Fast Self-Stabilizing Ranking

Abstract—**We present a silent, self-stabilizing ranking protocol for the population protocol model of distributed computing where agents interact in randomly chosen pairs to solve a common task. We are given n anonymous agents, and the goal is to assign each agent a unique rank in $\{1,\ldots,n\}$. Given unique ranks, it is straightforward to select a designated leader. Thus our protocol is a self-stabilizing leader election protocol as well.**

Ranking requires at least n states per agent, hence the goal is to minimize the additional number of states, called overhead states. The core of our protocol is a space-efficient but *non-self-stabilizing* ranking protocol that requires only $n+\mathrm{O}(\log n)$ states. Our protocol stabilizes in $\mathrm{O}(n^2\log n)$ interactions w.h.p. and in expectation, using $n+\mathrm{O}(\log^2 n)$ states in total. Our stabilization time is asymptotically optimal (see Burman et al., PODC'21). In comparison to the currently best known ranking protocol by Burman et al., which requires $n+\Omega(n)$ states, our result exponentially improves the number of overhead states.

Index Terms—**Self-Stabilization, Ranking, Leader Election, Labeling, Population Protocols**

I. Introduction

The population protocol model [9] is a simple yet expressive computational model for distributed computing. A population of n anonymous *agents* is given and the agents execute a protocol to solve a common task. At any time each agent is on one state out of a given set of states. In a sequence of discrete time steps pairs of agents are chosen uniformly at random to interact. In each interaction, the selected agents update their states according to a common transition function. The required number of states and the number of interactions until a valid configuration is reached form the main performance criteria of population protocols. The model has many applications, for example the authors of Angluin et al. [9] motivate the model with sensor networks, where devices with limited resources are required to perform simple computations. Other examples entail chemical reaction networks [36] and DNA computing [23]. Also, certain biochemical regulatory processes in living cells can be modeled as population protocols [22]. We refer to the surveys by Alistarh and Gelashvili [7] and Elsässer and Radzik [26] for further details and applications.

For many computational tasks in this model we are confronted with the following dilemma: the lack of unique identifiers prevents us from solving many problems efficiently. Unfortunately assigning and maintaining unique identifiers is a notoriously difficult problem, especially if memory (which means in our model the number of states) is scarce, and the systems are prone to faults. In this paper, we consider the problem of self-stabilizing ranking in the population protocol model. It is assumed that the agents start in an arbitrary configuration, and the goal is to assign a unique rank from $\{1,\ldots,n\}$ to each agent. We are interested in self-stabilizing protocols, which are protocols that are guaranteed to eventually reach a valid configuration. This has to hold for any arbitrary initial configuration, including configurations following transient faults.

The ranking problem is closely related to leader election. Here the goal is to reach a configuration where exactly one agent is in a leader state while all other agents are in so-called follower states. Given unique ranks it is straightforward to select a leader, e.g., by declaring the agent with rank 1 to be the leader. For the ranking problem n is a trivial lower bound on the size of the state space needed by any ranking protocol. This holds since each of the indistinguishable agents has to be able to adopt any of the n ranks. We, therefore, refer to the states that are required in addition to storing the n ranks as overhead states.

Results in a Nutshell. We present a novel self-stabilizing protocol for the ranking problem which stabilizes in $\mathrm{O}(n^2\log n)$ interactions w.h.p. It belongs to the natural class of so-called *silent* protocols which are protocols where, at a certain point, no agent changes its state any longer. Note our time complexity is optimal within this class [20]. Our algorithm uses only $\mathrm{O}(\log^2 n)$ overhead states in addition to the n states required to store the ranks of the agents. We underscore the prohibitive nature of this stringent memory restriction: the *additive* overhead of size $\mathrm{O}(\log^2 n)$ does not allow agents to hold their rank together with any additional piece of information. Indeed, even a single additional bit would immediately double the state space size. For example, this rules out that ranked agents participate in a phase-clock to synchronize the protocol, or store the information whether they are a leader or not. Similarly, if a leader exits which is then used to assign the ranks (as in, e.g., [20, 28]), leader agent cannot store any information about the ranks assigned so far, including its own rank. In order to overcome this issue, our protocol still works with a leader, but our leader is blissfully unaware of its special state. This piece of information is only revealed to the "unaware" leader once it communicates with an unranked agent u: depending on the combination of its own state and the state of u, the unaware leader realizes which role it has, which allows the agent to assign the correct rank to u. Finally, after ranking, our protocol solves the problem of self-stabilizing leader election by selecting the agent with the lowest rank as the leader.

II. Related Work

Ranking Protocols. The ranking problem occurs in different communication models and under various assumptions [1, 4, 12, 17, 32]. In this overview we focus on results in the population protocol model. Ranking in population protocols typically considers so-called *safe* and *silent* protocols. In a safe protocol, once an agent receives a rank, this rank is never changed. In a silent protocol the population eventually reaches a final configuration in which agents no longer change their states.

Beauquier et al. [11] present a population protocol for a generalization of the ranking problem. Their protocol distributes m unique labels with $m \geq n$ among the agents. For $m = n$, this corresponds to the ranking problem. The authors focus on the feasibility of the solution but do not analyze the time needed for the population to stabilize. Another set of self-stabilizing ranking algorithm is provided by Burman et al. [18]. Again, their focus lies on feasibility rather than optimizing the time and space complexity.

Gasieniec et al. [28] present two safe and silent ranking protocols, one for a range of $[1, (1+\epsilon)\cdot n]$ and one for the optimal range $[1, n]$. The first protocol requires $\mathrm{O}(n \log(n)/\epsilon)$ interactions w.h.p. and uses $(2+\epsilon)n + \mathrm{O}(n^{\alpha})$ states for an arbitrary constant α. While the protocol used $\Omega(n)$ overhead states, for $\epsilon = \Omega(1)$ the protocol only requires an asymptotically optimal number of $\mathrm{O}(n \log n)$ interactions. In addition to the upper bound, the authors show for this protocol a lower bound, which can also be generalized to a wider class of protocols: to assign ranks from the range $[1, n+r]$ the expected number of interactions is at least $n \cdot (n-1)/(2(r+1))$. For the optimal range $[1, n]$, the authors present a protocol which requires $\mathrm{O}(n^3)$ interactions in expectation and $n + 5\sqrt{n} + \mathrm{O}(n^c)$ states, where c can be an arbitrarily small constant. A generalization of the protocol is parameterized by $\epsilon \geq n^{-1}$ and uses $(1 + 9\sqrt{\epsilon}) \cdot n + \mathrm{O}(\log \log n)$ states and $\mathrm{O}(n^2/\epsilon)$ interactions w.h.p. Finally, the authors show a general lower bound for safe and silent protocols that even holds if a designated leader agent is present from the beginning: any safe and silent protocol that uses a range of $[1, n]$ and produces a valid ranking with probability larger than $1 - 1/n$ requires at least $n + \sqrt{n-1} - 1$ states (see the full version [27] of [28]). Note that none of these protocols are self-stabilizing.

Very recently, Gasieniec et al. [29] present a different approach for deanonymizing a population that is orthogonal to the previously mentioned approaches. Here, all agents agree on a common coordinate system and assign themselves unique points in that system.

Leader Election. Leader election is a prominent problem in the population protocol model that is closely related to ranking. A long series of papers on non-self-stabilizing leader election [2, 3, 6, 14, 16, 30, 31, 39] has lead to the currently best known protocol by Berenbrink et al. [13] which uses $\mathrm{O}(\log \log n)$ states and stabilizes in asymptotically optimal $\mathrm{O}(n \cdot \log n)$ interactions in expectation. Sudo et al. [37] present a so-called loosely-stabilizing protocol that elects a leader starting from any arbitrary initial configuration. Informally, a loosely-stabilizing protocol converges quickly from an arbitrary initial configuration to a valid configuration with a unique leader and then remains in a valid configuration for a long time. Their protocol improves on earlier works on loosely-stabilizing leader election [38, 40].

To the best of our knowledge, all efficient self-stabilizing leader election protocols in the population protocol model are based on self-stabilizing ranking protocols. These protocols compute a ranking which trivially implies a leader. In this setting, Cai et al. [21] present a silent self-stabilizing leader election protocol that requires $\mathrm{O}(n^3)$ interactions w.h.p. using n states. Furthermore, Burman et al. [20] present three self-stabilizing protocols for leader election based on ranking. The first one is silent and requires $\mathrm{O}(n^2 \log n)$ interactions w.h.p. using $\mathrm{O}(n)$ states. The second protocol and the third protocol are both non-silent. The second one needs only $\mathrm{O}(n \log n)$ interactions w.h.p. at the expense of an exponential number of $\exp(\mathrm{O}(n^{\log n} \cdot \log n))$ states. The third one allows a trade-off between the number of states and the running time. It requires $\mathrm{O}(Hn^{1+1/(H+1)})$ interactions w.h.p. and $\mathrm{O}(n^{\Theta(n^H)} \cdot \log n)$ states. Finally, note that Cai et al. [21] show that any self-stabilizing leader election protocol requires at least n states and Burman et al. [20] show that every silent leader election protocol requires $\Omega(n^2)$ interactions in expectation (and $\Omega(n^2 \log n)$ interactions w.h.p.). Thus, the silent self-stabilizing leader election protocol implied by our ranking protocol matches the lower bound on the time complexity from [20] and almost matches the state complexity from Cai et al. [21], except for the *additive* $\mathrm{O}(\log^2 n)$ states.

Ranking of Anonymous Networks. Another related problem is assigning a rank to all nodes of an anonymous network. The network is modeled by a connected graph $G = (V, E)$ whose edges may change over time. Time proceeds in synchronous rounds, and in each round, a node $v \in V$ can send a message to all its neighbors. The nodes have no identifiers, but usually, they differentiate between their neighbors based on *port numbers.* Kowalski and Mosteiro [34] present a (non-self-stabilizing) leader election protocol that runs in $\mathrm{O}(t_{\mathrm{mix}} \log^2 n)$ time w.h.p., where t_{mix} is the mixing time of simple random walk on G. Di Luna and Viglietta [24, 25] consider the "reverse" problem of determining the number of nodes n given a predetermined leader. Their algorithm takes $\mathrm{O}(n)$ rounds. Note that algorithms for population protocols can usually be transferred to anonymous networks: the authors of [8] present a general framework that simulates a population protocol on a graph. Their approach is random-walk based and the runtime depends on the properties of the graph G such as diameter or conductance. The converse direction is not straightforward: in an anonymous network *all* nodes can communicate with *all* their neighbors in each time step, while in a population protocol *exactly one pair* of agents interacts per time step.

III. Model and Results

We consider a set V of n agents. Each agent v has a state $x(v)$ from a state space Q. A *configuration* is a vector $(x(v))_{v \in V} \in Q^n$ that maps each agent $v \in V$ to its state $x(v)$. Time is measured in discrete steps. In each time step, two agents are chosen uniformly at random to interact. The two chosen agents update their states according to a common transition function. The configuration $\vec{X}_0$ at time 0 is called *initial configuration.* Due to the random interactions, the configuration at time $t > 0$ is a random vector $\vec{X}_t \in Q^n$. Since we consider self-stabilization, the initial configuration is arbitrary. The goal is to reach a *valid configuration* in which all agents have a unique rank from $[n]$ defined as $[n] = \{1, \dots, n\}$. A valid configuration is called *stable* if no sequence of interactions exists that changes the output on any agent. We say that a protocol stabilizes after τ steps if $\vec{X}_\tau$ is valid and stable.

A population protocol with state space Q is *self-stabilizing* with respect to a set of configurations $C_L \subset Q^n$ if and only if it fulfills the following two properties.

- *Closure:* If $\vec{X}_t \in C_L$ for some t, then $\vec{X}_{t+1} \in C_L$. If additionally $\vec{X}_{t+1} = \vec{X}_t$, i.e., no agent changes its state, the protocol is *silent*.
- *Probabilistic Stabilization:* For every $\vec{X}_t \in Q^n$ we have $\lim_{\tau\to\infty} \Pr\left[\vec{X}_{t+\tau} \in C_L\right] = 1$. Note that in contrast to other models, we cannot guarantee deterministic stabilization for population protocols due to the random interactions.

For our protocols, we let C_L be the set of all permutations of $[n]$. That is, C_L is the set of all configurations in which every agent is assigned a rank, and all ranks are unique. Together with an output function that maps a rank of 1 to "leader" and any other state to "follower" this immediately carries over to self-stabilizing leader election.

The following theorem is our first main result. The corresponding protocol SPACEEFFICIENTRANKING is introduced and analyzed in Section IV.

Theorem 1. SPACEEFFICIENTRANKING *is a silent population protocol with* $n + \Theta(\log n)$ *states that reaches a valid ranking in* $\mathrm{O}(n^2 \log n)$ *interactions w.h.p.*

In our second main result, we transform the protocol from Theorem 1 into a self-stabilizing protocol. Most notably, we amend it with error-detection and a resetting mechanism. We present the required changes and analyze the corresponding protocol RANKING+ in Section V.

Theorem 2. RANKING+ *is a silent population protocol for self-stabilizing ranking that requires* $n + \mathrm{O}(\log^2 n)$ *states and stabilizes in* $\mathrm{O}(n^2 \cdot \log n)$ *interactions w.h.p.*

IV. NON-SELF-STABILIZING RANKING

In this section we describe our non-self-stabilizing ranking protocol SPACEEFFICIENTRANKING. We assume that the exact value of n is known. This is in fact necessary for leader election and thus also for ranking by Theorem 1 in [21].

Our protocol uses the following state space, where Q_{LE} is the state space of the leader election protocol by Gasieniec and Stachowiak [30], and $\uplus$ is the disjoint union of two sets:

$$Q_{\text{Ranking}} = \underbrace{Q_{\text{LE}}}_{\mathtt{q}_{\text{LE}}} \times \underbrace{\{0,1\}}_{\mathtt{leaderDone}} \uplus \underbrace{\{1,\dots,\lceil c_{\text{wait}} \cdot \log n\rceil\}}_{\mathtt{waitCount}} \uplus \underbrace{\{1,\dots,\lceil \log n\rceil\}}_{\mathtt{phase}} \uplus \underbrace{\{1,\dots,n\}}_{\mathtt{rank}}.$$

The expression $\mathtt{q}_{\text{LE}}(v) \in Q_{\text{LE}}$ contains the leader-election state of v and is initialized to the initial state $q_0 \in Q_{\text{LE}}$ as described in [15]. The variable $\mathtt{leaderDone}(v)$ indicates if v has finished the leader election, it is initialized with 0. The variables $\mathtt{waitCount}(v)$ and $\mathtt{phase}(v)$ are both used to guide our ranking protocol and $\mathtt{rank}(v)$ is used to store the rank of agent v. These values are all initialized with $\perp$, indicating that the value is as-yet undefined. Throughout the run of the protocol, each agent can have exactly one value of $\mathtt{q}_{\text{LE}}(v)$, $\mathtt{waitCount}(v)$, $\mathtt{phase}(v)$, or $\mathtt{rank}(v)$ be *not* equal to $\perp$, and $\mathtt{leaderDone}(v) \neq \perp$ if and only if $\mathtt{q}_{\text{LE}}(v) \neq \perp$. We call these agents leader-electing agents, waiting agents, phase agents, and ranked agents, respectively.

Our protocol is divided into two parts, a *wrapper* protocol that elects a unique leader and the actual ranking protocol that assigns the ranks given a leader. The wrapper protocol is given as SPACEEFFICIENTRANKING in Protocol 1. It elects a unique leader and transitions to the actual ranking protocol RANKING, given in Protocol 2. The wrapper protocol begins by electing a unique leader using the protocol from [30]. Similarly to [15], we assume that the leader election protocol provides, additionally to the state $\mathtt{q}_{\text{LE}}(v) \in Q_{\text{LE}}$, a variable $\mathtt{leaderDone}(v)$ which is set to 1 when agent v assumes that the leader election is done. When all agents have $\mathtt{leaderDone}(v) = 1$ there is w.h.p. exactly one leader agent ℓ. Unfortunately, the leader agent ℓ cannot be identified by storing an additional bit, as this would immediately double the state space size to $2n$ states per agent. Nevertheless, as we will see later, the protocol maintains (w.h.p.) the invariant that exactly one agent *acts* as leader at any given time. During the course of the protocol, the leader agent ℓ will alternate between being the sole waiting agent and being the unaware leader. The leader agent ℓ will assign ranks to unranked agents unless it is waiting.

The formal description of the transition between the leader-election protocol and our core ranking protocol is given in Protocol 1. It works as follows. Whenever two leader-electing agents interact, they follow the transition function of the leader election protocol (lines 1–2). The real ranking protocol is started by the leader ℓ as soon as $\mathtt{leaderDone}(\ell)$ is set to 1. This initiates a one-way epidemic informing all agents that Phase 1 starts (lines 7–9). All agents transition out of their leader-election state and into a state in which only values of our main ranking protocol (phase, wait count, or rank) are set. The leader ℓ, as soon as $\mathtt{leaderDone}(\ell) = 1$, forgets its leader-election

Protocol 1 SPACEEFFICIENTRANKING(u, v)

```
Leader Election
1  if $\mathtt{q}_{\text{LE}}(u), \mathtt{q}_{\text{LE}}(v) \neq \perp$ then        ▷ two leader-electing agents interact
2    execute ELECTLEADER$(u, v)$
The Leader is Done
3  if $\exists \ell \in \{u, v\} : (\mathtt{isLeader}(\ell) = \mathtt{leaderDone}(\ell) = 1)$ then
4      $(\mathtt{q}_{\text{LE}}(\ell), \mathtt{leaderDone}(\ell)) \leftarrow (\perp, \perp)$        ▷ $\ell$ forgets LE state
5      $\mathtt{waitCount}(\ell) \leftarrow \lceil c_{\text{wait}} \cdot \log n \rceil$   ▷ $\ell$ becomes a waiting agent
6      return
Propagate Start of Ranking
7  if $\mathtt{q}_{\text{LE}}(w) \neq \perp$ and $\mathtt{q}_{\text{LE}}(x) = \perp$ for a $\{w, x\} = \{u, v\}$ then
8      $(\mathtt{q}_{\text{LE}}(w), \mathtt{leaderDone}(w)) \leftarrow (\perp, \perp)$ ▷ $w$ forgets its LE state
9      $\mathtt{phase}(w) \leftarrow 1$                        ▷ $w$ becomes a phase agent
Ranking Protocol
10 if $\mathtt{q}_{\text{LE}}(u), \mathtt{q}_{\text{LE}}(v) = \perp$ then        ▷ non-leader-electing agents interact
11     execute RANKING$(u, v)$
```

Protocol 2 RANKING(u, v)

```
1  if $\texttt{phase}(v) = \bot$ then return                    ▷ if $v$ has rank, do nothing
2  if $\texttt{rank}(u) \neq \bot$ then
3      let $k = \texttt{phase}(v)$
4      if $1 \leq \texttt{rank}(u) \leq f_k - f_{k+1}$ then     ▷ $u$ may assign a rank to $v$
5          $(\texttt{phase}(v), \texttt{rank}(v)) \leftarrow (\bot, f_{k+1} + \texttt{rank}(u))$
6          if $\texttt{rank}(u) < f_k - f_{k+1}$ then            ▷ phase not done
7              $\texttt{rank}(u) \leftarrow \texttt{rank}(u) + 1$
8          else if $k < \lceil \log_2(k) \rceil$           ▷ $u$ reached end of non-final phase
9              $(\texttt{rank}(u), \texttt{waitCount}(u)) \leftarrow (\bot, \lceil c_{\text{wait}} \cdot \log n \rceil)$
10     if $\texttt{rank}(u) = f_k$ then                        ▷ $u$ has last rank in phase $k$
11         $\texttt{phase}(v) \leftarrow \texttt{phase}(v) + 1$
12 if $\texttt{phase}(u) \neq \bot$ then
13         ▷ if both agents are phase agents, broadcast more advanced phase
14     $\texttt{phase}(u), \texttt{phase}(v) \leftarrow \max\{\texttt{phase}(u), \texttt{phase}(v)\}$
15 if $\texttt{waitCount}(u) \neq \bot$ then
16             ▷ decrement wait counter and ultimately transition to rank 1
17     $\texttt{waitCount}(u) \leftarrow \texttt{waitCount}(u) - 1$
18     if $\texttt{waitCount}(u) = 0$ then
19         $\texttt{waitCount}(u), \texttt{rank}(u) \leftarrow \bot, 1$
```

state (setting it to $\bot$) and sets $\texttt{waitCount}(\ell) = \lceil c_{\text{wait}} \log n \rceil$ (lines 3–5). At that time the agents are w.h.p. in a configuration with the following property (see Lemma 3): a unique leader agent ℓ has $\texttt{waitCount}(\ell) = \lceil c_{\text{wait}} \log n \rceil$, and all other agents are either in a state of Q_{LE} where $\texttt{isLeader}(v) = 0$, or have $\texttt{phase}(u) = 1$. We call this set of configurations C_{SR} for start ranking.

Having reached a configuration in C_{SR}, the actual ranking protocol RANKING described in Protocol 2 takes over. Whenever the agent ℓ with $\texttt{waitCount}(\ell) \neq \bot$ interacts with a phase agent, it decrements $\texttt{waitCount}(\ell)$ (lines 15–17). As soon as this counter reaches zero, ℓ assigns itself rank 1, thereby taking on the role of unaware leader (lines 18–19). The now-unaware leader ℓ with $\texttt{rank}(\ell) = 1$ starts the ranking. At that point each agent $v \neq \ell$ has $\texttt{phase}(v) = 1$ w.h.p.

The ranking is performed in $\log_2(n)$ *phases*. At the beginning of a phase, the leader ℓ has $\texttt{rank}(\ell) = 1$, and each agent v that is not yet ranked stores the number of the current phase in its value $\texttt{phase}(v)$. In the first phase, ranks $\lceil n/2 \rceil + 1, \ldots, n$ are assigned. We write f_k for the final (and maximal) rank that is assigned in phase k. In our protocol, $f_1 = n$, and $f_i = \lceil f_{i-1}/2 \rceil$ for all $i > 1$. Note that if n is a power of two, then $f_k = n/2^{k-1}$. In general, in the kth phase, ranks $f_{k+1} + 1, \ldots, f_k$ are assigned.

As an invariant, in each of these phases, there is at most one leader ℓ. While this leader stores a rank $\texttt{rank}(\ell) \leq f_{k+1}$, it only learns that it is a leader when it interacts with another agent v with $\texttt{phase}(v) = k$. The leader ℓ is responsible for the ranking in the actual phase k. Let us consider an interaction between ℓ and another agent v that stores phase k (meaning v is still not ranked). If $\texttt{rank}(\ell) = r \leq f_k - f_{k+1}$ two things happen (see lines 4–9). First, agent v sets $\texttt{rank}(v) = f_{k+1} + r$. Second, as long as $r < f_{k+1} - f_k$, agent ℓ increments $\texttt{rank}(\ell)$ by 1. If $r = f_k - f_{k+1}$, then agent ℓ knows that v has received the largest rank of phase k. In that case ℓ forgets its rank ($\texttt{rank}(\ell) = \bot$) and sets its state to $\texttt{waitCount}(\ell) = \lceil c_{\text{wait}} \log n \rceil$. In the case ℓ just finished the final phase $k = \lceil \log_2 n \rceil$, we know that $\texttt{rank}(\ell) = 1$ and the protocol is silent from now on (since no phase agents remain). Otherwise (if $k < \lceil \log_2 n \rceil$), the assignment of rank f_k triggers a transition to phase $k+1$ (see lines 10–11). As soon as the agent u with $\texttt{rank}(u) = f_k$ encounters an agent v with $\texttt{phase}(v) = k$, v increments $\texttt{phase}(v)$ to $k+1$. The incremented phase is spread via a one-way epidemic among phase agents only (see lines 12–14). Whenever two such agents meet, they propagate the higher phase. Note that ranked agents cannot participate in the one-way epidemic as this would require additional state. As soon as ℓ (now being a waiting agent) reaches $\texttt{waitCount}(\ell) = 0$ it starts ranking again by assigning itself Rank 1 again. At that time all phase agents have an updated phase counter w.h.p.

A. Analysis

In this section, we show Theorem 1. First, we calculate the number of states used by Protocol 1. Since these states of ranked agents, waiting agents, phase agents, and leader-electing agents are disjoint, the protocol uses $n + \lceil c_{\text{wait}} \cdot \log n \rceil + \lceil \log n \rceil + 2|Q_{\text{LE}}| = n + \Theta(\log n)$ states (as $|Q_{\text{LE}}| = \mathrm{O}(\log \log n)$), as claimed. It remains to show the correctness of the protocol and to calculate its runtime. The fact that the protocol is silent follows directly from the definition of the protocol.

The proof is split into two parts. First we show in Lemma 3 that after $\mathrm{O}(n \log n)$ interactions a state in C_{SR} (configurations in which the actual ranking is started) is reached. Then we show in Lemma 4 that in another $\mathrm{O}(n^2 \log n)$ interactions each agent receives a unique rank. Theorem 1 then follows directly from Lemmas 3 and 4.

Lemma 3. *W.h.p., there is a $\tau = \mathrm{O}(n \log^2 n)$ such that $\vec{X}_{t+\tau} \in C_{\text{SR}}$.*

Lemma 3 is mostly a direct consequence of the correctness of the leader election protocol.

Lemma 4. *Let c be a sufficiently large constant, and assume $\vec{X}_t \in C_{\text{SR}}$ and $c_{\text{wait}} \geq 24 + 48\gamma$. Then, there is a $\tau \in [c \cdot n^2 \log n]$ such that $\vec{X}_{t+\tau}$ is a configuration in C_{L}.*

The main idea of the proof is to show that the protocol alternates between so-called initial waiting and initial ranking configurations ($C_{k,\text{wait}}$ and $C_{k,\text{rank}}$, defined below in Definition 5). The proof of the lemma is divided into two cases. In Lemma 6 we show that, starting with an initial waiting configuration in $C_{k,\text{wait}}$, the protocol is in an initial ranking configuration after $\mathrm{O}(2^k n \log n)$ interactions. Also, starting with an initial ranking configuration $\in C_{k,\text{rank}}$, the protocol is in an initial waiting configuration after $\mathrm{O}(n^2 + 2^k n \log n)$ interactions (Lemma 7). From this Lemma 4 follows via induction over the number of phases. Note that the number of interactions in a waiting phase increases with k. This is because the one-way epidemics are restricted to the unranked agents. We prove Lemmas 6 and 7 in the remainder of this section.To state them, we need the following technical definition, with the largest rank in phase k, f_k, defined as above as $f_1 = n$ and $f_i = \lceil f_{i-1}/2 \rceil$ for $i > 1$.

Definition 5. *Let $k \in [\lceil \log_2 n \rceil]$ be a phase index.*

1) *The set of* initial waiting configurations *for phase $k = 1$, called $C_{1,\mathrm{wait}}$ is C_{SR}.*
2) *The set of* initial waiting configurations *for phase $k > 1$, called $C_{k,\mathrm{wait}}$, is the set of configurations with*
   a) *a unique waiting agent ℓ, which has* $\texttt{waitCount}(\ell) = \lceil c_{\mathrm{wait}} \cdot \log n \rceil$.
   b) *for each $i \in [f_k + 1, n]$, a unique agent u_i with* $\texttt{rank}(u_i) = i$*, and these are the only ranked agents,*
   c) $\texttt{phase}(w) \le k$ *for all phase agents,*
   d) *no leader-electing agents.*
3) *The set of* initial ranking configurations *called $C_{k,\mathrm{rank}}$ for phase k is defined as the set of configurations with*
   a) *a unique unaware leader ℓ having* $\texttt{rank}(\ell) = 1$,
   b) *for each $i \in [f_{k-1} + 1, n]$ a unique agent u_i having* $\texttt{rank}(u_i) = i$*, and these and the unaware leader are the only ranked agents,*
   c) $\texttt{phase}(w) = k$ *for all phase agents,*
   d) *no leader-electing or waiting agents.*

Lemma 6. *For any k with $1 \le k \le \lceil \log_2 n \rceil$ and any $\gamma > 0$ the following statement holds. Assume that $\vec{X}_t$ is an arbitrary configuration in $C_{k,\mathrm{wait}}$ and $c_{\mathrm{wait}} \ge 24 + 48\gamma$. Then with probability of at least $1 - 5n^{-\gamma}$ there is a $\tau \le (c_{\mathrm{wait}} + \gamma)2^k \cdot n \log n$ such that $\vec{X}_{t+\tau} \in C_{k,\mathrm{rank}}$.*

Proof. Case $k > 1$: Let ℓ be the unique waiting agent at the time t where $\vec{X}_t \in C_{k,\mathrm{wait}}$, and let T_{wait} be defined such that $t + T_{\mathrm{wait}}$ is the first time after t at which ℓ becomes ranked again. At all times in the interval $[t, t + T_{\mathrm{wait}}]$ there are $n - 1 - (n - f_k) = f_k - 1$ phase agents. This holds since there are $n - f_k$ ranked agents and one waiting agent (agent ℓ) in the population, the other agents are phase agents. Since $\texttt{waitCount}(\ell)$ is decremented every time ℓ meets a phase agent, after $\lceil c_{\mathrm{wait}} \log n \rceil$ such meetings ℓ becomes ranked again. Thus, T_{wait} has the negative binomial distribution $\mathrm{NegBin}\Big(\lceil c_{\mathrm{wait}} \log n \rceil, \frac{f_k - 1}{n(n-1)}\Big)$. From the upper and lower tail bounds (see Lemma 12 in Section A) we get

$$\Pr\left[T_{\mathrm{wait}} \le \frac{n^2}{f_k - 1} \cdot (c_{\mathrm{wait}} + \gamma) \log n\right] \ge 1 - n^{-\gamma} \text{ and } \quad (1)$$

$$\Pr\left[T_{\mathrm{wait}} > \frac{1}{4} \cdot \frac{n(n-1)}{f_k - 1} \cdot c_{\mathrm{wait}} \log n\right] \ge 1 - n^{-c_{\mathrm{wait}}/6}. \quad (2)$$

Taking into account that $f_k \ge 2$ for all k, we have $1/(f_k - 1) \le 2/f_k \le 2^k/n$, which together with the upper bound on T_{wait} above yields $T_{\mathrm{wait}} \le (c_{\mathrm{wait}} + \gamma) \cdot 2^k \cdot n \log n$.

Next we prove that with probability at least $1 - 3n^{-\gamma}$, at time $t + T_{\mathrm{wait}}$ all phase agents w have $\texttt{phase}(w) = k$ (remember, no phase agent can become ranked $t + T_{\mathrm{wait}}$). Each phase agent w switches $\texttt{phase}(w)$ to k by being prompted via a one-way epidemic spread among the phase agents. The initiator of this epidemic is the ranked agent v with $\texttt{rank}(v) = f_k$.

Consider now a modified protocol in which, as long as the epidemic did not reach all phase agents, ℓ is not allowed to assign any rank to the agents. Let T_{OWE} be defined such that $t + T_{\mathrm{OWE}}$ is the first time after t at which the one-way epidemic reaches all phase agents. Clearly, if $T_{\mathrm{OWE}} < T_{\mathrm{wait}}$, then the original and the modified protocol behave identically. For the modified protocol the upper tail bound (Lemma 14 in Section A with $m = f_k$) gives

$$\Pr\left[T_{\mathrm{OWE}} > 3\frac{n^2}{f_k} \cdot (\log(f_k) + 2\gamma \log n)\right] \le 2n^{-\gamma}. \quad (3)$$

Assuming $n \ge 2$ the above bound implies $T_{\mathrm{OWE}} \le 6 \cdot \frac{n(n-1)}{f_k - 1} \cdot (2\gamma + 1) \log n$. The assumption $c_{\mathrm{wait}} \ge 24 + 48\gamma$ ensures that this upper bound is not larger than the lower bound on T_{wait} given in Equation (2). Note that c_{wait} is also large enough to ensure $c_{\mathrm{wait}}/6 \ge \gamma$ and by union bound $\Pr[T_{\mathrm{OWE}} \le T_{\mathrm{wait}}] \ge 1 - 3n^{-\gamma}$, which concludes the proof of the case $k > 1$.

Case $k = 1$: For the analysis, we consider the following modified protocol. If two or more leaders are elected by the leader election protocol (lines 1–2 in Protocol 1), the agent u which sets $\texttt{leaderDone}(u)$ to 1 first, remains the leader, while any other elected leader v loses its leader role instead of setting $\texttt{leaderDone}(v)$ to 1. Clearly, if in the original protocol one leader is elected, then the two protocols behave identically. However, in the modified protocol we enforce that in fact at most one leader is elected. Apart from the modification described above, the two protocols have the same transition function.

Let ℓ be the unique leader elected in the modified protocol. Then, ℓ starts a one-way epidemic, which is spread among the whole population. Similarly to the case $k > 1$, define T_{wait} such that $t + T_{\mathrm{wait}}$ is the first time at which ℓ becomes ranked. Also, let T_{OWE} be defined such that $t + T_{\mathrm{OWE}}$ is the first time at which the one-way epidemic reaches all agents. We know that the waiting agent ℓ decrements its wait counter when interacting with any of the other $n - 1 = f_k - 1$ agents. Thus, also in the case $k = 1$, we obtain the same upper and lower tail bounds on T_{wait} as in Equations (1) and (2). For T_{OWE} we obtain the same bound as in Equation (3). Thus, in the modified protocol at time $t + T_{\mathrm{wait}}$ with $T_{\mathrm{wait}} \le (c_{\mathrm{wait}} + \gamma) \cdot n \log n$ we have $\vec{X}_{t+T_{\mathrm{wait}}} \in C_{1,\mathrm{rank}}$ with probability $1 - 4n^{-\gamma}$. Since the original protocol elects a single leader with probability at least $1 - n^{-\gamma}$ applying union bound concludes the proof. □

Lemma 7. *For any k with $2 \le k \le \lceil \log_2 n \rceil$ and $\gamma > 0$ the following statement holds. Assume that $\vec{X}_t$ is an arbitrary configuration in $C_{k,\mathrm{rank}}$. Then with probability at least $1 - n^{-\gamma}$ there is a $\tau \le 2n^2 + 2\gamma 2^k n \log n$ such that $\vec{X}_{t+\tau} \in C_{k+1,\mathrm{wait}}$ when $k < \lceil \log_2 n \rceil$, or $\vec{X}_{t+\tau} \in C_{\mathrm{L}}$ when $k = \lceil \log_2 n \rceil$.*

Proof. For $1 \le i \le f_k - f_{k+1}$, let $C_{k,i}$ be the set of configurations with

1) a unique unaware leader ℓ, which has $\texttt{rank}(\ell) = i$,
2) for each $i \in [f_{k+1}+1, f_{k+1}+i-1] \cup [f_k+1, n]$ a unique agent u_i with $\texttt{rank}(u_i) = i$, and these and the unaware leader are the only ranked agents,
3) $\texttt{phase}(k)$ for all phase agents, and
4) no leader-electing agents

By definition of the protocol, in any of these configurations, the only kind of interaction which will change the configuration is that between the unique unaware leader and one of the phase agents (lines 4–9 in Protocol 2). Assume for now that $\vec{X}_{t'}$ is an arbitrary configuration in $C_{k,i}$. Then if $1 \leq i < f_k - f_{k+1}$, the next configuration not in $C_{k,i}$ is in $C_{k,i+1}$. If $i = f_k - f_{k+1}$ and $k < \lceil \log_2 n \rceil$, the next configuration not in $C_{k,i}$ is in $C_{k+1,\mathrm{wait}}$; and if $i = f_k - f_{k+1}$ and $k = \lceil \log_2 n \rceil$, the next configuration not in $C_{k,i}$ is a valid ranking, i.e., in C_{L}.

Now in $C_{k,i}$ there is one unaware leader and $n-1-(n-f_k)-(i-1) = f_k - i$ phase agents, so in total $f_k - i$ out of $n(n-1)$ possible ordered interaction pairs will lead to the next class of configurations. Let $T_{k,i}$ denote the number of interactions between the first time step at which the configuration is in $C_{k,i}$ and the first time step at which the configuration leaves $C_{k,i}$. Then, $T_{k,i}$ has distribution $\mathrm{Geom}\left(\frac{f_k-i}{n(n-1)}\right)$, and the $T_{k,i}$ are independent as the corresponding time steps are disjoint.

In the following analysis we write $X \preceq Y$ when X is stochastically dominated by Y. The total time to reach $C_{k+1,\mathrm{wait}}$ if $k < \lceil \log_2 n \rceil$, or C_{L} if $k = \lceil \log_2 n \rceil$) from $C_{k,\mathrm{rank}}$ is the sum of independent geometric random variables. Since $\mathrm{Geom}(p) \preceq \mathrm{Geom}(q)$ for $p \geq q$ and by definition of the negative binomial distribution, for all k,

$$\sum_{i=1}^{f_k-f_{k+1}} T_{k,i} \sim \sum_{i=1}^{f_k-f_{k+1}} \mathrm{Geom}\left(\frac{f_k-i}{n(n-1)}\right) \preceq \sum_{i=1}^{f_k-f_{k+1}} \mathrm{Geom}\left(\frac{f_{k+1}}{n^2}\right) \sim \mathrm{NegBin}\left(f_k - f_{k+1}, \frac{f_{k+1}}{n^2}\right),$$

with the geometric random variables in the sums being independent. By stochastic domination and the tail bound on negative binomial random variables (see Lemma 12 in Section A) we thus have

$$\Pr\left[\sum_{i=1}^{f_k-f_{k+1}} T_{k,i} \leq \frac{2n^2}{f_{k+1}}(f_k - f_{k+1} + \gamma \log n)\right] \geq 1 - n^{-\gamma}.$$

Since $f_{k+1} \geq n \cdot 2^{-k}$ and $f_k - f_{k+1} \leq n2^{-k}$, this is further upper-bounded by

$$\frac{2n^2}{n2^{-k}}\left(n2^{-k} + \gamma \log n\right) = 2n^2 + 2\gamma \cdot 2^k \log n. \qquad \square$$

V. Self-Stabilizing Ranking

In this section we present our self-stabilizing ranking protocol called StableRanking, which is based on Ranking. Starting from *any* possible configuration, it ranks all agents in $\mathrm{O}(n^2 \log n)$ interactions, w.h.p. However, self-stabilization comes at a cost: it increases the memory complexity by an additive $\mathrm{O}(\log^2 n)$ number of states. The core idea behind the protocol is to run the ranking protocol from the previous section and reset it whenever we detect an error, i.e., a doubly assigned label, or if the protocol does not make any progress. To this end, StableRanking is divided into three *subprotocols*, FastLeaderElection, PropagateReset, and Ranking+. Both FastLeaderElection and Ranking+ are randomized and rely on a synthetic random coin (cf. [14]). In our state space, the coin is implemented under the variable $\texttt{coin}(v) \in \{0,1\}$ that is flipped every time the agent is activated. Intuitively speaking, in each iteration, the coin shows *heads* ($\texttt{coin}(v) = 1$) with probability roughly $1/2$, and *tails* ($\texttt{coin}(v) = 0$) otherwise after *warming up* for $\mathrm{O}(n \log\log n)$ steps. The protocol StableRanking uses the state space shown in Protocol 3.

The remainder of this section is structured as follows. We first describe the subprotocols PropagateReset, FastLeaderElection, and Ranking+ in Sections V-A to V-C, and then we present the analysis of our algorithm in Section V-D.

A. PropagateReset

First, we will briefly discuss the resetting protocol PropagateReset from [20], which we use here in an almost black-box-like manner. PropagateReset is responsible for restarting the protocol whenever an error in either Ranking+ or FastLeaderElection occurs. To be precise, whenever an agent detects an error, PropagateReset resets the agents to a configuration where all agents start to elect a leader. Each agent $v \in V$ has two counters $\texttt{resetCount}(v) \in [0, R_{\mathsf{max}}]$ and $\texttt{delayCount}(v) \in [0, D_{\mathsf{max}}]$ where $R_{\mathsf{max}}, D_{\mathsf{max}} \in \Theta(\log n)$. We will fix the values of R_{max} and D_{max} in our analysis.

The protocol works as follows. Based on the counters $\texttt{resetCount}(v)$ and $\texttt{delayCount}(v)$, the agents are divided into three classes. If $\texttt{resetCount}(v) = \bot$, we say the agent is *computing* (meaning it executes FastLeaderElection or Ranking+), and if $\texttt{resetCount}(v) > 0$, we say the agent is *propagating*. Finally, if $\texttt{delayCount}(v) > 0$ and $\texttt{resetCount}(v) = 0$, we say the agent is *dormant*.

If a computing agent v has to restart the protocol, it sets $\texttt{resetCount}(v) = R_{\mathsf{max}}$ and all its other variables except for $\texttt{coin}(v)$ to $\bot$. If $\texttt{coin}(v) \neq \bot$, the value of $\texttt{coin}(v)$ is maintained; otherwise, we initialize $\texttt{coin}(v)$ to 0. We call v the *triggered* agent, a configuration containing a triggered agent a *triggered configuration* and the set of all of these configurations C_{T}. We will write TriggerReset(v) for a routine triggering a reset for v as described.

A triggered agent v starts a one-way epidemic that will eventually turn all computing agents into dormant agents as follows. If a propagating agent v interacts with computing agent w, it decreases its $\texttt{resetCount}(v)$ by 1 and w becomes propagating by setting $(\texttt{resetCount}(w), \texttt{delayCount}(w)) = (\texttt{resetCount}(v), D_{\mathsf{max}})$ and all other values (except the coin) to $\bot$. If two propagating agents v and w interact, they set $\texttt{resetCount}(v)$ and $\texttt{resetCount}(w)$ to $\max\{\texttt{resetCount}(v), \texttt{resetCount}(w)\} - 1$ (unless both

Protocol 3 StableRanking(u, v).

This protocol uses the following set of states, where $\uplus$ is the disjoint union of two sets:

$$Q = \overset{\textsc{Rank}}{\underbrace{[n]}_{\texttt{rank}}} \uplus \overset{\textsc{Coin}}{\underbrace{\{0,1\}}_{\texttt{coin}}} \times \Big(\overset{\textsc{PropagateReset}}{\underbrace{\underbrace{[\Theta(\log n)]}_{\texttt{resetCount}} \times \underbrace{[\Theta(\log n)]}_{\texttt{delayCount}}}_{Q_{\mathsf{Reset}}}} \uplus \overset{\textsc{FastLE}}{\underbrace{\underbrace{[\Theta(\log^2 n)]}_{Q_{\mathrm{SLE}}}}_{Q_{\mathsf{LeaderElect}}}} \uplus \overset{\textsc{Ranking+}}{\underbrace{\underbrace{[\Theta(\log n)]}_{\texttt{aliveCount}} \times \underbrace{[\Theta(\log n)]}_{\text{non-}\texttt{rank}\text{ state(s) from }\textsc{Ranking}}}_{Q_{\mathsf{Main}}}} \Big)$$

```
execute PropagateReset($u, v$)                                ▷ if applicable, propagate resets and transition into leader election
if $\texttt{leaderDone}(u) \neq \bot \neq \texttt{leaderDone}(v)$ then
    execute ElectLeader($u, v$)                               ▷ elect leader, handle leader's transition to waiting
if $\texttt{leaderDone}(w) \neq \bot$ and $X(x) \in Q_{\mathrm{Main}}$ for a $\{w, x\} = \{u, v\}$    ▷ agent executing leader election meets agent executing main protocol
    set all of $w$'s state except $\texttt{coin}(w)$ to $\bot$          ▷ $w$ forgets its leader election states and becomes a phase agent
    $\texttt{phase}(w), \texttt{aliveCount}(w) \leftarrow 1, L_{\mathsf{max}}$
if $X(u) \in Q_{\mathrm{Main}}$ and $X(v) \in Q_{\mathrm{Main}}$        ▷ when two agents having main states interaction, execute the extended ranking protocol
    execute Ranking+($u$, $v$)
if $\texttt{coin}(v) \neq \bot$ then                                   ▷ toggle $v$'s coin if it has one
    $\texttt{coin}(v) \leftarrow 1 - \texttt{coin}(v)$
```

are 0). Finally, if a propagating agent v interacts with dormant agent, it decreases $\texttt{resetCount}(v)$ by 1 and if a dormant agent v interacts with an arbitrary agent, it decreases $\texttt{delayCount}(v)$ by 1.

As soon as an agent w reaches $\texttt{delayCount}(w) = 0$, it forgets the state associated with PropagateReset and initializes the state of the leader election protocol, where the value of $\texttt{coin}(w)$ is maintained.

B. FastLeaderElection

Unfortunately, we cannot use the leader election protocol by Gasieniec and Stachowiak [30] for our self-stabilizing algorithm. In a self-stabilizing setting we need to cope with bad initializations resulting in no leader being elected. A simple way of dealing with this is by attaching a *timer* $\texttt{LECount}(v)$ to each agent. The timer is initialized to the maximal number L_{max} of interactions that are w.h.p. required by any agent in the protocol—for [30], this is $\Theta(\log^2 n)$. Agents in the leader election phase decrement $\texttt{LECount}(v)$ whenever they interact. If an agent's timer ever reaches 0, it triggers a reset. However, this simple trick blows up the state space by a multiplicative factor of L_{max}. For the above-mentioned protocol [30], this would result in a state space of size $\mathrm{O}(\log^2 n \log\log n)$, which is slightly too large. To mitigate this, we present a simple and fast protocol FastLeaderElection (similar to the lottery game of [2]). In a nutshell, the protocol works as follows. Again, each agent v stores two bits $\texttt{isLeader}(v)$ and $\texttt{leaderDone}(v)$ that denote whether they are the leader and finished with the protocol execution. Recall that all unranked agents, even the *dormant* agents, have a variable called $\texttt{coin}(v)$ that is *flipped* on each activation. An agent v will declare itself to be the leader ($\texttt{isLeader}(v) = 1$) if it observes $\lceil \log n \rceil$ *heads* ($\texttt{coin}(w) = 1$) on its interaction partners in a row. This requires each agent to only store $\lceil \log n \rceil$ bits, one for each coin flip. With constant probability, there is *exactly one* agent that archives this, i.e., there is exactly one leader. This leader starts a broadcast that lets all agents start the ranking protocol just as before. Note that, if there are two or more agents, this will trigger a reset within $\mathrm{O}(n^2)$ interactions w.h.p., as this produces many duplicate labels. Furthermore, to avoid being stuck in a configuration without leaders, we use the aforementioned variable $\texttt{LECount}(v)$ to count each agent's interactions. If it reaches 0 on one agent before it starts ranking, the agent also triggers a reset. As detailed description of the protocol including the pseudocode can be found in Section C.

C. Ranking+

Similarly to Ranking, each agent can have exactly one value of $\texttt{waitCount}(v)$, $\texttt{phase}(v)$, or $\texttt{rank}(v)$ *not* equal to $\bot$. We again call agents having one of these values waiting agents, phase agents, and ranked agents, respectively. A leader ℓ alternates between the states waiting ($\texttt{waitCount} > 0$) and being unaware leader (there is an agent v with $\texttt{rank}(\ell) \leq n \cdot 2^{\texttt{phase}(v)}$). The counter $\texttt{aliveCount}$ we be used to check if the protocol is still making progress. The protocol is defined in Protocol 4.

At its core, Ranking+ extends Ranking by triggering a reset (via PropagateReset) as soon as one of the three following errors occurs: First, two agents have the same rank, which is detected when they interact directly (line 1). Second, more than two agents are waiting, which is also detected via direct interaction (line 2). Third, the protocol cannot assign more ranks, As we will show later, in this case $\texttt{aliveCount}(v)$ will reach 0 for at least one agent (line 9).

Note that it is unfortunately not always possible to detect if there are two or more leaders. However, unless they "acci-

Protocol 4 RANKING+(u, v).

This protocol uses the following set of *main states*, written Q_{Main}, where $\uplus$ is the disjoint union of two sets:

$$Q_{\text{Main}} = \underbrace{\{1,\dots,n\}}_{\texttt{rank}} \uplus \underbrace{\{0,1\}}_{\texttt{coin}} \times \underbrace{\{1,\dots,L_{\mathsf{max}}\}}_{\texttt{aliveCount}} \times \left(\underbrace{\{1,\dots,c_{\text{wait}} \cdot \log n\}}_{\texttt{waitCount}} \uplus \underbrace{\{1,\dots,\lceil \log n \rceil\}}_{\texttt{phase}} \right).$$

```
error detection
1  if $\texttt{rank}(u) = \texttt{rank}(v) \neq \bot$        ▷ if $u$ and $v$ have the same rank or two waiting agents meet, trigger a reset and do nothing else
2          or $(\texttt{waitCount}(u) \neq \bot$ and $\texttt{waitCount}(v) \neq \bot)$ then
3      execute TRIGGERRESET($u$)
4      return
liveness checking
5  if $\texttt{aliveCount}(u) \neq \bot \neq \texttt{aliveCount}(v)$ then        ▷ if both $u$ and $v$ check liveness, adopt maximum counter minus one
6      $\texttt{aliveCount}(u), \texttt{aliveCount}(v) \leftarrow \max\{\texttt{aliveCount}(u), \texttt{aliveCount}(v)\} - 1$
7  if $\texttt{rank}(u) \in \{n-1, n\}$ and $\texttt{aliveCount}(v) \neq \bot$ then        ▷ when meeting an agent ranked $\geq n-1$, decrement counter if present
8      $\texttt{aliveCount}(v) \leftarrow \texttt{aliveCount}(v) - 1$
9  if $\texttt{aliveCount}(v) = 0$ then        ▷ if the counter hits zero, trigger a reset and do nothing else
10     execute TRIGGERRESET($u$)
11     return
12 if $\texttt{coin}(v) = 0$ then        ▷ if $v$'s coin is 0, reset the counter if we *could* have made progress
13     if $\texttt{waitCount}(u) \neq \bot$ or $\left(\texttt{rank}(u) \neq \bot \neq \texttt{phase}(v)\right.$ and $\left.\texttt{rank}(u) \leq \left\lfloor n \cdot 2^{-\texttt{phase}(v)} \right\rfloor\right)$ then
14         $\texttt{aliveCount}(v) \leftarrow \lceil c_{\text{live}} \cdot \log n \rceil$
base protocol
15 else if $\texttt{coin}(v) = 1$ then        ▷ if $v$'s coin is 1, execute the base protocol
16     execute RANKING($u, v$)
17     if $u$ became waiting in the RANKING protocol then
18         $(\texttt{coin}(u), \texttt{aliveCount}(u)) \leftarrow (0, L_{\mathsf{max}})$
```

dentally" produce a correct ranking, this case will ultimately result in one of the three kinds of errors above being produced and detected.

It remains to explain how RANKING+ detects the third error. The idea of $\texttt{aliveCount}(v)$ is that it will reach the value 0 whenever the protocol does not make any progress. The variable $\texttt{aliveCount}$ is decremented for two reasons. First, whenever two unranked agents meet, they update their $\texttt{aliveCount}$ to the maximum of their respective counts minus 1 (lines 5–6). Second, an unranked agent u also decrements $\texttt{aliveCount}(u)$ when encountering an agent ranked $n-1$ or n (lines 7–8). This is necessary to reduce $\texttt{aliveCount}$ in the case where u is the only unranked agent. If there is no interaction that resets $\texttt{aliveCount}(v)$, the counter will eventually reach 0.

To explain how RANKING+ actually detects the third error we distinguish between two cases: the leader is either waiting or an unaware leader. In the first case, whenever the waiting leader ℓ encounters a phase agent v, it resets $\texttt{aliveCount}(v)$. In the second case, ideally we would like to reset the counter to its maximum value whenever the unaware leader ℓ assigns a new rank to an agent v. Unfortunately, neither ℓ nor v have state space left to store $\texttt{aliveCount}$. To circumvent this problem we do the following. Whenever a ranked agent u interacts with an unranked agent v, agent u will determine if it is the unaware leader ℓ (by checking if $\texttt{rank}(u) \leq n \cdot 2^{\texttt{phase}(v)}$). If the inequality is fulfilled u decides between either assigning a rank to v (if $\texttt{coin}(v) = 1$, lines 15–16) or setting $\texttt{aliveCount}(v)$ to L_{max} (if $\texttt{coin}(v) = 0$, lines 12–14).

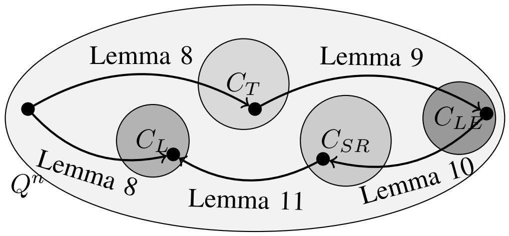

Fig. 1. A high-level overview of the self-stabilizing algorithm.

D. Analysis

In this section we show Theorem 2. Similarly to the proof of Theorem 1 we define a set of configurations which are safe entry states of the subprotocols of our protocol.

The set C_{LE} contains the configurations from which we can safely start FASTLEADERELECTION. We call this set leader election configurations. The set C_{T} contains the triggered configurations. These are configurations in which we start

PropagateReset. The set $C_{\mathrm{SR}+}$ contains safe ranking configurations defined as the configurations from which we can safely call Ranking+. Finally, C_{L} contains all legal configurations in which all agents have a unique rank. The following proof of Theorem 2 essentially tracks movement of the protocol through these sets until a configuration in C_{L} is reached, see Figure 1 for an overview of these sets.

Proof of Theorem 2. For the state space, see that the three subprotocols have $\mathrm{O}(\log^2 n)$ overhead states. Together with the coin and the ranks, we have $|Q| = n + \mathrm{O}(\log^2 n)$ as required.

The analysis of the correctness and running time of the protocol is split into Lemmas 8 to 11. In Lemma 8, we show that when $\vec{X}_t \notin C_L$ is an arbitrary configuration, within $\mathrm{O}(n^2 \log n)$ interactions the protocol either reaches configuration in C_{T} or C_{L}, w.h.p. Lemma 9 shows that, when the protocol is an arbitrary configuration of C_{T}, it reaches a configuration from C_{LE} within $\mathrm{O}(n \log n)$ interactions, w.h.p. Lemma 10 shows that, when the protocol is in an arbitrary configuration of C_{LE}, it reaches a configuration from $C_{\mathrm{SR}+}$ within $\mathrm{O}(n^2 \log n)$ interactions, w.h.p. Finally, Lemma 11 shows that from a configuration in $C_{\mathrm{SR}+}$, the protocol will reach a correct ranking configuration C_{L} in $\mathrm{O}(n^2 \log n)$ interactions, w.h.p., directly implying the correctness and running time.

We conclude with the protocol's closure. When the protocol is in a legal configuration $\vec{X}_t \in C_L$, all pairs of agents u and v have distinct ranks. In this case, they do not change their states. This can be seen by inspecting the protocol as two ranked agents only perform an action when they have equal rank. Therefore, $\vec{X}_{t+1} = \vec{X}_t$ and the protocol fulfills the closure property and is silent. □

Lemma 8. *Let c_1 be a sufficiently large constant, and assume that $\vec{X}_t$ is an arbitrary configuration not in C_{L}. Then, w.h.p., there is a $\tau \leq c_1 \cdot n^2 \log n$ such that either $\vec{X}_{t+\tau} \in C_{\mathrm{L}}$ or one the configurations $\vec{X}_t, \dots, \vec{X}_{t+\tau}$ contains a triggered agent (i.e., is in C_{T}).*

Proof sketch. First, we show that after a *preparation phase* having $\mathrm{O}(n \log^2 n)$ rounds, the protocol will be in a configuration where all agents are in a main state (i.e., either waiting agents, phase agents, or ranked agents). The remainder of the proof is then divided into two parts, both using a potential function. This potential function allows us to treat the various ways to make progress (assigning ranks, advancing saved phases, or resetting) in a unified manner without excessive case distinctions.

We call a pair of agents $u \neq v$ a *productive pair* if it fulfills the condition in line 13 of Protocol 4. That is, the protocol *could* make progress if the phase agent's coin shows 1. Our potential function Φ_t is then defined as 0 if there is no productive pair or there is a resetting agent in $\vec{X}_t$ and as $\sum_{v \in [n]\colon \mathtt{phase}_t(v) \neq \bot} 2^{-\mathtt{phase}_t(v)}$ otherwise.

In the first part of the proof we show that the potential drops to zero within $\mathrm{O}(n^2 \log n)$ rounds w.h.p. In the second part of the proof we then show that once the potential has reached zero, the protocol is either in a configuration in C_{L} or will reset within $\mathrm{O}(n^2 \log n)$ further interactions w.h.p. This directly implies Lemma 8.

For the first part of the proof we define a notion of *good time steps*, for which we can show an expected drop of the potential Φ in $\Omega(\Phi/n^2)$, leading to a geometric decay with decay factor $1 - \Omega(n^{-2})$. There are multiple ways in which a time step can be good. A time step is good if there is a directly detectable error, i.e., there are two agents having the same rank, or two waiting agents. In this case there is at least an n^{-2} probability of detection, in which case the potential immediately drops to zero. A time step is also good if an agent can increase its saved phase by some interaction. If this is the case, it must be true in particular for an agent v with the lowest saved phase. Letting s be the number of phase agents in the configuration, v contributes at least Φ/s to the potential, and this contribution will at least halve when v increases its phase. By case analysis, one can see that the probability of this occurring is in $\Omega(s/n^2)$. s cancels out between the probability and the drop in potential in this event, and we get the desired drop in potential in expectation. Lastly, a time step is good if an agent can be ranked in some interaction, and a sufficient proportion of phase agents have their coin showing 1 (so that they can actually get ranked). Assuming we are not in the previous case as well, *all* phase agents have the same phase and may be ranked in an interaction. So an agent gets ranked with probability in $\Omega(s/n^2)$. As each phase agent contributes exactly Φ/s to the potential, and this potential contribution drops to 0 when it ceases to be a phase agent, we get the desired drop in potential as in the previous case.

Finally, we need to show that there are enough good time steps within a time interval of size $\mathrm{O}(n^2 \log n)$. This involves the analysis of the synthetic coin among phase agents. Because this subpopulation shrinks (due to agents getting ranked) and can become quite small (even $o(\log n)$), and agents are removed from the subpopulation depending on the current value of the coin, we cannot use established techniques for this analysis. The full proof can be found in Section D-A.

For the second part of the proof, we need to show that if the potential Φ has reached zero but the protocol is not in a configuration in C_{L}, a reset is triggered within $\mathrm{O}(n^2 \log n)$ interactions. The challenge is to show that the protocol triggers a reset if there are no productive pairs. Here, we proceed by case distinction: either there are two agents with the same rank, a single unranked agent, or multiple unranked agents. In the proof we use an argument adapted from [20, Lemma 3.3], which, in turn, is adapted from [5, Lemma 1]. □

Lemma 9. *Let c_2 be a sufficiently large constant, and assume that $\vec{X}_t$ is an arbitrary configuration in C_{T}, i.e., containing a triggered agent. Then, w.h.p., there is a $\tau \leq c_2 \cdot n \log n$ such that $\vec{X}_{t+\tau} \in C_{\mathrm{LE}}$.*

*Proof sketch.*The lemma follows more or less directly from the correctness of the PropagateReset protocol [20], we also need to ensure that the synthetic coin used by the leader election is sufficiently "warmed up" by the time the reset has

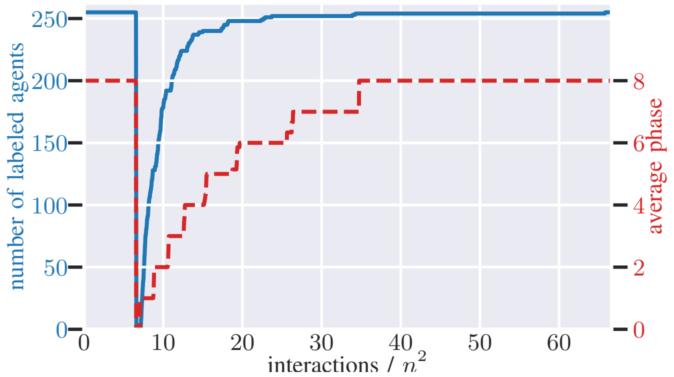

Fig. 2. Number of ranked agents (blue), and average of the phase counters stored by unranked agents (red, dashed), as a function of the number of interactions. The protocol (for $n = 256$) is initialized as follows: 255 agents are ranked (with ranks $2, \dots, 256$), and one agent is a phase agent with maximum liveness counter.

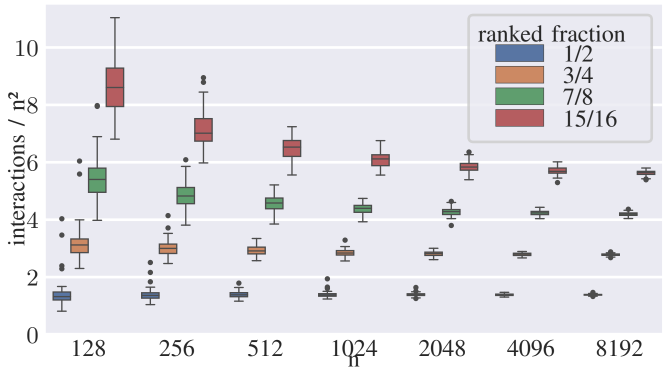

Fig. 3. Number of interactions (normalized by n^2) needed to reach a state in which $c \cdot n$ agents are ranked, starting from the following configuration: there is one agent in rank 1 (the unaware leader), and all other agents are still in a leader election state. We performed 100 simulations per value of $n \in \{2^i \mid i \in \mathbb{N}, 7 \leq i \leq 13\}$.

run its course. However, this can be achieved by letting the agents be dormant long enough. The detailed proof is given in Section D-B.

Lemma 10. *Let c_3 be a sufficiently large constant, and assume that $\vec{X}_t$ is an arbitrary configuration in C_{LE}. Then, w.h.p., there is a $\tau \leq c_3 \cdot n^2 \log n$ such that $\vec{X}_{t+\tau} \in C_{\mathrm{SR+}}$.*

Proof sketch. Here we have to show that, from an arbitrary leader election configuration, we reach a safe ranking configuration. Note that our simple leader election protocol has a constant failure probability (it can elect none or several leader). If the leader election protocol does not elect a leader is starts over again and we have to how that w.h.p. this does not happen too often. If several leaders are selected, all these leaders start the first phase of the ranking algorithm. Thus, there will be agents receiving the same rank. This will be detected within $\mathrm{O}(n^2)$ interactions by RANKING+ resulting in a reset. Details of the proof can be found in Section D-C.

Lemma 11. *Let c_4 be a sufficiently large constant, and assume that $\vec{X}_t$ is an arbitrary configuration in $C_{\mathrm{SR+}}$. Then, w.h.p., there is a $\tau \leq c_3 \cdot n^2 \log n$ such that $\vec{X}_{t+\tau} \in C_{\mathrm{L}}$.*

Proof sketch. This proof follows along the lines of the analysis of the non-self-stabilizing protocol in Section IV-A. The main added difficulty is to show that w.h.p. no agent v reaches a state where $\mathtt{aliveCount}(v) = 0$ which would inadvertently trigger a reset. Similarly to the proof of Lemma 8, this requires an analysis of the synthetic coin among the shrinking and potentially tiny subpopulation of phase agents, and additionally the analysis of one-way epidemics in the same setting. The detailed proof is given in Section D-D.

VI. Simulation Results

We implemented a simulation of our population protocol in Rust (with $c_{\text{wait}} = 2$ and $c_{\text{live}} = D_{\mathsf{max}}/\log_2(n) = 4$). In Figure 2 we show how the protocol resets and then quickly resumes assigning ranks starting from an invalid initialization. The chosen initialization can be considered to be worst-case as it needs $\Theta(n^2 \log n)$ interactions to reset (in expectation). The figure shows that most of the runtime is taken up by ranking the final few agents, with successive phases taking increasingly longer. This is to be expected, as the process is a coupon collection process. Accordingly, it should take about as long to rank half the agents as it takes to rank the next quarter, the next eight, and so on. Figure 3 confirms this; there, we consider the number of interactions to rank constant fractions of agents for various n and fractions. After $\Theta(n^2)$ interactions, constant fractions of agents are ranked, much faster than the $\Theta(n^2 \log(n))$ interactions needed to rank *all* agents.

VII. Conclusion

We present a self-stabilizing protocol for the population protocol model that solves the ranking problem. Our protocol is silent and requires $\mathrm{O}(n^2 \log n)$ interactions w.h.p. using $n+\mathrm{O}(\log^2 n)$ states. It is an open question to solve the ranking problem either in $\Theta(n^2)$ interactions in expectation using $n + \mathrm{O}(\log(n))$ states or in $\Theta(n^2 \log n)$ interactions w.h.p. using $n+o(\log n)$ states. Another question is if it is possible to improve on the $\Theta(\log^2 n)$ overhead for self-stabilizing leader election while keeping the number of iterations at $\mathrm{O}(n^2 \log n)$. Finally, it is also open whether time-optimal (non-silent) protocols exist that use only subexponentially many states, improving upon the protocol by Burman et al. [20].

References

[1] D. Alistarh, J. Aspnes, K. Censor-Hillel, S. Gilbert, and M. Zadimoghaddam, “Optimal-time adaptive strong renaming, with applications to counting,” in *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC*, 2011, pp. 239–248.

[2] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, and R. L. Rivest, “Time-space trade-offs in population protocols,” in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2017, pp. 2560–2579.

[3] D. Alistarh, J. Aspnes, and R. Gelashvili, “Space-optimal majority in population protocols,” in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018, pp. 2221–2239.

[4] D. Alistarh, H. Attiya, S. Gilbert, A. Giurgiu, and R. Guerraoui, “Fast randomized test-and-set and renaming,” in *Distributed Computing, 24th International Symposium, DISC*, 2010, pp. 94–108.

[5] D. Alistarh, B. Dudek, A. Kosowski, D. Soloveichik, and P. Uznanski, "Robust detection in leak-prone population protocols," in *DNA Computing and Molecular Programming - 23rd International Conference, DNA*, 2017, pp. 155–171.

[6] D. Alistarh and R. Gelashvili, "Polylogarithmic-time leader election in population protocols," in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP, Part II*, 2015, pp. 479–491.

[7] D. Alistarh and R. Gelashvili, "Recent algorithmic advances in population protocols," *SIGACT News*, vol. 49, no. 3, pp. 63–73, 2018.

[8] D. Alistarh, R. Gelashvili, and J. Rybicki, "Fast graphical population protocols," in *25th International Conference on Principles of Distributed Systems, OPODIS*, 2021, 14:1–14:18.

[9] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta, "Computation in networks of passively mobile finite-state sensors," *Distributed Comput.*, vol. 18, no. 4, pp. 235–253, 2006.

[10] D. Angluin, J. Aspnes, and D. Eisenstat, "Fast computation by population protocols with a leader," *Distributed Comput.*, vol. 21, no. 3, pp. 183–199, 2008.

[11] J. Beauquier, J. Burman, L. Rosaz, and B. Rozoy, "Non-deterministic population protocols," in *Principles of Distributed Systems, 16th International Conference, OPODIS*, 2012, pp. 61–75.

[12] P. Berenbrink, A. Brinkmann, R. Elsässer, T. Friedetzky, and L. Nagel, "Randomized renaming in shared memory systems," *J. Parallel Distributed Comput.*, vol. 150, pp. 112–120, 2021.

[13] P. Berenbrink, G. Giakkoupis, and P. Kling, "Optimal time and space leader election in population protocols," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2020, pp. 119–129.

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

[15] P. Berenbrink, D. Kaaser, and T. Radzik, "On counting the population size," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC*, 2019, pp. 43–52.

[16] A. Bilke, C. Cooper, R. Elsässer, and T. Radzik, "Brief announcement: Population protocols for leader election and exact majority with $o(\log^2 n)$ states and $o(\log^2 n)$ convergence time," in *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC*, 2017, pp. 451–453.

[17] A. Brodsky, F. Ellen, and P. Woelfel, "Fully-adaptive algorithms for long-lived renaming," *Distributed Comput.*, vol. 24, no. 2, pp. 119–134, 2011.

[18] J. Burman, J. Beauquier, and D. Sohier, "Space-optimal naming in population protocols," in *33rd International Symposium on Distributed Computing, DISC*, 2019, 9:1–9:16.

[19] J. Burman, H. Chen, H. Chen, D. Doty, T. Nowak, E. E. Severson, and C. Xu, *Time-optimal self-stabilizing leader election in population protocols*, 2019. arXiv: 1907.06068 [cs.DC].

[20] J. Burman, H. Chen, H. Chen, D. Doty, T. Nowak, E. E. Severson, and C. Xu, "Time-optimal self-stabilizing leader election in population protocols," in *PODC '21: ACM Symposium on Principles of Distributed Computing*, 2021, pp. 33–44.

[21] S. Cai, T. Izumi, and K. Wada, "How to prove impossibility under global fairness: On space complexity of self-stabilizing leader election on a population protocol model," *Theory Comput. Syst.*, vol. 50, no. 3, pp. 433–445, 2012.

[22] L. Cardelli and A. Csikász-Nagy, "The cell cycle switch computes approximate majority," *scientific reports*, vol. 2, no. 656, pp. 1–9, 2012.

[23] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig, "Programmable chemical controllers made from dna," *nature nanotechnology*, vol. 8, no. 10, pp. 755–762, 2013.

[24] G. Di Luna and G. Viglietta, "Computing in anonymous dynamic networks is linear," in *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2022, pp. 1122–1133.

[25] G. Di Luna and G. Viglietta, "Efficient computation in congested anonymous dynamic networks," in *49th International Symposium on Mathematical Foundations of Computer Science, MFCS*, 2024, 49:1–49:19.

[26] R. Elsässer and T. Radzik, "Recent results in population protocols for exact majority and leader election," *Bull. EATCS*, vol. 126, 2018.

[27] L. Gasieniec, J. Jansson, C. Levcopoulos, and A. Lingas, *Efficient assignment of identities in anonymous populations*, 2021. arXiv: 2105.12083 [cs.DC].

[28] L. Gasieniec, J. Jansson, C. Levcopoulos, and A. Lingas, "Efficient assignment of identities in anonymous populations," in *25th International Conference on Principles of Distributed Systems, OPODIS*, 2021, 12:1–12:21.

[29] L. Gasieniec, L. Kuszner, E. Latif, R. Parasuraman, P. Spirakis, and G. Stachowiak, *Anonymous distributed localisation via spatial population protocols*, 2024. arXiv: 2411.08434 [cs.DC].

[30] L. Gasieniec and G. Stachowiak, "Fast space optimal leader election in population protocols," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018, pp. 2653–2667.

[31] L. Gasieniec, G. Stachowiak, and P. Uznanski, "Almost logarithmic-time space optimal leader election in population protocols," in *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA*, 2019, pp. 93–102.

[32] G. Giakkoupis and P. Woelfel, "On the time and space complexity of randomized test-and-set," in *ACM Symposium on Principles of Distributed Computing, PODC*, 2012, pp. 19–28.

[33] T. Hagerup and C. Rüb, "A guided tour of chernoff bounds," *Inf. Process. Lett.*, vol. 33, no. 6, pp. 305–308, 1990.

[34] D. R. Kowalski and M. A. Mosteiro, "Time and communication complexity of leader election in anonymous networks," in *41st IEEE International Conference on Distributed Computing Systems, ICDCS*, 2021, pp. 449–460.

[35] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017, ISBN: 978-1-107-15488-9.

[36] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck, "Computation with finite stochastic chemical reaction networks," *Nat. Comput.*, vol. 7, no. 4, pp. 615–633, 2008.

[37] Y. Sudo, R. Eguchi, T. Izumi, and T. Masuzawa, "Time-optimal loosely-stabilizing leader election in population protocols," in *35th International Symposium on Distributed Computing, DISC*, 2021, 40:1–40:17.

[38] Y. Sudo, J. Nakamura, Y. Yamauchi, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely-stabilizing leader election in a population protocol model," *Theor. Comput. Sci.*, vol. 444, pp. 100–112, 2012.

[39] Y. Sudo, F. Ooshita, T. Izumi, H. Kakugawa, and T. Masuzawa, "Time-optimal leader election in population protocols," *IEEE Trans. Parallel Distributed Syst.*, vol. 31, no. 11, pp. 2620–2632, 2020.

[40] Y. Sudo, F. Ooshita, H. Kakugawa, T. Masuzawa, A. K. Datta, and L. L. Larmore, "Loosely-stabilizing leader election with polylogarithmic convergence time," *Theor. Comput. Sci.*, vol. 806, pp. 617–631, 2020.

Appendix A
Tail Bounds

First, we give a tail bound for the negative binomial distribution.

Lemma 12. *Let $X \sim \mathrm{NegBin}(r, p)$ have negative binomial distribution with parameters $r \geq 1$ and $p \in (0, 1)$.*

1) *For $\gamma > 0$ and $n \geq 1$, $\Pr\Big[X > \frac{1}{p} \cdot 2 \cdot (r + \gamma \log n)\Big] \leq n^{-\gamma}$.*
2) $\Pr\Big[X \leq \frac{1}{2} \cdot \frac{r}{p}\Big] \leq \exp\big(-\frac{r}{6}\big)$.

Next, we give a bound on the running time of coupon collection. We let $\mathrm{CouponCollector}(n)$ be the distribution of the number of trials needed in the coupon collector's problem with n coupons. The following lemma is a standard upper tail bound on this distribution.

Lemma 13 (cf., e.g., Section 3.3.1 in [35])**.** *Let $1 \leq k \leq n$, and $\gamma > 0$. Then for $X \sim \mathrm{CouponCollector}(k)$ we have*

$$\Pr[X > k(\log(k) + \gamma \log n)] \leq n^{-\gamma}.$$

Finally we give upper and lower bounds for the running time of one-way epidemics based on Lemma 2 from [10]. Write $\mathrm{OWE}(n, m)$ for the distribution of the number of interactions needed to perform a one-way epidemic among a subset of m agents in a total population of n agents, where one of the m agents is initially infected.

Lemma 14. *For $X \sim \mathrm{OWE}(n, m)$ for $2 \leq m \leq n$, and all $\gamma > 0$, we have*

$$\Pr\left[X > 3\frac{n^2}{m} \cdot (\log(m) + 2\gamma \log(n))\right] \leq 2n^{-\gamma}.$$

Appendix B
Omitted Proofs of Section IV

In this appendix we present the omitted proofs from Section IV. We first state a lemma that establishes the correctness of leader election based on the protocol by Gasieniec and Stachowiak [30].

Lemma 15 (cf. Lemma 2.4 in [15], citing [30])**.** *There is a population protocol using $\mathrm{O}(\log\log n)$ states w.h.p. electing a unique leader in $\mathrm{O}(n \log^2 n)$ interactions w.h.p.: after at most $\mathrm{O}(n \log^2 n)$ interactions, there is, w.h.p., an agent ℓ with* $\texttt{leaderDone}(\ell) = 1$ *and* $\texttt{isLeader}(\ell) = 1$*, and at that time, w.h.p., all other agents $v \neq \ell$ have* $\texttt{isLeader}(u) = 0$.

We now give the full technical proofs for the lemmas from Section IV.

Proof of Lemma 3. Recall that at time $t = 0$, all agents v start with $\mathtt{q}_{\mathtt{LE}}(v) = q_0$ where q_0 is the initial state of the leader election protocol as described by [15], and with $\texttt{leaderDone}(v) = 0$. Whenever two agents with $\mathtt{q}_{\mathtt{LE}}(v) \neq \perp$ interact, they follow the transition function of the leader election (see lines 1–2 of Protocol 1). As long as no agent ℓ transitions into a state with $\texttt{isLeader}(\ell) = 1$ and $\texttt{leaderDone}(\ell) = 1$, these are the only transitions taking place, since the one-way epidemic setting $\mathtt{q}_{\mathtt{LE}}(v)$ and $\texttt{leaderDone}(v)$ to $\perp$ (see lines 7–9) requires an initially infected agent, which is ℓ (see lines 3–5). By Lemma 15, w.h.p. this takes place at a time τ with $\tau = \mathrm{O}(n \log^2 n)$; the leader ℓ is then immediately transitioned into the state where $\texttt{waitCount}(\ell) = \lceil c_{\mathrm{wait}} \cdot \log n \rceil$ in the same interaction (line 5). And at that point, by Lemma 15, all other agents $v \neq \ell$ have $\texttt{isLeader}(v) = 0$. So after the interaction at time τ, the configuration fulfills all the requirements of a safe ranking configuration, yielding the claim. □

Proof of Lemma 4. We show for any constant $\gamma > 0$ that, assuming that $\vec{X}_t$ is an arbitrary configuration in C_{SR}, there is, with probability at least $1 - \mathrm{O}(\log n \cdot n^{-\gamma})$, a $\tau \leq 4(c_{\mathrm{wait}} + 5\gamma + 1)n^2 \lceil \log_2 n \rceil$ such that $\vec{X}_{t+\tau} \in C_{\mathrm{L}}$, which implies the claim.

First, we show by induction that for all $1 \leq k \leq \lceil \log_2 n \rceil =: k_{\max}$, with probability at least $p_k = 1 - 6(k-1)n^{-\gamma}$ there is a $\tau \leq \tau_{k,\max} = 4(k-1)n^2 + (c_{\mathrm{wait}} + 5\gamma) n \log n \cdot \sum_{k'=1}^{k-1} 2^k$ such that $\vec{X}_{t+\tau} \in C_{k,\mathrm{wait}}$. For $k = 1$, the statement is true with $\tau_k = 0$ since $\vec{X}_t \in C_{\mathrm{SR}} = C_{1,\mathrm{wait}}$ by assumption and definition of $C_{1,\mathrm{wait}}$.

So assume the statement is true for some $k < k_{\max}$. Then with probability at least p_k, there is a $\tau_k \leq \tau_{k,\max}$ such that $\vec{X}_{t+\tau} \in C_{k,\mathrm{wait}}$. Assuming this is the case, by Lemma 6, with probability at least $1 - 5n^{-\gamma}$ there is a $\tau' \leq (c_{\mathrm{wait}} + \gamma) 2^k n \log n$ such that $\vec{X}_{t+\tau+\tau'} \in C_{k,\mathrm{rank}}$. And assuming that *this* is the case, there is, by Lemma 7, with probability at least $1 - n^{-\gamma}$, a $\tau'' \leq 2n^2 + 2\gamma 2^k n \log n$ such that $C_{k+1,\mathrm{wait}}$ (since $k < k_{\max}$). So overall by the union bound, with probability at least $p_k - 6n^{-\gamma} = p_{k+1}$, there is a $\tau''' = \tau + \tau' + \tau'' \leq \tau_{k,\max} + 4n^2 + (c_{\mathrm{wait}} + 5\gamma) 2^k n \log n = \tau_{k+1,\max}$ such that $\vec{X}_{t+\tau'''} \in C_{k+1,\mathrm{wait}}$.

Using the statement proven by induction and analogously applying Lemmas 6 and 7 a final time for $k = k_{\max}$, we see by the union bound that with probability at least $1 - 6k_{\max} n^{-\gamma}$, there is a τ with $\vec{X}_{t+\tau} \in C_{\mathrm{L}}$, where

$$\tau \leq \tau_{k_{\max},\max} + 4n^2 + (c_{\mathrm{wait}} + 5\gamma) 2^{k_{\max}} n \log n = 4k_{\max} n^2 + (c_{\mathrm{wait}} + 5\gamma) n \log n \cdot \sum_{k=1}^{k_{\max}} 2^k.$$

Protocol 5 FASTLEADERELECTION(u, v):

The protocol uses the following state space.

$$Q_{\mathrm{LE}} = \underbrace{\{1, \ldots, L_{\mathsf{max}}\}}_{\texttt{LECount}} \times \underbrace{\{1, \ldots, \lceil \log n \rceil\})}_{\texttt{coinCount}} \times \underbrace{\{0,1\}}_{\texttt{leaderDone}} \times \underbrace{\{0,1\}}_{\texttt{isLeader}}$$

```
──────────── leader election phase ────────────
LECount(u) ← LECount(u) − 1
if coin(v) = 0 then leaderDone(u) ← 1                   ▷ if random coin is 0, u will not be leader
if leaderDone(u) = 1 then return                        ▷ if leaderDone(u) is 1, do nothing
if coinCount(u) > 0 then
    coinCount(u) ← coinCount(u) − 1                     ▷ u counts coins with value 1
else                                                    ▷ u observed ⌈log n⌉ coins with value 1
    isLeader(u) ← 1                                     ▷ u becomes leader
    leaderDone(u) ← 1                                   ▷ u stops looking further
──────────── transition to main phase ────────────
if LECount(u) ≥ L_max/2 and isLeader(u) = 1 then        ▷ leader was elected fast enough
    LECount(u), leaderDone(u), isLeader(u), coinCount(u) ← ⊥
    (waitCount(u), aliveCount(u)) ← (⌈c_wait log n⌉, L_max)   ▷ u starts main phase as waiting leader
    return
if LECount(u) = 0 then                                  ▷ leader was not elected fast enough
    LECount(u), leaderDone(u), isLeader(u), coinCount(u) ← ⊥
    execute TRIGGERRESET(u)                             ▷ trigger a reset
```

Now as $2^{k_{\max}} = 2^{\lceil \log_2 n \rceil} < 2n$, and the geometric series sums to $\sum_{k=1}^{k_{\max}} 2^k = 2^{k_{\max}+1} - 1 < 2 \cdot 2^{k_{\max}} < 4n$, we have

$$\tau \leq 4n^2 \lceil \log_2 n \rceil + 4(c_{\text{wait}} + 5\gamma) n^2 \log n \leq 4(c_{\text{wait}} + 5\gamma + 1) n^2 \lceil \log_2 n \rceil,$$

as claimed. □

APPENDIX C
DESCRIPTION OF FASTLEADERELECTION

In this section, we describe FASTLEADERELECTION, a simple protocol that elects a leader with constant probability and otherwise triggers reset. The protocol and its state space are formally defined in Protocol 5. Each agent $v \in V$ has a counter $\texttt{LECount}(v) \in [0, L_{\mathsf{max}}]$, a counter $\texttt{coinCount}(v) \in [\lceil \log n \rceil]$, and two flags $\texttt{leaderDone}(v) \in \{0,1\}$ and $\texttt{isLeader}(v) \in \{0,1\}$ as variables. Slightly abusing notation, Q_{LE} is again the state space used by the leader election. Just as R_{max} and D_{max}, we will bound $L_{\mathsf{max}} = c_{\text{live}} \cdot \log n$ for some $c_{\text{live}} > 0$ in the analysis. As the bounds for the other variables are fixed, it holds $|Q_{\mathrm{LE}}| \in \mathrm{O}(\log^2 n)$.

Recall that all unranked agents, even the *dormant* agents, have a variable called $\texttt{coin}(v)$ that is flipped on each activation. In a nutshell, the protocol works as follows: an agent $v \in V$ will declare itself to be the leader ($\texttt{isLeader}(v) = 1$) if it observes $\lceil \log n \rceil$ *heads* ($\texttt{coin}(w) = 1$) in a row. Furthermore, to avoid being stuck in a configuration without leaders, we use $\texttt{LECount}(v)$ to count each agent interactions. If it reaches 0, the protocol triggers a reset.

More precisely, the protocol uses the following interactions. Whenever an agent $v \in V$ interacts with another agent $w \in V$, the following happens. If $\texttt{leaderDone}(v) = 1$, v has already decided if it is a leader and will not consider the coin. Otherwise, it observes its coin $\texttt{coin}(w)$. If $\texttt{coin}(w) = 0$, it sets $\texttt{leaderDone}(v) = 1$. If $\texttt{coin}(w) = 1$, it decrements $\texttt{coinCount}(v)$ by 1. If $\texttt{coinCount}(v)$ reaches 0, v has seen $\lceil \log n \rceil$ *heads* in a row, and becomes leader. To this end, it sets $\texttt{isLeader}(v) = 1$ and $\texttt{leaderDone}(v) = 1$. Finally, v decrements $\texttt{LECount}(v)$ by 1. If it reaches $L_{\mathsf{max}}/2$ and v is the leader, v assumes that it is the unique leader. Therefore, it transitions to the main protocol by turning into a waiting agent. This will start a one-way epidemic that lets all other agents enter a state from Q_{Main}. Finally, if $\texttt{LECount}(v)$ reaches 0 and v is not a leader, v assumes that no leader was elected. Therefore, it triggers a reset.

We assume that every agent v starts the protocol in a state $q_{0,i}$ for $i \in \{0,1\}$ where

$$(\texttt{LECount}(v), \texttt{coinCount}(v), \texttt{leaderDone}(v), \texttt{isLeader}(v), \texttt{coin}(v)) = (L_{\mathsf{max}}, \lceil \log n \rceil, 0, 0, \perp),$$

and all other fields are $\perp$.

Appendix D
Omitted Proofs from Section V-D

A. Proof of Lemma 8

In this section, we show that starting from an *arbitrary* configuration, the protocol will either reach a correct ranking ($\in C_{\mathrm{L}}$) or trigger a reset within $\mathrm{O}(n^2 \log n)$ interactions. To this end, we divide the execution of the protocol into two phases, the *preparation phase* and the *main phase*. With Q_{Main} being the main states defined at the beginning of Section V-C, we let C_{Main} be the set of configurations where all agents have a state in Q_{Main}, calling them *main configurations*. We say that the protocol is in the main phase when it is in a main configuration. We let C_{Prep} be the complement of C_{Main}, i.e., the set of configurations where at least one agent has state not in Q_{Main}, calling these *prep configurations*. We say that the protocol is in the prep phase when it is in a main configuration. As C_{Main} and C_{Prep} partition all configurations by definition, so the system is always in one of the two phases. The following observation will be useful throughout.

Observation 16. *Assuming that $\vec{X}_t$ is an arbitrary configuration in C_{Main}, $\vec{X}_{t+1}$ will either contain a triggered agent or be in C_{Main}.*

First, we will show that we quickly, i.e., within only $\mathrm{O}(n \log^2 n)$ interactions, leave the preparation phase and start with the main phase.

Lemma 17. *Let c be a sufficiently large constant, and assume $\vec{X}_t \in C_{\mathrm{Prep}}$. Then with probability $1 - \mathrm{O}(\frac{1}{n})$, there is a $\tau \in [c \cdot n \log^2 n]$ such that either $\vec{X}_{t+\tau} \in C_{\mathrm{Main}}$ or that the protocol resets at time $t + \tau$.*

Proof. If there are still propagating agents, by the properties of the reset protocol, these agents will eventually become dormant. Thus, within $\mathrm{O}(nR_{\mathsf{max}})$ steps, we will reach a configuration where all agents are dormant, or in a state of Q_{LE} of FastLeaderElection, or in a state from Q_{Main}. Note that any interaction with a dormant or electing agent with an agent from Q_{Main} will change both agent's states to a state from Q_{Main}. Thus, if one agent is in Q_{Main}, all agents will be in Q_{Main} after $\mathrm{O}(n \log n)$ steps. Therefore, it remains to show that either one agent enters Q_{Main} or triggers a reset. Note that all dormant agents decrease their $\mathtt{delayCount}(v)$ by one on every interaction. Thus, within $\mathrm{O}(nD_{\mathsf{max}})$ steps, we will reach a configuration where all agents are in a state of Q_{LE} of FastLeaderElection. All leader-electing agents decrease their $\mathtt{LECount}(v)$ by one on every interaction. Thus, within $\mathrm{O}(nL_{\mathsf{max}})$ steps, they will either trigger a reset or one agent becomes leader and switches to Q_{Main} (which triggers an epidemic that turns all agents to Q_{Main}). □

Recall from the proof sketch of Lemma 8 that we call a pair of agents $u \neq v$ a *productive pair* if it fulfills the condition in line 13 of Protocol 4 (ignoring the coin), i.e., if it is a pair where the protocol *could* make progress if the phase agent's coin shows 1. There are hence two ways for a pair of agents to be productive. Either u may assign a rank to v when interacting (ignoring the valid of $\mathtt{coin}(v)$ here), i.e., when $\mathtt{rank}_t(u) \neq \bot \neq \mathtt{phase}_t(v)$, and $\mathtt{rank}_t(u) \leq \lfloor n \cdot 2^{-\mathtt{phase}_t(v)} \rfloor$, in which case we also call it a *rank-assigning pair*; or u is waiting and v has a phase, i.e., $\mathtt{waitCount}_t(u) \neq \bot \neq \mathtt{phase}_t(v)$. Recall that we define the potential Φ_t as 0 if there is no productive pair or there is a resetting agent in $\vec{X}_t$ and as

$$\Phi_t = \sum_{v \in [n]\colon\ \mathtt{phase}_t(v) \neq \bot} 2^{-\mathtt{phase}_t(v)}$$

otherwise.

Our main lemma is two-fold. The first part states that the potential will drop to 0 within $\mathrm{O}(n^2 \log n)$ interactions w.h.p. when it is currently in a main configuration. We prove this part in Section D-A1. The second part states that once the potential has hit 0, the protocol will either reach a stable configuration or reset within a further $\mathrm{O}(n^2 \log n)$ interactions w.h.p. We prove that part in Section D-A2.

Lemma 18. *Let c be a sufficiently large constant independent of c_{wait} or c_{live}.*

1) *Assume $\vec{X}_t$ is an arbitrary configuration in C_{Main}. Then w.h.p. there is a $\tau \leq c \cdot c_{\mathrm{wait}} \cdot n^2 \log n$ such that $\Phi_{t+\tau} = 0$.*
2) *Assume $\vec{X}_t$ is an arbitrary configuration where $\Phi_t = 0$. Then w.h.p. there is a $\tau \leq c \cdot c_{\mathrm{live}} \cdot n^2 \log n$ such that either $\vec{X}_{t+\tau}$ is stable ($\in C_{\mathrm{L}}$) or contains a resetting agent.*

1) Proof of Lemma 18, Part 1: Potential Drops Quickly:

For our proof, we need the following definition of *good time steps*; we show below that the expected value of the potential will decay geometrically by a factor of $1 - \Omega(n^{-2})$ in good time steps. Recall that a waiting agent v ($\mathtt{waitCount}(v) \neq \bot$) is a "leader" which is currently (supposed to be) waiting out a phase transition, and that above we called a pair of agents $u \neq v$ *rank-assigning* when $\mathtt{rank}(t)u \neq \bot$, $\mathtt{phase}(t)v \neq \bot$, and $\mathtt{rank}_t(u) \leq \lfloor n \cdot 2^{-\mathtt{phase}_t(v)} \rfloor$.

Definition 19. *A time step t is* good *if $\vec{X}_t \in C_{\mathrm{Main}}$ and one of the following statements holds.*

1) *There is a duplicate rank, i.e., there are agents $u \neq v$ such that $\mathtt{rank}_t(u) = \mathtt{rank}_t(v) \neq \bot$.*

2) *There are two (or more) waiting agents, i.e., there are agents $u \neq v$ such that* $\texttt{waitCount}_t(u) \neq \bot$ *and* $\texttt{waitCount}_t(v) \neq \bot$.
3) *There is a pair of agents where upon interacting, one of the agent's phases will increase. I.e., there are agents u, v with* $\texttt{phase}_t(v) \neq \bot$, *and either* $\texttt{phase}_t(v) < \texttt{phase}_t(u) \neq \bot$, *or* $\texttt{rank}_t(u) = f_k$ *for a* $k > \texttt{phase}_t(v)$.
4) *There is a rank-assigning pair; and for all agents u, v with* $\texttt{phase}_t(u) \neq \bot$ *and* $\texttt{phase}_t(v) \neq \bot$, *we have* $\texttt{phase}_t(u) = \texttt{phase}_t(v)$*; and at least a quarter of all phase agents have* $\texttt{coin}(v) = 1$.

We write $\mathcal{G}_t$ for the event that t is a good time step.

Recall that we assume that $\vec{X}_t \in C_{\text{Main}}$. Then since Φ takes non-negative integers as values, and by Markov's inequality, we have

$$\Pr\left[\bigvee_{\tau \in [c \cdot n^2 \log n]} \Phi_{t+\tau} = 0\right] = \Pr\left[\min_{\tau \in [c \cdot n^2 \log n]} \Phi_{t+\tau} = 0\right] \leq \mathbb{E}[\min_{\tau \in [c \cdot n^2 \log n]} \Phi_{t+\tau}].$$

So it is sufficient to show that this last expected value is in $\mathrm{O}(n^{-1})$.

First, we show that the potential Φ_t exhibits a multiplicative drop in expectation whenever a round is good.

Lemma 20. *For any t', $0 < \phi \in \mathbb{N}$,*

$$\mathbb{E}[\Phi_{t'+1} \mid \Phi_{t'} = \phi, \mathcal{G}_{t'}] \leq \left(1 - \frac{1}{4n^2}\right) \cdot \phi.$$

Proof. First, note that since we assume that the time step is good, we are in a main configuration, so that no phase agent can decrease its phase, and no non-phase agent can become a phase agent. And as a consequence, we have $\Phi_{t'+1} \leq \Phi_{t'}$.

We proceed by case distinction over the four alternatives by which a round can be good, and show that in each case, the potential decreases by a factor $1 - 1/(4n^2)$ in expectation.

Cases 1 and 2. When there is a duplicate rank at time t', there is at least a $\frac{1}{n(n-1)} \geq n^{-2}$ chance of two agents with the same rank interacting, in which case they will initiate a reset and the potential drops to 0. The same holds when there are two (or more) waiting agents. So conditioning on time t' being good for either of those reasons, the expected value of $\Phi_{t'}$ is at most

$$\frac{1}{n^2} \cdot 0 + \left(1 - \frac{1}{n^2}\right) \cdot \phi \leq \left(1 - \frac{1}{4n^2}\right) \cdot \phi.$$

Case 3. In this case, there is at least one pair of agents where upon interacting, one of the agents will increase its phase; w.l.o.g., we may assume that this is an agent having the minimum currently saved phase. Let s be the number of phase agents, and let $\ell \geq 1$ be the number of agents having the minimum currently saved phase. Now if $\ell \geq s/2$, the probability of one of the ℓ agents increasing its phase is at least $\ell/(n(n-1)) \geq s/(2n^2)$ (since if one of those ℓ agents can increase its phase in an interaction, all of them can). Otherwise, if $\ell < s/2$, there are at least $s/2$ agents saving a phase $\geq \ell$, so there is also at least an $\ell \cdot s/2/(n(n-1)) \geq s/(2n^2)$ chance of an interactions where on of the ℓ agents will increase its phase. Since the potential contribution of an agent is decreasing in its phase, the ℓ agents each have an above-average contribution to the potential, and their potential contribution will drop by at least one half when their phase increases. Hence, the expected value of $\Phi_{t'}$ in this case is at most

$$\frac{s}{2n^2} \cdot \phi \cdot \left(1 - \frac{1}{2s}\right) + \left(1 - \frac{s}{2n^2}\right) \cdot \phi = \phi - \phi \cdot \frac{s}{4sn^2} = \left(1 - \frac{1}{4n^2}\right) \cdot \phi.$$

Case 4. The final case is that in which t' is good because there is a rank-assigning pair, no agent can increment its phase (and hence all saved phases are equal), and at least a quarter of phase agents has $\texttt{coin}_{t'}(v) = 1$. Since all s phase agents save the same phase and there is a rank-assigning pair u, v, there must in fact be at least s such pairs. Assume w.l.o.g. that $\texttt{rank}_{t'}(u) \neq \bot$ and $\texttt{phase}_{t'}(v) \neq \bot$; since all saved phases are equal, it holds for all v' with $\texttt{phase}_{t'}(v') \neq \bot$ that $\texttt{phase}_{t'}(v') = \texttt{phase}_{t'}(v)$, and hence u, v' is also a rank-assigning pair. Since at least a quarter of phase agents has $\texttt{coin}_{t'}(v) = 1$, we know that there are at least $s/4$ pairs of agents which, when interacting, would lead to a phase agent to become ranked (and hence no longer a phase agent). So conditioning on this case, and using the fact that as all saved phases are equal, each phase agent has equal contribution to the potential, the expected value of $\Phi_{t'}$ is at most

$$\frac{s}{4n(n-1)} \cdot \phi \cdot \left(1 - \frac{1}{s}\right) + \left(1 - \frac{s}{4n(n-1)}\right) \cdot \phi = \phi - \phi \cdot \frac{s}{4sn(n-1)} \leq \left(1 - \frac{1}{4n^2}\right)\phi.$$

Since we have seen that this holds in all four cases, we are done. □

To see that the expected minimum value of the potential over $\mathrm{O}(n^2 \log n)$ rounds is small, we need a lower bound on the number of good time steps in such an interval.

To that end, the following lemma considers the number B_t of time steps which are *not* good in the time interval starting with t and ending when $\Phi = 0$. To be precise, letting $T = \min\{t' \geq t \mid \Phi_{t'} = 0\}$ be the next time after t where the potential is 0, we let

$$B_t = \sum_{t'=t}^{T-1} \mathbb{1}\{t' \text{ is good}\}.$$

We show $B_t = \mathrm{O}(n^2 \log n)$ w.h.p.; hence, to ensure that there are $\Omega(n^2 \log n)$ *good* time steps in expectation, a time interval of $\mathrm{O}(n^2 \log n)$ will indeed suffice.

Lemma 21. *There is a sufficiently large $c > 0$ and a sufficiently small $0 < c' < 1$ such that such that for any t, $\Pr\big[B_t \leq c \cdot (1 + c_{\text{wait}}) \cdot n^2 \log n + (1 - c')(T - t)\big] \geq 1 - \mathrm{O}(n^{-2})$.*

Proof sketch. Assume that a time step $\tau \in [t, T)$ is *not* good. Since $\tau \in [t, T)$, we know that the protocol has not reset in the interval $[t, \tau)$. Since furthermore τ is not good, there is at most one waiting agent, there are no duplicate ranks, no interaction increasing a phase agent's saved phase, and one of the following is true:

- either less than a quarter of phase agents' coins show heads,
- or there is *no* rank-assigning pair; but since there is still a productive pair (as $\Phi_\tau > 0$ since $\tau < T$), there must be a single waiting agent (since there cannot be two).

For both cases, we need to bound the expected number of time steps where the two cases hold separately.

Let us consider the first case. We divide the time steps after time t into non-overlapping epochs of $2n$ time steps each. By Lemma 39, for each of these time steps, if the protocol is in C_{Main} at the beginning of the epoch (which it will be unless a reset is triggered by Observation 16), there is a constant probability p of either there being a reset in the epoch or there being at least a quarter of phase agents' coins showing heads for at least a constant fraction c of the epoch's time steps. Hence, when considering at least $c'n$ epochs for some large constant c' depending on p, a Chernoff bound guarantees that either there is a reset in these epochs or for at least a constant fraction of time steps during the epochs, at least a quarter of phase agents' coins showing heads. Since we do not make any guarantees about there possibly being less than $cn^2 \log n$ non-good time steps for a sufficiently large c, the assumption on the number of epochs is safe.

Now for the second case. Let t be a time where this case occurs, but where it hasn't occurred in the previous time step. We bound the number of steps until the unique waiting agent transitions out of a waiting state (or a reset is triggered). Until this happens, besides a reset triggering, the only change that can occur (besides liveness checker values or coins changing) is that the waiting agent decreases its counter by one in an interaction with a phase agent. Since there is no phase-increasing interaction, all phase agents must be saving the same phase, let this be k. Since there is no productive pair, there is no agent u with $\bot \neq \mathtt{rank}_t(u) \leq \lfloor n \cdot 2^{-k} \rfloor$; and since there are no duplicate ranks, there can be at most $n - \lfloor n \cdot 2^{-k} \rfloor$ ranked agents. So there must be at least $\lfloor n \cdot 2^{-k} \rfloor - 1$ phase agents (because there is one waiting agent as well); However, there is at least one phase agent, since otherwise, there would be no productive pair and $\Phi_t = 0$. So the time until the waiting agent transitions out of a waiting state (or there is a reset) is stochastically dominated by a negative binomial random variable $\mathrm{NegBin}(r, p)$ with $r = \lceil c_{\text{wait}} \cdot \log n \rceil$ and $p = \max\{1, \lfloor n \cdot 2^{-k} \rfloor - 1\}/n^2)$. By Lemma 12, with probability at least $1 - 2n^{-\gamma}$ this is at most

$$\frac{2}{p} \cdot (r + 2 \log n) \leq \frac{n^2}{\max\{1, n \cdot 2^{-k} - 1\}} \cdot (\lceil c_{\text{wait}} \log n \rceil + 2 \log n) \leq c(c_{\text{wait}} + 1) n 2^k \log n$$

for sufficiently large n and some constant c. Now if this case occurs at most once for each possible $k \in [\lceil \log_2 n \rceil]$, this would give a total time of $c(c_{\text{wait}} + 1) n \log n \sum_{k=1}^{\lceil \log_2 n \rceil} 2^k$, and with the sum being at most $4n$ (see the proof of Lemma 4 in Section B), we have the claimed time. Otherwise, consider the first time that the case occurs a second time for some value of k. Then w.h.p. after the time bound above, the waiting agent reaches rank 1, and at that time all phase agents will save phase k, meaning that formerly waiting agent is an unaware leader, and will assign rank $f_{k+1} + 1$ on the next interaction with a phase agent. But this rank will already have been assigned after the first time this case was encountered for this value of k. And since all saved phases are at least k and cannot decrease without going through a reset, this rank persisted, and there are now two agents having the same rank. So within at most $\mathrm{Geom}(1/(n(n-1)))$ rounds, these agents will interact and trigger a reset (if a reset doesn't occur before then). As this is in $\mathrm{O}(n^2 \log n)$ w.h.p., we are done.

Since these two cases cover all relevant bad time steps, this proves the claim. □

We are now ready to show that $\mathbb{E}[\min_{\tau \in [cn^2 \log n]} \Phi_{t+\tau}] = \mathrm{O}(n^{-1})$, as required.

Let $T_{G,i}$ be the ith time step after t which is good. By the law of total expectation, Write $\Phi_{t,t'} := \min_{t \leq \tau \leq t'} \Phi_\tau$ for the minimum of Φ_τ over the time interval $[t, t']$. Clearly, $\Phi_{t,t'}$ is monotonically non-increasing in t', and non-negative. Furthermore, note that $\Phi_{t'+1} > \Phi_{t'}$ iff $\Phi_{t,t'} = 0$, because the protocol was in a main configuration at time t, and for the potential to increase, a non-phase agent needs to become a phase agent or a phase agent must decrease its phase, which can only happen after a reset. Hence, also, $\Phi_{t'} > 0$ if and only if $\Phi_{t,t'} > 0$, and in that case, $\Phi_{t'} = \Phi_{t,t'}$. So

$$\begin{aligned}\mathbb{E}[\Phi_{t,T_{G,i}+1}] &\leq \sum_{\phi>0} \Pr\bigl[\Phi_{T_{G,i}} = \phi\bigr] \cdot \left(\Pr[\mathcal{G}_{T_{G,i}} \mid \Phi_{T_{G,i}} = \phi] \cdot \left(1 - \frac{1}{4n^2}\right)\phi + \Pr[\neg\mathcal{G}_{T_{G,i}} \mid \Phi_{T_{G,i}} = \phi] \cdot \phi\right) \\ &= \sum_{\phi>0} \Pr\bigl[\Phi_{T_{G,i}} = \phi\bigr]\left(1 - \frac{1}{4n^2}\right)\phi = \left(1 - \frac{1}{4n^2}\right) \cdot \sum_{\phi\geq 0} \Pr\bigl[\Phi_{t,T_{G,i}} = \phi\bigr]\phi = \left(1 - \frac{1}{4n^2}\right)\mathbb{E}[\Phi_{t,T_{G,i}}].\end{aligned}$$

Applying this repeatedly and using the monotonicity of $\Phi_{t,t'}$ in t', we obtain

$$\mathbb{E}[\Phi_{t,T_{G,i}+1}] \leq \left(1 - \frac{1}{4n^2}\right)^i \mathbb{E}[\Phi_{t,t}] \leq \exp\left(-\frac{i}{4n^2}\right) \cdot n,$$

so that

$$\mathbb{E}[\Phi_{t,T_{G,8n^2 \log n}}] \leq \exp(-2\log n) \cdot n = n^{-1}.$$

Finally, Lemma 21 implies that

$$\Pr\bigl[T_{G,6n^2 \log n} \leq t + (8 + c \cdot c_{\text{wait}})n^2 \log n\bigr] \geq 1 - n^{-2}.$$

So that indeed

$$\Pr\left[\min_{\tau\in[(8+c\cdot c_{\text{wait}})n^2 \log n]} \Phi_{t+\tau} > 0\right] \leq n^{-1} + n^{-2}.$$

2) Proof of Lemma 18, Part 2: Recall that we assume that $\Phi_t = 0$. So by definition of Φ, either the protocol is resetting at time t, or there are no productive pairs at time t. If the protocol is resetting at time t, the proof of this part of Lemma 18 is already done with $\tau = 0$, so only the latter case remains. In the following we call a non-legal configuration with no productive pairs a *dead configuration*.

Observation 22. *Assume the protocol is in a dead configuration at time t. Then from time t until the protocol resets (if ever), all of the following hold:*

1) *All waiting agents remain waiting,*
2) *no ranked agent changes its rank, and*
3) *there are no productive pairs, i.e., the configuration remains dead.*

Lemma 23. *Assume that at time t, there are no productive pairs and that the configuration is not stable, i.e., we are in a dead configuration. Then w.h.p. within $\mathrm{O}(n^2 \log n)$ interactions, the protocol will reset.*

The proof of this lemma follows from the following lemmas.

Lemma 24. *Let c be a sufficiently large constant. Assume the protocol is in a main configuration without productive pairs, but with duplicate ranks, at time t. Then w.h.p. there is a time $\tau \in [t, t + c \cdot n^2 \log n]$ such that the protocol resets at time τ.*

Proof. According to Observation 22, none of the ranked agents will change its state unless the protocol resets. Let u and v be two agents with the same rank. According to Protocol 4 if u and v are selected for interaction, then the protocol resets (unless there was a reset between step t and the time step in which u and v meet). We know that in each time step u and v are selected with probability $2/(n(n-1))$, independently of the agents selected for interaction in any other step. Clearly, u and v do not interact in $c \cdot n^2 \log n$ time steps with probability

$$\left(1 - \frac{2}{n(n-1)}\right)^{c\cdot n^2 \log n} < \frac{1}{n^2}$$

whenever c is large enough. This implies that w.h.p. the protocol resets in some time step $\tau \in [t, t + c \cdot n^2 \log n]$. □

Lemma 25. *Let c be a sufficiently large constant. Assume the protocol is in a main configuration without productive pairs, and with a single agent without a rank, at time t. Then w.h.p. there is a time $\tau \in [t, t + c \cdot c_{\text{live}} \cdot n^2 \log n]$ such that the protocol resets at time τ.*

Proof. Let u be the agent without a rank. According to Observation 22, $\texttt{waitCount}(u) \neq \perp$ or $\texttt{phase}(u) \neq \perp$ in all steps $\tau \geq t$ until the protocol resets.

As we only have one agent without a rank, there must be two agents with the same rank, or at least one of the ranks $n-1$ or n are assigned at time t. If there are two agents with the same rank, Lemma 24 implies that w.h.p. there is a time $\tau \in [t, t + c \cdot n^2 \log n]$ such that the protocol resets at time τ. Otherwise, agent u decrements $\texttt{aliveCount}(u)$ every time step in which it interacts with an agent with rank $n-1$ or n. We know that u interacts with such an agent in a time step with probability at least $2/(n(n-1))$, independently of any other time step. Thus, applying Chernoff bounds [33], we obtain that with probability at least $1-n^{-2}$, $c_{\text{live}} \cdot \log n + 1$ such interactions will occur within $c \cdot c_{\text{live}} \cdot n^2 \log n$ steps, if c is large enough (c_{live} is the constant specified in Protocol 4). This implies that within $c \cdot c_{\text{live}} \cdot n^2 \log n$ time steps, $\texttt{aliveCount}(u)$ reaches 0 w.h.p., leading to a reset according to Protocol 4. □

Lemma 26. *Let c be a sufficiently large constant. Assume the protocol is in a main configuration without productive pairs, and with two or more agents without a rank, at time t. Then w.h.p. there is a time $\tau \in [t, t + c \cdot c_{\text{live}} \cdot n^2 \log n]$ such that the protocol resets at time τ.*

Proof. We adapt a proof from [20, Lemma 3.3], which, in turn, is adapted from [5, Lemma 1].

For an agent v define $C_\tau(v)$ to be $-\infty$ if $\texttt{aliveCount}(v) = \bot$ at time τ or if the protocol resets at a time between t and τ; otherwise let $C_\tau(v)$ be the value of $\texttt{aliveCount}(v)$ at time τ. Furthermore, define $\Gamma_\tau(v) = 3^{C_\tau(v)}$ (thus $\Gamma_\tau(v) = 0$ if $C_\tau(v) = -\infty$), and let $\Gamma_t = \sum_{v \in V} \Gamma_t(v)$. As long as there are no productive pairs, the only way the values $C_\tau(v)$ (and thus Γ_τ) can change is if there is a reset in step τ (in which case $\Gamma_{\tau+1}$ becomes 0) or if two agents with $\texttt{aliveCount}(\cdot) \neq \bot$ (i.e., two unranked agents) interact in step τ.

Now let $k \geq 2$ be the number of agents without a rank at time t. Due to Observation 22 all these agents remain unranked until the protocol resets. The probability that two agents without a rank, say v and u, interact at a given time (before a reset) is $\frac{k(k-1)}{n(n-1)}$. In such an interaction, they both reset their $\texttt{aliveCounts}$ to the maximum of the $\texttt{aliveCounts}$ of u and v, minus one. Then,

$$\Gamma_{\tau+1}(u) + \Gamma_{\tau+1}(v) = 2 \cdot 3^{\max\{C_\tau(u), C_\tau(v)\}-1} \leq \frac{2}{3} \max\{3^{C_\tau(u)}, 3^{C_\tau(v)}\} \leq \frac{2}{3} \cdot (\Gamma_\tau(u) + \Gamma_\tau(v)).$$

Conditioned on the event that in some step τ two unranked agents are chosen for interaction, for two arbitrary but fixed unranked agents u, v we have $\Gamma_{\tau+1} \leq \Gamma_\tau - (\Gamma_\tau(u) + \Gamma_\tau(v))/3$ with probability $2/(k(k-1))$. Then,

$$\mathbb{E}[\Gamma_{\tau+1}] \leq \left(1 - \frac{(k-1)k}{n(n-1)}\right) \cdot \mathbb{E}[\Gamma_\tau] + \frac{(k-1)k}{n(n-1)} \cdot \left(\mathbb{E}[\Gamma_\tau] - \frac{2}{3k} \cdot \mathbb{E}[\Gamma_\tau]\right) \leq \left(1 - \frac{2k-2}{3n^2}\right) \cdot \mathbb{E}[\Gamma_\tau].$$

The value of Γ is always at most $n \cdot n^{2c_{\text{live}}}$, so in $c \cdot c_{\text{live}} n^2/(k-1) \cdot \log n \leq c \cdot c_{\text{live}} n^2 \log n$ time steps, the expected value of Γ will decrease below $n^{-c'}$ for any predefined constant c' if c is large enough. Applying now Markov's inequality, we obtain the lemma. □

B. Proof of Lemma 9

Recall that we need to show that when $\vec{X}_t \in C_{\text{T}}$ is a triggered configuration, the protocol will enter a leader-electing configuration ($\in C_{\text{LE}}$, see Section D-C below) within $\mathrm{O}(n \log n)$ interactions.

The following lemma, which describes the behavior of PROPAGATERESET, uses the notion of an *awakening configuration*, which is the first partially computing configuration reachable from a fully dormant configuration.

Lemma 27 (Corollary of Theorem 3.4 in [19]). *Let $R_{\mathsf{max}} = 60 \ln n$ and $D_{\mathsf{max}} = \Omega(\log n + R_{\mathsf{max}})$. Starting from a triggered configuration*[1]*, we reach an awakening configuration in $\Theta(D_{\mathsf{max}} n)$ interactions*[2] *with probability at least $1 - \mathrm{O}(1/n)$.*

Now recall that even while PROPAGATERESET is running, responding agents flip their coin on every interaction, and that our definition of C_{LE} requires that the difference in numbers between coins showing 1 and 0 is at most $\frac{n}{4 \log n}$. As PROPAGATERESET takes at least $\Omega(n D_{\mathsf{max}})$ interactions, and $D_{\mathsf{max}} = \Omega(\log n)$ with the leading constant being our choice, the following lemma shows that this coin property indeed holds with sufficient probability. As the proof of the Lemma is entirely analogous to that found in [14] (replacing occurrences of $\log \log \log n$ with $\log \log n$), we omit it here.

Lemma 28 (cf. Lemma 3 in [14]). *Let $\gamma > 0$ and consider an interaction t with $n \log(4 \log n)/2 \leq t \leq n^\gamma$. Then the number of coins that equal zero at the beginning of interaction t lies with probability at least $1 - n^{-\gamma}$ in $(1 \pm 1/(4 \log n)) \cdot n/2$.*

C. Proof of Lemma 10

We start from a configuration that results from executing PROPAGATERESET protocol. Intuitively, these are all configurations the population is ready to execute FASTLEADERELECTION. To be precise, all agents are either dormant (and wait to start the protocol) *or* are in the initial state of the leader election. Furthermore, the values of the flip bits have converged to a distribution such that roughly half of all agents have either bit. In other words, all agents have been dormant long enough for the bit to settle, and first, the agent has just woken up from being dormant. Formally, we define these configurations as follows.

[1] Note that this is called a “partially-triggered configuration in [19].

[2] In [19, Theorem 3.4], this is just an upper bound. However, as the $\texttt{delayCount}$ of some agent has to decrease from D_{max} to 0, and $D_{\mathsf{max}} = \Omega(\log n)$ with a sufficiently large constant, it indeed also takes at least $\Omega(n D_{\mathsf{max}})$ interactions for this to occur.

Definition 29 (C_{LE})**.** *In a leader electing configuration $\vec{X} \in C_{\mathrm{LE}}$, all agents are either dormant, i.e., it holds* $\texttt{delayCount}(v) \geq 0$*, or are in an initial state $q_{0,i}$ ($i \in \{0,1\}$) for* FASTLEADERELECTION*, i.e., their variables have the following values*

$$(\texttt{isLeader}(v), \texttt{leaderDone}(v), \texttt{coinCount}(v), \mathit{LECount}(v), \texttt{coin}(v)) = (0, 0, \lceil \log n \rceil, L_{\mathsf{max}}, i).$$

Furthermore, the following (global) property holds for the agents' coins:

$$\big| |\{v \in V \mid \texttt{coin}(v) = 1\}| - |\{v \in V \mid \texttt{coin}(v) = 0\}| \big| \leq \frac{n}{4 \log n}.$$

Starting from such a configuration, the population will reach configuration with a *unique* leader with constant probability.

Lemma 30 (Prob. for Unique Leader)**.** *Suppose that the following (global) property holds for the agents' coins:*

$$\big| |\{v \in V \mid \texttt{coin}(v) = 1\}| - |\{v \in V \mid \texttt{coin}(v) = 0\}| \big| \leq \frac{n}{4 \log n}.$$

Then, with probability greater than $^1/_{8e}$, there is exactly one agent that sets $\texttt{isLeader}(v) = 1$.

Proof. Fix an agent $v \in V$ and let $L_v \in \{0,1\}$ be the indicator that v is a leader and $U_v \in \{0,1\}$ be the indicator that v is the unique leader. Note that $\log n \leq \lceil \log n \rceil \leq \log 2n \leq 2 \log n$. Then, the probability that v becomes a leader is lower bounded by

$$\Pr[L_v = 1] \geq \left(\left(1 - \frac{1}{4 \log n}\right) \frac{1}{2} \right)^{\lceil \log n \rceil} \geq \left(1 - \frac{1}{4 \log n}\right)^{\lceil \log n \rceil} \frac{1}{2^{\lceil \log n \rceil}} \geq \left(1 - \frac{1}{4 \log n}\right)^{\lceil \log n \rceil} \frac{1}{2n} \geq \left(1 - \frac{2 \log n}{4 \log n}\right) \frac{1}{2n}$$
$$\geq \frac{1}{4n}.$$

Furthermore, using an analogous calculation, it is upper bounded by

$$\begin{aligned} \Pr[L_v = 1] &\leq \left(\left(1 + \frac{1}{4 \log n}\right) \frac{1}{2} \right)^{\lceil \log n \rceil} \leq \left(1 + \frac{1}{4 \log n}\right)^{\lceil \log n \rceil} \frac{1}{2^{\lceil \log n \rceil}} \leq \left(1 + \frac{1}{4 \log n}\right)^{2 \log n} \frac{1}{n} \\ &\leq \left(1 + \frac{1}{4 \log n}\right)^{4 \log n \cdot \frac{1}{2}} \frac{1}{n} \leq \frac{\sqrt{e}}{n} \leq \frac{2}{n}. \end{aligned}$$

The first inequality in the last line follows from the fact that $(1 + \frac{x}{n})^n \leq e^x$. Note that these bounds are the same for all agents. Thus, the probability that v is the *unique* leader is lower can be bounded as follows

$$\Pr[U_v = 1] = \Pr[L_v = 1] \prod_{w \in V \setminus \{v\}} (1 - \Pr[L_w = 1]) m \geq \frac{1}{4n} \left(1 - \frac{2}{n}\right)^{n-1} \geq \frac{1}{4n} \left(1 - \frac{2}{n}\right)^{n} \geq \frac{1}{4n} \left(e^{-1} \left(1 - \frac{4}{n}\right) \right) \geq \frac{1}{8en}.$$

Here, we used that $(1 - \frac{x}{n})^n \geq e^{-x}(1 - \frac{x^2}{n})$ and $1 - \frac{4}{n} \geq \frac{1}{2}$ for $n \geq 8$. As all event U_v are disjoint, we can sum them up and get

$$\Pr\left[\sum_{v \in V} U_v = 1 \right] = \sum_{v \in V} \Pr[U_v = 1] \geq \sum_{v \in V} \frac{1}{8e} \cdot \frac{1}{n} = \frac{1}{8e}.$$

Thus, the lemma follows. □

Recall that, we trigger a reset if any agent is activated L_{max} times. If we choose $L_{\mathsf{max}} \in \Theta(\log n)$ large enough, the leader will be elected before any agent interacts $(1/8) \cdot L_{\mathsf{max}}$ times. Thus, no agent *accidentally* starts with executing the main protocol before there is a unique leader.

Formally, we define these configurations as follows.

Definition 31. *$C_{\mathrm{SR}+}$ is the set of configurations where all agents' variables have the following values.*

$$\mathit{LECount}(v) \geq (7/8) \cdot L_{\mathsf{max}} \qquad \textit{Counter was not decreased too much.}$$
$$\texttt{leaderDone}(v) = 1 \qquad \textit{No (additional) agent will be leader.}$$

Furthermore, there is exactly one *agent $l \in V$ with* $\texttt{isLeader}(l) = 1$*, and for* all other *agents $w \in V \setminus \{l\}$, it holds* $\texttt{isLeader}(w) = 0$.

Using standard arguments, we can show the following lemma.

Lemma 32. *Let c be a sufficiently large constant, and assume $\vec{X}_t \in C_{\mathrm{LE}}$. Then, with constant probability, there is a $\tau \in [c \cdot n^2 \cdot \log n]$ such that $\vec{X}_{t+\tau} \in C_{\mathrm{SR}+}$. Otherwise, with probability $1/2^d$, there is a $\tau' \in [d \cdot c \cdot \cdot n^2]$ such that $\vec{X}_{t+\tau'} \in C_{\mathrm{LE}}$.*

Proof. We begin with the first two properties and show that there is time step where all agents have $\texttt{LECount}(v) \geq (7/8) \cdot L_{\mathsf{max}}$ and $\texttt{leaderDone}(v) = 1$. Recall that in a configuration from C_{LE}, there is precisely one agent in a state $q_{0,i} \in Q_{\text{LE}}$, and all others are dormant. In each interaction between a dormant agent and an agent with a state from Q_{LE}, the dormant agent will always switch to a state from Q_{LE}. Furthermore, whenever two dormant agents interact, they either remain dormant (if it holds $\texttt{delayCount}(v) > 0$ for both of them) or they switch to an initial state $q_{0,i} \in Q_{\text{LE}}$. Thus, the time in which all agents switch to a state from Q_{LE} is upper bounded by the time of one-way epidemic, namely $4\gamma \cdot n \cdot \log n$ interactions with probability $1 - \mathrm{O}(1/n^{\gamma})$. Furthermore, note that each agent $v \in V$ switches from being dormant sets $\texttt{LECount}(v) = L_{\mathsf{max}}$ upon initialization.

Let t_1 be the time step where the last agent $v \in V$ switches to $q_{0,i} \in Q_{\text{LE}}$. From this point on, each interaction between v and w, will decrease $\texttt{coinCount}(v)$ unless $\texttt{leaderDone}(v) = 1$. Once, it holds $\texttt{coinCount}(v) = 0$, it also sets $\texttt{leaderDone}(v) = 1$. After $(1+\gamma) \cdot n \log n$ steps each agent has been activated $\lceil \log n \rceil$ times with probability $1 - n^{-\gamma}$ and it holds $\texttt{leaderDone}(v) = 1$ for all $v \in V$. Thus, after $t_1 \leq 4\gamma \cdot n \cdot \log n$ global steps for waking up and $t_2 \leq (1+\gamma) \cdot n \log n$ global steps for flipping coins, all agents have $\texttt{leaderDone}(v) = 1$.

During all this time, each agent was activated $t_1 + t_2 \leq (5\gamma + 1) \cdot n \cdot \log n$ times on expectation. For a large enough choice of $L_{\mathsf{max}} > 100\gamma \log n$, no agent was activated less than $(1/8) \cdot L_{\mathsf{max}}$ times with probability $1 - \mathrm{O}(n^{-\gamma})$. This follows from a straightforward application of the Chernoff bound [33].

Now, we get to the second property, the number of leaders. By Lemma 30 we know that exactly one leader is elected with constant probability. In this case, we are done. Therefore, it remains to show that we reset if have no or more than one leader.

Case 1: No leader. If no leader is elected after $\Theta(L_{\mathsf{max}} \cdot n)$ interactions, one agent $v \in V$ must have been activated L_{max} times and it holds $\texttt{LECount}(v) = 0$. As this agent is not a leader, it will trigger a reset.

Case 2: Two or more leaders. Suppose we have two or more leaders. In the following we assume w.l.o.g. that at least two leaders start waiting. Otherwise, we are in Case 1 or the protocol *overwrites* the other leaders. We will show that in this case, either two leaders assign rank $r_1 = \lceil \frac{n}{2} \rceil + 1$ or trigger a reset in $\mathrm{O}(n^2)$ steps.

First, we argue that two leaders enter the waiting state at most $\mathrm{O}(n \log n)$ steps apart, w.h.p. Suppose that the first leader starts to wait in some step t_1. Then, the epidemic that turns any other leader into a phase agent reaches every other agent $\mathrm{O}(n \log n)$ steps later. Thus, if a second leader also starts waiting in some step $t_2 \geq t_1$, we can assume $|t_2 - t_1| \in \mathrm{O}(n \log n)$.

Let now ℓ_1 and ℓ_2 be the first two leaders that stop waiting and start ranking. We claim that both agents stop waiting while all other unranked agents are still in the first phase. This is trivially true for ℓ_1. To prove this for ℓ_2, note that ℓ_1 must interact with $n/2$ agents to finish the first ranking phase. This requires at least $N = \Omega(\frac{n^2}{\log n})$ global time, w.h.p. If t_1 is the time where any leader started waiting, at time $t_1 + N$, all agents are still in the first phase (unless, of course, we triggered a reset). We argue that ℓ_2 stops waiting before this time. During this time, at least half the agents remain unranked (if no other leader wakes up). Thus, if the second leader has not stopped waiting up until time $t_1 + N$, it would have either interacted with $\Theta(\frac{N}{n}) \in \omega(W_{\mathsf{max}})$ unranked agents or with another waiting leader, w.h.p. Therefore, it either stopped waiting within $\mathrm{O}(n^2)$ steps or triggered a reset.

Thus, if no reset is triggered, the first two leaders both wake up in the first phase and assign r_1. In every round after this, these two agents interact with probability $\frac{1}{n(n-1)}$. Therefore, the probability that they have not interacted after τ steps is dominated by the geometric distribution with parameter $1/n^2$.

□

Note that this lemma implies that after $\mathrm{O}(\log n)$ failed attempts to enter $C_{\text{SR}+}$, we succeed. The time spent in these failed attempts is $\mathrm{O}(n^2 \log n)$.

D. Proof of Lemma 11

Recall that C_{L} is the set of all configurations where all agents have unique rank.

Lemma 33. *Let c be a sufficiently large constant, and assume $\vec{X}_t \in C_{\text{SR}+}$. Then,* unless we trigger a reset*, there is a $\tau \in [c \cdot n^2 \log n]$ such that $\vec{X}_{t+\tau} \in C_{\text{L}}$.*

Proof. As we have exactly one leader and no more leaders will be elected, this leader will eventually start waiting. This starts an epidemic that turns all agents into phase-counting agents. Recall that these agents *forget* their value of $\texttt{LECount}(v)$. Since $\texttt{LECount}(v) \geq (7/8) \cdot L_{\mathsf{max}}$ for all agents, the epidemic will reach them before they perform $(7/8) \cdot L_{\mathsf{max}}$ interactions, w.h.p., for a large enough choice of L_{max}. Thus, eventually, one agent is waiting, and all agents are phase-counting. Given that there is no reset in the next $\mathrm{O}(n^2 \log n)$ interactions, the lemma follows from Lemma 4. □

Note that this lemma only holds under the premise that during execution, we never trigger a reset. So, in the remainder of the proof, we show that we do not reset w.h.p. We must consider all rules that could potentially trigger a reset when starting in an arbitrary configuration in $C_{\text{SR}+}$ and argue why these rules are *not* triggered with sufficient probability. Recall that a reset can *either* is triggered through two agents with the same label, an emergency reset issued by a waiting agent *or* by a phase counting agent whose liveness counter expires. We look at these three rules separately.

Rule 1: Reset Through Duplicate Labels. First, we note that the protocol, if started from $C_{\mathrm{SR}+}$ will never assign a label twice, w.h.p. Therefore, w.h.p., no agent will trigger a reset because of an interaction with the same label.

Rule 2: Reset Through Waiting Agent. A waiting or phase counting agent triggers a reset if it interacts with an agent of label n or $n-1$ more than L_{max} times before being labeled itself. This case can be ruled out through an appropriate choice of the tunable variable L_{max}. By Lemma 4 (and waiting for the right coins slows it down by a constant factor) the protocol stabilizes in $\mathrm{O}(W_{\mathsf{max}} \cdot n^2)$ interactions, w.h.p. Thus, after $\mathrm{O}(W_{\mathsf{max}} \cdot n^2)$ interactions, every agent has a label unless a reset is triggered. We first show the following claim.

Claim 34. *There is a constant $c_1 > 0$ that does not depend on L_{max}, s.t., every pair of agents interacts at most $c_1 \cdot W_{\mathsf{max}}$ times before all agents are labeled, w.h.p.*

Proof. Condition on the event that after $\mathrm{O}(W_{\mathsf{max}} \cdot n^2)$ interactions, every agent has a label. This happens w.h.p. Recall that the probability of two agents interacting in a given step is $1/n(n-1)$ as we use the uniform scheduler. Thus, for a given pair (v, w) of agents, the expected number of interactions between v and w within $\mathrm{O}(W_{\mathsf{max}} \cdot n^2)$ steps is $\mathrm{O}(W_{\mathsf{max}})$. As interactions are independent and can be modeled as binary random variables, we can apply the Chernoff bound [33]. For each pair v, w, we can use this well-known bound to show that the probability that v and w interact more than $c_1 \cdot W_{\mathsf{max}}$ times (where c_1 is large constant that depends on the constants hidden in $\mathrm{O}(W_{\mathsf{max}})$) during this time is at most $1/n^{c_2}$ where c_2 depends on c_1. A union bound over all $n(n-1)$ pairs of agents yields that the probability of *any* pair interacting more than $c_1 \cdot W_{\mathsf{max}}$ times is less than $1/n^{c_2-2}$. Thus, an appropriate choice of c_1 yields the lemma. □

From this, we can conclude that any agent $v \in V$ interacts with any set of two agents $w_1, w_2 \in V$ at most $2 \cdot c_1 \cdot W_{\mathsf{max}}$ times w.h.p. By choosing $L_{\mathsf{max}} > 2 \cdot c_1 \cdot W_{\mathsf{max}}$, we conclude that, w.h.p., no waiting agent interacts L_{max} times with any two specific agents. This includes, in particular, the agents with labels n and $n-1$ at any point during the execution. So, no reset is triggered, w.h.p.

Rule 3: Reset Through Liveness Checker. Finally, we argue why a phase counting agent will not trigger a reset, w.h.p. This (arguably) requires the most intricate proof. To this end, consider a configuration $\vec{X}_t$ and assume w.l.o.g. that we have k phase counting agents. Recall that these agents do *not* have a label and count the phase. In the following, we denote the set of phase counting agents in configuration $\vec{X}_{t'}$ as $P_{t'} \subset V$. Note that $P_t \subseteq P_{t+1}$ (unless we trigger a reset) and the number of phase counting agents is monotonically decreasing. Furthermore, we use ℓ to denote the agent that assigns the next label when interacting with an agent from P or is waiting. Recall from the analysis that there is always *exactly* one such agent in every configuration, w.h.p. Finally, we will divide time into *phases* of $\tau = 4 \cdot \frac{n^2}{k}$ interactions, s.t., phase i includes all configurations from $\vec{X}_{t+i\tau}$ to $\vec{X}_{t+(i+1)\tau-1}$. We call $T = c \cdot 4 \log n$ continuous phases an *epoch*. Here $c > 1$ is a tunable constant we will fix in the analysis. We will show the following lemma.

Lemma 35. *Consider an epoch $\vec{X}_t, \ldots, \vec{X}_{t+T\cdot\tau}$ and assume for all agents it holds $\geq (7/8) \cdot L_{\mathsf{max}}$ in $\vec{X}_t$. Then, w.h.p., it holds that*

1) *For all $t' \in [0, T \cdot \tau]$ and all $v \in P_{t+t'}$, it holds $\texttt{liveCount}(v) > 0$.*
2) *For all $v \in P_{t+T\cdot\tau}$, it holds $\texttt{liveCount}(v) \geq (7/8) \cdot L_{\mathsf{max}}$.*

Informally, this lemma states that during an epoch, no reset is triggered, and given that all agents have a high counter value in the beginning, they have a high counter value at the end. Let $\mathcal{E}_i$ be the event that no reset is triggered in the ith epoch *and* for all $v \in P$ at the end of the epoch, it holds $\texttt{liveCount}(v) \geq (7/8) \cdot L_{\mathsf{max}}$. Note that each epoch is of length at least $\mathrm{O}(n \log n)$. Therefore, if the event $\mathcal{E}_i$ holds for $\eta \in \mathrm{O}(n)$ consecutive epochs, we do not trigger a reset within $\mathrm{O}(n^2 \log n)$ interactions. Thus, we will show that for some $c > 1$ that

$$\Pr\left[\bigcap_{i=1}^{\eta} \mathcal{E}_i\right] \geq 1 - n^{-c}. \tag{4}$$

Let $\mathcal{P}_t$ be the event that for all $v \in P_t$, it holds $\texttt{liveCount}(v) \geq (7/8) \cdot L_{\mathsf{max}}$ Furthermore, $S_i = (t, k)$ is the event that the ith starts in step t with k phase counting agents. By Lemma 35, for *all* possible choices of t and k, it holds for some universal $c' > 1$ that

$$\Pr[\mathcal{E}_i \mid S_i = (t, k) \cap \mathcal{P}_t] \geq 1 - n^{-c'}.$$

In particular, the lemma holds conditioned on everything else that happened before step t as only $\mathcal{P}_t$ and S_i are relevant for the lemma. Let now $\mathcal{S}_i$ be the set of all possible realizations of S_i. Then, using the chain rule of conditional probability and the law of total probability, we get

$$\begin{aligned}
\Pr\left[\bigcap_{i=1}^{\eta}\mathcal{E}_i\right] &= \prod_{i=1}^{\eta}\Pr\left[\mathcal{E}_i \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right] \\
&= \prod_{i=1}^{\eta}\sum_{(t,k)\in\mathcal{S}_i}\Pr\left[S_i=(t,k) \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right]\Pr\left[\mathcal{E}_i \mid S_i\bigcap_{j=1}^{i-1}\mathcal{E}_j\right] \\
&= \prod_{i=1}^{\eta}\sum_{(t,k)\in\mathcal{S}_i}\Pr\left[S_i=(t,k) \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right]\Pr[\mathcal{E}_i \mid S_i=(t,k)\cap\mathcal{P}_t] \\
&\geq \prod_{i=1}^{\eta}\sum_{(t,k)\in\mathcal{S}_i}\Pr\left[S_i=(t,k) \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right](1-n^{-c'}) \\
&\geq \prod_{i=1}^{\eta}(1-n^{-c'})\prod_{i=1}^{\eta}\sum_{(t,k)\in\mathcal{S}_i}\Pr\left[S_i=(t,k) \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right] \\
&\geq (1-n^{-c'})^{\eta}\prod_{i=1}^{\eta}\sum_{(t,k)\in\mathcal{S}_i}\Pr\left[S_i=(t,k) \mid \bigcap_{j=1}^{i-1}\mathcal{E}_j\right] \\
&\geq (1-n^{-c'})^{\eta}\prod_{i=1}^{\eta}1 \geq (1-n^{-c'-2}).
\end{aligned}$$

Thus, inequality (4) holds for $c=c'+2$ and the lemma follows.

Proof of Lemma 35. Before we start with our main argument, let us quickly recall the behavior of these agents. If they interact with the labeling agent ℓ, they either get labeled (if their coin is 0) or reset their liveness counter to L_{max} (if their coin is 1) . If ℓ is waiting, they always reset their liveness counter to L_{max}. Furthermore, whenever two phase-counting agents interact, they agree on the maximum of their respective liveness counters and decrease them by one. We suppose that the agents perform a slightly different protocol for this proof. For the sake of argument, assume that agents have infinite memory and act like agents in a message-passing system. The adapted protocol works as follows: when agent ℓ interacts with an agent $v\in P$ whose coin is 1, it starts a broadcast b. A broadcast b is a message that contains l_b, a counter that stores how often it has been forwarded since its creation. Initially, the counter is set to $l_b=0$. Whenever two agents $v,w\in P$ interact, they forward all broadcasts they know and increase their counters by 1. Assume we run both protocols simultaneously, letting the same agents interact in each step. Call the resulting protocol, the *coupled* protocol. Then, the connection between broadcast protocol and our protocol is as follows.

Claim 36. *Consider a broadcast b that has been forwarded l_b times. In the coupled protocol, the liveness counter of an agent $v\in V$ that has received b has a value of at least $L_{\mathsf{max}}-l_b$ in the original protocol.*

Proof. This can be shown through a simple induction on the lifespan of a broadcast b.

1) For the base case, recall broadcast b is created by interacting with ℓ. Suppose an agent v interacts with agent ℓ and creates b. Its initial count is $l_b=0$. This interaction also sets v's counter to $L_{\mathsf{max}}=L_{\mathsf{max}}-l_b$. Thus, initially, our claim holds.
2) For the step, suppose that agents v and w interact. W.l.o.g., let v have the higher liveness counter. Then, v must have a broadcast message b_v with counter l_{b_v} such that $\texttt{liveCount}(v)\geq L_{\mathsf{max}}-l_{b_v}$. After the interaction, v has decreased its counter by one and increased l_{b_v} by 1 and w has the same value as v and also knows b_v. Thus, the claim follows.

This proves the claim. □

Therefore, we can use the broadcast time to bound the drift of the liveness counters. We show the following claim.

Claim 37. *Suppose a broadcast b was received by an agent $v\in V$ within T' phases. For any choice $T'\geq c\log n$, it holds that*

$$\Pr[l_b\leq 200T']\geq 1-n^{-c}. \tag{5}$$

Proof. Fix a broadcast b started in configuration $\vec{X}_{t_0}$ by some agent $v_1\in V$ interacting with ℓ and is known by agent v in step $t_0+T'\cdot\tau$. If v receives the broadcast with value $l_b=l$, we can create a *witness sequence* that *proves* that v has the broadcast b and the current counter is l. Seeking formalization, there must be the sequence $W = ((v_1, w_1, t_1), \dots, (v_l, w_l, t_l))$ of time steps $t_1, \dots, t_l \in [T' \cdot \tau]^l$ and pairs $(v_1, w_1), \dots, (v_l, w_l) \in P^l$. If $v_i = v_{i+1}$, agent v_i has sent b to w_i in step t_i (and thereby increases its counter). Otherwise, If $v_{i+1} = w_i$, agent v_i has initially received b from w_{i+1} in step t_i (and thereby increases its counter). This distinction is necessary because the broadcast counter is increased on every interaction and not only in the step that it is received. Note that if the broadcast counter is l, we can construct a witness sequence of length l. We now bound the probability of such a sequence in general. First, we are interested in the probability of a fixed witness sequence W. Let $I(v_i, w_i, t_i)$ be the event that v_i interacts with w_i in step t_i. Note that for any pair $v_i, w_i \in V$ and any step t_i, it holds regardless of everything that happened before step t that

$$\Pr[I(v_i, w_i, t_i)] \leq \frac{1}{n(n-1)}.$$

Therefore, by the chain of conditional probability

$$\Pr[W]\Pr\left[\bigcap_{i=1}^{l} I(v_i, w_i, t_i\right] = \prod_{i=1}^{l} \Pr[I(v_i, w_i, t_i \mid \bigcap_{j<i} I(v_j, v_i, t_j] \leq \prod_{i=1}^{l} \frac{1}{n(n-1)} = \left(\frac{1}{n(n-1)}\right)^l.$$

Let $\mathcal{W}_l$ be set of all witness sequences of length l. We want to count how many of these sequences can exist. For a better approximation, recall that two consecutive members (v_i, w_i, t_i) and $(v_{i+1}, w_{i+1}, t_{i+1})$ share an agent. Thus, for each i, we can define a bit f_i such that $f_i = 0$ if $v_{i+1} = w_i$ and $f_i = 1$ if $v_{i=1} = v_i$. Therefore, any member of a sequence can be expressed as $(v_i, f_i, t_i) \in P \times [1] \times [T'\tau]$ instead without losing information. For $l \geq c \cdot 200T'$, it holds that

$$\begin{aligned}
|\mathcal{W}_l| &\leq \underbrace{\binom{k}{l}}_{\text{Choices of } v} \cdot \underbrace{\binom{T' \cdot \tau}{l}}_{\text{Choices of } t} \cdot \underbrace{\sum_{h=1}^{l} \binom{l-h}{h}}_{\text{Choices of } f} \\
&\leq \binom{k}{l} \cdot \binom{T' \cdot \tau}{l} \cdot 2^l && \sum \binom{n-i}{i} = 2^n - 1 \\
&\leq \left(\frac{e \cdot k}{l}\right)^l \cdot \left(\frac{e \cdot T' \cdot \tau}{l}\right)^l \cdot 2^l && \binom{n}{i} \leq (\frac{en}{i})^i \\
&= \left(\frac{2e^2 \cdot k \cdot T' \cdot \tau}{l}\right)^l = \left(\frac{8e^2 \cdot k \cdot T' \cdot n(n-1)}{k \cdot l}\right)^l && \tau = 4 \cdot \frac{n(n-1)}{k} \\
&= \left(\frac{8e^2 \cdot T' \cdot n(n-1)}{l}\right)^l \leq \left(\frac{8e^2 \cdot T' \cdot n(n-1)}{200 \cdot T'}\right)^l && l \geq 200T' \\
&= \left(\frac{8e^2 \cdot n(n-1)}{200}\right)^l \leq \left(\frac{n(n-1)}{3}\right)^l && 8e^2 < 60.
\end{aligned}$$

Finally, let $\mathcal{B}$ be the event that there is a witness sequence of length more than $c \cdot 200T'$. We use the union bound to show

$$\begin{aligned}
\Pr[\mathcal{B}] = \Pr\left[\bigcup_{l=c \cdot 200T'}^{T' \cdot \tau} \bigcup_{W \in \mathcal{W}_l} W\right] &\leq \sum_{l=c \cdot 200T'}^{T' \cdot \tau} \sum_{W \in \mathcal{W}_l} \Pr[W] \leq \sum_{l=c \cdot 200T'}^{T' \cdot \tau} \left(\frac{n(n-1)}{3}\right)^l \left(\frac{1}{n(n-1)}\right)^l \\
&\leq \sum_{l=c \cdot 200T'}^{T' \cdot \tau} \left(\frac{1}{3}\right)^l \leq T' \cdot \tau \cdot \left(\frac{1}{3}\right)^{c \cdot 200T'} \leq n^3 \cdot \left(\frac{1}{3}\right)^{c200 \log n} \leq \left(\frac{1}{n}\right)^{200c-3}.
\end{aligned}$$

Therefore, a sequence of this length does not exist, w.h.p. This proves the claim. □

To prove Lemma 35, we must show that v receives a broadcast started in the epoch within the epoch. It is easy to see that the broadcast time of broadcast message b is bounded by a two-way epidemic, which is bounded by a one-way epidemic on the unranked agents. However, we need to take into account that during the protocol's execution an unranked agent gets ranked and, therefore, stops spreading the broadcast. Even worse, all agent holding a broadcast b could get a rank and broadcast dies out completely. As it turns, we can still show the following.

Claim 38. *Let $\mathcal{P} \subseteq P$ be a set of agents that are unranked at the end of the epoch. Then, w.h.p., all these agents received at least one broadcast within the epoch.*

Proof. First, note that we can assume that $|\mathcal{P}| \geq k/2$, w.h.p. Suppose the number of unranked agents halves at any point in the epoch. Then, ℓ will start waiting (if it hasn't before). In particular, ℓ will wait (and therefore not rank any agents) until it interacts with any unranked agent W_{max} times. Given that there are k unranked agents initially, this takes at least $\mathrm{O}(\frac{n(n-1)}{k} \cdot W_{\mathsf{max}})$ interaction on expectation and w.h.p. For a large enough choice of W_{max}, this exceeds whatever time is remaining in the epoch. Therefore, no further agents get ranked.

Going forward, we say that a time step t is good if a constant fraction of $\rho \cdot |\mathcal{P}|$ agents in $\mathcal{P}$ have a coin that shows *heads*. According to Lemma 40, each epoch has a constant fraction of good time steps unless there is a reset. This follows because an episode of length $2n$ has a constant fraction of good steps with constant probability and the individual episodes are independent of one another as they assume an arbitrary distribution of the coins in the beginning of each episode. Therefore, as the other two rules do not trigger a reset w.h.p., and during the epoch no counter gets to 0, w.h.p., we can assume that a constant fraction of the steps is good, w.h.p. In the remainder, we only focus on these good time steps.

Denote the event that at $\Theta(\frac{T\tau}{)}$ time steps are good and at least $k/2$ agents stay unranked during the epoch as $\mathcal{E}$, i.e., they do not interact with ℓ if their coin is tails. Let $T(\mathcal{P})$ be the number of good time steps until all agents are informed. Let $I_t \subset \mathcal{P} \cup \{\ell\}$ be the set of agents that, in step t, know of a broadcast started after the beginning of the epoch. Note that this set always contains ℓ as it starts all broadcasts. Suppose that there are i uninformed agents in $\mathcal{P}$ that know none of these broadcasts. Then, the probability that the number of informed agents increases is in good step t is

$$\begin{aligned}
\Pr[|I_t| = |I_t| + 1 \mid \{I_t = i\} \cap \{t \textit{ good }\}] &\geq \underbrace{\frac{(i-1)(m-i)}{n(n-1)}}_{\substack{v \in I_t \cap \mathcal{P} \textit{ interacts} \\ \textit{with } w \notin I_t \cap \mathcal{P}}} + \underbrace{\frac{\rho(m-i)}{n(n-1)}}_{\ell \textit{ creates new broadcast}} \\
&\geq \frac{\rho(i-1)(m-i) + \rho \cdot (m-i)}{n(n-1)} \\
&\geq \rho\left(\frac{(i-1)(m-i) + (m-i)}{n(n-1)}\right) \\
&= \rho\left(\frac{i(m-i)}{n(n-1)}\right).
\end{aligned}$$

Furthermore, by the law of total probability it holds that

$$\begin{aligned}
\Pr[|I_t| = |I_t| + 1 \mid \{|I_t| = i\} \cap \{t \textit{ good }\} \cap \mathcal{E}] &\geq \Pr[|I_t| = |I_t| + 1 \mid \{|I_t| = i\} \cap \{t \textit{ good }\}] - \Pr[\mathcal{E}] \\
&\geq \rho\left(\frac{i(m-i)}{n(n-1)}\right) - \frac{1}{n^c} \geq \frac{\rho}{2}\left(\frac{i(m-i)}{n(n-1)}\right).
\end{aligned}$$

As $\mathcal{P}$ has at least$k/2$ agents, the number of good steps required to inform all agents in $\mathcal{P}$ is stochastically dominated by

$$T(\mathcal{P}) \mid \mathcal{E} \prec Y = \sum_{i \in [m-1]} X_i, \text{ with } X_i \sim \mathrm{Geom}\left(\frac{\rho}{4} \cdot \frac{i(k-i)}{n(n-1)}\right) \text{ independent.}$$

Therefore, we have $\Pr\Big[T(\mathcal{P}) \geq c \cdot \rho^{-1} 24 \frac{n^2}{k} \log n \mid \mathcal{E}\Big] \leq n^{-c}$. This means, we require $c \cdot \rho^{-1} 24 \frac{n^2}{k} \log n$ good steps until all agents in $\mathcal{P}$ received the broadcast. Thus, if we pick the length of our epoch larger than $c \cdot \rho^{-1} \cdot C \cdot \log n$ where C is a large constant, it holds $c \cdot \rho^{-1} 24 \frac{n^2}{k} \log n \leq \Theta(T\tau)$. As we conditioned on $\Theta(T\tau)$ good steps in an epoch, all agents in $\mathcal{P}$ must receive a broadcast, w.h.p. □

Recall that we can choose L_{max} as large as we want. Thus, by choosing $L_{\mathsf{max}} \geq 1600 \cdot T$, we can combine all of our lemmas and get

$$\begin{aligned}
1 - n^{-c} &\leq \Pr[\exists b : l_b \leq c200T] = \Pr[\texttt{liveCount}(v) \leq L_{\mathsf{max}} - c200T] \\
&= \Pr[\texttt{liveCount}(v) \leq L_{\mathsf{max}} - ({}^{1}\!/\!{}_{8}) \cdot L_{\mathsf{max}}] = \Pr[\texttt{liveCount}(v) \leq ({}^{7}\!/\!{}_{8}) \cdot L_{\mathsf{max}}].
\end{aligned}$$
□

E. Auxiliary results: analysis of the phase agents' coin

For a fixed t where we assume that $\vec{X}_t$ is a configuration in C_{Main}, and $t' \geq t$, we let $K_{t'}$ be the number of phase agents at time t', and $H_{t'}$ be the number of phase agents whose coin shows 1 at time t'.

Lemma 39. *Assume that $\vec{X}_t$ is a configuration in C_{Main}. Then there are constants $0 < p, c < 1$ such that with probability at least p, either a reset is triggered within the $2n$ rounds following t, or for at least a c_1 fraction of the next $2n$ rounds, $H_{t'} \geq K_{t'}/4$—i.e., for $S_t = \{t' \in (t, t+2n] \mid H_{t'} \geq K_{t'}/4\}$ we have $|S_t| \geq c \cdot 2n$.*

Proof. For $\tau \geq 0$ consider the random variable

$$R_{t+\tau} = \begin{cases} 1, & \text{if for any time } t' \in (t, t+\tau], \\ & \text{a reset was triggered at a time } t' \text{ or } K_{t'} = 0, \\ H_t/K_t, & \text{otherwise.} \end{cases}$$

We now consider the contribution to the expected change of R by various events, noting that we want to bound this from below:

- In any case, if $R_{t+\tau}$ is 1 due to the first branch of the definition, then $R_{t+\tau} = R_{t+\tau}$.
- If a reset is triggered or the last phase agent disappears at time $t+\tau+1$, then $R_{t+\tau+1} = 1 \geq \min\{1, R_{t+\tau+1} + 1/K_{t+\tau}\}$.
- If a phase agent v with $\texttt{coin}(v) = 0$ is chosen as respondent at time $t+\tau$ (with probability $(K_{t+\tau} - H_{t+\tau})/n = (K_{t+\tau} - R_{t+\tau}K_{t+\tau})/n$) and none of the earlier cases holds, then its coin is toggled and then $R_{t+\tau+1} = R_{t+\tau} + 1/K_{t+\tau}$.
- If a phase agent v with $\texttt{coin}(v) = 1$ is chosen as respondent at time $t+\tau$ (with probability $H_{t+\tau}/n = R_{t+\tau}K_{t+\tau}/n$) and none of the earlier cases holds, then
  - either its coin is toggled to tails and $R_{t+\tau+1} = R_{t+\tau} - 1/K_{t+\tau}$,
  - or it is ranked and $R_{t+\tau+1} = (H_{t+\tau} - 1)/(K_{t+\tau} - 1) \geq R_{t+\tau} - 1/K_{t+\tau}$.

Combining all of this, one can see that the additive $1/K_{t+\tau}$ terms in the changes in R and $K_{t+\tau}$s in the denominators of the probabilities cancel to obtain

$$\mathbb{E}[R_{t+\tau+1} \mid R_{t+\tau}] \geq R_{t+\tau} + \frac{1 - 2R_{t+\tau}}{n} = 0.5 + (R_{t+\tau} - 0.5) - \frac{2(R_{t+\tau} - 0.5)}{n}.$$

From this, induction over τ and using $R_{t+\tau} \geq 0$ gets us

$$\mathbb{E}[0.5 - R_{t+\tau+1} \mid R_t] \leq \left(1 - \frac{2}{n}\right) \cdot \mathbb{E}[0.5 - R_{t+\tau} \mid R_t] \leq (0.5 - R_t) \cdot \left(1 - \frac{2}{n}\right)^{\tau} \leq \frac{1}{2} \cdot \left(1 - \frac{2}{n}\right)^{\tau}.$$

For $\tau = 3n/2$, this yields

$$\mathbb{E}[(0.5 - R_{t+3n/2}) \mid R_t] \leq \frac{1}{2} \cdot \exp\left(-\frac{2\tau}{n}\right) = \frac{e^{-3}}{2} \leq \frac{1}{40},$$

and hence applying Markov's inequality on $(1 - R_{t+3n/2})$ (which is ≥ 0)

$$\Pr[R_{t+3n/2} \geq 1/3 \mid R_t] = 1 - \Pr\big[1 - R_{t+3n/2} \geq 2/3\big]R_t \geq 1 - \frac{0.525}{2/3} = 0.2125.$$

Now condition on $R_{t+3n/2} \geq 1/3$. If this is because in the interval $(t, t+3n/2]$ a reset was triggered or there were no phase agents, we are already done. Otherwise, there is an $\Omega(1)$ probability that in the next $n/2$ rounds, at most $K_{t+3n/2}/12$ of the $H_{t+3n/2} \geq K_{t+3n/2}/3$ phase agents v having $\texttt{coin}(v) = 1$ at time $t+3n/2$ are selected as respondents in the following $n/2$ rounds, and this is independent from the prior rounds. Hence, with at least that probability, from time $t+3n/2$ to time $t+2n$, at least a $\frac{1}{3} - \frac{1}{12}$ fraction of phase agents still has $\texttt{coin}(v) = 1$, as claimed. □

We need a similar result for the case where we want to show that no $\texttt{aliveCount}(v)$ reaches 0, w.h.p., when the protocol is in a well-formed configuration for ranking. There, we care about the number of rounds where there is at least a $1/4$ fraction of phase agents' coins showing tails (i.e., 0), so that we can ensure that some $\texttt{aliveCount}(v)$ is reset often enough. Here, unlike above, a phase agent having its coin at 1 being ranked doesn't work in our favor. However, as long as we are in the good case, this will only happen with probability $\leq 1/n$ in any interaction because there is at most one unaware leader.

So for a time interval of size $2n$, there is an $\Omega(1)$ probability of the leader never being selected as initiator or responder, in which case no agent is ranked throughout the interval.

Lemma 40. *Assume that $\vec{X}_t$ is a configuration in C_{Main}. Then there are constants $0 < p, c < 1$ such that for n sufficiently large, with probability at least p, either a reset is triggered within the $2n$ rounds following t, or there is more than one unaware leader in any of the $2n$ rounds following t, or for a c_1 fraction of the next $2n$ rounds, $H_{t'} \leq 3K_{t'}/4$—i.e., for $S_t = \{\tau \in (t, t+2n] \mid H_t \leq 3K_t/4\}$ we have $\Pr[|S_t| \geq c \cdot 2n] \geq p$.*

Proof. For $\tau \geq 0$ consider the random variable

$$R_{t+\tau} = \begin{cases} 0, & \text{if for any time } t' \in (t, t+\tau], \text{ a reset was triggered at a time } t', \text{ or} \\ & \text{there was more than one unaware leader, or } K_{t'} = 0, \\ H_t/K_t, & \text{otherwise.} \end{cases}$$

We now consider the contribution to the expected change of R by various events, noting that we want to bound this from above:

- In any case, if $R_{t+\tau}$ is 0 due to the first branch of the definition, then $R_{t+\tau+1} = R_{t+\tau}$.

- If a reset is triggered or the last phase agent disappears at time $t+\tau+1$, then $R_{t+\tau+1} = 0 \le \max\{0, R_{t+\tau} - 1/K_{t+\tau}\}$.
- If a phase agent v with $\texttt{coin}(v) = 0$ is chosen as respondent at time $t+\tau$ (with probability $(K_{t+\tau} - H_{t+\tau})/n = (K_{t+\tau} - R_{t+\tau}K_{t+\tau})/n$) and none of the earlier cases holds, then its coin is toggled and then $R_{t+\tau+1} = R_{t+\tau} + 1/K_{t+\tau}$.
- If a phase agent v with $\texttt{coin}(v) = 1$ is chosen as respondent at time $t+\tau$ (with probability $H_{t+\tau}/n = R_{t+\tau}K_{t+\tau}/n$) and none of the earlier cases holds, then
  - either its coin is toggled to tails and $R_{t+\tau+1} = R_{t+\tau} - 1/K_{t+\tau}$,
  - or it is ranked and $R_{t+\tau+1} = (H_{t+\tau} - 1)/(K_{t+\tau} - 1) = R_{t+\tau} - \frac{1-R_{t+\tau}}{K_{t+\tau}-1} \le R_{t+\tau}$; however, this can only happen with probability $\le 1/(n-1)$ as there is at most one unaware leader in this case.

Again, combining all of this, one can see that the additive $1/K_{t+\tau}$ terms in the changes in R and $K_{t+\tau}$s in the denominators of the probabilities cancel to obtain

$$\mathbb{E}[R_{t+\tau+1} \mid R_{t+\tau}] \le R_{t+\tau} + \frac{1 - \left(2 - \frac{1}{n-1}\right)R_{t+\tau}}{n} = a + (R_{t+\tau} - a) - \frac{\left(2 - \frac{1}{n}\right)(R_{t+\tau} - a)}{n}.$$

where $a = \left(2 - \frac{1}{n}\right)^{-1} = \frac{n-1}{2n-3}$. Then, induction over τ and using $R_{t+\tau} \le 1$ gets us

$$\mathbb{E}[R_{t+\tau+1} \mid R_t] \le a + \left(1 - \frac{2 - \frac{1}{n}}{n}\right) \cdot \mathbb{E}[R_{t+\tau} - a \mid R_t] \le a + (R_t - a) \cdot \left(1 - \frac{2 - \frac{1}{n}}{n}\right)^{\tau} \le a + (1 - a) \cdot \left(1 - \frac{2 - \frac{1}{n}}{n}\right)^{\tau}.$$

Now $a = \frac{n-1}{2n-3} = \frac{1}{2} + \frac{1}{4n-6} = \frac{1}{2} + o(1)$, and so for $\tau = 3n/2$ we have

$$\mathbb{E}[R_{t+\tau+1} \mid R_t] \le a + (1 - a) \cdot \exp\left(-\frac{(2 - o(1)\tau}{n}\right) = \frac{1}{2} + o(1) + \left(\frac{1}{2} - o(1)\right) \cdot \exp(-3 + o(1)),$$

which is at most 0.525 for sufficiently large n. An argument analogous to that at the end of the proof of Lemma 39 yields the claim. □