(v) = 0$. In this case, however, u remains a contender with maximal counter. It might also happen that the maximal counter value increases to $M + 1$ while v still has $c(v) = M$. In that case, however, u will have $c(u) = M + 1$ and thus u will be a contender with maximal counter value.

Case 2: $c(v) < M$**.**
Since both v and u do not have maximal counter value, the number of contenders having maximal counters cannot decrease.

Together, these two cases yield the induction step and the lemma follows. ◀

It is easy to see that the maximum counter values are spread through the system like messages in the case of push/pull broadcasting (see, e.g., [10]). The following observation is an adaption of the results for randomized broadcasting algorithms to our setting.

For any interaction t let $\overline{C}(t)$ and $\underline{C}(t)$ denote the maximal and minimal counter value of all nodes after interaction t.

▶ **Observation 7.** *Fix an interaction t. With probability at least $1-n^{-3}$ the maximal counter is broadcast to all nodes in $4n\log n$ interactions: $\underline{C}(t+4n\log n) \geq \overline{C}(t)$.*

We use this observation to obtain the following corollary.

▶ **Corollary 8.** *Fix an interaction t. Let $\mathcal{E}_t$ be the event that $\underline{C}(t+4n\log n) < \overline{C}(t)$. We have* $\Pr\left[\bigcup_{t=1}^{\Theta(n\cdot(\log n)^2)} \mathcal{E}_t\right] \leq 1/n$.

Proof. Note that for an interaction t the event $\mathcal{E}_t$ is precisely the complementary event from the one characterized in Observation 7. Therefore, $\Pr[\mathcal{E}_t] \leq n^{-3}$. The corollary follows from union bound over the $\Theta\big(n\cdot(\log n)^2\big)$ interactions. ◀

From Observation 7 and Corollary 8 we obtain that whenever a contender increments its counter, after at most $4n\log n$ interactions, all nodes have at least the same value.

▶ **Lemma 9.** *Let v_1, v_2 with $v_1 \neq v_2$ be two contenders which are both in the tournament phase. With constant probability $p_{\mathrm{L9}} = \Theta(1)$ one of the two contenders becomes a minion after $8n\log n$ interactions.*

Proof. W.l.o.g. assume that $c(v_1) \geq c(v_2)$. We split the $8n\log n$ interactions into two parts and show that with constant probability v_1 increments its counter in the first part, while v_2 does not increment its counter at all in both parts.

In each interaction, the probability that v_1 is selected interacts with a marked node is $\Theta(1/(n\log n))$. This follows directly from the number of marked nodes, see Proposition 2. Let p_1 be the probability that v interacts with a marked node in $4n\log n$ interactions. For the complementary event we get

$$\overline{p_1} = 1 - p_1 = \left(1 - \Theta\left(\frac{1}{n\log n}\right)\right)^{4n\log n} = e^{-\Theta(1)}$$

and thus $p_1 = \Theta(1)$.

Let p_2 be the probability that v_2 does not interact with a marked node and thus does not increment its counter in all $8n\log n$ interactions. The numbers of interactions of node v_1 and node v_2 are not independent, they are negatively correlated. To obtain a lower bound on p_2, we assume that in the worst case v_1 does not interact at all. Under this assumption, the probability that v_2 is selected is $2/(n-1)$ and thus the probability that v_2 does not interact with a marked node is at least

$$p_2 \geq \left(1 - \frac{2}{n-1}\cdot\Theta\left(\frac{1}{\log n}\right)\right)^{8n\log n} = \Theta(1).$$

From Corollary 8 we obtain that in the second part of the $8n \log n$ interactions v_2 will *see* a counter value which is as least as large as the counter value of v_1 after the first part of the interactions, with high probability. We use union bound on above probabilities and the result from Corollary 8 and conclude that with constant probability $p_{\mathrm{L9}} \geq p_1 \cdot p_2 - 1/n = \Theta(1)$ the node v_2 becomes a minion. ◀

Together with Lemma 6 and Corollary 8, the above lemma forms the basis for the proof of our main theorem, Theorem 1. Our main result is restated as follows.

▶ **Theorem 1.** *With high probability and in expectation, the protocol defined in Algorithm 1 elects exactly one leader in* $\mathrm{O}\big(n \cdot (\log n)^2\big)$ *interactions. Furthermore, the protocol eventually reaches and stays in a configuration where exactly one leader contender is left with probability* 1.

Proof. Let v_1 and v_2 be an arbitrary but fixed pair of contenders. We denote the first interaction when both v_1 and v_2 have entered the tournament phase as t_0. From Proposition 2 we obtain that with high probability $t_0 = \mathrm{O}(n \log n)$. Starting with interaction t_0, we consider $5/\log(1/(1-p_{\mathrm{L9}})) \cdot \log n = \Theta(\log n)$ so-called *periods* consisting of $8n \log n$ interactions each. More precisely, the i-th period consists of interactions in $[t_{i-1}, t_i)$ for $1 \leq i \leq 5/\log(1/(1-p_{\mathrm{L9}})) \cdot \log n$, where $t_i = t_0 + i \cdot 8n \log n$.

From Lemma 9 we know that with constant probability p_{L9} either v_1 or v_2 becomes a minion in each period. Therefore, with constant probability $1 - p_{\mathrm{L9}}$ both nodes v_1 and v_2 remain contenders in one period. After $5/\log(1/(1-p_{\mathrm{L9}})) \cdot \log n = \Theta(\log n)$ periods, the probability that v_1 and v_2 both remain contenders is at most $1/n^5$.

We take the union bound over all n^2 pairs of nodes and obtain a probability of at least $1 - 1/n^3$ that from each pair at least one node becomes a minion. From Lemma 6 we know that we always have at least one contender. Obviously, in any pair of nodes this contender cannot be the one to become a minion. Together, this implies that we have at least one contender and after $t_0 + \Theta(\log n) \cdot 8n \log n = \mathrm{O}\big(n \cdot (\log n)^2\big)$ interactions we have exactly one remaining contender, with high probability. All other nodes become minions with high probability, which shows the first part of the theorem.

To argue that the claimed run time also holds in expectation, we observe that the definition of the algorithm includes a *backup protocol* based on the flip bits. Observe that a similar approach to use a backup protocol has also been described in [3] and in [7]. Intuitively, whenever two contenders with the same counter value interact, there is a constant probability that one of them becomes a minion due to the flip bits. This backup protocol reduces the number of contenders to one in $\mathrm{O}\big(n^2 \log n\big)$ interactions in expectation. This follows from the coupon collector's problem. Since the probability that our main protocol fails and thus the backup protocol is actually needed is at most $\mathrm{O}\big(1/n^3\big)$, we obtain that our result also holds in expectation.

Finally, to show that the protocol eventually reaches a state where exactly one contender – the *leader* – is left, we observe the following. From any state of the system which is reachable over a sequence of interactions from the initial configuration, it is straight forward to specify a finite sequence of interactions such that all but one nodes become a minion. That means, at any time we have a positive probability to reach a stable state within finitely many interactions, and thus with probability 1 eventually only one contender will be left. ◀

References

1 Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L. Rivest. Time-Space Trade-offs in Population Protocols. In *Proc. SODA*, pages 2560–2579, 2017.

2 Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. *CoRR*, abs/1704.04947, 2017.

3 Dan Alistarh and Rati Gelashvili. Polylogarithmic-Time Leader Election in Population Protocols. In *Proc. ICALP*, pages 479–491, 2015.

4 Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006.

5 Dana Angluin, James Aspnes, and David Eisenstat. Stably Computable Predicates Are Semilinear. In *Proc. PODC*, pages 292–299, New York, NY, USA, 2006.

6 Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

7 Andreas Bilke, Colin Cooper, Robert Elsässer, and Tomasz Radzik. Population protocols for leader election and exact majority with $O(\log^2 n)$ states and $O(\log^2 n)$ convergence time. *CoRR*, abs/1705.01146, 2017.

8 David Doty and David Soloveichik. Stable leader election in population protocols requires linear time. *CoRR*, abs/1502.04246, 2015.

9 Leszek Gasieniec and Grzegorz Stachowiak. Fast Space Optimal Leader Election in Population Protocols. *CoRR*, abs/1704.07649, 2017.

10 Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. Randomized Rumor Spreading. In *Proc. FOCS*, pages 565–574, 2000.

11 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

A Appendix

Let $\mathbb{N}$ be the natural numbers including zero.

▶ **Definition 10** (Poisson Random Variable)**.** A discrete *Poisson random variable* Y with parameter λ is given by the following probability distribution on $\mathbb{N}$: $\Pr[Y = k] = e^{-\lambda}\lambda^k/k!$ for all $k \in \mathbb{N}$.

▶ **Lemma 11** (Chernoff Bounds [11] (Th. 4.4, Th. 4.5))**.** *Let $Z_1, \dots, Z_n$ be independent Poisson trials such that $Pr(Z_i) = p_i$. Let $Z = \sum_{i=1}^n Z_i$ and $\mu_L \leq \mathbb{E}[Z] \leq \mu_U$.*[3] *Then,*

- $\Pr[Z \geq (1+\delta)\mu_U] \leq e^{-\mu_U \delta^2/3}$ *for* $0 < \delta \leq 1$ *and*
- $\Pr[Z \leq (1-\delta)\mu_L] \leq e^{-\mu_L \delta^2/2}$ *for* $0 < \delta < 1$.

▶ **Lemma 12** (Chernoff for Poisson Variables [11] (Th. 5.4))**.** *Let X be a Poisson random variable with parameter λ.*

- *If $x > \lambda$, then* $\Pr(X \geq x) \leq \frac{e^{-\lambda}(e\lambda)^x}{x^x}$.
- *If $x < \lambda$, then* $\Pr(X \leq x) \leq \frac{e^{-\lambda}(e\lambda)^x}{x^x}$.

[3] While [11] states these bounds in terms of $\mu = \mathbb{E}[Z]$, it is easy to see and also mentioned in [11] that the Chernoff bounds hold also for suitable lower and upper bounds on $\mathbb{E}[Z]$.

Proof of Equation (1). Let Y be a Poisson random variable with parameter λ. Using the Taylor series $e^x = \sum_{k\in\mathbb{N}} \frac{x^k}{k!}$, we obtain

$$\begin{aligned}
\Pr[Y \text{ is even}] &= \sum_{k\in\mathbb{N}} \Pr[Y = 2k] = \sum_{k\in\mathbb{N}} \frac{e^{-\lambda}\lambda^{2k}}{(2k)!} \\
&= \frac{e^{-\lambda}}{2}\left(\sum_{k\in\mathbb{N}} \frac{\lambda^{2k}}{(2k)!} + \sum_{k\in\mathbb{N}} \frac{\lambda^{2k}}{(2k)!}\right) \\
&= \frac{e^{-\lambda}}{2}\left(\sum_{k\in\mathbb{N}} \frac{\lambda^{2k}}{(2k)!} + \sum_{k\in\mathbb{N}} \frac{\lambda^{2k+1}}{(2k+1)!} + \sum_{k\in\mathbb{N}} \frac{\lambda^{2k}}{(2k)!} - \sum_{k\in\mathbb{N}} \frac{\lambda^{2k+1}}{(2k+1)!}\right) \\
&= \frac{e^{-\lambda}}{2}\left(\sum_{k\in\mathbb{N}} \frac{\lambda^{k}}{(k)!} + \sum_{k\in\mathbb{N}} \frac{(-\lambda)^{k}}{(k)!}\right) \\
&= \frac{e^{-\lambda}}{2}(e^{\lambda} + e^{-\lambda}) \\
&= \frac{1}{2}(1 + e^{-2\lambda}) \ .
\end{aligned}$$

◀