PAPER Special Section on Foundations of Computer Science —Frontiers of Theory of Computation and Algorithm—

Loosely Stabilizing Leader Election on Arbitrary Graphs in Population Protocols without Identifiers or Random Numbers*

Yuichi SUDO[†a)], ***Nonmember***, **Fukuhito OOSHITA**[††], **Hirotsugu KAKUGAWA**[†††], ***and*** **Toshimitsu MASUZAWA**[†], ***Members***

SUMMARY We consider the leader election problem in the population protocol model, which Angluin et al. proposed in 2004. A self-stabilizing leader election is impossible for complete graphs, arbitrary graphs, trees, lines, degree-bounded graphs, and so on unless the protocol knows the exact number of nodes. In 2009, to circumvent the impossibility, we introduced the concept of *loose stabilization*, 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.,* an exponentially long 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 or random numbers. Given upper bounds N and Δ of the number of nodes n and the maximum degree of nodes δ, respectively, the proposed protocol reaches a safe configuration within $O(mn^2 d \log n + mN\Delta^2 \log N)$ 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.

key words: *population protocols, leader election, loose stabilization*

1. Introduction

This paper addresses the leader election problem in the population protocol model. The *population protocol* (PP) model, which was presented by Angluin et al. [2], represents wireless sensor networks of mobile sensing devices that cannot control their own 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*. One example represented by this model is a flock of birds, where each bird is equipped with a sensing device with a small transmission range. Two devices can communicate (*i.e.,* interact) with each other only when the corresponding birds come sufficiently close to each other. This unique but meaningful model has attracted a lot of attention and has been used in numerous studies.

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 thereafter the population keeps 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 that the unique leader exists [2]–[4], and (ii) self-stabilization tolerates any finite number of transient faults, which suits systems consisting of numerous cheap and unreliable nodes. (Such systems are the original motivation of the PP model.) However, SS-LE is impossible in many cases in the PP model: no protocol solves SS-LE for complete graphs, trees, lines, degree-bounded graphs, and so on unless the exact number of agents n is available to agents in advance [4].

Hence, many researchers conducting studies of SS-LE have taken 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. [5] proved that n states of each agent are necessary and sufficient to solve SS-LE for a complete graph of n agents. Mizoguchi et al. [6] and Xu et al. [7] improved the space complexity by adopting the mediated population protocol model [8] and the PP_k model [9], respectively. The other approach is to use *oracles*, a kind of failure detectors. Fischer and Jiang [10] took this approach for the first time. They introduced oracle $\Omega?$, which informs all agents whether a leader exists or not, and proposed two protocols that solve SS-LE for rings and complete graphs by using $\Omega?$. Beauquier et al. [11] presented an SS-LE protocol for arbitrary graphs that uses two copies of $\Omega?$. Canepa et al. [12] proposed two SS-LE protocols that use $\Omega?$ 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.

In our previous works [13], [14] we 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

Manuscript received April 4, 2019.
Manuscript revised September 22, 2019.
Manuscript publicized November 27, 2019.

[†]The authors are with Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871 Japan.
[††]The author is with Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.
[†††]The author is with Faculty of Science and Technology, Ryukoku University, Otsu-shi, 520–2194 Japan.

*An extended abstract of this paper [1] appears in the proceedings of the 19th International Conference on Principles of Distributed Sysems (OPODIS 2015). This work was supported by JSPS KAKENHI Grant Numbers 16K00018, 17K19977, 18K18000, and 18K11167 and JST SICORP Grant Number JPMJSC1606.

a) E-mail: y-sudou@ist.osaka-u.ac.jp

DOI: 10.1587/transinf.2019FCP0003

Table 1 Loosely stabilizing leader election for arbitrary graphs

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

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 a given upper bound N of n. Protocols P_{ID} and P_{RD} solve leader election for arbitrary graphs using agent identifiers and random numbers, respectively, given N. All three protocols are practically equivalent to an 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).

Many of the results on population protocols assume uniform probabilistic distribution for the interactions of agents; that is, every interaction occurs uniformly at random [2], [3], [13]–[18]. This assumption has 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 of SS-LE [2] in the PP model still holds even with this assumption.

In the field of population protocols, many efforts have been devoted to devising protocols for a complete graph, *i.e.,* a population where every pair of agents interacts infinitely often. However, several works [2], [4], [11], [12], [19], [20] give protocols that work for an arbitrary graph $G(V, E)$ where E specifies the set of interactable pairs. Each pair of agents $(u, v) \in E$ has interactions infinitely often, while each pair of agents $(u', v') \notin E$ never has an interaction. A set of considerable graphs depends on an application. A probability distribution which determines the probability that each pair of agents interact at each step also depends on an application. In this paper, from the theoretical interest, we focus on an arbitrary graph where every interactable pair of agents has the same probability to be selected to interact at each step. This setting is of interest because it generalizes the setting of complete graphs and the uniformly random scheduler for them. As an example of a practical application for the setting, we can consider a network consisting of a huge number of tiny and cheap devices where each device repeatedly tries to initiate an interaction (*i.e.,* wireless communication) with another device at the same intervals. For some pair of devices, the time gap between the (periodic) initiating trials of the two devices may be too small; that is, they try to initiate wireless communication almost simultaneously, and they fail even when they come sufficiently close to each other. Thus, such pairs of agents never have interactions, while the other pairs of agents have interactions with the same rate. In this case, it is natural to consider an arbitrary (general) graph and the uniformly random scheduler that selects each edge (*i.e.,* a pair of agents) with the same probability at each step.

1.1 Our Contribution

In this paper, we give a loosely stabilizing protocol P_{AR} for leader election in arbitrary graphs without agent identifiers or random numbers (or a model with a weaker assumption than P_{ID} or P_{RD} [14]). Thus, we succeed in removing the assumptions of the unique identifiers and random number generators for a loosely stabilizing leader election on arbitrary graphs in the PP model. Removing these assumptions is important because the unique identifiers or random number generators may be difficult to realize in weak devices, such as the tiny devices with restricted capability constituting a network of the PP model.

The expected convergence time (*i.e.,* the expected number of interactions needed to reach a safe configuration) and the expected holding time (*i.e.,* the expected number of interactions to keep the specification after reaching a safe configuration) of P_{ID}, P_{RD}, and P_{AR} are shown in Table 1, where Δ is a given upper bound of the maximum degree of agents δ and d is the diameter of the graph. All the protocols, including P_{AR}, keep the unique leader for an exponentially long time 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 [5]. Furthermore, Izumi [15] proved that loosely stabilizing leader election with polynomial convergence time and an 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 with $\Theta(\log N)$ agent memory is not well-motivated because $\Theta(\log n)$ 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 among their identifiers. Actually, many works assume anonymity and agent memory space of $O(\log n)$ or more (*e.g.,* [4]–[7], [13], [14], [16], [21], [22]). In this paper, we analyze the 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 makes a considerable contribution: The token with a countdown 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 the token and the colors of viruses are newly introduced to remove the assumption of random number generators.

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 the expected time until two tokens performing random walks meet in the PP model. The analysis can be applied with a slight modification to estimate the expected time until a token visits all nodes by random walks. We believe that the analysis techniques are of significant importance because existing analyses 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 analyze the PP model.

Angluin et al. [2] proved 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 we can easily obtain a loosely stabilizing leader election protocol on arbitrary graphs using their transformer and an existing loosely stabilizing leader election protocol on complete graphs [13]. 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 an execution where an elected leader moves among the population forever, which does not satisfy the specification of the leader election. Furthermore, this transformer depends on the assumption that all the agents have a common initial state at the start of the execution. We cannot assume any specific initial state since we consider loose stabilization. That is, our protocol works from any arbitrary configuration (*i.e.,* any global state). Hence, we cannot use the transformer [2] to achieve our goal.

2. Preliminaries

In this section, we define the model that 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 assume that G is undirected, *i.e.,* $(u, v) \in E \Leftrightarrow (v, u) \in E$. We define $n = |V|$, $m = |E|$, $\delta_v = |\{(u, w) \in E \mid v \in \{u, w\}\}|$ for each $v \in V$, and $\delta = \max_{v \in V} \delta_v$.

A *protocol* $P(Q, Y, T, O)$ consists of a finite set Q of states, a finite set Y of output symbols, transition function $T : Q \times Q \to Q \times Q$, and output function $O : Q \to Y$. 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 $C : V \to Q$ that specifies the states of all the agents. We denote the set of all configurations of protocol P by $C_{\mathrm{all}}(P)$. We say that configuration C changes to C' by an 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* $\boldsymbol{\Gamma} = \Gamma_0, \Gamma_1, \ldots$: 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 $\boldsymbol{\Gamma}$, the *execution* of protocol P is defined as $\Xi_P(C_0, \boldsymbol{\Gamma}) = C_0, C_1, \ldots$ such that $C_t \xrightarrow{\Gamma_t} C_{t+1}$ for all $t \geq 0$. We denote $\Xi_P(C_0, \boldsymbol{\Gamma})$ 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. The specification of leader election, denoted by LE, requires that there exists one agent v such that v is always a leader and all other agents are always followers throughout an execution. We define *expected holding time* $\mathrm{EHT}_P(C, LE)$ as the expected number of interactions during which an execution $\Xi_P(C)$ starting from a configuration $C \in C_{\mathrm{all}}(P)$ keeps LE (*i.e.,* the expected number of interactions until $\Xi_P(C)$ deviates from LE). For any set $\mathcal{S} \subseteq C_{\mathrm{all}}(P)$ of configurations, we also define *expected convergence time* $\mathrm{ECT}_P(C, \mathcal{S})$ as the expected number of interactions required for the population to enter a configuration in $\mathcal{S}$ in an execution $\Xi_P(C)$ starting from a configuration $C \in C_{\mathrm{all}}(P)$.

Definition (Loosely stabilizing leader election [13]): A protocol P is an (α, β)-loosely stabilizing leader election protocol if there exists set $\mathcal{S}$ of configurations satisfying $\max_{C \in C_{\mathrm{all}}(P)} \mathrm{ECT}_P(C, \mathcal{S}) \leq \alpha$ and $\min_{C \in \mathcal{S}} \mathrm{EHT}_P(C, LE) \geq \beta$.

We write the natural logarithm of x as $\ln x$. and do not indicate the base of logarithm in an asymptotical expression such as $O(\log n)$.

Finally, we define *round time*. We define the first round time $\mathrm{RT}_{\boldsymbol{\Gamma}}(1)$ 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 $\mathrm{RT}_{\boldsymbol{\Gamma}}(i)$ as the minimum t satisfying $\forall e \in E,\ \mathrm{RT}_{\boldsymbol{\Gamma}}(i-1) < \exists t' \leq t,\ \Gamma_{t'} = e$. For simplicity, we define $\mathrm{RT}_{\boldsymbol{\Gamma}}(0) = -1$. Note that all agents join at least one interaction during $\Gamma_{\mathrm{RT}_{\boldsymbol{\Gamma}}(i)+1}, \ldots, \Gamma_{\mathrm{RT}_{\boldsymbol{\Gamma}}(i+1)}$ for each $i \geq 0$.

Lemma 1 (in [23]): $\Pr(\mathrm{RT}_{\boldsymbol{\Gamma}}(i) < im(1 + \ln n)) \geq 1 - ne^{-i/4}$ for any $i \geq 1$.

Proof : The lemma follows from the proof of Lemma 15 in [23] with a slight modification.[†] □

3. Loosely Stabilizing Leader Election Protocol

In this section, we give a loosely stabilizing leader election

[†]Lemma 15 in [23] claims $\Pr(\mathrm{RT}_{\boldsymbol{\Gamma}}(i) < 2im\lceil \ln n \rceil) \geq 1 - ne^{-i/4}$ for any $i \geq 1$. We obtain $\Pr(\mathrm{RT}_{\boldsymbol{\Gamma}}(i) < im(1 + \ln n)) \geq 1 - ne^{-i/4}$ replacing $2im\lceil \ln n \rceil$ with $im(1 + \ln n)$ in all the inequalities that appear in the proof of Lemma 15 in [23].

Algorithm 1 Leader Election P_{AR}

Variables of each agent:

$\mathtt{leader} \in \{\top, \bot\}$, $\mathtt{token} \in \{\top, \bot\}$, $\mathtt{clr} \in \{\mathrm{BL}, \mathrm{WH}\}$

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

Output function O:

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

Interaction between initiator u_0 and responder u_1:

1: $u_0.\mathtt{timer_L} \leftarrow u_1.\mathtt{timer_L} \leftarrow \max(u_0.\mathtt{timer_L} - 1,\ u_1.\mathtt{timer_L} - 1,\ 0)$
2: **if** $u_0.\mathtt{leader} = \top$ **or** $u_1.\mathtt{leader} = \top$ **then**
3: $\quad u_0.\mathtt{timer_L} \leftarrow u_1.\mathtt{timer_L} \leftarrow t_{\max}$ // a leader resets leader timer
4: **else if** $u_0.\mathtt{timer_L} = 0$ **then**
5: $\quad u_0.\mathtt{leader} \leftarrow \top$ // a new leader is created at timeout
6: $\quad u_0.\mathtt{timer_L} \leftarrow u_1.\mathtt{timer_L} \leftarrow t_{\max}$
7: **end if**

8: $u_0.\mathtt{timer_T} \leftarrow u_1.\mathtt{timer_T} \leftarrow \max(u_0.\mathtt{timer_T} - 1,\ u_1.\mathtt{timer_T} - 1,\ 0)$
9: **if** $u_0.\mathtt{token} = \top$ **or** $u_1.\mathtt{token} = \top$ **then**
10: $\quad u_0.\mathtt{timer_T} \leftarrow u_1.\mathtt{timer_T} \leftarrow t_{\max}$ // a token resets token timer
11: **else if** $u_0.\mathtt{timer_T} = 0$ **then**
12: $\quad u_0.\mathtt{token} \leftarrow \top$ // a new token is created at timeout
13: $\quad u_0.\mathtt{timer_T} \leftarrow u_1.\mathtt{timer_T} \leftarrow t_{\max}$
14: **end if**

15: $u_0.\mathtt{token} \leftrightarrow u_1.\mathtt{token}$ // a token moves between agents
16: $u_0.\mathtt{timer_E} \leftrightarrow u_1.\mathtt{timer_E}$
17: **forall** $i \in \{0, 1\}$ **do** $u_i.\mathtt{timer_E} \leftarrow \max(0, u_i.\mathtt{timer_E} - 1)$ **endfor**
18: **if** $u_0.\mathtt{token} = \top$ **and** $u_1.\mathtt{token} = \top$ **then** $u_1.\mathtt{token} \leftarrow \bot$ **endif**

19: **if** $\exists i \in \{0, 1\} : u_i.\mathtt{timer_V} > 0 \wedge u_{1-i}.\mathtt{timer_V} = 0 \wedge u_i.\mathtt{clr} \neq u_{1-i}.\mathtt{clr}$ **then**
20: $\quad u_{1-i}.\mathtt{clr} \leftarrow u_i.\mathtt{clr}$
21: $\quad u_{1-i}.\mathtt{leader} \leftarrow \bot$
22: **end if**
23: $u_0.\mathtt{timer_V} \leftarrow u_1.\mathtt{timer_V} \leftarrow \max(u_0.\mathtt{timer_V} - 1,\ u_1.\mathtt{timer_V} - 1,\ 0)$

24: **if** $\exists i \in \{0, 1\} : u_i.\mathtt{leader} = \top \wedge u_i.\mathtt{token} = \top \wedge u_i.\mathtt{timer_E} = 0$ **then**
25: $\quad$ **if** $u_i.\mathtt{clr} = \mathrm{BL}$ **then** $u_i.\mathtt{clr} \leftarrow \mathrm{WH}$ **else** $u_i.\mathtt{clr} \leftarrow \mathrm{BL}$ **endif**
26: $\quad u_i.\mathtt{timer_V} \leftarrow t_{\mathrm{virus}}$
27: $\quad u_i.\mathtt{timer_E} \leftarrow t_{\mathrm{epi}}$
28: **end if**

protocol P_{AR}. It elects exactly one leader within polynomial time and keeps the unique leader for an exponentially long time for arbitrary graphs without identifiers or random numbers, given upper bounds N and Δ of n and δ, respectively.

The pseudocode of P_{AR} is described in Algorithm 1. In the pseudocode, we use $X \leftarrow Y$ to represent the substitution of a value Y to a variable X and use $W \leftrightarrow Z$ to represent swapping the values of two variables W and Z. A state of an agent is described by a collection of variables, and a transition function is described by a pseudocode that updates variables of the initiator x and the responder y. We denote the value of variable $\mathtt{var}$ of agent $v \in V$ by $v.\mathtt{var}$. We also denote the value of $\mathtt{var}$ in state $q \in Q$ by $q.\mathtt{var}$. In P_{AR}, each agent has three binary variables, $\mathtt{leader} \in \{\top, \bot\}$, $\mathtt{token} \in \{\top, \bot\}$, and $\mathtt{clr} \in \{\mathrm{BL}, \mathrm{WH}\}$, and four timers, $\mathtt{timer_L}$, $\mathtt{timer_T}$, $\mathtt{timer_V}$, and $\mathtt{timer_E}$. The output function defines leaders as follows: an 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 $v.\mathtt{clr} = \mathrm{BL}$, 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 15–18), virus creation (Lines 24–28), and virus propagation (Lines 19–23). 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 or 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 no leader exists. 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 no leader exists 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 occurs and a leader is created. When a leader exists, the timeout rarely happens since all agents keep high timer values almost all the time thanks to the timer reset and the larger value propagation. Therefore, this mechanism rarely ruins the stability of the unique leader.

A naive way to reduce the number of leaders to one is as follows: When two leaders meet, one remains a leader and the other becomes a follower. This simple mechanism works on a complete graph, but does not work on an arbitrary graph because two leaders u and v never meet forever if $(u, v) \notin E$. Therefore, we do not adopt this simple mechanism.

Instead, P_{AR} reduces the number of leaders to one 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 such that exactly one agent eventually remains a leader (*i.e.,* survives a virus war). Specifically, 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 an epidemic timer of zero value; the virus-propagation part propagates the virus to the whole population, which changes leaders to followers.

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 15–18) aims to re-

duce 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 (Line 18). Hence, the number of tokens eventually becomes one. Each token has an epidemic timer ($\texttt{timer}_\texttt{E}$). The epidemic timer is decremented by one every time the token moves (Lines 16–17), and thus, it eventually becomes zero. Note that the number of tokens never becomes zero once a token exists because the number of tokens decreases only when two tokens meet at an interaction.

The virus-creation part (Lines 24–28) creates a new virus when a leader meets a token with an epidemic timer of zero value. We call this event *virus creation*. Specifically, if a token with $\texttt{timer}_\texttt{E} = 0$ moves to a leader agent, the leader changes its color from black to white or from white to black (Line 25) and creates a new virus with full value TTL (Time To Live), *i.e.,* $\texttt{timer}_\texttt{V} = t_{\text{virus}}$ (Line 26). The leader also resets the epidemic timer of the token (Line 27), which enables periodical occurrence of epidemics.

The virus-propagation part (Lines 19–23) propagates a virus from agent to agent and reduces the number of leaders. When an agent has a virus (*i.e.,* $v.\texttt{timer}_\texttt{V} > 0$), we regard $v.\texttt{timer}_\texttt{V}$ as the TTL of the virus. A virus vanishes from the agent when its TTL becomes zero. In the same way as $\texttt{timer}_\texttt{L}$ and $\texttt{timer}_\texttt{T}$, a virus propagates at interactions in the larger value propagation fashion (Line 23). Moreover, a virus has the power to change the colors of agents and kill leaders. Specifically, if an agent u with a virus interacts with an agent v without a virus such that $u.\texttt{clr} \neq v.\texttt{clr}$, v changes its color to $u.\texttt{clr}$ (Line 20). At this time, if v is a leader, the virus kills v (*i.e.,* changes v from a leader to a follower) (Line 21). We say that an virus on an agent w is black (resp. white) if $w.\texttt{clr} = \text{BL}$ (resp. $w.\texttt{clr} = \text{WH}$). Note that in an execution of P_{AR}, a white virus kills only black leaders and a black virus kills only white leaders.

Once a new virus is created in the virus-creation part, the virus propagates to the whole population within a short time. However, the value of $\texttt{timer}_\texttt{V}$ is reset to t_{virus} 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. Then, the leader changes its color to black, and the black virus propagates to the whole population. Since a black virus kills only white leaders, the leader is never killed by the virus until it becomes white. Therefore, if we set t_{epi} to a sufficiently high value, suicide rarely happens because the black virus vanishes from the population before the leader becomes white.

Protocol P_{AR} correctly works if $t_{\max}$ and t_{virus} are sufficiently large, and t_{epi} is sufficiently greater than t_{virus}. When no leader exists, the leader-creation part eventually creates a leader by timeout. In the following, let us consider the case in which 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 a 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 to the whole population and turns 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 the 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 previous 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

In this section, we analyze the expected holding time and the expected convergence time of P_{AR}. The notations and assumptions used in this paper are summarized in Table 2. We denote $C_{\text{all}}(P_{\text{AR}})$ simply by C_{all} throughout this section.

Table 2 Notations and assumptions for P_{AR}

Notations	
τ :	$\lfloor t_{\text{virus}}/(4\delta) \rfloor$
$\text{PROP}_L(i)$:	$C_i \in \mathcal{L}_{\text{exist}} \Rightarrow C_{i+2m\tau} \in \mathcal{L}_{\text{half}}$
$\text{PROP}_T(i)$:	$C_i \in \mathcal{T}_{\text{exist}} \Rightarrow C_{i+2m\tau} \in \mathcal{T}_{\text{half}}$
$\text{HALF}(i)$:	$\text{HALF}(i) = 1$ if every agent joins only less than $t_{\max}/2$ interactions among $\Gamma_i, \ldots, \Gamma_{i+2m\tau-1}$; $\text{HALF}(i) = 0$ otherwise.
$\#_{TI}(v, t_1, t_2)$:	the number of interactions among $\Gamma_{t_1}, \ldots, \Gamma_{t_2}$ involving the token that agent v has in configuration C_{t_1}
Assumptions	
	$n \geq 3$
	$t_{\text{virus}} \geq 4\delta \max(d, 7 \ln n)$
	$t_{\max} = 2t_{\text{virus}}$
	$8\delta t_{\text{virus}} \lceil \ln n \rceil \leq t_{\text{epi}} \leq \tau e^\tau / n^4$

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} are sufficiently large and t_{epi} is sufficiently greater than t_{virus}. Specifically, we assume $t_{\mathrm{virus}} \geq 4\delta \max(d, 7 \ln n)$, $t_{\max} = 2t_{\mathrm{virus}}$, and $t_{\mathrm{epi}} \geq 8\delta t_{\mathrm{virus}} \lceil \ln n \rceil$, 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: $t_{\mathrm{epi}} \leq \tau e^{\tau}/n^4$, where $\tau = \lfloor t_{\mathrm{virus}}/(4\delta) \rfloor = \lfloor t_{\max}/(8\delta) \rfloor$. Otherwise, even when a leader exists, the leader timeout happens with non-negligible probability within a resulting very long epidemic interval. This severely harms our analysis of the convergence time. We will prove the following equations under these assumptions:

$$\max_{C \in \mathcal{C}_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}}) = O(mn^2 d \log n + mt_{\mathrm{epi}}), \tag{1}$$

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

where $\mathcal{S}_{\mathrm{AR}}$ is the set of configurations we define later. Thus, given a sufficiently large N, we obtain an $(O(mn^2 d \log n + mN\Delta^2 \ln N), \Omega(Ne^N))$-loosely stabilizing leader election protocol by assigning $t_{\mathrm{virus}} = 4\Delta N$ and $t_{\mathrm{epi}} = 8\Delta t_{\mathrm{virus}} \lceil \ln N \rceil$. (Remind that N and Δ are given upper bounds of n and the maximum degree $\delta = \max_{v \in V} \delta_v$.)

To prove (1) and (2), define 10 sets of configurations:

$$\begin{aligned}
\mathcal{L}_{\mathrm{one}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \#_L(C) = 1\},\\
\mathcal{T}_{\mathrm{one}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \#_T(C) = 1\},\\
\mathcal{L}_{\mathrm{exist}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \#_L(C) \geq 1\},\\
\mathcal{T}_{\mathrm{exist}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \#_T(C) \geq 1\},\\
\mathcal{L}_{\mathrm{half}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \forall v \in V,\ C(v).\mathtt{timer}_{\mathrm{L}} > t_{\max}/2\},\\
\mathcal{T}_{\mathrm{half}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \forall v \in V,\ C(v).\mathtt{timer}_{\mathrm{T}} > t_{\max}/2\},\\
\mathcal{V}_{\mathrm{same}} &= \left\{ C \in \mathcal{C}_{\mathrm{all}} \;\middle|\; \begin{array}{l} \exists u, \forall v \in V,\ C(u).\mathtt{leader} = \top \\ \wedge\ (C(v).\mathtt{timer}_{\mathrm{V}} > 0 \\ \quad \Rightarrow C(u).\mathtt{clr} = C(v).\mathtt{clr})\} \end{array} \right\},\\
\mathcal{V}_{\mathrm{zero}} &= \{C \in \mathcal{C}_{\mathrm{all}} \mid \forall v \in V,\ C(v).\mathtt{timer}_{\mathrm{V}} = 0\},\\
\mathcal{E}_{\mathrm{half}} &= \left\{ C \in \mathcal{C}_{\mathrm{all}} \;\middle|\; \begin{array}{l} \forall v \in V,\ C(v).\mathtt{token} = \top \\ \quad \Rightarrow C(v).\mathtt{timer}_{\mathrm{E}} > t_{\mathrm{epi}}/2 \end{array} \right\},\\
\mathcal{S}_{\mathrm{AR}} &= \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{same}} \cap (\mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}),
\end{aligned}$$

where $\#_L(C)$ and $\#_T(C)$ denote the number of leaders and the number of tokens in configuration C, respectively. Note that $\mathcal{V}_{\mathrm{same}}$ is the set of configurations where there is a leader 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 an epidemic timer whose value is greater than $t_{\mathrm{epi}}/2$.

4.1 Expected Holding Time

The goal of this subsection is to prove (2). First, we give a sufficient condition to satisfy (2) in Lemma 2.

Lemma 2: Let $C_0 \in \mathcal{S}_{\mathrm{AR}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. We have (2) if

$$\begin{aligned}
&\Pr(C_0, \ldots, C_{8m\delta\tau\lceil \ln n \rceil} \in LE \wedge C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{S}_{\mathrm{AR}})\\
&\qquad = 1 - O(n\delta \log n \cdot e^{-\tau}).
\end{aligned} \tag{3}$$

Proof : Let $A = \min_{C_0 \in \mathcal{S}_{\mathrm{AR}}} \mathrm{EHT}_{P_{\mathrm{AR}}}(C_0, LE)$. If (3) holds, we have $A \geq (1 - O(n\delta \log n \cdot e^{-\tau}))(8m\delta\tau\lceil \ln n \rceil + A)$. Solving this inequality gives $A \geq \tau e^{\tau}$. □

By Lemma 2, our goal is now to prove (3). In what follows, we give five conditions such that satisfying all the conditions yields (3) (Lemma 3) and show that all the five conditions hold with probability at least $1 - O(n\delta \log n \cdot e^{-\tau})$ (Lemmas 4, 5, 6, 7, and 8).

To specify the conditions, we introduce four random variables $\mathrm{PROP}_L(i)$, $\mathrm{PROP}_T(i)$, $\mathrm{HALF}(i)$, and $\#_{TI}(v, t_1, t_2)$. The first three are binary random variables:

- $\mathrm{PROP}_L(i) = 1$ if $C_i \in \mathcal{L}_{\mathrm{exist}} \Rightarrow C_{i+2m\tau} \in \mathcal{L}_{\mathrm{half}}$ holds, otherwise $\mathrm{PROP}_L(i) = 0$.
- $\mathrm{PROP}_T(i) = 1$ if $C_i \in \mathcal{T}_{\mathrm{exist}} \Rightarrow C_{i+2m\tau} \in \mathcal{T}_{\mathrm{half}}$, otherwise $\mathrm{PROP}_T(i) = 0$.
- $\mathrm{HALF}(i) = 1$ if every agent joins less than $t_{\max}/2$ interactions among $\Gamma_i, \ldots, \Gamma_{i+2m\tau-1}$, otherwise $\mathrm{HALF}(i) = 0$.

Roughly speaking, $\mathrm{PROP}_L(i) = 1$ ($\mathrm{PROP}_T(i) = 1$) means that high values of $\mathtt{timer}_{\mathrm{L}}$ ($\mathtt{timer}_{\mathrm{T}}$) propagate from a leader (a token, respectively) to all the agents during $2m\tau$ interactions $\Gamma_i, \Gamma_{i+1}, \ldots, \Gamma_{i+2m\tau-1}$, and $\mathrm{HALF}(i) = 1$ means that every agent does not interact so much during those $2m\tau$ interactions. Note that $\mathrm{PROP}_L(i) = 1$ ($\mathrm{PROP}_T(i) = 1$) unconditionally holds when no leader (token, respectively) exists in C_i. Next, we define random variable $\#_{TI}(v, t_1, t_2)$ for agent $v \in V$ and integers t_1 and t_2, $(0 \leq t_1 < t_2)$. This variable is meaningful only if v has a token when interaction Γ_{t_1} occurs. Intuitively, $\#_{TI}(v, t_1, t_2)$ represents the number of interactions that the token joins during $\Gamma_{t_1}, \ldots, \Gamma_{t_2-1}$ (or the number of times the token moves during the period). Formally, 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 $t > t_1$.

Now, we can describe the five conditions (A), (B), (C), (D), and (E) in the form of the following lemma.

Lemma 3: 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 $C_0, \ldots, C_{8m\delta\tau\lceil \ln n \rceil} \in LE$ and $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{S}_{\mathrm{AR}}$ if the following conditions hold:
(A) $\#_{TI}(v_T, 0, 8m\delta\tau\lceil \ln n \rceil) < t_{\mathrm{epi}}/2$,
(B) $\mathrm{PROP}_L(2im\tau) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \ln n \rceil - 1$,
(C) $\mathrm{PROP}_T(2im\tau) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \ln n \rceil - 1$,
(D) $\mathrm{HALF}(2im\tau) = 1$ for all $i = 0, 1, \ldots, 4\delta\lceil \ln n \rceil - 1$, and
(E) $\mathrm{RT}_{\boldsymbol{\Gamma}}(t_{\mathrm{virus}}) < 8m\delta\tau\lceil \ln n \rceil$.

Proof : The number of tokens decreases only when two

Table 3 Convergence steps in P_{AR}

	Convergence Step	Convergence Time (#interactions)	Failure Probability	Lemmas
1.	$C_{\mathrm{all}} \to \mathcal{T}_{\mathrm{exist}} \cap \mathcal{T}_{\mathrm{half}}$	$mt_{\mathrm{epi}}/2$	$o(1)$	Lemma 12
2.	$\to \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}}$	$3mn^2 d \log_2 n$	$o(1)$	Lemma 14
3.	$\to (\mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}})$	$mt_{\mathrm{epi}}/2$	$o(1)$	Lemma 15
4.	$\to \mathcal{S}_{\mathrm{AR}}$	$5mt_{\mathrm{epi}}/2 + mn^2 d$	$o(1)$	Lemma 16
total	$C_{\mathrm{all}} \to \mathcal{S}_{\mathrm{AR}}$	$4(mt_{\mathrm{epi}} + mn^2 d \log_2 n)$	$o(1)$	

tokens meet, and it increases only when a token timeout happens. Clearly, conditions (C) and (D) guarantee that a token timeout never happens during $C_0, C_1, \ldots, C_{8m\delta\tau\lceil \ln n \rceil}$; thus, the number of tokens is always one during the period.

Therefore, condition (A) guarantees that the population is always in $\mathcal{V}_{\mathrm{same}}$ during $C_0, C_1, \ldots, C_{8m\delta\tau\lceil \ln n \rceil}$ for the following reason. Note that $C_0 \in \mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}$. In the case that $C_0 \in \mathcal{E}_{\mathrm{half}}$, no agent creates a new virus during the period by (A). Since $C_0 \in \mathcal{V}_{\mathrm{same}}$, the population always stays in $\mathcal{V}_{\mathrm{same}}$ during the period in this case. In the case that $C_0 \in \mathcal{V}_{\mathrm{zero}}$, the virus creation happens at most once by (A). If the virus creation does not happen, the population always stays in $\mathcal{V}_{\mathrm{same}}$ during the period since $C_0 \in \mathcal{V}_{\mathrm{same}}$. Otherwise, the leader that creates a virus and all the agents that are infected by the virus have the same color during this period. Therefore, the population always stays in $\mathcal{V}_{\mathrm{same}}$ in this period since $C_0 \in \mathcal{V}_{\mathrm{zero}}$.

The population does not reach $\neg\mathcal{L}_{\mathrm{exist}}$ as long as the population stays in $\mathcal{V}_{\mathrm{same}}$; That is, the last surviving leader is never killed while the population is in $\mathcal{V}_{\mathrm{same}}$. Therefore, there is always at least one leader during $C_0, C_1, \ldots, C_{8m\delta\tau\lceil \ln n \rceil}$. Thus, conditions (B) and (D) yield that a leader timeout does not happen during this period. Hence, $C_0, C_1, \ldots, C_{8m\delta\tau\lceil \ln n \rceil} \in LE$.

Now, all we have to do is to show that $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{S}_{\mathrm{AR}}$. Remind that $\mathcal{S}_{\mathrm{AR}} = \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{same}} \cap (\mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}))$ and we have shown $C_i \in \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{V}_{\mathrm{same}}$ for all $i = 0, 1, \ldots, 8m\delta\tau\lceil \ln n \rceil$. Moreover, this yields $C_{8m\delta\tau\lceil \ln n \rceil - 2m\tau} \in \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}}$; thus, we have $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}}$ by conditions (B), (C), and (D). Hence, it suffices to show that $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{E}_{\mathrm{half}} \cup \mathcal{V}_{\mathrm{zero}}$. If the virus creation happens during $C_0, C_1, \ldots, C_{8m\delta\tau\lceil \ln n \rceil}$, then condition (A) yields $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{E}_{\mathrm{half}}$. Consider the other case, *i.e.,* the case in which the virus creation never happens during this period. The maximum value of $\mathtt{timer}_{\mathrm{V}}$ in the population (*i.e.,* $\max_{v \in V} v.\mathtt{timer}_{\mathrm{V}}$) decreases at least by one at each round. Hence, condition (E) yields $C_{8m\delta\tau\lceil \ln n \rceil} \in \mathcal{V}_{\mathrm{zero}}$. □

Each of the five conditions (A), (B), (C), (D), and (E) holds with probability at least $1 - O(n\delta \log n \cdot e^{-\tau})$, as claimed by Lemmas 4, 5, 6, 7, and 8, respectively. Fortunately, these lemmas can be proven by a simple application of Chernoff bounds and/or by techniques used in [23]. See Appendix for the proofs of these lemmas.

Lemma 4: Let $C_0 \in \mathcal{T}_{\mathrm{one}}$ 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, $\Pr(\#_{TI}(v_T, 0, 8m\delta\tau\lceil \ln n \rceil) < t_{\mathrm{epi}}/2) \geq 1 - e^{-\tau}$.

Lemma 5 (in [23]): $\Pr(\mathrm{PROP}_L(i) = 1) \geq 1 - 2ne^{-\tau}$ for any $i \geq 0$.

Lemma 6: $\Pr(\mathrm{PROP}_T(i) = 1) \geq 1 - 2ne^{-\tau}$ for any $i \geq 0$.

Lemma 7 (in [23]): $\Pr(\mathrm{HALF}(i) = 1) \geq 1 - ne^{-\tau}$ for any $i \geq 0$.

Lemma 8: $\Pr(\mathrm{RT}_{\Gamma}(t_{\mathrm{virus}}) < 8m\delta\tau\lceil \ln n \rceil) \geq 1 - ne^{-\tau}$.

Lemma 9: $\min_{C \in \mathcal{S}_{\mathrm{AR}}} \mathrm{EHT}_{P_{\mathrm{AR}}}(C, LE) = \Omega(\tau e^{\tau})$.

Proof : The lemma follows from Lemmas 2, 3, 4, 5, 6, 7, and 8. □

4.2 Expected Convergence Time

The goal of this subsection is to prove (1). First, we give a sufficient condition to satisfy (1) in Lemma 10.

Lemma 10: Let $C_0 \in C_{\mathrm{all}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. We have $\max_{C \in C_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}}) = O(mt_{\mathrm{epi}} + mn^2 d \log n)$ if the following holds:

$$\Pr(\exists i \in O(mt_{\mathrm{epi}} + mn^2 d \log n), C_i \in \mathcal{S}_{\mathrm{AR}}) = \Omega(1). \quad (4)$$

Proof : Let $B = \max_{C \in C_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}})$. If (4) holds, we have $B = O(mt_{\mathrm{epi}} + mn^2 d \log n)) + (1 - \Omega(1)) \cdot B$. Solving this equality gives $B = O(mt_{\mathrm{epi}} + mn^2 d \log n)$. □

By lemma 10, our goal is now to show (4). We will show (4) with the four convergence steps shown in Table 3. This table shows the convergence time and failure probability of each step; for example, the third row means that letting $C_0 \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$, we have $\Pr(\exists i \leq mt_{\mathrm{epi}}/2,\ C_i \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}}) \geq 1 - o(1)$. We show the analyses for those four steps as Lemmas 12, 14, 15, and 16. In total, these lemmas together show that an execution of P_{AR} starting from any configuration reaches $\mathcal{S}_{\mathrm{AR}}$ within $4(mt_{\mathrm{epi}} + mn^2 d \log_2 n)$ interactions with probability $1 - o(1) \subset \Omega(1)$, *i.e.,* we obtain (4).

We analyze the four steps by assuming that $\mathrm{PROP}_L() = \mathrm{PROP}_T() = \mathrm{HALF}() = 1$ always hold during the $4(mt_{\mathrm{epi}} + mn^2 d \log_2 n)$ interactions. This assumption is justified by the following lemma.

Lemma 11: $\Pr(\forall i \leq 4(mt_{\mathrm{epi}} + mn^2 d \log_2 n), \mathrm{PROP}_L(i) = \mathrm{PROP}_T(i) = \mathrm{HALF}(i) = 1) = 1 - o(1)$.

Proof : By Lemmas 5, 6, and 7, the error probability is at most $4(mt_{\mathrm{epi}} + mn^2 d \log_2 n) \cdot 5ne^{-\tau} = o(1)$ because $t_{\mathrm{epi}} \leq \tau e^{\tau}/n^4$ and $\tau \geq 7 \ln n - 1$. □

4.2.1 First Step

Lemma 12: Let $C_0 \in C_{\mathrm{all}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$.

Then, $\Pr(\exists i \le mt_{\mathrm{epi}}/2,\ C_i \in \mathcal{T}_{\mathrm{exist}} \cap \mathcal{T}_{\mathrm{half}}) = 1 - o(1)$.

Proof : If no token exists in C_0, a token is created by a token timeout within $t_{\max}$ rounds because every round decreases $\max_{v \in V} C(v).\mathtt{timer}_{\mathrm{T}}$ at least by one when no token exists. By Lemma 1, $t_{\max}$ rounds finish within $16m\delta\tau\lceil \ln n \rceil$ interactions with probability $1-o(1)$. (See Lemma 18 in Appendix for the complete proof.) By Lemma 11, once a token exists, the population reaches a configuration in $\mathcal{T}_{\mathrm{exist}} \cap \mathcal{T}_{\mathrm{half}}$ within $2m\tau$ interactions with probability $1-o(1)$. The lemma follows from $16m\delta\tau\lceil \ln n \rceil + 2m\tau < mt_{\mathrm{epi}}/2$. □

4.2.2 Second Step

To obtain a sufficient number of interactions to reach $\mathcal{T}_{\mathrm{one}}$ from $\mathcal{T}_{\mathrm{exist}}$, we analyze a number of interactions until two tokens meet in the population (Lemma 13). The result of the second step (Lemma 14) follows from Lemma 13.

Lemma 13: Let C_0 be a configuration where two or more tokens exist. For any two of those tokens, the expected number of interactions until either of the two tokens disappears is at most $mn^2d/2$ in execution $\Xi_{P_{\mathrm{AR}}}(C_0)$.

Proof : Let $u, v \in V$ be two distinct agents such that both of them 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, we ignore this event because this just reduces the expected number of interactions until one of the two tokens vanish.) Consider the pair of random walks by the two tokens on population G, *i.e.,* a Markov chain (u_t, v_t) in which a state of the chain is a pair of agents (the locations of the two token) 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 any 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. 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 $h_{x,y} \le mn^2/2$ for any states x and y satisfying $x \to y$. Let $w_0, w_1, \ldots, w_l$ ($w_0 = u, w_l = v, l \le d$) be the shortest path from u to v. The expected time until the two tokens meet is bounded by $\left(\sum_{i=0}^{l-2} h_{(w_i,w_l),(w_{i+1},w_l)}\right) h_{(w_{l-1},w_l),(w_l,w_{l-1})} \le mn^2d/2$. □

Lemma 14: Let $C_0 \in \mathcal{T}_{\mathrm{exist}} \cap \mathcal{T}_{\mathrm{half}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Then, $\Pr(\exists i \le 3mn^2d \log_2 n,\ C_i \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}}) \ge 1 - o(1)$.

Proof : For any pair of tokens in C_0, at least one of the two tokens disappears within the first mn^2d interactions with probability at least $1/2$, by Markov's inequality and Lemma 13. Hence, at least one of them disappears within the first $\lfloor 3mn^2d \log_2 n \rfloor$ interactions with probability at least $1 - 2^{-3\log_2 n + 1} = 1 - O(n^{-3})$. Since there are at most $n(n-1)$ pairs of tokens in C_0, exactly one of the tokens survives after the first $\lfloor 3mn^2d \log_2 n \rfloor$ interactions with probability $1 - O(1/n) = 1 - o(1)$, by the union bound. Moreover, Lemma 11 guarantees that no token is created by a token timeout during the period with probability $1 - o(1)$. Thus, the population reaches $\mathcal{T}_{\mathrm{one}}$ during the period with probability $1 - o(1)$. Lemma 11 also guarantees that the population is in $\mathcal{T}_{\mathrm{half}}$ at that time. □

4.2.3 Third Step

By Lemma 8, all viruses on the population vanish within $8m\delta\tau\lceil \ln n \rceil$ interactions with probability $1 - o(1)$ unless a new virus is created during the period. Even if a new virus is created during the period, the next $8m\delta\tau\lceil \ln n \rceil$ interactions are enough for the population to reach $\mathcal{V}_{\mathrm{zero}}$, with probability $1 - o(1)$. A virus creation may happen again during these $8m\delta\tau\lceil \ln n \rceil$ interactions; however, it requires that the unique token joins more than t_{epi} ($\ge 64m\delta\tau\lceil \ln n \rceil$) interactions among the $8m\delta\tau\lceil \ln n \rceil$ interactions. Since the unique token joins each interaction with probability $2/m$, a Chernoff bound shows that such an event happens only with probability $o(1)$ (See Lemma 4). After that, even if there is no leader in the population, a leader timeout happens and a new leader is created within $16m\delta\tau\lceil \ln n \rceil$ interactions with probability $1 - o(1)$. This is because $t_{\max}$ rounds finish within $16m\delta\tau\lceil \ln n \rceil$ interactions with probability $1 - o(1)$ by Lemma 1. (See Lemma 18 in Appendix for the complete proof.) Since $32m\delta\tau\lceil \ln n \rceil \le mt_{\mathrm{epi}}/2$, we have obtained the following lemma; Thus, we have finished the third step.

Lemma 15: Let $C_0 \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Then, $\Pr(\exists i \le mt_{\mathrm{epi}}/2,\ C_i \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}}) \ge 1 - o(1)$.

4.2.4 Forth Step

Let $C_0 \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. By Lemma 11, we can assume that no token timeout happens and no leader timeout happens for a sufficiently long period as long as at least one leader exists. We expect that the execution $\Xi_{P_{\mathrm{AR}}}(C_0)$ proceeds in the following way in this order.

- (A1) The epidemic timer of the unique token becomes zero,
- (A2) The unique token meets a leader, and a new virus is created,
- (A3) The population reaches a configuration where all agents have a virus,
- (A4) All viruses vanish from the population,
- (B1) The epidemic timer of the unique token becomes zero,

- (B2) The unique token meets a leader and a new virus is created,
- (B3) The population reaches a configuration where all agents have a virus before the epidemic timer of the unique token decreases to $t_{\mathrm{epi}}/2$.

The population reaches a configuration in $\mathcal{S}_{\mathrm{AR}}$ when (A1), (A2),..., (B3) happen in this order in $\Xi_{P_{\mathrm{AR}}}(C_0)$. Let v_A and v_B be the leaders that create viruses in (A2) and (B2), respectively. ($v_A = v_B$ may hold.) Since $C_0 \in \mathcal{V}_{\mathrm{zero}}$, v_A is the only agent that has a virus just after ($A2$) happens. Without loss of generality, suppose that v_A is black at this time. Thereafter, every agent becomes black when it meets an agent with a virus. Hence, all agents are black when (A3) happens. Thereafter, when v_B creates a virus in (B2), its color is changed from black to white. At this time, all other leaders are black. Therefore, each of them is killed and becomes a follower when it meets an agent with a virus. Hence, only v_B is a leader, and all agents are white when (B3) happens. Therefore, $C_t \in \mathcal{L}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{one}} \cap \mathcal{V}_{\mathrm{same}} \cap \mathcal{E}_{\mathrm{half}}$ holds where C_t is a configuration just after (B3) happens. Since we can assume $\mathrm{PROP}_L(t-2m\tau) = \mathrm{PROP}_T(t-2m\tau) = 1$ by Lemma 11, $C_t \in \mathcal{L}_{\mathrm{half}} \cap \mathcal{T}_{\mathrm{half}}$ also holds; thus, $C_t \in \mathcal{S}_{\mathrm{AR}}$.

It is reasonable to expect that (A1), (A2),..., (B3) happen in this order. (A1) and (A2) must happen in this order. We can prove that a virus created by v_A propagates to the whole population within $2m\tau$ interactions with probability at least $1-o(1)$, in the same way as Lemma 5.[†] After that, all viruses eventually vanish from the population (A4). As we discussed above (in the third step), this happens before the epidemic timer of the unique token becomes zero again, *i.e.,* (B1) happens, with probability $1-o(1)$. (B2) also must eventually happen because there is at least one leader when (A4) happens; actually, v_A must be a leader at this time. The virus created by v_B propagates to the whole population within $2m\tau$ interactions with probability $1-o(1)$, again. (B3) may fail if the epidemic timer of the unique token decreases to $t_{\mathrm{epi}}/2$ within the $2m\tau$ interactions. However, such an event happens only with probability $1-o(1)$. (See Lemma 4). Thus, the above sequence (A1), (A2),..., (B3) happens in this order with probability $1-o(1)$.

To finish the analysis of the forth step, we need to analyze a sufficient number of interactions for each of (A1), (A2), ..., (B3).

- (A1), (B1): The epidemic timer of the unique token decreases by one if it joins an interaction. The unique token joins each interaction with probability at least $2/m$. Therefore, the epidemic timer of the unique token becomes zero within $mt_{\mathrm{epi}}/2$ interactions on average. We can easily prove by a Chernoff bound that it is done within mt_{epi} interactions (the double of the average) with probability $1-o(1)$. (See Lemma 19 in Appendix for the complete proof.)
- (A2), (B2): Once the epidemic timer of the unique token becomes zero, the token meets a leader within $mnd/2$ interactions on average. We can prove this in almost the same way as Lemma 13. (See Lemma 20 in Appendix for the complete proof.) Therefore, (A2) (and (B2)) finishes within $mn^2d/2$ interactions with probability $1-1/n = 1-o(1)$, by Markov's inequality.
- (A3), (B3): As mentioned above, (A3) (and (B3)) finishes within $2m\tau \le mt_{\mathrm{epi}}/8$ interactions with probability $1-o(1)$.
- (A4): By Lemma 8, (A4) finishes within $8m\delta\tau\lceil \ln n\rceil \le mt_{\mathrm{epi}}/8$ interactions with probability $1-o(1)$.

In total, with probability $1-o(1)$, (A1), (A2), ..., (B3) finish within $2(mt_{\mathrm{epi}} + mn^2d/2 + mt_{\mathrm{epi}}/8) + mt_{\mathrm{epi}}/8 \le 5mt_{\mathrm{epi}}/2 + mn^2d$ interactions. The population is in $\mathcal{S}_{\mathrm{AR}}$ at this time. To conclude, we have shown the following lemma.

Lemma 16: Let $C_0 \in \mathcal{T}_{\mathrm{one}} \cap \mathcal{T}_{\mathrm{half}} \cap \mathcal{V}_{\mathrm{zero}} \cap \mathcal{L}_{\mathrm{exist}}$ and $\Xi_{P_{\mathrm{AR}}}(C_0) = C_0, C_1, \ldots$. Then, $\Pr(\exists i \le 5mt_{\mathrm{epi}}/2 + mn^2d, C_i \in \mathcal{S}_{\mathrm{AR}}) \ge 1-o(1)$.

Lemma 17: $\max_{C\in\mathcal{C}_{\mathrm{all}}} \mathrm{ECT}_{P_{\mathrm{AR}}}(C, \mathcal{S}_{\mathrm{AR}}) = O(mt_{\mathrm{epi}} + mn^2d\log n)$.

Proof : Lemmas 12, 14, 15, and 16 yield that an execution of P_{AR} starting from any configuration reaches $\mathcal{S}_{\mathrm{AR}}$ within $4(mt_{\mathrm{epi}} + mn^2d\log_2 n)$ interactions with probability $1-o(1) \subset \Omega(1)$. Hence, the lemma follows from Lemma 10. □

Lemmas 9 and 17 give the following theorem.

Theorem 1: Protocol P_{AR} is an $(O(mn^2d\log n + mt_{\mathrm{epi}}), \Omega(\tau e^{\tau}))$- loosely stabilizing leader election protocol for arbitrary graphs G when $t_{\mathrm{virus}} \ge 4\delta\max(d, \lceil 7\ln n\rceil)$, $t_{\max} = 2t_{\mathrm{virus}}$, and $4\delta t_{\max}\lceil \ln n\rceil \le t_{\mathrm{epi}} \le \tau e^{\tau}/n^4$.

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 or random numbers, unlike our previous protocols. Given upper bounds N of n and Δ of δ, the population reaches a safe configuration within $O(mn^2d\log n + mN\Delta^2\log N)$ expected interactions and, after that, keeps a unique leader for $\Omega(Ne^N)$ expected interactions.

[†]Lemma 5 guarantees that if a leader exists, a high value of $\mathtt{timer}_{\mathrm{L}}$ propagates to the whole population, and the population reaches, within $2m\tau$ interactions, a configuration where $v.\mathtt{timer}_{\mathrm{L}} > t_{\max}/2$ for all $v \in V$, with probability $1-o(1)$. Since $t_{\max} = 2t_{\mathrm{virus}}$, this lemma is essentially the same as what we need to show here: if an agent with $\mathtt{timer}_{\mathrm{V}} = t_{\mathrm{virus}}$ exists, then the population reaches a configuration where $v.\mathtt{timer}_{\mathrm{V}} > 0$ for all $v \in V$ within $2m\tau$ interactions, with probability $1-o(1)$.

References

[1] Y. Sudo, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely-stabilizing leader election on arbitrary graphs in population protocols without identifiers nor random numbers," International Conference on Principles of Distributed Systems, 2015.

[2] D. Angluin, J. Aspnes, Z. Diamadi, M.J. Fischer, and R. Peralta, "Computation in networks of passively mobile finite-state sensors,"

Distributed Computing, vol.18, no.4, pp.235–253, 2006.
[3] D. Angluin, J. Aspnes, and D. Eisenstat, "Fast computation by population protocols with a leader," Distributed Computing, vol.21, no.3, pp.183–199, 2008.
[4] D. Angluin, J. Aspnes, M.J. Fischer, and H. Jiang, "Self-stabilizing population protocols," ACM Transactions on Autonomous and Adaptive Systems, vol.3, no.4, p.13, 2008.
[5] 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, vol.50, no.3, pp.433–445, 2012.
[6] R. Mizoguchi, H. Ono, S. Kijima, and M. Yamashita, "On space complexity of self-stabilizing leader election in mediated population protocol," Distributed Computing, vol.25, no.6, pp.451–460, 2012.
[7] X. Xu, Y. Yamauchi, S. Kijima, and M. Yamashita, "Space complexity of self-stabilizing leader election in population protocol based on k-interaction," Symposium on Self-Stabilizing Systems, pp.86–97, 2013.
[8] O. Michail, I. Chatzigiannakis, and P.G. Spirakis, "Mediated population protocols," Theoretical Computer Science, vol.412, no.22, pp.2434–2450, 2011.
[9] J. Beauquier, J. Burman, L. Rosaz, and B. Rozoy, "Non-deterministic population protocols," International Conference on Principles of Distributed Systems, pp.61–75, 2012.
[10] M. Fischer and H. Jiang, "Self-stabilizing leader election in networks of finite-state anonymous agents," International Conference on Principles of Distributed Systems, pp.395–409, 2006.
[11] J. Beauquier, P. Blanchard, and J. Burman, "Self-stabilizing leader election in population protocols over arbitrary communication graphs," International Conference on Principles of Distributed Systems, pp.38–52, 2013.
[12] D. Canepa and M.G. Potop-Butucaru, "Stabilizing leader election in population protocols," 2007. http://hal.inria.fr/inria-00166632.
[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, vol.444, pp.100–112, 2012.
[14] Y. Sudo, F. Ooshita, H. Kakugawa, T. Masuzawa, A.K. Datta, and L.L. Larmore, "Loosely-stabilizing leader election for arbitrary graphs in population protocol model," IEEE Trans. Parallel Distrib. Syst., vol.30, no.6, pp.1359–1373, 2019.
[15] T. Izumi, "On space and time complexity of loosely-stabilizing leader election," International Colloquium on Structural Information and Communication Complexity, pp.299–312, 2015.
[16] Y. Sudo, F. Ooshita, H. Kakugawa, T. Masuzawa, A.K. Datta, and L.L. Larmore, "Loosely-stabilizing leader election with polylogarithmic convergence time," 22nd International Conference on Principles of Distributed Systems (OPODIS 2018), pp.30:1–30:16, 2019.
[17] D. Alistarh, R. Gelashvili, and M. Vojnović, "Fast and exact majority in population protocols," the 34th ACM Symposium on Principles of Distributed Computing, pp.47–56, 2015.
[18] L. Gasieniec and G. Staehowiak, "Fast space optimal leader election in population protocols," Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pp.2653–2667, SIAM, 2018.
[19] G.B. Mertzios, S.E. Nikoletseas, C.L. Raptopoulos, and P.G. Spirakis, "Determining majority in networks with local interactions and very small local memory," International Colloquium on Automata, Languages, and Programming, pp.871–882, Springer, 2014.
[20] G. Cordasco and L. Gargano, "Space-optimal proportion consensus with population protocols," International Symposium on Stabilization, Safety, and Security of Distributed Systems, pp.384–398, Springer, 2017.
[21] D. Alistarh and R. Gelashvili, "Polylogarithmic-time leader election in population protocols," Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, pp.479–491, Springer, 2015.
[22] L. Gąsieniec, G. Stachowiak, and P. Uznański, "Almost logarithmic-time space optimal leader election in population protocols," arXiv preprint arXiv: 1802.06867, 2018.
[23] Y. Sudo, F. Ooshita, H. Kakugawa, and T. Masuzawa, "Loosely-stabilizing leader election on arbitrary graphs in population protocols," International Conference on Principles of Distributed Systems, pp.339–354, 2014.
[24] M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2005.

Appendix A: Lemmas

Lemma 4: 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, $\Pr(\#_{TI}(v_T, 0, 8m\delta\tau\lceil\ln n\rceil) < t_{\text{epi}}/2) \geq 1 - e^{-\tau}$.

Proof : For every $i \geq 0$, the unique 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, $\#_{TI}(v_T, 0, 8m\delta\tau\lceil\ln n\rceil)$ is bounded by a binomial random variable $X \sim B(8m\delta\tau\lceil\ln n\rceil, \delta/m)$. We have

$$\begin{aligned}\Pr(X \geq t_{\text{epi}}/2) &\leq \Pr(X \geq 16\delta^2\tau\lceil\ln n\rceil)\\ &= \Pr(X \geq 2\mathbf{E}[X])\\ &\leq e^{-\mathbf{E}[X]/3} = e^{-8\delta^2\tau\lceil\ln n\rceil/3} \leq e^{-\tau},\end{aligned}$$

where we use $t_{\text{epi}} \geq 32\delta^2\tau\lceil\ln n\rceil$ for the first inequality and use the Chernoff Bound in the form of Lemma 21 (in Appendix) with $\kappa = 1$ for the second inequality, which gives the lemma. □

Lemma 5 (in [23]): $\Pr(\text{PROP}_L(i) = 1) \geq 1 - 2ne^{-\tau}$ for any $i \geq 0$.

Proof : This lemma follows from the proof of Lemma 5 in [23]. That proof claims that if there is a leader in a configuration, then the $\texttt{timer}_{\text{L}}$ of every agent is more than $t_{\max}/2$ in the configuration reached by the next $2m\tau$ interactions with probability at least $1 - 2ne^{\tau}$. □

Lemma 6: $\Pr(\text{PROP}_T(i) = 1) \geq 1 - 2ne^{-\tau}$ for any $i \geq 0$.

Proof : The same argument as the proof of Lemma 5 gives the lemma. □

Lemma 7 (in [23]): $\Pr(\text{HALF}(i) = 1) \geq 1 - ne^{-\tau}$ for any $i \geq 0$.

Proof : The lemma follows from the proof of Lemma 18 in [23]. □

Lemma 8: $\Pr(\text{RT}_{\mathbf{\Gamma}}(t_{\text{virus}}) < 8m\delta\tau\lceil\ln n\rceil) \geq 1 - ne^{-\tau}$.

Proof : By Lemma 1, we have

$$\begin{aligned}&\Pr(\text{RT}_{\mathbf{\Gamma}}(t_{\text{virus}}) < 8m\delta\tau\lceil\ln n\rceil)\\ &\quad\geq \Pr(\text{RT}_{\mathbf{\Gamma}}(4\delta(1+\tau)) < 8m\delta\tau\lceil\ln n\rceil)\\ &\quad\geq \Pr(\text{RT}_{\mathbf{\Gamma}}(4\delta(1+\tau)) < 4m\delta(1+\tau)(1+\lceil\ln n\rceil))\\ &\quad\geq 1 - ne^{-\delta(\tau+1)},\end{aligned}$$

where we use $t_{\text{virus}} \leq 4\delta(1+\tau)$ for the first inequality and

use $(1+\tau)(1+\lceil \ln n \rceil) \leq 2\tau\lceil \ln n \rceil$ when $\tau \geq 3$ and $n \geq 3$ for the second inequality. □

Lemma 18: $\Pr(\mathrm{RT}_{\mathbf{\Gamma}}(t_{\max}) < 16m\delta\tau\lceil \ln n \rceil) \geq 1 - ne^{-\tau}$.

Proof : The lemma follows from almost the same calculation as the above proof for Lemma 8. □

Lemma 19: $\Pr(\#_{TI}(v, t, t + mt_{\mathrm{epi}}) < t_{\mathrm{epi}}) < e^{-\delta\tau}$ holds for any agent $v \in V$ and integer $t \geq 0$.

Proof : Every agent joins each interaction Γ_i with probability at least $2/m$. Hence, letting X be the binomial random variable such that $X \sim B(mt_{\mathrm{epi}}, 2/m)$, we have

$$\begin{aligned}\Pr(\#_{TI}(v, t, t + mt_{\mathrm{epi}}) < t_{\mathrm{epi}}) &\leq \Pr(X < t_{\mathrm{epi}}) \\ &\leq \Pr(X < E[X]/2) \\ &\leq e^{-t_{\mathrm{epi}}/4} \leq e^{-\delta\tau},\end{aligned}$$

where we use the Chernoff Bound in the form of Lemma 22 with $\kappa = \frac{1}{2}$ for the third inequality. □

Lemma 20: Let C_0 be a configuration where exactly one token exists and v be an agent. In the execution $\Xi_{P_{\mathrm{AR}}}(C_0)$, the expected number of interactions until the unique tokens reaches v is at most $mnd/2$.

Proof : Consider a random walk of the unique token on population G, *i.e.,* a Markov chain (u_t) in which a state of the chain is an agent in V (the location of the unique token) in G. 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 $(x, y) \in E$, $P_{x,y} = 1 - (2/m)\delta_x$ 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 any state y. Thus, $\pi = (\pi(x_1), \pi(x_2), \ldots, \pi(x_n)) = n^{-1}(1, 1, \ldots, 1)$ is the stationary distribution of the chain ($\pi P = \pi$), where $x_1, x_2, \ldots, x_n$ are all the states of the chain. We denote the expected number of transition steps from state x to state y by $h_{x,y}$. We have $h_{y,y} = 1/\pi(y) = n$ for any state y. We also have $h_{y,y} = 1 + \sum_{(y,z)\in E}(2/m)\cdot h_{z,y}$. Hence, we obtain $\sum_{(y,z)\in E} h_{z,y} = mn/2 - m/2$. Thus, we have $h_{x,y} \leq mn/2$ for any states x and y satisfying $(x, y) \in E$. Let $w_0, w_1, \ldots, w_l$ ($w_0 = u, w_l = v, l \leq d$) be the shortest path from u to v, where u is the agent that has the unique token in C_0. The expected time until the unique token reaches v is bounded by $\sum_{i=0}^{l-1} h_{w_i,w_{i+1}} \leq mnd/2$. □

Appendix B: Chernoff Bounds

Two variants of Chernoff bounds [24] used in several proofs of this paper are quoted below.

Lemma 21 (Eq. (4.2) in [24]): 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 22 (Eq. (4.5) in [24]): The following inequality holds for any binomial random variable X and κ, $0 < \kappa \leq 1$:

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

Yuichi Sudo received the B.E., M.E., and Ph.D. degrees in computer science from Osaka University in 2009, 2011, and 2014. He worked at NTT Corporation and was engaged in research on network security during 2011–2017. He is now an assistant professor at the Graduate School of Information Science and Technology, Osaka University. His research interests include distributed algorithms, graph theory, and network security.

Fukuhito Ooshita received the M.E. and D.I. degrees in computer science from Osaka University in 2002 and 2006. He had been an assistant professor in the Graduate School of Information Science and Technology at Osaka University during 2003–2015. He is now an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST). His research interests include parallel algorithms and distributed algorithms. He is a member of ACM, IEEE, IEICE, and IPSJ.

Hirotsugu Kakugawa received the B.E. degree in engineering in 1990 from Yamaguchi University, and the M.E. and D.E. degrees in information engineering in 1992, 1995 respectively from Hiroshima University. He is currently a professor of Ryukoku University. He is a member of the IEEE Computer Society and the Information Processing Society of Japan.

Toshimitsu Masuzawa received the B.E., M.E. and D.E. degrees in computer science from Osaka University in 1982, 1984 and 1987. He had worked at Osaka University during 1987–1994, and was an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) during 1994–2000. He is now a professor of Graduate School of Information Science and Technology, Osaka University. He was also a visiting associate professor of Department of Computer Science, Cornell University between 1993-1994. His research interests include distributed algorithms, parallel algorithms and graph theory. He is a member of ACM, IEEE, IEICE and IPSJ.