Vol.2009-AL-124 No.5
2009/5/11

Loosely-stabilizing Leader Election in Population Protocol Model

Yuichi Sudo ,[†1] Junya Nakamura ,[†1]
Yukiko Yamauchi ,[†2] Fukuhito Ooshita ,[†1]
Hirotsugu Kakugawa [†1] and Toshimitsu Masuzawa [†1]

A self-stabilizing protocol guarantees that, starting from an arbitrary initial configuration, a system eventually comes to satisfy its specification and keeps the specification forever. Although self-stabilizing protocols show excellent fault-tolerance against any transient faults (e.g. memory crash), designing self-stabilizing protocols is difficult and, what is worse, might be impossible due to the severe requirements. To circumvent the difficulty and impossibility, we introduce in this paper a novel notion of *loose-stabilization*. The loose-stabilization relaxes the closure requirement; starting from an arbitrary configuration, a system comes to satisfy its specification in a relatively short time, and it keeps the specification *for a long time, though not forever*. To show effectiveness and feasibility of the new concept, we present a probabilistic loosely-stabilizing leader election protocol in the Probabilistic Population Protocol (PPP) model of complete networks. The protocol elects a unique leader within $O(nN \log n)$ expected steps starting from any configuration, and keeps the unique leader for $\Omega(Ne^N)$ expected steps, where n is the network size (not known to the protocol) and N is a known upper bound of n. This result proves that introduction of the loose-stabilization circumvents the already-known impossibility result; the self-stabilizing leader election in the PPP model of complete networks cannot be solved without knowledge on the exact network size.

1. Introduction

A distributed system is a collection of autonomous computational entities (processes) connected by communication links. Fault tolerance of distributed systems has attracted more and more attention since distributed systems are prone to faults. A self-stabilizing system[6] has a desirable property that, even when any transient fault (*e.g.* memory crash at processes) hits the system, it can autonomously recover from that fault. The notion of self-stabilization is described as follows: (i) starting from an arbitrary initial configuration, a system eventually reaches a *safe configuration* (*convergence*), and (ii) once a system reaches a safe configuration, then it keeps its specification forever (*closure*). Although self-stabilizing systems provide excellent fault-tolerance as mentioned above, designing self-stabilizing protocols is difficult and, what is worse, might be impossible due to the severe requirements.

To circumvent this difficulty and impossibility, many researchers have tried to relax the severe requirement of self-stabilization and proposed the following variants. *Probabilistic self-stabilization*[8] guarantees convergence to a safe configuration with probability 1 starting from an arbitrary configuration. *Quasi-stabilization*[9] guarantees convergence to a safe configuration only when all processes in the system start with the program counters of value 0. *Weak-stabilization*[7] guarantees that starting from an arbitrary configuration there exists an execution that reaches a safe configuration. Devismes et al.[5] investigated the relations among self, probabilistic and weak stabilization. A notable characteristic common to all the above variants is that they relax only the convergence requirement but not the closure requirement of self-stabilization.

In this paper, we adopt Probabilistic Population Protocol (PPP) model[1),2)] as a distributed system model. The population protocol model[1)] is one of abstractions that represent wireless sensor networks consisting of mobile sensing devices. In this model, two devices communicate with each other only when they come sufficiently close to each other (we call this event an interaction). For example, population protocol model can represent a flock of birds such that each bird is equipped with a sensing device of small transmission range. In such a sensor network, each device can communicate with another device only when the corresponding birds come sufficiently close to each other. The PPP model is a population protocol model with the assumption that any interaction occurs uniformly at random.

Our contribution. To circumvent difficulty and impossibility in designing self-stabilizing protocols, we introduce a novel notion of *loose-stabilization*, which relaxes the closure requirement of self-stabilization. To the best of our knowledge,

†1 Graduate School of Information Science and Technology, Osaka University
†2 Graduate School of Information Science, Nara Institute of Science and Technology

this is the first trial to relax the closure requirement and not the convergence requirement. Intuitively, the notion of loose-stabilization is described as follows: (i) starting from an arbitrary configuration, a system reaches a *loosely-safe configuration* within a short time (*convergence*), and (ii) once a system reaches a loosely-safe configuration, then it keeps its specification for a long time (*loose-closure*). In other words, we relax the closure requirement by allowing a system to deviate from its specification even after a loosely-safe configuration but only after a long period satisfying the specification. The requirement of fast convergence is added to guarantee that most of the system running time should satisfy the specification. Actually, the loose-stabilization is practically equivalent to self-stabilization if the specification is kept for a significantly long time (e.g. exponential order with the network size) after the loosely-safe configuration.

Several definitions with the above notion can be formulated, and in this paper, we give a concrete definition of *probabilistic loose-stabilization*, which ensures the fast convergence and the long period satisfying the specification in terms of *expected time*.

To show effectiveness and feasibility of the loose-stabilization, we present a probabilistic loosely-stabilizing leader election protocol in the PPP model of complete networks. The protocol assumes that each device knows an upper bound, say N, of the network size. Starting from any configuration, the protocol elects a unique leader within $O(nN \log n)$ expected steps, and then, keeps the unique leader for $\Omega(Ne^N)$ expected steps where n is the actual network size. This result discloses an evidence that introduction of the loose-stabilization can circumvent impossibility results on self-stabilization; the self-stabilizing leader election in the PPP model of complete networks cannot be solved even in a probabilistic way without knowledge of the exact network size[4]. The proposed protocol uses $O(\log N)$ space per device while prior papers on population protocols usually do not allow each device to use more than constant space (with respect to n). However, the importance of our protocol is never impaired by this fact because the above impossibility holds even if each device can use infinite space.

2. Preliminaries

In this section, we show the definition of probabilistic population protocol model and define the concept of probabilistic loose-stabilization. We use some definitions in 1), 3).

A *population* consists of a collection of finite state sensing devices called *agents*. Each agent has its own state and updates the state by communication with other agents in pairs, called *interactions*. We represent a population by simple directed graph $G(V, E)$: $V = \{0, 1, \ldots, n-1\}$ $(n \geq 2)$ is a set of agents and $E \subseteq V \times V$ is a set of possible interactions. If $(u, v) \in E$, agents u and v can interact with each other in such a way that u serves as an *initiator* and v serves as a *responder*. In this paper, we assume that a population $G(V, E)$ is a complete graph, *i.e.*, $(u, v) \in E$ for any distinct agents $u, v \in V$.

A *protocol* $P(Q, Y, O, \delta)$ consists of a finite set of states Q, a finite set of output symbols Y, an output function $O : Q \to Y$, and a transition function $\delta : Q \times Q \to Q \times Q$. The *output of an agent* is determined by O: when the state of an agent is $p \in Q$, the output of the agent is $O(p)$. When an interaction between two agents happens, δ determines the next states of the two agents after the interaction. For agent u with state p and agent v with state q, $\delta(p, q) = (p', q')$ represents that the states of u and v after the interaction (u, v) are p' and q' respectively.

A *configuration* is a mapping $C : V \to Q$ that specifies the states of all agents in a population. The output of a configuration C is defined as a composite function $O \circ C : V \to Y$, denoted by $O(C)$. Let C and D be configurations, and let u and v be distinct agents. We say that C changes to D by an interaction $r = (u, v)$, denoted by $C \xrightarrow{r} D$, if we have $(D(u), D(v)) = \delta(C(u), C(v))$ and $D(w) = C(w)$ for all $w \in V$ except u and v.[⋆1] We denote by $\mathcal{C}_{\mathrm{all}}(P)$ the set of all configurations of P.

An *interaction sequence* $\gamma = (u_0, v_0), (u_1, v_1), \ldots$ is an infinite sequence of interactions. For each $t \geq 0$, we denote u_t and v_t by $\gamma_1(t)$ and $\gamma_2(t)$ respectively, and denote (u_t, v_t) by $\gamma(t)$. We call $\gamma(t)$ *the interaction at time t in γ*. We say that agent v *joins in* interaction $\gamma(t)$ when $v \in \{\gamma_1(t), \gamma_2(t)\}$.

Given an interaction sequence γ and an initial configuration C_0, the *execution* $\Xi_P(C_0, \gamma)$ of a protocol P is uniquely defined as $\Xi_P(C_0, \gamma) = C_0, C_1, \ldots$ s.t. $\forall t \geq$

⋆1 This definition implies that interactions between two agents happen sequentially, that is, exactly one pair of agents interact at any time.

$0,\ C_t \overset{\gamma(t)}{\rightarrow} C_{t+1}$.

A scheduler determines which interaction happens at each time t ($t \geq 0$). In this paper, we consider a uniformly random scheduler: the interaction at each time is chosen at random, independently and uniformly from all possible interactions. We represent the choice of this scheduler by the interaction sequence Γ: each $\Gamma(t)$ is a random variable such that $\Pr(\Gamma(t) = (u, v)) = \frac{1}{|E|}$ for any arbitrary interactions $(u, v) \in E$ and for any integer $t \geq 0$.

2.1 Behavior

In this section, we define *behavior* to describe the specification of a problem. A *trace* T on population $G(V, E)$ is a finite or infinite sequence of assignments from V to Z, where Z is a set of symbols. We call Z the *alphabet* of T. If $Z = Q$ for protocol $P(Q, Y, O, \delta)$ then we say that T is a *configuration trace* of P.[⋆1] Let $T = C_0, C_1, \ldots$ be a finite or infinite configuration trace of P. The *output trace* of T for P is $OT_P(T) = O(C_0), O(C_1), \ldots$.

For a finite trace $T = \lambda_0, \lambda_1, \ldots, \lambda_{l-1}$, we define the length of T as $|T| = l$. For an infinite trace T', we define $|T'| = \infty$. Let $T = \lambda_0, \lambda_1, \ldots$ be a finite or infinite trace. The *sub-trace* $T_{x,y}$ ($0 \leq x \leq y \leq |T| - 1$) [⋆2] is a sequence of assignments $T_{x,y} = \lambda_x, \lambda_{x+1}, \ldots, \lambda_y$. The *prefix* of T, $T_{0,l}$ ($0 \leq l \leq |T| - 1$) is denoted by $T_{\mathrm{pre}}(l)$.

A *behavior* $B(Z)$ on population $G(V, E)$ is a set of traces on G that have an identical alphabet Z. (We use the notation B if Z is clear from context.) We define a *problem* as a behavior that specifies the set of all legitimate output traces for the problem. Let $B(Y)$ be a behavior and let T be a configuration trace of $P(Q, Y, O, \delta)$. Trace T is legitimate for the problem defined by B iff $OT_P(T) \in B$. We say that a behavior B is *canonical* if $T_{x,y} \in B$ for any trace $T \in B$ and any x, y ($0 \leq x \leq y < |T|$).

Definition 1 (Leader Election Problem) We denote by le the set of all assignment $\omega : V \rightarrow \{F, L\}$ such that for some $v_l \in V$, $\omega(v_l) = L$ and for all $v \neq v_l$, $\omega(v) = F$. The leader election behavior $LE(\{F, L\})$ on population $G(V, E)$ is the set of all traces $T = \omega, \omega, \ldots$ ($1 \leq |T| \leq \infty$) such that ω belongs to le.

⋆2 Note that y can be ∞ if $|T| = \infty$.

Informally, LE requires that any legitimate execution of a protocol for leader election has one static leader agent with the output symbol L and $n - 1$ non-leader (follower) agents with the output symbol F through its all configuration. Clearly, LE is canonical.

2.2 Probabilistic Loose-stabilization

In this section, we define the notion of *probabilistic loose-stabilization*.

Let $P(Q, Y, O, \delta)$ be a protocol and $B(Y)$ be a canonical behavior. Let $T = D_0, D_1, \ldots$ be a finite or infinite configuration trace of P. If there exists an integer t ($t \geq 0$) such that $OT_P(T_{\mathrm{pre}}(t)) \in B$ and $OT_P(T_{\mathrm{pre}}(t+1)) \notin B$, the *maintenance trace* $MT_P(T, B)$ is defined by $T_{\mathrm{pre}}(t)$. If such t does not exist, we define $MT_P(T, B)$ as follows: if $OT(T_{\mathrm{pre}}(0)) \in B$ then $MT_P(T, B) = T$, otherwise $MT_P(T, B) = \varepsilon$, where ε is the empty trace ($|\varepsilon| = 0$). Let C_0 be a configuration of P. We denote $\mathbf{E}[|MT_P(\Xi_P(C_0, \Gamma), B)|]$ by $EMT_P(C_0, B)$. Intuitively, when an execution of P starts from C_0, the execution satisfies the specification defined by B during $EMT_P(C_0, B)$ expected interactions.

Let $\mathcal{C}$ be a set of configurations of P. If there exists an integer t such that $D_i \notin \mathcal{C}$ for all i ($i = 0, 1, \ldots, t$) and $D_{t+1} \in \mathcal{C}$, the *convergence trace* $CT_P(T, \mathcal{C})$ is defined by $T_{\mathrm{pre}}(t)$. If such t does not exist, we define $CT_P(T, \mathcal{C})$ as follows: if $D_0 \in \mathcal{C}$ then $CT_P(T, \mathcal{C}) = \varepsilon$, otherwise $CT_P(T, \mathcal{C}) = T$. We denote $\mathbf{E}[|CT_P(\Xi_P(C_0, \Gamma), \mathcal{C})|]$ by $ECT_P(C_0, \mathcal{C})$. Intuitively, when an execution of P starts from C_0, the execution reaches a configuration of $\mathcal{C}$ within $ECT_P(C_0, \mathcal{C})$ expected interactions.

Definition 2 (Probabilistic Loose-stabilization) Let α and β be real numbers. A protocol $P(Q, Y, O, \delta)$ is (α, β)-probabilistic loosely-stabilizing for a canonical behavior $B(Y)$ and a nonempty set of configurations $\mathcal{S}$ if the following equations hold:

$$\max_{C \in \mathcal{C}_{\mathrm{all}}(P)} ECT_P(C, \mathcal{S}) \leq \alpha,$$
$$\min_{C \in \mathcal{S}} EMT_P(C, B) \geq \beta.$$

We say that a configuration C of P is a *β-loosely-safe configuration* for P and B when $EMT_P(C, B) \geq \beta$. Clearly, $\mathcal{S}$ in the above definition consists of β-loosely-safe configurations for P and B.

Intuitively, a (α, β)-probabilistic loosely-stabilizing protocol is quite useful if β is sufficiently large (*e.g.* exponential order with n) and α is relatively small (*e.g.*

R1.	$((l,*),(l,*))$	$\to$	$((l,s),(-,s))$
R2.	$((l,*),(-,*))$	$\to$	$((l,s),(-,s))$
R3.	$((-,*),(l,*))$	$\to$	$((-,s),(l,s))$
R4.	$((-,0),(-,0))$	$\to$	$((l,s),(-,s))$
R5.	$((-,i),(-,j))$	$\to$	$((-,f),(-,f))$ $(0 \le i,j \le s,\ f = \max(i,j)-1)$

Fig. 1 the transition function δ of P_{LE}

low polynomial order with n).

3. Probabilistic loosely-stabilizing Leader Election

3.1 The Proposed Protocol

In this section, we present a probabilistic loosely-stabilizing protocol $P_{LE}(Q,\{F,L\},O,\delta)$ that solves the leader election problem with knowledge of an upper bound N of the network size n. The protocol has a design parameter s and becomes a probabilistic loosely-stabilizing protocol when s is adequately set depending on N (Theorem 2).

Each agent has one *leader bit* and a *timer* that takes an integer value in $[0,s]$, *i.e.* $Q = \{-,l\} \times \{0,1,\ldots,s\}$. We define the output function O as follows: if the leader bit of an agent is l, then the output of the agent is L, otherwise F. We call an agent with the leader bit l $(-)$ a leader (non-leader, respectively). We describe the transition function δ by pattern rules in Figure 1. Given any pair of states (p,q), the pair of the next states $\delta(p,q)$ is defined as follows: (i) if (p,q) matches the left side of exactly one rule, $\delta(p,q)$ is determined by the right side of the rule, and (ii) if there are two or more matched rules, $\delta(p,q)$ is determined by the right side of the matched rule with the smallest rule number. The symbol $*$ means "don't care", that is, $*$ matches any value of the timer. Note that this five rules are collectively exhaustive.

If two leaders interact, one remains a leader and the other becomes a non-leader (R1). If a leader and a non-leader interact, the leader bits of both the agents do not change (R2, R3). In every interaction in which one or two leaders join, the timers of both the agents are reset to the full value s (R1, R2, and R3). We call this event *timer reset*. A new leader is created only when two non-leaders with timer value 0 interact (R4). We call this event *timeout*. If two non-leaders interact where either or both agents have non-zero timer, then at least one of the two agents decrements its timer value by 1 (R5). R5 plays another role of *propagating the higher timer value*: intuitively, when two non-leaders interact, the timer of a lower value is set to the other (higher) value (minus 1).

In a configuration containing at least one leader, timeout rarely happens because of frequent occurrences of timer reset and propagations of high timer values. On the other hand, in a configuration containing no leader, timeout happens in a relatively short time because of no possibility of timer reset. Hence, starting from any configuration, removing leaders by R1 or creating a leader by R4 eventually bring the system to a configuration with exactly one leader. The following two properties hold clearly: (i) once a configuration with one or more leaders is reached, the number of leaders cannot become 0 thereafter, and (ii) once a unique leader is elected, P_{LE} keeps the unique leader until the next timeout happens.

3.2 Epidemic and Virtual Agents

In this section, we introduce the notion of *epidemic* (presented in 2)) and *virtual agents* for the proof in Section 3.3.

We define $\mathcal{L}_{\text{one}}$ as a set of configurations in which there exists exactly one leader in the population. Let C_0 be a configuration in $\mathcal{L}_{\text{one}}$, and let $v_l \in V$ be the unique leader in C_0. Let γ be an interaction sequence. The *epidemic function* $I_{C_0,\gamma}(t)$ $(t = 0,1,\ldots)$ that returns a set of agents is defined as follows: $I_{C_0,\gamma}(0) = \{v_l\}$, and $I_{C_0,\gamma}(t) = I_{C_0,\gamma}(t-1) \cap Add_{C_0,\gamma}(t)$ for any $t \ge 1$, where

$$Add_{C_0,\gamma}(t) = \begin{cases} \{\gamma_1(t-1),\gamma_2(t-1)\} & \text{if } I_{C_0,\gamma}(t-1) \cap \{\gamma_1(t-1),\gamma_2(t-1)\} \\ \emptyset & \text{otherwise.} \end{cases}$$

We say that, if $v \in I_{C_0,\gamma}(t)$, v is *infected* at time t in the epidemic starting from C_0 under γ, otherwise v is *infection-free* at time t in that epidemic. At time 0, only v_l is infected, and an infection-free agent becomes infected when it interacts with an infected agent. Once an agent becomes infected, it remains infected thereafter.

In the following, we define the *virtual agent* $VA_{C_0,\gamma}(v)$ of each agent $v \in V$. We assume that all agents eventually become infected, that is, $I_{C_0,\gamma}(t') = V$ holds

for some $t' \geq 0$. The virtual agent $VA_{C_0,\gamma}(v)$ is not defined if no such t' exists for C_0 and γ. Let v be any agent other than v_l. The *infected time* $T_{C_0,\gamma}(v)$ of v is an integer $i \geq 0$ that satisfies $v \notin I_{C_0,\gamma}(i)$ and $v \in I_{C_0,\gamma}(i+1)$. The *parent* of v, denoted by $P_{C_0,\gamma}(v)$, is the agent that infects v. It is formally defined as agent u such that $\{u\} = \{\gamma_1(T_{C_0,\gamma}(v)), \gamma_2(T_{C_0,\gamma}(v))\} \setminus \{v\}$. We define agent $P^k_{C_0,\gamma}(v)$ $(k \geq 0)$ as follows: $P^0_{C_0,\gamma}(v) = v$, and $P^k_{C_0,\gamma}(v) = P_{C_0,\gamma}(P^{k-1}_{C_0,\gamma}(v))$ for $k \geq 1$. Intuitively, $P^k_{C_0,\gamma}(v)$ is v's ancestor k generations back. Obviously, there exists an integer $m \geq 0$ such that $P^m_{C_0,\gamma}(v) = v_l$. For each $0 \leq i \leq m$, let w_i be $P^{m-i}_{C_0,\gamma}(v)$. Note that $w_0 = v_l$ and $w_m = v$. The *infecting path* of v is defined as $v_l = w_0 \to w_1 \to \cdots \to w_m = v$. Let t_i $(1 \leq i \leq m)$ be $T_{C_0,\gamma}(w_i)$. The virtual agent $VA_{C_0,\gamma}(v)$ is a virtual entity that migrates from v_l to v through the infecting path of v. This notion is formalized as *the location of the virtual agent* $L_{C_0,\gamma}(v,t)$ $(t \geq 0)$,which is defined as follows:

$$L_{C_0,\gamma}(v,t) = \begin{cases} v_l & (0 \leq t \leq t_1) \\ w_i & (t_i + 1 \leq t \leq t_{i+1},\ 1 \leq i \leq m-1) \\ v & (t \geq t_m + 1 = T_{C_0,\gamma}(v) + 1). \end{cases}$$

For the leader agent v_l, we define $L_{C_0,\gamma}(v_l,t) = v_l$ for any $t \geq 0$.

Let v be an agent in V.[⋆1] For simplicity, we denote the virtual agent $VA_{C_0,\gamma}(v)$ by v' here. We say that the virtual agent v' joins in interaction $\gamma(t)$ if agent $L_{C_0,\gamma}(v,t)$ joins in $\gamma(t)$, and we define indicator variable $VJ_{C_0,\gamma}(v,t)$ for any $t \geq 0$ as follows: if v' joins in $\gamma(t)$, then $VJ_{C_0,\gamma}(v,t) = 1$, otherwise $VJ_{C_0,\gamma}(v,t) = 0$. *The number of virtual interactions* of v is defined as $VI_{C_0,\gamma}(v,t) = \sum_{i=0}^{t-1} VJ_{C_0,\gamma}(v,i)$. Intuitively, $VI_{C_0,\gamma}(v,t)$ is the number of interactions in which v' joins between time 0 and time $t-1$.

In the rest of this section, we prove two lemmas. Informally, these two lemmas assure that the virtual agent v' brings an large timer value to v with high probability when v' reaches v through the infecting path of v. For state p, we denote the second element (timer) of p by $p.time$.

Lemma 1 *Let C_0 be a configuration in $\mathcal{L}_{\text{one}}$ and let γ be an interaction sequence. Let $\Xi_{P_{LE}}(C_0,\gamma) = C_0, C_1, \ldots$. The following predicate holds for any agent $v \in V$ and any $t \geq 0$:*

$$I_{C_0,\gamma}(t) = V \ \Rightarrow\ C_t(v).time \geq s - VI_{C_0,\gamma}(v,t).$$

Proof Sketch. Assume $I_{C_0,\gamma}(t) = V$. Let v_l be the unique leader in C_0 and t_{first} be the first time at which v_l have interaction, i.e. $t_{first} = \min\{i \geq 0 \mid v_l \in \{\gamma_1(i), \gamma_2(i)\}\}$. Then, it is easily shown by induction with respect to i that $C_i(L_{C_0,\gamma}(v,i)).time \geq s - VI_{C_0,\gamma}(v,i)$ holds for any integer $i \geq t_{first}+1$ (we omit the proof). Since $I_{C_0,\gamma}(t) = V$, $t \geq t_{first}+1$ and $v = L_{C_0,\gamma}(v,t)$ clearly hold. Hence, we have $C_t(v).time = C_t(L_{C_0,\gamma}(v,t)).time \geq s - VI_{C_0,\gamma}(v,t)$. □

The following lemma probabilistically bounds the number of virtual interactions of each agent by a certain binomial distribution. Recall that random variable Γ is the interaction sequence that represents the choice of uniformly random scheduler.

Lemma 2 *Let C_0 be a configuration in $\mathcal{L}_{\text{one}}$ and let $X(i)$ be a binomial random variable such that $X(i) \sim B(i, \frac{4}{n})$ for integer $i \geq 0$. $\Pr(VI_{C_0,\Gamma}(v,t) \geq j+n-1 \mid I_{C_0,\Gamma}(t) = V) \leq \Pr(X(t) \geq j)$ holds for any $v \in V$ and any integers $t \geq n$ and $j \geq 0$.*

Proof. Assume $I_{C_0,\Gamma}(t) = V$ and let $v_l \in V$ be the unique leader in C_0. We define the *infecting time set* IT as $\bigcup_{v \in V \setminus \{v_l\}} \{T_{C_0,\Gamma}(v)\}$, and the *non-infecting time set* NIT as $\{0,1,\ldots,t-1\} \setminus IT$. Let v be any agent in V, and let $NVI = \sum_{t' \in NIT} VJ_{C_0,\Gamma}(v,t')$. Since $|IT| = n-1$, the inequality $VI_{C_0,\Gamma}(v,t) \leq NVI + n - 1$ immediately follows. Therefore, it is sufficient for our proof to show $\Pr(NVI \geq j \mid I_{C_0,\Gamma}(t) = V) \leq \Pr(X(t) \geq j)$.
Let t' be any integer in $[0, t-1]$ and let $m = |I_{C_0,\Gamma}(t')|$. If $t' \in NIT$, the interaction $\Gamma(t')$ must be an interaction such that both agents $\Gamma_1(t')$ and $\Gamma_2(t')$ belong to $I_{C_0,\Gamma}(t')$ or both the agents belong to $V \setminus I_{C_0,\Gamma}(t')$. Thus, letting ${}_0C_2 = {}_1C_2 = 0$, we have

$$\Pr(VJ_{C_0,\Gamma}(v,t') = 1 \mid I_{C_0,\Gamma}(t) = V \wedge t' \in NIT \wedge L_{C_0,\Gamma}(v,t') \in I_{C_0,\Gamma}(t')) = \frac{m-1}{{}_mC_2 + {}_{n-m}C_2},$$

$$\Pr(VJ_{C_0,\Gamma}(v,t') = 1 \mid I_{C_0,\Gamma}(t) = V \wedge t' \in NIT \wedge L_{C_0,\Gamma}(v,t') \notin I_{C_0,\Gamma}(t')) = \frac{n-m-1}{{}_mC_2 + {}_{n-m}C_2}.$$

These inequalities lead $\Pr(VJ_{C_0,\Gamma}(v,t') = 1 \mid I_{C_0,\Gamma}(t) = V \wedge t' \in NIT) \leq 4/n$

⋆1 Note that v can be v_l.

because $\frac{m-1}{{}_mC_2+_{n-m}C_2} \leq \frac{4}{n}$ and $\frac{n-m-1}{{}_mC_2+_{n-m}C_2} \leq \frac{4}{n}$ hold. Note that this upper bound $4/n$ of the probability is independent from the interaction at any time other than t'. Hence, for any set S of $t-n+1$ distinct integers in $[0, t-1]$, we have

$$\Pr\left(\sum_{t' \in NIT} VJ_{C_0,\Gamma}(v,t') \geq j \mid I_{C_0,\Gamma}(t) = V \ \wedge \ NIT = S\right) \leq \Pr(X(t-n+1) \geq j).$$

Therefore, following inequality holds and so does the lemma.

$$\begin{aligned}\Pr(NVI \geq j \mid I_{C_0,\Gamma}(t) = V) &= \Pr\left(\sum_{t' \in NIT} VJ_{C_0,\Gamma}(v,t') \geq j \mid I_{C_0,\Gamma}(t) = V\right) \\ &\leq \Pr(X(t-n+1) \geq j) \\ &\leq \Pr(X(t) \geq j).\end{aligned}$$

□

3.3 Analysis and Proofs

Assume that we set design parameter s so that s is multiple of 96 and $s \geq \max(3n, 96(2\ln n + \ln 24))$ holds. In this section, we prove that under this assumption, P_{LE} is $(O(ns\log n), \Omega(se^{s/96}))$-probabilistic loosely-stabilizing for behavior LE and $\mathcal{S}_{\mathrm{half}}$, where $\mathcal{S}_{\mathrm{half}}$ is the set of all configurations in which there exists exactly one leader and the timer value of every agent is greater than or equal to $s/2$. To claim it, we prove the following two expressions:

$$\max_{C \in \mathcal{C}_{\mathrm{all}}(P_{LE})} ECT_{P_{LE}}(C, \mathcal{S}_{\mathrm{half}}) \in O(ns\log n), \tag{1}$$

$$\min_{C \in \mathcal{S}_{\mathrm{half}}} EMT_{P_{LE}}(C, LE) \in \Omega\left(s \cdot \exp\left(\frac{s}{96}\right)\right). \tag{2}$$

First, we prove Expr.(2). In the following, we denote $\mathcal{C}_{\mathrm{all}}(P_{LE})$ by $\mathcal{C}_{\mathrm{all}}$ for simplicity.

Lemma 3 Expr.(2) *holds if the following equation holds for any configuration* C_0 *in* $\mathcal{S}_{\mathrm{half}}$*:*

$$\Pr\left((\Xi_{P_{LE}}(C_0,\Gamma))_{\mathrm{pre}}\left(\frac{ns}{48}\right) \in LE \ \wedge \ C_{\frac{ns}{48}} \in \mathcal{S}_{\mathrm{half}}\right) \geq 1 - 2n \cdot \exp\left(-\frac{s}{96}\right), \tag{3}$$

where $\Xi_{P_{LE}}(C_0,\Gamma) = C_0, C_1, \ldots, C_{\frac{ns}{48}}, \ldots$.

Proof. Assume that Expr.(3) holds for any configuration in $\mathcal{S}_{\mathrm{half}}$. Then the inequality $EMT_{P_{LE}}(C_0, LE) \geq (1-2ne^{-s/96})(\frac{ns}{48} + \min_{C \in \mathcal{S}_{\mathrm{half}}} EMT_{P_{LE}}(C, LE)))$ clearly holds for any configuration $C_0 \in \mathcal{S}_{\mathrm{half}}$. Hence, we have

$$\begin{aligned}&\min_{C \in \mathcal{S}_{\mathrm{half}}} EMT_{P_{LE}}(C, LE) \\ &\geq \left(1 - 2n \cdot \exp\left(-\frac{s}{96}\right)\right)\left(\frac{ns}{48} + \min_{C \in \mathcal{S}_{\mathrm{half}}} EMT_{P_{LE}}(C, LE)\right).\end{aligned}$$

Solving this inequality gives us Expr.(2). □

In the following, we show that Expr.(3) holds for any configuration $C_0 \in \mathcal{S}_{\mathrm{half}}$. Firstly, we prove the probability of $C_{\frac{ns}{48}} \in \mathcal{S}_{\mathrm{half}}$ is sufficiently close to 1 (Lemma 4,5 and Corollary 1), and secondly, we prove that the probability of $(\Xi_{P_{LE}}(C_0,\Gamma))_{\mathrm{pre}}(\frac{ns}{48}) \in LE$ is sufficiently close to 1 (Lemma 6 and Corollary 2).

Lemma 4 *Let* C_0 *be a configuration in* $\mathcal{L}_{\mathrm{one}}$*. The following inequality holds:*

$$\Pr\left(\max_{v \in V} VI_{C_0,\Gamma}\left(v, \frac{ns}{48}\right) \leq \frac{s}{2} \ \mid \ I_{C_0,\Gamma}\left(\frac{ns}{48}\right) = V\right) \geq 1 - n \cdot \exp\left(-\frac{s}{36}\right). \tag{4}$$

Proof. Applying Chernoff bounds, $\Pr(Y \geq (1+\delta)\mathbf{E}[Y]) \leq \exp(-\delta^2\mathbf{E}[Y]/3)$ holds for any binomial random variable Y and any real number δ $(0 \leq \delta \leq 1)$. (See Expr.4.2 in 10).) Let X be an binomial variable such that $X \sim B(\frac{ns}{48}, \frac{4}{n})$. It follows from the above inequality that $\Pr(X \geq \frac{s}{6}) \leq \exp(-s/36)$. Let v be any agent. By Lemma 2 and the assumption $s \geq 3n$, we have

$$\begin{aligned}&\Pr\left(VI_{C_0,\Gamma}\left(v, \frac{ns}{48}\right) \geq \frac{s}{2} \mid I_{C_0,\Gamma}\left(\frac{ns}{48}\right) = V\right) \\ &\leq \Pr\left(VI_{C_0,\Gamma}\left(v, \frac{ns}{48}\right) \geq \frac{s}{6} + n - 1 \mid I_{C_0,\Gamma}\left(\frac{ns}{48}\right) = V\right) \qquad \because \frac{s}{2} \geq \frac{s}{6} + n - 1 \\ &\leq \Pr\left(X \geq \frac{s}{6}\right) \leq \exp\left(-\frac{s}{36}\right).\end{aligned}$$

We obtain (4) by summing up all above probabilities with respect to $v \in V$. □

Lemma 5 $\Pr(I_{C_0,\Gamma}\left(\frac{ns}{48}\right) = V) \geq 1 - n \cdot \exp\left(-\frac{s}{96}\right)$ *holds for any configuration* C_0 *in* $\mathcal{L}_{\mathrm{one}}$*.*

Proof. For each k $(2 \leq k \leq n)$, we define $T(k)$ as integer t such that $|I_{C_0,\Gamma}(t-1)| = k-1$ and $|I_{C_0,\Gamma}(t)| = k$, and define $T(1) = 0$. Intuitively, $T(k)$ is the first time at which there exists k infected agents in the population. Let $X_{\mathrm{pre}} = T(\lceil \frac{n+1}{2} \rceil)$ and $X_{\mathrm{post}} = T(n) - T(n - \lceil \frac{n+1}{2} \rceil + 1)$. Angluin *et al.* found in 2) that $T(k)$ and $T(n) - T(n-k+1)$ have the same probability distribution for any k $(1 \leq k \leq n)$. Hence, so do X_{pre} and X_{post}. And, $X_{\mathrm{pre}} + X_{\mathrm{post}} \geq T(n)$ holds

because $\lceil \frac{n+1}{2} \rceil \geq n - \lceil \frac{n+1}{2} \rceil + 1$. We denote $T(n - \lceil \frac{n+1}{2} \rceil + 1)$ by T_{half} and let $X_v = \max(T_{C_0,\Gamma}(v) - T_{\mathrm{half}}, 0)$ for any agent v. Informally, X_v is the number of interactions that occurs between time T_{half} and the time at which agent v becomes infected. Consider the case $v \notin I_{C_0,\Gamma}(T_{\mathrm{half}})$. At any time $t \geq T_{\mathrm{half}}$, at least $n - \lceil \frac{n+1}{2} \rceil + 1$ ($\geq \frac{n}{2}$) agents are infected. Therefore, each interaction at time $t \geq T_{\mathrm{half}}$ infects v with the probability of at least $\frac{1}{{}_nC_2} \cdot \frac{n}{2} \geq \frac{1}{n}$, and hence, we have $\Pr(X_v > \frac{ns}{96}) \leq (1 - \frac{1}{n})^{ns/96} \leq \exp(-\frac{s}{96})$. Since the number of infection-free agent at time T_{half} is at most $\frac{n}{2}$, $\Pr(X_{\mathrm{post}} > \frac{ns}{96}) \leq \Pr(\bigvee_{v \in V}(X_v \geq \frac{ns}{96})) \leq \sum_{v \in V} \Pr(X_v \geq \frac{ns}{96}) \leq \frac{n}{2} \cdot \exp(-\frac{s}{96})$. By the equivalence of the distribution of X_{pre} and X_{post}, we have

$$\begin{aligned}\Pr\left(I_{C_0,\Gamma}\left(\frac{ns}{48}\right) \neq V\right) &= \Pr\left(T(n) > \frac{ns}{48}\right) \\ &\leq \Pr\left(X_{\mathrm{pre}} > \frac{ns}{96}\right) + \Pr\left(X_{\mathrm{post}} > \frac{ns}{96}\right) \\ &\leq n \cdot \exp\left(-\frac{s}{96}\right).\end{aligned}$$

□

We define $\mathcal{L}_{\mathrm{half}}$ to be the set of all configurations in which there exists at least one leader and the timer value of every agent is greater than or equal to $s/2$. Note that $\mathcal{S}_{\mathrm{half}} = \mathcal{L}_{\mathrm{half}} \cap \mathcal{L}_{\mathrm{one}}$. The following corollary is directly obtained from Lemmas 1, 4, and 5.

Corollary 1 *Let C_0 be a configuration in $\mathcal{L}_{\mathrm{one}}$ and let $\Xi_{P_{LE}}(C_0, \Gamma) = C_0, C_1, \ldots, C_{\frac{ns}{48}}, \ldots$. Then, $\Pr(C_{\frac{ns}{48}} \in \mathcal{L}_{\mathrm{half}}) \geq 1 - n \cdot \exp(-s/36) - n \cdot \exp(-s/96)$ holds.*

We define $RJ_\gamma(v,t)$ for any $v \in V$ and any $t \geq 0$ as follows: if v joins in $\gamma(t)$, $RJ_\gamma(v,t) = 1$, otherwise $RJ_\gamma(v,t) = 0$. *The number of real interactions* of v is defined by $RI_\gamma(v,t) = \sum_{i=0}^{t-1} RJ_\gamma(v,t)$. Intuitively, $RI_\gamma(v,t)$ is the number of interactions in which v joins between time 0 and time $t-1$.

Lemma 6 $\Pr(\max_{v \in V} RI_\Gamma(v, \frac{ns}{48}) \leq \frac{s}{2}) \geq 1 - n \cdot \exp(-s/4)$ *holds.*

Proof. For any integer $t \geq 0$ and any agent $v \in V$, the probability that v joins in $\Gamma(t)$ is $\frac{2}{n}$. Hence, $RI_\Gamma\left(v, \frac{ns}{48}\right) \sim B(\frac{ns}{48}, \frac{2}{n})$. Applying Chernoff bounds, $\Pr(Y \geq R) \leq 2^{-R}$ holds for any binomial random variable Y and any real number $R \geq 6 \cdot \mathbf{E}[Y]$. (See Expr.4.3 in 10)). Since $\frac{s}{2} \geq 6\mathbf{E}[RI_\Gamma(v, \frac{ns}{48})]$ and $\ln 2 \geq \frac{1}{2}$ hold,

$$\begin{aligned}\Pr\left(\max_{v \in V} RI_\Gamma\left(v, \frac{ns}{48}\right) \geq \frac{s}{2}\right) &\leq \sum_{v \in V} \Pr\left(RI_\Gamma\left(v, \frac{ns}{48}\right) \geq \frac{s}{2}\right) \\ &\leq n \cdot 2^{-s/2} \leq n \cdot \exp\left(-\frac{s \ln 2}{2}\right) \leq n \cdot \exp\left(-\frac{s}{4}\right).\end{aligned}$$

□

Corollary 2 $\Pr((\Xi_{P_{LE}}(C_0,\Gamma))_{\mathrm{pre}}(\frac{ns}{48}) \in LE) \geq 1 - n \cdot \exp(-s/4)$ *holds for any configuration C_0 in $\mathcal{S}_{\mathrm{half}}$.*

Proof. Recall that an execution of P_{LE} staring from a configuration in $\mathcal{L}_{\mathrm{one}}$ keeps its unique leader until next timeout happens (Section 3.1). Since $C_0 \in \mathcal{S}_{\mathrm{half}}$, timeout happens by time $\frac{ns}{48} - 1$ only when some agent joins in at least $\frac{s}{2} + 1$ interactions between time 0 and time $\frac{ns}{48} - 1$. Therefore, the corollary follows from Lemma 6. □

Theorem 1 *Any configuration in $\mathcal{S}_{\mathrm{half}}$ is $\Omega(se^{s/96})$-loosely-safe configuration for LE and P_{LE},* i.e. Expr.(2) *holds.*

Proof. Under the assumption $s \geq 96(2\ln n + \ln 24)$ and $n \geq 2$, we have $\exp(-\frac{s}{4}) + \exp(-\frac{s}{36}) \leq \exp(-\frac{s}{96})$. Hence, $\exp(-\frac{s}{4}) + \exp(-\frac{s}{36}) + \exp(-\frac{s}{96}) \leq 2\exp(-\frac{s}{96})$ follows. Therefore, Expr.(3) holds for any configuration $C_0 \in \mathcal{S}_{\mathrm{half}}$ from Corollaries 1 and 2. Hence, we obtain Expr.(2) by Lemma 3. □

Next, we show Expr.(1) to complete our proof. We denote by $\mathcal{L}$ the set of all configurations in which there exists at least one leader. The following inequality clearly holds:

$$\begin{aligned}&\max_{C \in \mathcal{C}_{\mathrm{all}}} ECT_{P_{LE}}(C, \mathcal{S}_{\mathrm{half}}) \\ &\leq \max_{C \in \mathcal{C}_{\mathrm{all}}} ECT_{P_{LE}}(C, \mathcal{L}) + \max_{C \in \mathcal{L}} ECT_{P_{LE}}(C, \mathcal{L}_{\mathrm{half}}) + \max_{C \in \mathcal{L}_{\mathrm{half}}} ECT_{P_{LE}}(C, \mathcal{S}_{\mathrm{half}}).\end{aligned}$$

Therefore, for obtaining Expr.(1), it suffices to show that each term in the right side of the above inequality belongs to $O(ns \log n)$. This is proven by the following three lemmas. (We omit the proofs of Lemma 8 and 9 due to the lack of space.)

Lemma 7 $\max_{C \in \mathcal{C}_{\mathrm{all}}} ECT_{P_{LE}}(C, \mathcal{L})$ *belongs to $O(ns \log n)$.*

Proof. We define $\nu(C,i)$ $(0 \leq i \leq s)$ as the number of agents with timer value i in configuration C, *i.e.* $\nu(C,i) = |\{v \in V \mid C(v).time = i\}|$. For any integer i, j $(0 \leq i \leq s,\ 1 \leq j \leq n)$ we denote by $\mathcal{W}_{i,j}$ the set of all configurations in which there exists no leader, the maximum timer value of all agents is i, and

$\nu(C,i) = j$ holds.[⋆1] For any set of configurations $\mathcal{X} \in \mathcal{C}_{\rm all}$, we denote the complement set $\mathcal{C}_{\rm all} \setminus \mathcal{X}$ by $\overline{\mathcal{X}}$. Note that $\overline{\mathcal{L}} = \bigcup_{i=0}^{s} \bigcup_{j=1}^{n} \mathcal{W}_{i,j}$.

Let $w_{i,j}$ be $\max_{C \in \mathcal{W}_{i,j}} ECT_{P_{LE}}(C, \overline{\mathcal{W}_{i,j}})$. By the definition of P_{LE}, no interaction increments the maximum timer value of all agents as long as there exists no leader in the population. Therefore, once an execution of P_{LE} reaches a configurations in $\overline{\mathcal{W}_{i,j}}$ from a configuration in $\mathcal{W}_{i,j}$, the execution cannot reach any configuration in $\mathcal{W}_{i,j}$ thereafter. Hence, the inequality $\max_{C \in \mathcal{C}_{\rm all}} ECT_{P_{LE}}(C, \mathcal{L}) \leq w_{0,n} + \sum_{i=1}^{s} \sum_{j=1}^{n} w_{i,j}$ holds. With simple calculation, we can obtain $w_{i,j} \leq n^2/(j(2n-j))$ when $1 \leq i \leq s,\ 1 \leq j \leq n$. Therefore, we have

$$w_{i,j} \leq \frac{n^2}{j(2n-j)} = 1 + \frac{(n-j)^2}{j(2n-j)} \leq 1 + \frac{n-j}{j} = \frac{n}{j}.$$

Clearly, $w_{0,n}$ is 1 with the probability 1. Hence, we obtain

$$\max_{C \in C_{\rm all}} ECT_{P_{LE}}(C, \mathcal{L}) \leq w_{0,n} + \sum_{i=1}^{s} \sum_{j=1}^{n} w_{i,j} \leq 1 + ns \cdot H(n) \in O(ns \log n),$$

where H is the harmonic function. □

Lemma 8 $\max_{C \in \mathcal{L}} ECT_{P_{LE}}(C, \mathcal{L}_{\rm half})$ *belongs to* $O(ns)$.

Lemma 9 $\max_{C \in \mathcal{L}_{\rm half}} ECT_{P_{LE}}(C, \mathcal{S}_{\rm half})$ *belongs to* $O(ns)$.

Thus, we have Expr.(1). The following theorem is directly derived from Thorem 1 and Expr.(1).

Theorem 2 *P_{LE} is $(O(ns \log n), \Omega(se^{s/96}))$-probabilistic loosely-stabilizing for behavior LE and $\mathcal{S}_{half}$ if $s \geq \max(3n, 96(2 \ln n + \ln 24))$ holds.*

Recall that P_{LE} knows an upper bound N of n. When we set $s = \max(96N, 96(2 \ln N + \ln 24))$, P_{LE} realizes $(O(nN \log n), \Omega(Ne^N))$-probabilistic loose-stabilization for behavior LE and $\mathcal{S}_{\rm half}$.

4. Conclusion

In this paper, we introduced a novel concept of loose-stabilization and presented a probabilistic loosely-stabilizing leader election protocol in the PPP model of complete networks. Starting from an arbitrary configuration, the proposed protocol reaches a loosely-safe configuration within $O(nN \log n)$ expected steps, and then, it keeps a unique leader for $\Omega(Ne^N)$ expected steps, where n is the actual network size and N is a known upper bound of n. This protocol has practical significance from the following reason: the protocol can be practically considered to attain self-stabilization because of exponentially long time of keeping a unique leader while the self-stabilizing leader election in the PPP model of complete networks is impossible without knowledge of the exact network size[4].

Acknowledgments This work is supported in part by Global COE Program of MEXT, Grant-in-Aid for Scientific Research ((B)17300020, (B)19300017, (B)20300012) of JSPS, Grant-in-Aid for Young Scientists ((B)18700059) of JSPS, and the Kayamori Foundation of Informational Science Advancement.

References

1) Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M. and Peralta, R.: Computation in networks of passively mobile finite-state sensors, *Distributed Computing*, Vol.18, No.4, pp.235–253 (2006).
2) Angluin, D., Aspnes, J. and Eisenstat, D.: Fast Computation by Population Protocols with a Leader, *Proceedings of Distributed Computing, 20th International Symposium*, pp.61–75 (2006).
3) Angluin, D., Aspnes, J., Fischer, M. and Jiang, H.: Self-stabilizing Population Protocols, *Proceedings of Principles of Distributed Systems*, pp.103–117 (2006).
4) Cai, S., Izumi, T. and Wada, K.: Space Complexity of Self-Stabilizing Leader Election in Passively-Mobile Anonymous Agents, To be submitted.
5) Devismes, S., Tixeuil, S. and Yamashita, M.: Weak vs. Self vs. Probabilistic Stabilization, *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS 2008)*, pp.681–688 (2008).
6) Dijkstra, E.: Self-stabilizing systems in spite of distributed control, *Communications of the ACM*, Vol.17, No.11, pp.643–644 (1974).
7) Gouda, M.: The Theory of Weak Stabilization, *Proceedings of the 5th International Workshop on Self-Stabilizing Systems*, Springer, pp.114–123 (2001).
8) Israeli, A. and Jalfon, M.: Token management schemes and random walks yield self-stabilizing mutual exclusion, *Proceedings of the ninth annual ACM symposium on Principles of distributed computing*, ACM New York, NY, USA, pp. 119–131 (1990).
9) Lin, J., Huang, T., Yang, C. and Mou, N.: Quasi-self-stabilization of a distributed system assuming read/write atomicity, *Computers and Mathematics with Applications*, Vol.57, No.2, pp.184–194 (2009).
10) Mitzenmacher, M. and Upfal, E.: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press (2005).

⋆1 Note that $\mathcal{W}_{0,j} = \emptyset$ for any integer j $(1 \leq j < n)$