Loosely-Stabilizing Leader Election on Arbitrary Graphs in Population Protocols Without Identifiers nor Random Numbers[*]

Yuichi Sudo[1], Fukuhito Ooshita[2], Hirotsugu Kakugawa[3], and Toshimitsu Masuzawa[4]

1   NTT Secure Platform Laboratories, Tokyo, Japan; and
    Graduate School of Information Science and Technology, Osaka University,
    Osaka, Japan
    sudo.yuichi@lab.ntt.co.jp
2   Graduate School of Information Science, Nara Institute of Science and
    Technology, Nara, Japan
    f-oosita@is.naist.jp
3   Graduate School of Information Science and Technology, Osaka University,
    Osaka, Japan
    kakugawa@ist.osaka-u.ac.jp
4   Graduate School of Information Science and Technology, Osaka University,
    Osaka, Japan
    masuzawa@ist.osaka-u.ac.jp

Abstract

In the population protocol model Angluin et al. proposed in 2004, there exists no self-stabilizing leader election protocol for complete graphs, arbitrary graphs, trees, lines, degree-bounded graphs and so on unless the protocol knows the exact number of nodes. To circumvent the impossibility, we introduced the concept of *loose-stabilization* in 2009, which relaxes the closure requirement of self-stabilization. A loosely-stabilizing protocol guarantees that starting from any initial configuration a system reaches a safe configuration, and after that, the system keeps its specification (e.g. the unique leader) not forever, but for a sufficiently long time (e.g. exponentially large time with respect to the number of nodes). Our previous works presented two loosely-stabilizing leader election protocols for arbitrary graphs; One uses agent identifiers and the other uses random numbers to elect a unique leader. In this paper, we present a loosely-stabilizing protocol that solves leader election on arbitrary graphs without agent identifiers nor random numbers. By the combination of virus-propagation and token-circulation, the proposed protocol achieves polynomial convergence time and exponential holding time without such external entities. Specifically, given upper bounds N and Δ of the number of nodes n and the maximum degree of nodes δ respectively, it reaches a safe configuration within $O(mn^3d + mN\Delta^2 \log N)$ expected steps, and keeps the unique leader for $\Omega(Ne^N)$ expected steps where m is the number of edges and d is the diameter of the graph. To measure the time complexity of the protocol, we assume the uniformly random scheduler which is widely used in the field of the population protocols.

19th International Conference on Principles of Distributed Systems (OPODIS 2015).
Editors: Emmanuelle Anceaume, Christian Cachin, and Maria Potop-Gradinariu; Article No. 14; pp. 14:1–14:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1   Introduction

This paper focuses on self-stabilizing leader election in the population protocol model. The *population protocol* (PP) model, which was presented by Angluin et al. [1], represents wireless sensor networks of mobile sensing devices that cannot control their movement. Two devices (say *agents*) communicate with each other and change their states only when they come sufficiently close to each other (we call this event an *interaction*). Self-stabilizing leader election (SS-LE) requires that starting from any configuration, a system (say *population*) reaches a safe-configuration in which a unique leader is elected, and after that, the population has the unique leader forever. Self-stabilizing leader election is important in the PP model because (i) many population protocols in the literature work on the assumption of the unique leader [1, 2, 3], and (ii) self-stabilization tolerates any finite number of transient faults and this property suits systems consisting of numerous cheap and unreliable nodes. (Such systems are the original motivation of the PP model.) However, there exists strict impossibility of SS-LE in the PP model: no protocol solves SS-LE for complete graphs, arbitrary graphs, trees, lines, degree-bounded graphs and so on unless the number of agents n is available to agents in advance [3].

Therefore, many studies of SS-LE took either one of the following two approaches. One approach is to accept the assumption that the exact n is available and focus on the space complexity of the protocol. Cai et al. [6] proved that n states of each agent is necessary and sufficient to solve SS-LE for a complete graph of n agents. Mizoguchi et al. [12] and Xu et al. [15] improved the space-complexity by adopting the mediated population protocol model [10] and the PP_k model [5] respectively. The other approach is to use *oracles*, a kind of failure detectors. Fischer and Jiang [8] took this approach for the first time. They introduced oracle Ω? that informs all agents whether a leader exists or not and proposed two protocols that solve SS-LE for rings and complete graphs by using Ω?. Beauquier et al. [4] presented an SS-LE protocol for arbitrary graphs that uses two copies of Ω?. Canepa et al. [7] proposed two SS-LE protocols that use Ω? and consume only 1 bit of each agent: one is a deterministic protocol for trees and the other is a probabilistic protocol for arbitrary graphs although the position of the leader is not static and moves among the agents.

Our previous works [13, 14] took another approach to solve SS-LE. We introduced the concept of *loose-stabilization*, which relaxes the closure requirement of self-stabilization. Specifically, starting from any initial configuration, the population must reach a safe configuration within a relatively short time; after that, the specification of the problem (the unique leader) must be kept for a sufficiently long time, though not forever. We proposed three loosely-stabilizing protocols P_{LE}, P_{ID}, and P_{RD}. Protocol P_{LE} solves leader election for complete graphs whose size is no more than given upper bound N of n. Protocol P_{ID} and P_{RD} solve leader election for arbitrary graphs using agent identifiers and random numbers respectively, given N and upper bound Δ of the maximum degree of nodes δ. All the three protocols are practically equivalent to a SS-LE protocol since they keep the specification for an exponentially long time after reaching a safe configuration (and reaches a safe configuration within polynomial time).

Some works on population protocols assume the probabilistic distribution regarding the interactions of agents: any interaction occurs uniformly at random [1, 2, 9, 13, 14]. This assumption have been used mainly for evaluating the time complexity of protocols. We also adopt this assumption because the measure of time is crucial in the concept of loose-stabilization. The impossibility result for SS-LE [1] still holds even with this assumption.

■ **Table 1** Loosely-stabilizing leader election for arbitrary graphs.

Protocol	Convergence Time	Holding Time	Agent Memory	Requisite
P_{ID} [14]	$O(mN\Delta \log N)$	$\Omega(Ne^N)$	$O(\log N)$	agent identifiers
P_{RD} [14]	$O(mN^3\Delta^2 \log N)$	$\Omega(Ne^N)$	$O(\log N)$	random numbers
P_{AR} (proposed)	$O(mn^3d + mN\Delta^2 \log N)$	$\Omega(Ne^N)$	$O(\log N)$	–

Our Contribution

This paper proposes a loosely-stabilizing protocol P_{AR} for leader election in arbitrary graphs without agent identifiers nor random numbers (or a model with a weaker assumption than P_{ID} or P_{RD}). Thus, we succeed to remove the assumptions of unique identifiers and random number generators for a loosely-stabilizing leader election on arbitrary graphs in the PP model, which may be difficult to realize in weak computation models, like the PP model, consisting of huge number of tiny devices with restricted capability.

The expected convergence time and the expected holding time of P_{ID}, P_{RD}, and P_{AR} are shown in Table 1 where d is the diameter of the graph. All the protocols including P_{AR} keep the unique leader for an exponentially long time ($\Omega(Ne^N)$ interactions) after a safe configuration. Protocol P_{AR} consumes $O(\log N)$ bits of each agent's memory while any self-stabilizing protocol (which uses knowledge of exact n) consumes $\Omega(\log n)$ memory [6]. Furthermore, Izumi [9] proves that loosely-stabilizing leader election with polynomial convergence time and exponentially long holding time needs $\Omega(\log n)$ agent memory. Thus, P_{AR} is asymptotically space-optimal when N is polynomial in n. One may think that the model of anonymous agents and $O(\log N)$ agent memory is not well-motivated because $O(\log n)$ memory is sufficient to store an identifier. However, we believe that anonymity is still an important assumption: assigning distinct identifiers to a huge number of agents is not an easy task, and memory corruption may cause conflicts of identifiers of different agents. Actually, many works assume anonymity and agent memory space of $O(\log n)$ or more (e.g. [3, 6, 12, 13, 14, 15]). In this paper, we analyze time complexities for undirected graphs for simplicity, however, it works on any *directed* graphs without modifications.

While protocol P_{AR} is based on the virus war mechanism developed for P_{RD} [14], the key idea of P_{AR} is quite novel and has a considerable contribution: The token with a count-down timer circulates in the graph, and a leader creates and spreads a black or white virus when encountering the token with zero timer value. The idea of circulating tokens and the colors of viruses are newly introduced to remove the assumption of random number generators. This technique may be useful also for other problems and/or other models.

The formal analysis of the convergence time and the holding time is another main contribution of this paper, since analyzing such complexities of loosely-stabilizing protocols is a challenging task. In particular, we analyze in the expected time until two tokens performing random walks meet in the PP model. The analysis can be applied with slight modification to estimate the expected time until a token performing random walks visits all nodes. We believe that the analysis techniques are of significant importance because existing analysis for usual random walks cannot be applied to the population protocol model: the token always moves through an edge at each step in usual random walks while, in the population protocol model, the token moves at each step with a probability depending on the degree of the node the token currently exists on. Thus, the techniques we developed open up a new path to analysis of loosely-stabilizing protocols in the PP model.

Angluin et al. [1] proves that for any population protocol P working on complete graphs, there exists a protocol that simulates P on any arbitrary graph. One may think that this

simulator can translate our previous loosely-stabilizing algorithm for complete graphs [13] to a loosely-stabilizing algorithm that works for arbitrary graphs. However, it cannot work since, in this simulation, two agents swap their states when they have interactions. This swap is needed to simulate interactions between distant agents in an arbitrary graph, but it results in the execution where an elected leader moves among the population, which does not satisfy the specification of the leader election.

2   Preliminaries

This section defines the model we consider for this paper.

A *population* is a simple and weakly-connected directed graph $G(V, E)$ where V ($|V| \geq 2$) is a set of *agents* and $E \subseteq V \times V$ is a set of directed edges. Each edge represents a possible *interactions* (or communication between two agents): If $(u, v) \in E$, agents u and v can interact with each other where u serves as an *initiator* and v serves as a *responder*. We say that G is undirected if it satisfies $(u, v) \in E \Leftrightarrow (v, u) \in E$. We define $n = |V|$ and $m = |E|$.

A *protocol* $P(Q, Y, T, O)$ consists of a finite set Q of states, a finite set Y of output symbols, transition function $T : Q \times Q \rightarrow Q \times Q$, and output function $O : Q \rightarrow Y$. When an interaction between two agents occurs, T determines the next states of the two agents based on their current states. The *output of an agent* is determined by O: the output of agent v with state $q \in Q$ is $O(q)$.

A *configuration* is a mapping $C : V \rightarrow Q$ that specifies the states of all the agents. We denote the set of all configurations of protocol P by $\mathcal{C}_{\text{all}}(P)$. We say that configuration C changes to C' by interaction $e = (u, v)$, denoted by $C \xrightarrow{e} C'$, if we have $(C'(u), C'(v)) = T(C(u), C(v))$ and $C'(w) = C(w)$ for all $w \in V \setminus \{u, v\}$. A scheduler determines which interaction occurs at each time. In this paper, we consider a *uniformly random scheduler* $\Gamma = \Gamma_0, \Gamma_1, \ldots$: each $\Gamma_t \in E$ is a random variable such that $\Pr(\Gamma_t = (u, v)) = 1/m$ for any $t \geq 0$ and any $(u, v) \in E$. Given an initial configuration C_0 and Γ, the *execution* of protocol P is defined as $\Xi_P(C_0, \Gamma) = C_0, C_1, \ldots$ such that $C_t \xrightarrow{\Gamma_t} C_{t+1}$ for all $t \geq 0$. We denote $\Xi_P(C_0, \Gamma)$ simply by $\Xi_P(C_0)$ when no confusion occurs.

The leader election problem requires that every agent should output L or F which means "leader" or "follower" respectively. We say that a finite or infinite sequence of configurations $\xi = C_0, C_1, \ldots$ preserves a unique leader, denoted by $\xi \in LE$, if there exists $v \in V$ such that $O(C_t(v)) = L$ and $O(C_t(u)) = F$ for any $t \geq 0$ and $u \in V \setminus \{v\}$. For $\xi = C_0, C_1, \ldots$, the holding time of the leader $\text{HT}(\xi, LE)$ is defined as the maximum $t \in \mathbb{N}$ that satisfies $(C_0, C_1, \ldots, C_{t-1}) \in LE$. We define $\text{HT}(\xi, LE) = 0$ if $C_0 \notin LE$. We denote $\mathbf{E}[\text{HT}(\Xi_P(C), LE)]$ by $\text{EHT}_P(C, LE)$. Intuitively, $\text{EHT}_P(C, LE)$ is the expected number of interactions for which the population keeps the unique leader after protocol P starts from configuration C. For configuration sequence $\xi = C_0, C_1, \ldots$ and a set of configurations $\mathcal{C}$, we define convergence time $\text{CT}(\xi, \mathcal{C})$ as the minimum $t \in \mathbb{N}$ that satisfies $C_t \in \mathcal{C}$. We define $\text{CT}(\xi, \mathcal{C}) = |\xi|$ if $C_t \notin \mathcal{C}$ for any $t \geq 0$, where $|\xi|$ is the length of ξ (i.e. the number of configurations). We denote $\mathbf{E}[\text{CT}(\Xi_P(C), \mathcal{C})]$ by $\text{ECT}_P(C, \mathcal{C})$. Intuitively, $\text{ECT}_P(C, \mathcal{C})$ is the expected number of interactions by which the population reaches a configuration in $\mathcal{C}$ when starting from C.

▶ **Definition 1** (Loose-stabilizing leader election [13]). Protocol $P(Q, Y, T, O)$ is an (α, β)-loosely-stabilizing leader election protocol if there exists set $\mathcal{S}$ of configurations satisfying $\max_{C \in \mathcal{C}_{\text{all}}(P)} \text{ECT}_P(C, \mathcal{S}) \leq \alpha$ and $\min_{C \in \mathcal{S}} \text{EHT}_P(C, LE) \geq \beta$.

Chernoff Bounds

Two variants of Chernoff bounds [11] used in several proofs of this paper are quoted below.

▶ **Lemma 2** (Eq. (4.2) in [11])**.** *The following inequality holds for any binomial random variable X and any κ, $0 < \kappa \leq 1$:*

$$\Pr(X \geq (1 + \kappa)\mathbf{E}[X]) \leq e^{-\kappa^2 \mathbf{E}[X]/3}.$$

▶ **Lemma 3** (Eq. (4.5) in [11])**.** *The following inequality holds for any binomial random variable X and κ, $0 < \kappa \leq 1$:*

$$\Pr(X \leq (1 - \kappa)\mathbf{E}[X]) \leq e^{-\kappa^2 \mathbf{E}[X]/2}.$$

3   Loosely-stabilizing Leader Election Protocol

This section presents loosely-stabilizing leader election protocol P_{AR} for arbitrary undirected anonymous graphs without identifiers or random numbers. Symmetry breaking is not a key issue to elect a leader in the population protocol model since random scheduler breaks the symmetry of the population. (Global-fairness breaks the symmetry in the case of deterministic scheduler.) The challenging issue is to reduce the number of leaders to one while avoiding to remove all leaders from the population. Protocol P_{AR} solves this issue without identifiers or random numbers by virus-propagation and token-circulation. A leader tries to kill other leaders by creating and propagating a virus while a circulating token controls the frequency of creating a virus so that eventually exactly one agent remains a leader (i.e. survives a virus war).

Protocol P_{AR} is described in Protocol 1. A state of an agent is described by a collection of variables, and a transition function is described by a pseudo code that updates variables of initiator x and responder y. We denote the value of variable `var` of agent $v \in V$ by $v.\mathtt{var}$. We also denote the value of `var` in state $q \in Q$ by $q.\mathtt{var}$. In P_{AR}, each agent has three binary variables `leader` $\in \{\top, \bot\}$, `token` $\in \{\top, \bot\}$ and `color` $\in \{\mathrm{BLACK}, \mathrm{WHITE}\}$, and four timers $\mathtt{timer_L}$, $\mathtt{timer_T}$, $\mathtt{timer_V}$ and $\mathtt{timer_E}$. The output function defines leaders based on variable `leader` : agent v is a leader if $v.\mathtt{leader} = \top$, and a follower otherwise. We say that agent v has a token if $v.\mathtt{token} = \top$ and v has a virus if $v.\mathtt{timer_V} > 0$. We also say that v is black if $v.\mathtt{color} = \mathrm{BLACK}$, and v is white otherwise.

Protocol P_{AR} consists of five parts: leader-creation (Lines 1–7), token-creation (Lines 8–14), token-circulation (Lines 15–20), virus-creation (Lines 29–37), and virus-propagation (Lines 21–28). Our goal is to elect a unique leader in the population from an arbitrary initial configuration. The leader-creation part creates a leader when no leader exists in the population. The other four parts work together to reduce the number of leaders to one when two ore more leaders exist.

The leader-creation part aims to create a leader when no leader exists in the population. Each agent uses $\mathtt{timer_L}$ as the barometer for suspecting that there exists no leader. Specifically, when initiator x and responder y interact, they take the larger value of $x.\mathtt{timer_L}$ and $y.\mathtt{timer_L}$, decrease it by one, and substitute the decreased value into $x.\mathtt{timer_L}$ and $y.\mathtt{timer_L}$ (Line 1). We call this event *larger value propagation*. If x or y is a leader, both timers are reset to $t_{\max}$ (Lines 2–3). We call this event *timer reset*. When a timer becomes zero (i.e. timeout), agents x and y suspect that there exists no leader in the population. Then, x becomes a new leader with the full timer value $t_{\max}$ (Lines 5–6). When no leader exists, the population never experiences timer reset, thus, their timers keep on decreasing. Hence, the timeout eventually

---

Algorithm 1 Leader Election P_{AR}

Variables of each agent:

   leader $\in \{\top, \bot\}$, token $\in \{\top, \bot\}$, color $\in \{\mathrm{BLACK}, \mathrm{WHITE}\}$

   $\mathtt{timer_L} \in [0, t_{\max}]$, $\mathtt{timer_T} \in [0, t_{\max}]$, $\mathtt{timer_V} \in [0, t_{\mathrm{virus}}]$, $\mathtt{timer_E} \in [0, t_{\mathrm{epi}}]$

Output function O:

   if $v.\mathtt{leader} = \top$ holds, then the output of agent v is L, otherwise F.

Interaction between initiator x and responder y:

 1: $x.\mathtt{timer_L} \leftarrow y.\mathtt{timer_L} \leftarrow \max(x.\mathtt{timer_L} - 1,\ y.\mathtt{timer_L} - 1,\ 0)$

 2: **if** $x.\mathtt{leader} = \top$ **or** $y.\mathtt{leader} = \top$ **then**

 3:     $x.\mathtt{timer_L} \leftarrow y.\mathtt{timer_L} \leftarrow t_{\max}$ // a leader resets leader timer

 4: **else if** $x.\mathtt{timer_L} = 0$ **then**

 5:     $x.\mathtt{leader} \leftarrow \top$ // a new leader is created at timeout

 6:     $x.\mathtt{timer_L} \leftarrow y.\mathtt{timer_L} \leftarrow t_{\max}$

 7: **end if**

 8: $x.\mathtt{timer_T} \leftarrow y.\mathtt{timer_T} \leftarrow \max(x.\mathtt{timer_T} - 1,\ y.\mathtt{timer_T} - 1,\ 0)$

 9: **if** $x.\mathtt{token} = \top$ **or** $y.\mathtt{token} = \top$ **then**

10:     $x.\mathtt{timer_T} \leftarrow y.\mathtt{timer_T} \leftarrow t_{\max}$ // a token resets token timer

11: **else if** $x.\mathtt{timer_T} = 0$ **then**

12:     $x.\mathtt{token} \leftarrow \top$ // a new token is created at timeout

13:     $x.\mathtt{timer_T} \leftarrow y.\mathtt{timer_T} \leftarrow t_{\max}$

14: **end if**

15: $x.\mathtt{token} \leftrightarrow y.\mathtt{token}$ // a token moves between agents

16: $x.\mathtt{timer_E} \leftarrow \max(0, y.\mathtt{timer_E} - 1)$

17: $y.\mathtt{timer_E} \leftarrow \max(0, x.\mathtt{timer_E} - 1)$ // decrement and swap epidemic timers

18: **if** $x.\mathtt{token} = \top$ **and** $y.\mathtt{token} = \top$ **then**

19:     $y.\mathtt{token} \leftarrow \bot$

20: **end if**

21: **if** $x.\mathtt{timer_V} > 0$ **and** $y.\mathtt{timer_V} = 0$ **and** $x.\mathtt{color} \neq y.\mathtt{color}$ **then**

22:     $y.\mathtt{leader} \leftarrow \bot$

23:     $y.\mathtt{color} \leftarrow x.\mathtt{color}$

24: **else if** $x.\mathtt{timer_V} = 0$ **and** $y.\mathtt{timer_V} > 0$ **and** $x.\mathtt{color} \neq y.\mathtt{color}$ **then**

25:     $x.\mathtt{leader} \leftarrow \bot$

26:     $x.\mathtt{color} \leftarrow y.\mathtt{color}$

27: **end if**

28: $x.\mathtt{timer_V} \leftarrow y.\mathtt{timer_V} \leftarrow \max(x.\mathtt{timer_V} - 1,\ y.\mathtt{timer_V} - 1,\ 0)$

29: **if** $x.\mathtt{leader} = \top$ **and** $x.\mathtt{token} = \top$ **and** $x.\mathtt{timer_E} = 0$ **then**

30:     **if** $x.\mathtt{color} = \mathrm{BLACK}$ **then** $x.\mathtt{color} \leftarrow \mathrm{WHITE}$ **else** $x.\mathtt{color} \leftarrow \mathrm{BLACK}$ **endif**

31:     $x.\mathtt{timer_V} \leftarrow t_{\mathrm{virus}}$

32:     $x.\mathtt{timer_E} \leftarrow t_{\mathrm{epi}}$

33: **else if** $y.\mathtt{leader} = \top$ **and** $y.\mathtt{token} = \top$ **and** $y.\mathtt{timer_E} = 0$ **then**

34:     **if** $y.\mathtt{color} = \mathrm{BLACK}$ **then** $y.\mathtt{color} \leftarrow \mathrm{WHITE}$ **else** $y.\mathtt{color} \leftarrow \mathrm{BLACK}$ **endif**

35:     $y.\mathtt{timer_V} \leftarrow t_{\mathrm{virus}}$

36:     $y.\mathtt{timer_E} \leftarrow t_{\mathrm{epi}}$

37: **end if**

---

occurs and a leader is created. When a leader exists, the timeout rarely happens since all agents keep high timer values thanks to the timer reset and the larger value propagation. Therefore, this mechanism rarely ruins stability of the unique leader.

Protocol P_{AR} reduces the number of leaders to one as follows. The token-creation part creates a token when no token exists in the population; The token-circulation part reduces the number of tokens to one, circulates the unique token among the population, and decrements the epidemic timer ($\mathtt{timer_E}$) of the unique token every time it moves; The virus-creation part creates a new virus when a leader meets a token with epidemic timer of value zero; The virus-propagation part propagates the virus to the whole population, which changes leader agents to follower agents.

The token-creation part (Lines 8–14) creates a token in the same way as the leader-creation part when no token exists in the population. There is no difference between the two parts except that the former uses variable $\mathtt{timer_T}$ while the latter uses $\mathtt{timer_L}$.

The token-circulation part (Lines 15–20) aims to reduce the number of tokens to one, and circulates the unique token. A token moves between agents by interaction (Line 15). We can say that a token makes a random walk among the population since the scheduler randomly chooses two agents to interact at each time. Hence, two tokens eventually meet if two or more tokens exist in the population. When two agents interact and both agents have tokens, then either one of the two loses its token (Lines 18–20). Hence, the number of tokens eventually becomes one. Each token has an epidemic timer ($\mathtt{timer_E}$). The epidemic timer is decremented by one every time the token moves, and thus, it becomes zero eventually (Line 16–17). Note that the number of tokens never becomes zero once a token exists since the number of tokens decreases only when two tokens meet at an interaction.

A virus-creation part (Lines 29–37) creates a new virus when a leader meets a token with an epidemic timer of value zero. We call this event *virus creation*. Specifically, if a token with $\mathtt{timer_E} = 0$ moves to a leader agent, the leader changes its color from black to white or from white to black (Lines 30 and 34) and creates a new virus with full value TTL (Time To Live), i.e. $\mathtt{timer_V} = t_{\mathrm{virus}}$ (Lines 31 and 35). The leader also resets the epidemic timer of the token (Lines 32 and 36), which enables periodical occurrence of epidemics.

A virus-propagation part (Lines 21–28) propagates a virus from agent to agent and reduces the number of leaders. When an agent has a virus (i.e. $v.\mathtt{timer_V} > 0$), we regard that $v.\mathtt{timer_V}$ is the TTL of the virus. A virus vanishes from the agent when its TTL becomes zero. In the same way as $\mathtt{timer_L}$ and $\mathtt{timer_T}$, a virus propagates at interaction in the larger value propagation fashion (Line 28). Moreover, a virus has the power to change the colors of agents and kill leaders. Specifically, if an agent with a virus interacts an agent without a virus, the virus changes the color of the newly infected agent (Lines 23 and 26). At this time, if the newly infected agent is a leader, the virus kills the leader (i.e. changes the newly infected agent from a leader to a follower). Once a new virus is created at the virus-creation part, the virus propagates to the whole population within a short time. However, the value of $\mathtt{timer_V}$ is reset only when a new virus is created. Hence, viruses eventually vanish from the population if the frequency of epidemics, controlled by the value t_{epi}, is sufficiently low. The concept of colors helps to avoid the suicide of leaders, i.e. a leader is rarely killed by a virus that it creates. Consider that a white leader creates a virus. After that, the leader and any infected agent with the virus are black, thus the leader is never killed by the virus until another virus is created and the leader becomes white.

Protocol P_{AR} correctly works if $t_{\max}$ and t_{virus} is sufficiently large and t_{epi} is sufficiently greater than t_{virus}. When there exists no leader, the leader-creation part eventually creates a leader by timeout. In the following, let us consider the case that multiple leaders exist in the

population, and see how P_{AR} reduces these leaders to one. The token-creation and the token circulation parts eventually create the unique token and circulate it in the population. Since t_{epi} is sufficiently greater than t_{virus}, the population eventually reaches a configuration where no agent has virus. After that, the epidemic timer of the token keeps on decreasing and eventually becomes zero, and the token eventually moves to a leader in the population, which creates a new virus. This virus soon propagates among the whole population and turn all the agents to the ones with the same color (black or white). Let the color be black without loss of generality. Again, the virus vanishes, the epidemic timer of the token becomes zero, and the token moves to a leader in the same way. Then, the black leader becomes white and creates a new virus. It soon propagates to the whole population and changes all agents from black to white, which kills all other leaders. Then, we have the exactly one leader in the population.

Even after we have exactly one leader and one token, the population sometimes enters the wrong configuration where no leader exists, multiple leaders exist, or multiple tokens exist. These deviations are caused by the following events: (i) leader timeout happens, (ii) token timeout happens, or (iii) a new virus is created when viruses remain in the population. Cases (i) and (ii) rarely happens thanks to the timer reset, the larger value propagation, and the sufficiently large t_{max}, which is the reset value of leader timers and token timers. Case (iii) also rarely happens because t_{epi}, the reset value of the epidemic timer, is sufficiently larger than the reset value of a virus timer t_{virus}. As we shall see later, the expected time from a safe configuration to such a wrong configuration is exponential.

4   Complexity Analysis

This section analyzes the expected holding time and the expected convergence time of P_{AR}. Due to the lack of space, we present only proof sketches for the analyses of the expected convergence time. Complete proofs are left to the full paper. Notations and assumptions used in this paper are summarized in Table 2.

We have three parameters in P_{AR}: the reset values of timers t_{max}, t_{virus}, and t_{epi}. We mentioned that P_{AR} correctly works if t_{max} and t_{virus} is sufficiently large and t_{epi} is sufficiently greater than t_{virus}. Specifically, we assume $t_{\mathrm{max}} \geq 8\delta \max(d, \lceil 2\log mn^3 d \rceil)$, $t_{\mathrm{virus}} = t_{\mathrm{max}}/2$, and $t_{\mathrm{epi}} \geq 4\delta t_{\mathrm{max}} \lceil \log n \rceil$ where δ is the maximum degree of the agents and d is the diameter of population G. (Note that δ is an even number because G is undirected, i.e. $(u, v) \in E \Leftrightarrow (v, u) \in E$.) We also assume that t_{epi} is not extremely large: $t_{\mathrm{epi}} \leq \tau e^\tau/(9n)$ where $\tau = \lfloor t_{\mathrm{max}}/(8\delta) \rfloor$. Otherwise, even if a leader exists, the leader timeout happens with non-negligible probability within an exponentially long epidemic interval. This means that the protocol may not reduce the number of leaders to one at the convergence step. We also assume $n \geq 3$ because P_{AR} is obviously a self-stabilizing leader election protocol when $n = 2$. In the rest of this section, we prove the following equations under these assumptions:

$$\max_{C \in \mathcal{C}_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}}) = O(mn^3 d + mt_{\mathrm{epi}}), \tag{1}$$

$$\min_{C \in \mathcal{S}_{\mathrm{AR}}} \mathrm{EHT}_{P_{\mathrm{AR}}}(C, LE) = \Omega(\tau e^\tau), \tag{2}$$

where $\mathcal{S}_{\mathrm{AR}}$ is the set of configurations we define later. When upper bounds N and Δ of n and δ are available and we assign $t_{\mathrm{max}} = 8\Delta \max(N, \lceil 12\log N \rceil)$, $t_{\mathrm{epi}} = 4\Delta t_{\mathrm{max}} \lceil \log N \rceil$, then P_{AR} is an $(O(mn^3 d + mN\Delta^2 \log N), \Omega(Ne^N))$-loosely-stabilizing leader election protocol. (Note that this assignment satisfies the above assumptions.)

■ **Table 2** Notations and Assumptions for P_{AR}.

Notations	
$\tau:$	$\lfloor t_{\max}/(8\delta)\rfloor$
$\#_L(C):$	the number of leaders in configuration C
$\#_T(C):$	the number of tokens in configuration C
$\mathcal{L}_{\mathrm{one}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_L(C) = 1\}$
$\mathcal{T}_{\mathrm{one}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_T(C) = 1\}$
$\mathcal{L}_{\mathrm{exist}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_L(C) \geq 1\}$
$\mathcal{T}_{\mathrm{exist}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_T(C) \geq 1\}$
$\mathcal{L}_{\mathrm{half}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_L} > t_{\max}/2\}$
$\mathcal{T}_{\mathrm{half}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_T} > t_{\max}/2\}$
$\mathcal{V}_{\mathrm{same}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \exists u, \forall v \in V, \ C(u).\mathtt{leader} = \top$
	$\qquad\qquad \land\, (C(v).\mathtt{timer_V} > 0 \Rightarrow C(u).\mathtt{color} = C(v).\mathtt{color})\}$
$\mathcal{V}_{\mathrm{zero}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_V} = 0\}$
$\mathcal{E}_{\mathrm{half}}:$	$\{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{token} = \top \Rightarrow C(v).\mathtt{timer_E} > t_{\mathrm{epi}}/2\}$
$\mathcal{S}_{\mathrm{AR}}:$	$\mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{same}} \cap (\mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}})$
$\mathrm{PROP}_L(i):$	$C_{2m\tau(i+1)} \in \mathcal{L}_{\mathrm{half}} \lor C_{2m\tau i} \notin \mathcal{L}_{\mathrm{exist}}$
$\mathrm{PROP}_T(i):$	$C_{2m\tau(i+1)} \in \mathcal{T}_{\mathrm{half}} \lor C_{2m\tau i} \notin \mathcal{T}_{\mathrm{exist}}$
$\mathrm{HALF}(i):$	$\mathrm{HALF}(i) = 1$ if every agent joins only less than $t_{\max}/2$ interactions
	among $\Gamma_{2m\tau i}, \dots, \Gamma_{2m\tau(i+1)-1}$, otherwise $\mathrm{HALF}(i) = 0$.
$\#_{TI}(v, t_1, t_2):$	the number of interactions among $\Gamma_{t_1}, \dots, \Gamma_{t_2}$ involving the token that agent v has
	in configuration C_{t_1} (The definition is given just after Lemma 6.)
Assumptions	
	$n \geq 3$
	$t_{\max} \geq 8\delta \max(d, \ \lceil 2\log mn^3 d \rceil)$
	$t_{\mathrm{virus}} = t_{\max}/2$
	$4\delta t_{\max}\lceil \log n \rceil \leq t_{\mathrm{epi}} \leq \tau e^\tau/(9n)$

Before proving equations (1) and (2), we define ten sets of configurations:

$$\mathcal{L}_{\mathrm{one}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_L(C) = 1\},$$

$$\mathcal{T}_{\mathrm{one}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_T(C) = 1\},$$

$$\mathcal{L}_{\mathrm{exist}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_L(C) \geq 1\},$$

$$\mathcal{T}_{\mathrm{exist}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \#_T(C) \geq 1\},$$

$$\mathcal{L}_{\mathrm{half}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_L} > t_{\max}/2\},$$

$$\mathcal{T}_{\mathrm{half}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_T} > t_{\max}/2\},$$

$$\mathcal{V}_{\mathrm{same}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \exists u, \forall v \in V, \ C(u).\mathtt{leader} = \top$$
$$\qquad\qquad\qquad \land\, (C(v).\mathtt{timer_V} > 0 \Rightarrow C(u).\mathtt{color} = C(v).\mathtt{color})\},$$

$$\mathcal{V}_{\mathrm{zero}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{timer_V} = 0\},$$

$$\mathcal{E}_{\mathrm{half}} = \{C \in \mathcal{C}_{\mathrm{all}}(P_{\mathrm{AR}}) \mid \forall v \in V, \ C(v).\mathtt{token} = \top \Rightarrow C(v).\mathtt{timer_E} > t_{\mathrm{epi}}/2\},$$

$$\mathcal{S}_{\mathrm{AR}} = \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{same}} \cap (\mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}})$$

where $\#_L(C)$ and $\#_T(C)$ denote the number of leaders and tokens in configuration C, respectively. Note that $\mathcal{V}_{\mathrm{same}}$ is the set of configurations where there exists a leader agent such that every agent with a virus has the same color as the leader, and $\mathcal{E}_{\mathrm{half}}$ is the set of configurations where every token has the epidemic timer whose value is greater than $t_{\mathrm{epi}}/2$.

First, we analyze the expected holding time. Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. To prove (2), it is sufficient to show that both (i) $C_0, \ldots, C_{8m\delta\tau\lceil \log n \rceil} \in LE$ and (ii) $C_{8m\delta\tau\lceil \log n \rceil} \in \mathcal{S}_{\mathrm{AR}}$ hold with probability no less than $p_{\mathrm{suc}} = 1 - O(n\delta \log n \cdot e^{-\tau})$. Then, letting $A = \min_{C_0 \in \mathcal{S}_{\mathrm{AR}}} \mathrm{EHT}_{P_{\mathrm{AR}}}(C_0, LE)$, we have $A \geq 8m\delta\tau\lceil \log n \rceil p_{\mathrm{suc}}/(1 - p_{\mathrm{suc}}) = \Omega(\tau e^{\tau})$, since $A \geq p_{\mathrm{suc}}(8m\delta\tau\lceil \log n \rceil + A)$. We give five conditions such that satisfying all the conditions leads to above conditions (i) and (ii) (Lemma 10). After that, we analyze the probability that all the five conditions hold and prove that the probability is no less than $1 - O(n\delta \log n \cdot e^{-\tau})$.

We define three predicates $\mathrm{PROP}_L(i)$, $\mathrm{PROP}_T(i)$ and $\mathrm{HALF}(i)$ for any $i \geq 0$: $\mathrm{PROP}_L(i) = 1$ if $C_{2m\tau(i+1)} \in \mathcal{L}_{\mathrm{half}}$ or $C_{2m\tau i} \notin \mathcal{L}_{\mathrm{exist}}$, otherwise $\mathrm{PROP}_L(i) = 0$; $\mathrm{PROP}_T(i) = 1$ if $C_{2m\tau(i+1)} \in \mathcal{T}_{\mathrm{half}}$ or $C_{2m\tau i} \notin \mathcal{T}_{\mathrm{exist}}$, otherwise $\mathrm{PROP}_T(i) = 0$; $\mathrm{HALF}(i) = 1$ if every agent joins less than $t_{\max}/2$ interactions among $\Gamma_{2m\tau i}, \ldots, \Gamma_{2m\tau(i+1)-1}$, otherwise $\mathrm{HALF}(i) = 0$. Intuitively, $\mathrm{PROP}_L(i) = 1$ ($\mathrm{PROP}_T(i) = 1$) means that high value of $\mathtt{timer_L}$ ($\mathtt{timer_T}$) propagates from a leader (a token, respectively) to all the agents during $2m\tau$ interactions, and $\mathrm{HALF}(i) = 1$ means every agent does not interact so much during $2m\tau$ interactions. Note that $\mathrm{PROP}_L(i) = 1$ ($\mathrm{PROP}_T(i) = 1$) unconditionally holds when there exists no leader (token, respectively) in $C_{2m\tau i}$. In addition, we define binary random variable $TO_L(C_0, t_1, t_2)$ and $TO_T(C_0, t_1, t_2)$ for integers t_1 and t_2, $(0 \leq t_1 \leq t_2)$ as follows: $TO_L(C_0, t_1, t_2) = 1$ if there exists integer i ($t_1 \leq i < t_2$) satisfying $\#_L(C_i) < \#_L(C_{i+1})$, otherwise $TO_L(C_0, t_1, t_2) = 0$; $TO_T(C_0, t_1, t_2) = 1$ if there exists integer i ($t_1 \leq i < t_2$) satisfying $\#_T(C_i) < \#_T(C_{i+1})$, otherwise $TO_T(C_0, t_1, t_2) = 0$. Intuitively, variable $TO_L(C_0, t_1, t_2)$ (variable $TO_T(C_0, t_1, t_2)$) represents whether an interaction among $\Gamma_{t_1}, \ldots, \Gamma_{t_2-1}$ trigger the leader timeout (the token timeout, respectively) or not.

▶ **Lemma 4.** *Let $C_0 \in \mathcal{L}_{\mathrm{half}} \cap \mathcal{L}_{\mathrm{exist}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. We have $C_{2m\tau} \in \mathcal{L}_{\mathrm{half}}$ and $TO_L(C_0, 0, 2m\tau) = 0$ if $\mathrm{PROP}_L(0) = \mathrm{HALF}(0) = 1$.*

Proof. Since there exists a leader in C_0, $\mathrm{PROP}_L(0) = 1$ assures $C_{2m\tau} \in \mathcal{L}_{\mathrm{half}}$. Assumptions $C_0 \in \mathcal{L}_{\mathrm{half}}$ and $\mathrm{HALF}(0) = 1$ assures that the leader timeout does not happen by $\Gamma_0, \ldots, \Gamma_{2m\tau-1}$. ◀

▶ **Corollary 5.** *Let $C_0 \in \mathcal{L}_{\mathrm{half}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let $k \geq 1$ be any integer. We have $C_{2m\tau k} \in \mathcal{L}_{\mathrm{half}}$ and $TO_L(C_0, 0, 2m\tau k) = 0$ if $\mathrm{PROP}_L(i) = \mathrm{HALF}(i) = 1$ and $C_{2m\tau i} \in \mathcal{L}_{\mathrm{exist}}$ hold for all $i = 0, 1, \ldots, k-1$.*

Once a token exist in the population, the number of tokens never become zero after that. Hence, we have a simpler lemma as for the token timeout.

▶ **Lemma 6.** *Let $C_0 \in \mathcal{T}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{exist}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let $k \geq 1$ be any integer. We have $C_{2m\tau k} \in \mathcal{T}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{exist}}$ and $TO_T(C_0, 0, 2m\tau k) = 0$ if $\mathrm{PROP}_T(i) = \mathrm{HALF}(i) = 1$ holds for all $i = 0, 1, \ldots, k-1$.*

For agent $v \in V$ and integers t_1 and t_2, $(0 \leq t_1 < t_2)$, we define $\#_{TI}(v, t_1, t_2) = |\{t \in [t_1 + 1, t_2] \mid v_t \neq v_{t-1}\}|$ where $v_{t_1} = v$, and

$$v_t = \begin{cases} u & \text{if } \Gamma_{t-1} = (u, v_{t-1}) \\ w & \text{if } \Gamma_{t-1} = (v_{t-1}, w) \\ v_{t-1}. & \text{otherwise} \end{cases}$$

for $t > t_1$. Random variable $\#_{TI}(v, t_1, t_2)$ has a intuitive meaning if v has a token when interaction Γ_t occurs: Intuitively, $\#_{TI}(v, t_1, t_2)$ represents the number of interactions that the token involves during $\Gamma_{t_1}, \ldots, \Gamma_{t_2-1}$ (or the number of times the token moves during the period).

▶ **Lemma 7.** *Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let v_T be the agent that has the unique token in configuration C_0, and $t \geq 0$ be a non-negative integer. Then, we have $C_i \in \mathcal{V}_{\mathrm{same}}$ for all $i = 0, 1, \ldots, t$ if we have $\#_{TI}(v_T, 0, t) < t_{\mathrm{epi}}/2$ and $C_i \in \mathcal{T}_{\mathrm{one}}$ for all $i = 0, 1, \ldots, t$.*

Proof. Let v_L be the unique leader in configuration C_0, and we assume that the color of v_L and all agents with viruses are black without loss of generality (Note that $C_0 \in \mathcal{V}_{\mathrm{same}}$). Since $C_0 \in \mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}$, we prove the lemma for two cases $C_0 \in \mathcal{E}_{\mathrm{half}}$ and $C_0 \in \mathcal{V}_{\mathrm{zero}}$. In case $C_0 \in \mathcal{E}_{\mathrm{half}}$, the epidemic timer of the unique token never becomes zero in $C_0, \ldots, C_t$ because $\#_{TI}(v_T, 0, t) < t_{\mathrm{epi}}/2$. Therefore, a new virus is not created during $C_0, \ldots, C_t$, which assures that v_l and all agents with viruses are still black in $C_0, \ldots, C_t$. Thus, we have $C_i \in \mathcal{V}_{\mathrm{same}}$ for all $i = 1, 2, \ldots, t$. In case $C_0 \in \mathcal{V}_{\mathrm{zero}}$, the virus creation happens at most once during $C_0, \ldots, C_t$ because $\#_{TI}(v_T, 0, t), < t_{\mathrm{epi}}/2$ and $C_i \in \mathcal{T}_{\mathrm{one}}$ for all $i = 0, 1, \ldots, t$. If the virus creation does not happen, $C_i \in \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}} \subseteq \mathcal{V}_{\mathrm{same}}$ holds for all $i = 0, 1, \ldots, t$. If a leader meets a token with an epidemic timer of value zero and creates a new virus, the virus propagates from agent to agent. However, the virus makes all infected agents the same color as the leader that creates the virus, which assures $C_i \in \mathcal{V}_{\mathrm{same}}$ for all $i = 0, 1, \ldots, t$. ◀

The following lemma is directly obtained from Corollary 5 and Lemma 7.

▶ **Lemma 8.** *Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let v_T be the agent that has the unique token in configuration C_0, and $k \geq 0$ be any integer. Then, we have $C_{2m\tau k} \in \mathcal{L}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{same}}$ and $C_i \in \mathcal{L}_{\mathrm{one}}$ for all $i = 0, 1, \ldots, 2m\tau k$ if we have $\mathrm{PROP}_L(j) = \mathrm{HALF}(j) = 1$ for all $j = 0, 1, \ldots, k-1$, $\#_{TI}(v_T, 0, 2m\tau k) < t_{\mathrm{epi}}/2$, and $C_i \in \mathcal{T}_{\mathrm{one}}$ for all $i = 0, 1, \ldots, 2m\tau k$.*

We define the first round time $\mathrm{RT}_\Gamma(1)$ as the minimum t satisfying $\forall e \in E$, $0 \leq \exists t' \leq t$, $\Gamma_{t'} = e$. For any $i \geq 2$, we define the i-th round time $\mathrm{RT}_\Gamma(i)$ as the minimum t satisfying $\forall e \in E$, $\mathrm{RT}_\Gamma(i-1) < \exists t' \leq t$, $\Gamma_{t'} = e$.

▶ **Lemma 9.** *Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let $t \geq 0$ be any integer. We have $C_t \in \mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}$ if we have $\mathrm{RT}_\Gamma(t_{\mathrm{virus}}) < t$, $\#_{TI}(v_T, 0, t) < t_{\mathrm{epi}}/2$, and $\#_T(C_i) = 1$ for all $i = 0, 1, \ldots, t$.*

Proof. If a new virus is not created among $\Gamma_0, \ldots, \Gamma_t$, then all viruses in the initial configuration vanish during the period since each round decreases the maximum value of $\mathtt{timer_V}$ by at least one. Thus, $C_t \in \mathcal{V}_{\mathrm{zero}}$ holds. If some agent v creates a new virus at $\Gamma_{t'}$, then the epidemic timer of the unique token are reset at the same time. (Note that the unique token always exist in the population by the assumption of the lemma.) Thus, we have $C_{t'}(v).\mathtt{timer_E} = t_{\mathrm{epi}}$. Since $\#_{TI}(v, t', t) \leq \#_{TI}(v_T, 0, t) < t_{\mathrm{epi}}/2$, the epidemic timer of the unique token is no less than $t_{\mathrm{epi}} - t_{\mathrm{epi}}/2 = t_{\mathrm{epi}}/2$, which means $C_t \in \mathcal{E}_{\mathrm{half}}$. ◀

▶ **Lemma 10.** *Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Let v_T be the agent that has the unique token in configuration C_0. Then, we have both $C_0, \ldots, C_{8m\delta\tau\lceil \log n\rceil} \in LE$ and $C_{8m\delta\tau\lceil \log n\rceil} \in \mathcal{S}_{\mathrm{AR}}$ if the following conditions hold:*
(A) $\#_{TI}(v_T, 0, 8m\delta\tau\lceil \log n\rceil) < t_{\mathrm{epi}}/2$,
(B) $\mathrm{PROP}_L(i) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \log n\rceil - 1$,
(C) $\mathrm{PROP}_T(i) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \log n\rceil - 1$,
(D) $\mathrm{HALF}(i) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \log n\rceil - 1$, and
(E) $\mathrm{RT}_\Gamma(t_{\mathrm{virus}}) < 8m\delta\tau\lceil \log n\rceil$.

Proof. Assigning $k = 4\delta\lceil \log n\rceil$, we obtain $C_{8m\delta\tau\lceil \log n\rceil} \in \mathcal{T}_{\mathrm{half}}$ and $C_j \in \mathcal{T}_{\mathrm{one}}$ for all $j = 0, 1, \ldots, 8m\delta\tau\lceil \log n\rceil$ by Lemma 6 and Conditions (C) ad (D). From Lemma 8 and Conditions

(A), (B), and (D), the unique token assures that $C_{8m\delta\tau\lceil\log n\rceil} \in \mathcal{L}_{\text{half}} \cap \mathcal{V}_{\text{same}}$ and $C_j \in \mathcal{L}_{\text{one}}$ holds for $j = 0, 1, \ldots, 8m\delta\tau\lceil\log n\rceil$. Note that $C_j \in \mathcal{L}_{\text{one}}$ ($j = 0, 1, \ldots, 8m\delta\tau\lceil\log n\rceil$) guarantees not only that the number of leaders is one, but also that the unique leader is stable (i.e. $\exists v \in V, \forall i \in [0, 8m\delta\tau\lceil\log n\rceil], C_i(v).leader = \top$) because P_{AR} does not move the leader role from agent to agent at any one interaction. Hence, we have $C_0, \ldots, C_{8m\delta\tau\lceil\log n\rceil} \in LE$. We have $C_{8m\delta\tau\lceil\log n\rceil} \in \mathcal{E}_{\text{half}} \cup \mathcal{V}_{\text{zero}}$ from Lemma 9, Condition (A), Condition (E), and $C_j \in \mathcal{T}_{\text{one}}$ for all $j = 0, 1, \ldots, 8m\delta\tau\lceil\log n\rceil$. Thus, we have shown that $C_{8m\delta\tau\lceil\log n\rceil} \in \mathcal{L}_{\text{one}} \cap \mathcal{T}_{\text{one}} \cap \mathcal{L}_{\text{half}} \cap \mathcal{T}_{\text{half}} \cap \mathcal{V}_{\text{same}} \cap (\mathcal{E}_{\text{half}} \cup \mathcal{V}_{\text{zero}}) \subseteq \mathcal{S}_{\text{AR}}$ ◄

▶ **Lemma 11.** *Let $C_0 \in \mathcal{T}_{\text{one}}$ and $\Xi_{P_{\text{AR}}}(C_0) = C_0, C_1, \ldots$. Let v_T be the agent that has the unique token in configuration C_0. Then, we have $\Pr(\#_{TI}(v_T, 0, 8m\delta\tau\lceil\log n\rceil) < t_{\text{epi}}/2) \geq 1 - e^{-\delta\tau}$.*

Proof. For every $i \geq 0$, the token joins interaction Γ_i with probability at most δ/m regardless of the location of the token in C_i because any agent has at most δ edges. Thus, $\#_{TI}(v_T, 0, 8m\delta\tau\lceil\log n\rceil)$ is bounded by binomial random variable $X \sim B(8m\delta\tau\lceil\log n\rceil, \delta/m)$. We have

$$\begin{aligned}
\Pr(X \geq t_{\text{epi}}/2) &\leq \Pr(X \geq 16\delta^2\tau\lceil\log n\rceil) &&\because t_{\text{epi}} \geq 32\delta^2\tau\lceil\log n\rceil \\
&= \Pr(X \geq 2\mathbf{E}[X]) \\
&\leq e^{-\mathbf{E}[X]/3} &&\text{(By Chernoff Bound of Lemma 2 with } \kappa = 1) \\
&= e^{-8\delta^2\tau\lceil\log n\rceil/3} \\
&\leq e^{-\delta\tau},
\end{aligned}$$

which gives the lemma.   ◄

▶ **Lemma 12.** $\Pr(\text{PROP}_L(i) = 1) \geq 1 - 2ne^{-\tau}$ *for any $i \geq 0$.*

Proof. We assume $i = 0$ without loss of generality, and prove $\Pr(\text{PROP}_L(0) = 1) \geq 1 - 2ne^{-\tau}$. We have $\text{PROP}_L(0) = 1$ by the definition of PROP_L if no leader exists in C_0. Thus, it suffices to show $\Pr(C_{2m\tau}(v).\texttt{timer}_L > t_{\text{max}}/2) \geq 1 - 2e^{-\tau}$ for any agent $v \in V$ in case $C_0 \in \mathcal{L}_{\text{exist}}$. Let v_L be a leader agent in C_0. We denote the shortest path from v_L to v by $(v_0, v_1, \ldots, v_s)$ where $v_0 = v_L$, $v_s = v$, $0 \leq s \leq d$ and $(v_{j-1}, v_j) \in E$ for all $j = 1, 2, \ldots, s$. For any $t = 0, 1, \ldots, 2m\tau$, we define $v_{\text{head}}(t)$ as v_h with maximum $h \in [1, s]$ such that there exist $t_1, t_2, \ldots, t_h$ satisfying $0 \leq t_1 < t_2 < \cdots < t_h < t$ and $\Gamma_{t_j} \in \{(v_{j-1}, v_j), (v_j, v_{j-1})\}$ for $j = 1, 2, \ldots, h$. We define $v_{\text{head}}(t) = v_0$ if such h does not exist. Intuitively, $v_{\text{head}}(t)$ is the head of the agents in path $(v_0, v_1, \ldots, v_l)$ to which a large value of $\texttt{timer}_L$ is propagated from v_L to v. (Remind that v_L resets $\texttt{timer}_L$ to t_{max}.) We define $J(t)$ as the number of integers $j \in [0, \ldots, 2m\tau - 1]$ such that $v_{\text{head}}(j)$ joins interaction Γ_j. Intuitively, $J(t)$ is the number of interactions that the head agent joins among $\Gamma_0, \ldots, \Gamma_{2m\tau-1}$. Obviously, we have $C_t(v_{\text{head}}(t)).\texttt{timer} \geq t_{\text{max}} - J(t)$ for any $t = 0, 1, \ldots, 2m\tau$.

In what follows, we prove $\Pr(v_{\text{head}}(2m\tau) = v) \geq 1 - e^{-\tau}$ and $\Pr(J(2m\tau) < t_{\text{max}}/2) \geq 1 - e^{-\tau}$, which give $\Pr(C_{2m\tau}(v).\texttt{timer} > t_{\text{max}}/2) \geq 1 - 2e^{-\tau}$. For any $j = 1, \ldots, s$, a pair v_{j-1} and v_j interacts with probability $2/m$ at each interaction. Hence, we can say each interaction makes v_{head} forward with probability $2/m$. Therefore, by letting Z be a binomial

random variable such that $Z \sim B(2m\tau, 2/m)$, we have

$$
\begin{aligned}
\Pr(v_{\text{head}}(2m\tau) = v) &= 1 - \Pr(Z < s) \\
&\geq 1 - \Pr(Z < d) \\
&\geq 1 - \Pr\left(Z < \frac{1}{4} \cdot \mathbf{E}[Z]\right) && \because d \leq \tau = \frac{1}{4} \cdot \mathbf{E}[Z] \\
&\geq 1 - e^{-9\mathbf{E}[Z]/32} && \text{(By Chernoff bound of Lemma 3 with } \kappa = \frac{3}{4}) \\
&> 1 - e^{-\tau}.
\end{aligned}
$$

The probability that $v_{\text{head}}(t)$ joins interaction Γ_t is at most δ/m regardless of t. Hence, by letting Z' be a binomial random variable such that $Z' \sim B(2m\tau, \delta/m)$, we have

$$
\begin{aligned}
\Pr(J(2m\tau) < t_{\max}/2) &> 1 - \Pr(Z' \geq t_{\max}/2) \\
&> 1 - \Pr(Z' \geq 2\mathbf{E}[Z']) \\
&> 1 - e^{-\mathbf{E}[Z']/3} && \text{(By Chernoff bound of Lemma 2 with } \kappa = 1) \\
&= 1 - e^{-2\delta\tau/3} \\
&> 1 - e^{-\tau}. && \because \delta \geq 2
\end{aligned}
$$

Thus, we have shown $\Pr(C_{2m\tau}(v).\mathtt{timer_L} > t_{\max}/2) \geq 1 - 2e^{-\tau}$. ◄

▶ **Lemma 13.** $\Pr(\text{PROP}_T(i) = 1) \geq 1 - 2ne^{-\tau}$ *for any* $i \geq 0$.

Proof. The same argument as the proof of Lemma 12 gives the lemma. ◄

▶ **Lemma 14** (in [14]). *The probability that every* $v \in V$ *interacts only less than* $t_{\max}/2$ *times during* $2m\tau$ *interactions is at least* $1 - ne^{-\tau}$.

Proof. For any $v \in V$ and $i \geq 0$, v joins interaction Γ_i with probability at most δ/m. Thus, the number of interactions v joins during the $2m\tau$ interactions is bounded by binomial random variable $X \sim B(2m\tau, \delta/m)$. Applying Chernoff bound of Lemma 2 with $\kappa = 1$, we have

$$
\begin{aligned}
\Pr(X \geq t_{\max}/2) &\leq \Pr(X \geq 2\mathbf{E}[X]) && \because t_{\max} \geq 8\delta\tau \\
&\leq e^{-\mathbf{E}[X]/3} \\
&= e^{-2\delta\tau/3} && \text{(By Chernoff Bound of Lemma 2 with } \kappa = 1) \\
&\leq e^{-\tau}. && \because \delta \geq 2
\end{aligned}
$$

Summing up the probabilities for all $v \in V$ gives the lemma. ◄

▶ **Lemma 15** (in [14]). $\Pr(\text{HALF}(i) = 1) \geq 1 - ne^{-\tau}$ *for any* $i \geq 0$.

Proof. Each interaction is independent. Thus, Lemma 14 gives the lemma. ◄

▶ **Lemma 16** (in [14]). $\Pr(\text{RT}_\Gamma(i) < im(1 + \lceil \log n \rceil)) \geq 1 - ne^{-i/4}$ *holds for any* $i \geq 1$.

Proof. The proof in [14] can be used with slight modification. ◄

▶ **Lemma 17.** $\Pr(\text{RT}_\Gamma(t_{\text{virus}}) < 8m\delta\tau\lceil \log n \rceil) \geq 1 - ne^{-\delta(\tau+1)}$ *holds.*

Proof. By Lemma 16, we have

$$\Pr(\mathrm{RT}_\Gamma(t_{\mathrm{virus}}) < 8m\delta\tau\lceil\log n\rceil) \geq \Pr(\mathrm{RT}_\Gamma(4\delta(1+\tau)) < 8m\delta\tau\lceil\log n\rceil)$$
$$\geq \Pr(\mathrm{RT}_\Gamma(4\delta(1+\tau)) < 4m\delta(1+\tau)(1+\lceil\log n\rceil))$$
$$\geq 1 - ne^{-\delta(\tau+1)}$$

where we use $t_{\mathrm{virus}} \leq 4\delta(1+\tau)$ for the first inequality, and use $(1+\tau)(1+\lceil\log n\rceil) \leq 2\tau\lceil\log n\rceil$ when $\tau \geq 3$ and $n \geq 3$ for the second inequality. (Note that $\tau \geq \lceil 2\log mn^3 d\rceil \geq 10$.) ◄

▶ **Lemma 18.** $\min_{C\in\mathcal{S}_{\mathrm{AR}}} \mathrm{EHT}_{P_{\mathrm{AR}}}(C, LE) = \Omega(\tau e^\tau)$.

Proof. Probability p_{suc}, discussed in the beginning of this section, is at least $1 - e^{-\delta\tau} - 4\delta\lceil\log n\rceil(2ne^{-\tau} + 2ne^{-\tau} + ne^{-\tau}) - ne^{-\delta(\tau+1)} \geq 1 - 22n\delta\lceil\log n\rceil e^{-\tau}$ by Lemmas 10, 11, 12, 13, 15, and 17, which leads to the lemma. ◄

Next, we analyze the expected convergence time.

▶ **Lemma 19.** $\max_{C\in\mathcal{C}_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}}) = O(mt_{\mathrm{epi}} + mn^3 d)$.

Proof Sketch. In an execution of P_{AR}, the population converges to $\mathcal{S}_{\mathrm{AR}}$ starting from any configuration through the following convergence steps: (i) a token is created even when no token exists, (ii) the number of tokens become one, i.e. the unique token is elected, (iii) all viruses vanish from the population, (iv) the epidemic timer of the unique token becomes zero, (v) the unique token meets a leader and a new virus is created, (vi) a newly created virus propagates to the whole population and changes all agents to the ones with the same color (Let the color be black without loss of generality), (vii) the epidemic timer of the unique token becomes zero, (viii) the unique token meets a leader and a new virus is created, (ix) a newly created virus propagates to the whole population and makes all agents white, which kills all leaders other than the leader that creates the virus, and the population enters $\mathcal{S}_{\mathrm{AR}}$. Steps (ii), (iv) and (vii) require the dominant number of interactions. We will prove that the expected number of interactions until two tokens meet is $O(mn^2 d)$ in Lemma 20. The number of tokens is at most n, and the token timeout, which is the only event that increases the number of tokens, rarely happens once a token exists. Hence, the expected number of interactions Step (ii) requires is $O(mn^3 d)$. The expected number of interactions Step (iv) and (vii) require is $O(mt_{\mathrm{epi}})$ because the epidemic timer decreases by one as the token joins an interaction, and the unique token joins each interaction Γ_t with probability at least $2/m$. ◄

▶ **Lemma 20.** *Let C_0 be a configuration where two or more tokens exist. In execution $\Xi_{P_{\mathrm{AR}}}(C_0)$, the expected number of interactions until two tokens meet is at most $mn^2 d/2$.*

Proof. Let $u, v \in V$ be the distinct two agents both of which have tokens in C_0. We analyze the expected number of interactions until the two tokens meet. (One of the two tokens may vanish by meeting another token, however, this just reduces the expected number of interactions until any two tokens meet.) Consider the pair of random walks by the two tokens on population G, i.e. a Markov chain (u_t, v_t) in which the states of the chain are pairs of the agents in G. We denote $(a, b) \to (c, d)$ for agents $a, b, c, d \in V$ if $(a, c) \in E \wedge b = d$, or $(b, d) \in E \wedge a = c$, or $(a, b) \in E \wedge a = d \wedge b = c$. For any two states x and y, the transition probability $P_{x,y}$ of the chain is given by $P_{x,y} = 2/m$ if $x \to y$, $P_{x,y} = 1 - (2/m)|\{z \mid x \to z\}|$ if $x = y$, otherwise $P_{x,y} = 0$. The symmetry structure of the chain ($P_{x,y} = P_{y,x}$) gives $\sum_x P_{x,y} = 1$ for all state y. Thus, $\pi = (\pi(x_1), \pi(x_2), \ldots, \pi(x_{n(n-1)})) = \{n(n-1)\}^{-1}(1, 1, \ldots, 1)$ is the stationary distribution of the chain ($\pi P = \pi$) where $x_1, x_2, \ldots, x_{n(n-1)}$ are all the states of

the chain (i.e. all pairs of token locations). We denote the expected number of transition steps from state x to state y by $h_{x,y}$. We have $h_{y,y} = 1/\pi(y) = n(n-1)$ for any state y. We also have $h_{y,y} = 1 + \sum_{y \to z} (2/m) \cdot h_{z,y}$. Hence, we obtain $\sum_{y \to z} h_{z,y} = n(n-1)m/2 - m/2$. Thus, we have $h_{x,y} \leq mn^2/2$ for any states x and y satisfying $x \to y$. Let $w_0, w_1, \dots, w_l$ ($w_0 = u, w_l = v, l \leq d$) be the shortest path from u to v. The expected time until the two token meet is bounded by $\left(\sum_{i=0}^{l-2} h_{(w_i, w_l), (w_{i+1}, w_l)} \right) h_{(w_{l-1}, w_l), (w_l, w_{l-1})} \leq mn^2 d/2$. ◄

Lemmas 18 and 19 gives the following theorem.

▶ **Theorem 21.** *Protocol P_{AR} is an $(O(mt_{\mathrm{epi}} + mn^3 d), \Omega(\tau e^\tau))$- loosely-stabilizing leader election protocol for arbitrary graphs G when $t_{\max} \geq 8\delta \max(d, \lceil 2\log mn^3 d \rceil)$, $t_{\mathrm{virus}} = t_{\max}/2$, and $4\delta t_{\max} \lceil \log n \rceil \leq t_{\mathrm{epi}} \leq \tau e^\tau/(9n)$.*

Therefore, given an upper bounds N of n and upper bound Δ of δ, we have a $(O(mn^3 d + mN\Delta^2 \log N), \Omega(Ne^N))$ loosely-stabilizing leader election protocol for arbitrary graphs if we assign $t_{\max} = 8\Delta \max(N, \lceil 12 \log N \rceil)$, $t_{\mathrm{virus}} = t_{\max}/2$, $t_{\mathrm{epi}} = 4\Delta t_{\max} \lceil \log N \rceil$.

5 Conclusion

We have presented a loosely-stabilizing leader election protocol for arbitrary undirected graphs in the population protocol model. It does not use agent identifiers nor random numbers unlike our previous protocols. Given upper bounds N of n and Δ of δ, the population reaches a safe configuration within $O(mn^3 d + mN\Delta^2 \log N)$ expected interactions, and after that, keeps a unique leader for $\Omega(Ne^N)$ expected interactions. The restriction to *undirected* graph is only for simplicity of complexity analysis, and P_{AR} works on arbitrary *directed* graphs without modifications.

References

1   D. Angluin, J Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253, 2006. `doi:10.1007/s00446-005-0138-3`.

2   D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. In *DISC*, pages 61–75, 2006.

3   D. Angluin, J. Aspnes, M. J Fischer, and H. Jiang. Self-stabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems*, 3(4):13, 2008.

4   J. Beauquier, P. Blanchard, and J. Burman. Self-stabilizing leader election in population protocols over arbitrary communication graphs. In *OPODIS*, pages 38–52, 2013.

5   J. Beauquier, J. Burman, L. Rosaz, and B. Rozoy. Non-deterministic population protocols. In *OPODIS*, pages 61–75, 2012.

6   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 of Computing Systems*, 50(3):433–445, 2012.

7   D. Canepa and M. G. Potop-Butucaru. Stabilizing leader election in population protocols, 2007. URL: `http://hal.inria.fr/inria-00166632`.

8   M. J. Fischer and H. Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *OPODIS*, pages 395–409, 2006. `doi:10.1007/11945529_28`.

9   T. Izumi. On space and time complexity of loosely-stabilizing leader election. In *SIROCCO*, 2015.

10   O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Mediated population protocols. *Theoretical Computer Science*, 412(22):2434–2450, 2011.

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

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

13   Y. Sudo, J. Nakamura, Y. Yamauchi, F. Ooshita, H. Kakugawa, and T. Masuzawa. Loosely-stabilizing leader election in a population protocol model. *Theoretical Computer Science*, 444:100–112, 2012.

14   Y. Sudo, F. Ooshita, H. Kakugawa, and T. Masuzawa. Loosely-stabilizing leader election on arbitrary graphs in population protocols. In *OPODIS*, pages 339–354. Springer, 2014.

15   X. Xu, Y. Yamauchi, S. Kijima, and M. Yamashita. Space complexity of self-stabilizing leader election in population protocol based on k-interaction. In *SSS*, pages 86–97, 2013.