Self-Stabilizing Clock Synchronization in Probabilistic Networks

Bernadette Charron-Bost, Louis Penet de Monterno

Self-Stabilizing Clock Synchronization in Probabilistic Networks

Bernadette Charron-Bost
DI ENS, École Normale Supérieure, 75005 Paris, France

Louis Penet de Monterno
École polytechnique, IP Paris, 91128 Palaiseau, France

Abstract

We consider the fundamental problem of clock synchronization in a synchronous multi-agent system. Each agent holds a clock with an arbitrary initial value, and clocks must eventually indicate the same value, modulo some integer P. A known solution for this problem in dynamic networks is the self-stabilization *SAP* (for self-adaptive period) algorithm, which uses finite memory and relies solely on the assumption of a finite *dynamic diameter* in the communication network.

This paper extends the results on this algorithm to probabilistic communication networks: We introduce the concept of *strong connectivity with high probability* and we demonstrate that in any probabilistic communication network satisfying this hypothesis, the *SAP* algorithm synchronizes clocks with high probability. The proof of such a *probabilistic hyperproperty* is based on novel tools and relies on weak assumptions about the probabilistic communication network, making it applicable to a wide range of networks, including the classical PUSH model. We provide an upper bound on time and space complexity.

Building upon previous works by Feige et al. and Pittel, the paper provides solvability results and evaluates the stabilization time and space complexity of *SAP* in two specific cases of communication topologies.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Dynamic graph algorithms

Keywords and phrases Self-stabilization, Clock synchronization, Probabilistic networks

Digital Object Identifier 10.4230/LIPIcs..2023.

Acknowledgements We would like to thank Patrick Lambein-Monette, Stephan Merz, and Guillaume Prémel for very useful discussions.

1 Introduction

There is a considerable interest in distributed systems consisting of multiple, potentially mobile, agents. This is mainly motivated by the emergence of large scale networks, characterized by the lack of centralized control, the access to limited information and a time-varying connectivity. Control and optimization algorithms deployed in such networks should be completely distributed, relying only on local observations and information, and robust against unexpected changes in topology such as link or node failures.

A canonical problem in distributed control is *clock synchronization*: In a system where agents are equipped with local discrete clocks with common pulses, the objective is that all clocks eventually synchronize despite arbitrary initializations. That corresponds to synchronization in *phase*, as opposed to the problem of synchronization in *frequency* (e.g. for instance in [34, 21, 23, 32]).

Clock synchronization is a fundamental problem arising in a number of applications, both in engineering and natural systems. A synchronized clock is a fundamental basic block used in many engineering systems, e.g. in the universal self-stabilizing algorithm developed by Boldi and Vigna [9], or for deploying distributed algorithms structured into

synchronized *phases* (e.g., the *Two-Phase* and *Three-Phase Commit* algorithms [6], or many consensus algorithms [5, 20, 31, 14]). Clock synchronization also corresponds to a ubiquitous phenomenon in the natural world and finds numerous applications in physics and biology, e.g., the Kuramoto model for the synchronization of coupled oscillators [35], synchronous flashing fireflies, collective synchronization of pancreatic beta cells [29].

In the model we consider, we assume agents take steps in synchronous rounds, but do not have a consistent numbering of the rounds (e.g., agents regularly receive a common pulse). The communication pattern at each round is captured by a directed graph that may change continually from one round to the next, and the history of communications in the network is modeled as a whole by a *dynamic graph*, that is an infinite sequence of directed graphs with the same set of nodes. For a given set of agents (nodes), a communication model is thus naturally represented by a *subset* of dynamic graphs, and corresponds to a *probability distribution* on the set of dynamic graphs (or, equivalently, to a probability distribution on the communication graph at each round) in the probabilistic setting.

In deterministic communication models, Charron-Bost and Monterno [12] proposed a synchronization algorithm of periodic clocks, termed *SAP* (for self-adaptive period), which is finite state and self-stabilizing, i.e., the complete initial state of each agent (not just its clock value) can be arbitrary. This algorithm does not assume any global knowledge on the network, and tolerates time-varying topologies. It is proved to solve the mod P-synchronization problem in any dynamic network with a finite *dynamic diameter*, that is, from any time onward and for every pair of agents i and j, there is a temporal path of bounded length connecting i to j. In this paper, we focus instead on the probabilistic setting, and explore the behavior of the *SAP* algorithm in a general probabilistic network.

In a deterministic communication model, each execution of *SAP* is considered individually, and the *property* of mod-P synchronization must be satisfied by each possible execution in this model. By contrast, in the probabilistic framework, the set of executions of *SAP* is considered as a whole, and the correctness of *SAP* then corresponds to the following *hyperproperty* [16]: the probability of executions of *SAP* in which mod P-synchronization is achieved is greater than a chosen real $p \in [0, 1)$.

Unfortunately, verifying probabilistic hyperproperties is technical, and requires to develop novel proof strategies and new analysis tools. In particular, the notion of dynamic diameter, which is central in the correctness proof of *SAP* in [12], is no more relevant in a probabilistic framework. Indeed, we may see that in most of probabilistic networks, the dynamic diameter is almost surely infinite (cf. Section 2.4). Instead, we provide a suitable notion of *probabilistic diameter*, and define a probabilistic network to be *strongly connected with high probability*.

Importantly, *SAP* uses a finite but unbounded amount of memory, that is, in each execution, the memory usage of each node eventually stops growing. Regarding memory size issue, the *SAP* algorithm appears to be optimal since, in an unpublished companion paper [13], we prove that there is no self-stabilizing and bounded-memory algorithm solving the mod P-synchronization problem in a deterministic communication model if no bound on the dynamic diameter is known.

1.1 Related work

Self-stabilizing clocks have been extensively studied in different communication models and under different assumptions. In particular, clocks may be unbounded, in which case they are required to be eventually equal, instead of only congruent. The synchronization problem of unbounded clocks admits simple solutions in strongly connected networks, namely the *Min* and *Max* algorithms [22, 27].

Periodic clocks require more sophisticated synchronization mechanisms. In addition to strong connectivity and static networks, the pioneer papers on periodic clock synchronization [3, 28, 10, 1] all assume that a bound on the diameter is available. Then Boldi and Vigna [9] proposed a synchronization algorithm, based on a self-adaptive period mechanism, that dispenses with the latter assumption.

More recently, periodic clock synchronization has been studied in the *Beeping model* [17] in which agents have severely limited communication capabilities: given a static and connected bidirectional communication graph, in each round, each agent can either send a "beep" to all its neighbors or stay silent. A self-stabilizing algorithm has been proposed by Feldmann et al. [25], which is optimal both in time and space, but which requires to know a bound on the network size. As explained above, without this global information available at each node, such an algorithm does not exist: The best we can do with a self-stabilizing synchronization algorithm in a network of unknown dynamic diameter[1] is to use finite memory as achieved in *SAP*.

There are also numerous results for mod P-synchronization with faulty agents. The fault-tolerant solutions that have been proposed in various failure models, including the Byzantine failure model, using algorithmic schemes initially developed for consensus (e.g., see [18, 19]). They typically require a bidirectional connected (most of the time fully-connected) network.

All these works assume a *static* network. In [12], Charron-Bost et al. tackled the dynamic setting: they extended the method of self-adaptive periods developed in [9] to dynamic networks, and proposed the *SAP* algorithm for this type of networks with "dynamic disconnectivity".

For probabilistic communication models, the problem of clock synchronization has been addressed by Boczkowski et al. [8] and later on by Bastide et al. [4], both in the particular framework of the PULL model [30] over the fully-connected graph: In each round each agent receives a message from an agent sampled uniformly at random. Their focus is on minimizing message size and they both obtain a stabilization time of $O(\log n)$ in a network of size n. Unfortunately, the algorithms in these papers are specific to the PULL model, and their good performances highly rely on the assumption of a fully-connected network. Observe that, in the case where P is not a power of 2, those algorithms are not bounded-memory.

Clock synchronization has been studied in another probabilistic communication model, namely, the model of *population protocols*, consisting of a set of agents, interacting in randomly chosen pairs. This is basically an asynchronous model, where the synchronization task is quite different from the one studied in this paper since it amounts to implementing the abstraction of rounds [2]. In other words, the point in the population protocol model is to achieve synchronization in *frequency* instead of synchronization in *phase*.

1.2 Contribution

Our main contribution in this paper is the probabilistic correctness proof of the *SAP* algorithm: we show that it synchronizes periodic clocks in the large class of networks that are strongly connected with high probability. The probabilistic communication model is totally general in the sense that we only assume it to be *memoryless*, meaning that the communication graph at each round does not depend on the previous rounds. As a byproduct, our correctness proof provides upper bounds on the stabilization time and the space complexity of *SAP*.

[1] or unknown size in the static case.

Verifying probabilistic hyperproperties requires to develop novel proof strategies and analysis tools. In particular, we devise new parameters for probabilistic dynamic graphs, namely, a hierarchy of probabilistic diameters, and prove several basic properties on these diameters that are interesting on their own.

Finally, we apply our results for general probabilistic communication models to the classical PUSH model: Leveraging the fundamental properties of this rumor spreading model established by Feige et al. [24] and Pittel [33], we prove the probabilistic correctness of *SAP* in the PUSH model and provide an estimate of its stabilization time and its space complexity, first for a general bidirectional network, and then for the case of a fully-connected network.

2 Preliminaries

2.1 The computing model

We consider a networked system of n agents (nodes), denoted $1, 2, \ldots, n$. Computation proceeds in *synchronized rounds*, which are communication closed in the sense that no message received in round t is sent in a round different from t. In round $t\,(t = 1, 2, \cdots)$, each node successively (a) broadcasts a message at the beginning of round t, (b) receives some messages, and (c) undergoes an internal transition to a new state. Communications that occur at round t are modeled by a directed graph (digraph) $\mathbb{G}(t) = ([n], E_t)$ where $[n] = \{1, \ldots, n\}$. We assume a self-loop at each node in all these digraphs since a node can communicate with itself instantaneously. An infinite sequence of digraphs $\mathbb{G} = (\mathbb{G}(t))_{t \geqslant 1}$ is called a *dynamic graph* and the set of dynamic graphs of size n is denoted $\mathcal{G}_n$.

An *algorithm* $\mathcal{A}$ is given by a set $\mathcal{Q}$ of local states, a set of messages $\mathcal{M}$, a sending function $\sigma : \mathcal{Q} \to \mathcal{M}$, and a transition function $\tau : \mathcal{Q} \times \mathcal{M}^{\oplus} \to \mathcal{Q}$, where $\mathcal{M}^{\oplus}$ is the set of finite multisets over $\mathcal{M}$. Every state in $\mathcal{Q}$ is a possible initial state (self-stabilization model).

An *execution of* $\mathcal{A}$ with the dynamic graph $\mathbb{G}$ then proceeds as follows: In round $(t = 1, 2 \ldots)$, each node applies the sending function σ to its current state to generate the message to be broadcasted, then it receives the messages sent by its incoming neighbors in the digraph $\mathbb{G}(t)$, and finally applies the transition function τ to its current state and the list of messages it has just received to go to a next state. It is entirely determined by the collection of initial states and the dynamic graph $\mathbb{G}$.

Given an execution of $\mathcal{A}$, the value of any local variable x_i at the end of round t is denoted by $x_i(t)$, and $x_i(0)$ is the initial value of x_i.

2.2 Dynamic graphs and probability measure

Let us first recall that the *product* of two digraphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, denoted $G_1 \circ G_2$, is the digraph with the set of nodes V and with an edge (i, j) if there exists $k \in V$ such that $(i, k) \in E_1$ and $(k, j) \in E_2$. For any dynamic graph $\mathbb{G}$ and any integers $t' \geqslant t \geqslant 1$, we let

$$\mathbb{G}(t : t') \stackrel{\text{def}}{=} \mathbb{G}(t) \circ \cdots \circ \mathbb{G}(t').$$

By convention, $\mathbb{G}(t : t) = \mathbb{G}(t)$, and when $0 < t' < t$, $\mathbb{G}(t : t')$ is the digraph with only a self-loop at each node. The set of i's in-neighbors in $\mathbb{G}(t : t')$ is denoted by $\mathrm{In}_i(t : t')$, and simply by $\mathrm{In}_i(t)$ when $t' = t$. Every edge (i, j) in $\mathbb{G}(t : t')$ corresponds to a *path in the round interval* $[t, t']$: there exist $t' - t + 2$ nodes $i = k_0, k_1, \ldots, k_{t'-t+1} = j$ such that (k_r, k_{r+1}) is an edge of $\mathbb{G}(t + r)$ for each $r = 0, \ldots, t' - t$.

For a fixed $n \in \mathbb{N}^+$, and a triple $i \in [n]$, $t \in \mathbb{N}$, $\delta \in \mathbb{N}^+$, we let

$$\Gamma_i^{t,\delta} \stackrel{\text{def}}{=} \{\mathbb{G} \in \mathcal{G}_n \mid \forall j \in [n], (i,j) \text{ is an edge of } \mathbb{G}(t+1:t+\delta)\}.$$

If Σ_n denotes the Borel σ-algebra on $\mathcal{G}_n$, then $(\mathcal{G}_n, \Sigma_n)$ is a measurable space. Then, we consider a probability measure on $(\mathcal{G}_n, \Sigma_n)$, denoted $\Pr_n$, or simply Pr. The pair $(\mathcal{G}_n, \Pr_n)$ is called a *probabilistic communication network* of size n.

By analogy with the terminology used in game theory (e.g., see [7]), we say that Pr is *memoryless* if the random variables $\mathbb{G}(1), \mathbb{G}(2), \mathbb{G}(3), \ldots$ are mutually independent.

Each execution of an algorithm $\mathcal{A}$ in a system with n nodes is characterized by an initial global state in $\mathcal{Q}^n$ and a communication graph in $\mathcal{G}_n$. Hence, the set of executions of $\mathcal{A}$ starting at $q \in \mathcal{Q}^n$, denoted $\mathcal{E}_q(\mathcal{A})$, is isomorphic to $\mathcal{G}_n$. Thus Pr induces a probabilistic measure on $\mathcal{E}_q(\mathcal{A})$, which will also be denoted by Pr as no confusion can arise.

2.3 The mod *P*-synchronization problem

Let P and n be two positive integers, and let $\mathcal{A}$ be an algorithm where each node i maintains an *integer* variable C_i, called the clock of i. An execution of $\mathcal{A}$, over a network of n agents, is said *to achieve mod P-synchronization in τ rounds* if

$$\exists c \in \mathbb{N}, \forall t \geqslant \tau, \forall i \in [n],\ C_i(t) \equiv_P t + c,$$

in which case the network is said to be *synchronized* (for *mod P-synchronized*) *from round τ*, i.e.,

$$\forall t \geqslant \tau, \forall i, j \in [n],\quad C_i(t) \equiv_P C_j(t).$$

Then we say that the algorithm $\mathcal{A}$ with n agents *solves the mod P-synchronization problem in τ rounds with probability p* if for every initial state $q \in \mathcal{Q}^n$ of $\mathcal{A}$, the measure of the set of executions of $\mathcal{A}$ achieving mod P-synchronization in τ rounds is at least equal to p.

2.4 Probabilistic diameters

Let us fix a global state $q \in \mathcal{Q}^n$. For any real $p \in [0,1]$ and any integer $k \in [n]$, we define the *probabilistic order k diameter* as the minimum number of rounds required for k arbitrary nodes to communicate with all the nodes in the network with probability at least p. Formally, we let

$$\hat{D}^{(k)}(p) \stackrel{\text{def}}{=} \inf\{\delta \in \mathbb{N}^+ \mid \inf_{i_1,\cdots,i_k \in [n], t \in \mathbb{N}} \Pr(\Gamma_{i_1}^{t,\delta} \cap \cdots \cap \Gamma_{i_k}^{t,\delta}) \geqslant p\}.$$

Clearly, we have that $\hat{D}^{(1)}(p) \leqslant \cdots \leqslant \hat{D}^{(n)}(p)$, with equalities when $p = 1$. As a matter of fact, the probabilistic proof of the SAP_g algorithm that we develop in the following section only involves the probabilistic diameters $\hat{D}^{(1)}(p)$ and $\hat{D}^{(2)}(p)$.

As an example, let us consider the memoryless probability measure Pr on $\mathcal{G}_2$ defined by

$$\Pr(\mathbb{G}(t) = G_1) = \Pr(\mathbb{G}(t) = G_2) = \frac{1}{2},$$

where G_1 and G_2 are the two-node digraphs defined in Figure 1. For any round t and any positive integer δ, we have:

$$\Pr(\Gamma_1^{t,\delta}) = 1 - \Pr\left(\bigcap_{d=1}^{\delta} (\mathbb{G}(t+d) = G_2)\right) = 1 - \prod_{\ell=1}^{\delta} \Pr(\mathbb{G}(t+d) = G_2) = 1 - 2^{-\delta}.$$

(a) digraph G_1 **(b)** digraph G_2

Figure 1 Two digraphs.

Similarly, $\Pr(\Gamma_2^{t,\delta}) = 1 - 2^{-\delta}$. Moreover,

$$\begin{aligned}\Pr(\Gamma_1^{t,\delta} \cap \Gamma_2^{t,\delta}) &= 1 - \Pr(\overline{\Gamma_1^{t,\delta}} \cup \overline{\Gamma_2^{t,\delta}}) \\ &= 1 - \Pr\left(\bigcap_{d=1}^{\delta} (\mathbb{G}(t+d) = G_2) \cup \bigcap_{d=1}^{\delta} (\mathbb{G}(t+d) = G_1)\right) \\ &= 1 - 2^{-\delta+1}.\end{aligned}$$

Using the definition of the probabilistic diameters and the two equations above, we obtain the values of $\hat{D}^{(1)}(p)$ and $\hat{D}^{(2)}(p)$ in our example:

$$\hat{D}^{(1)}(p) = \inf\{\delta \in \mathbb{N}^+ \mid 1 - 2^{-\delta} \geqslant p\} = \lceil -\log_2(1-p) \rceil \text{ and}$$
$$\hat{D}^{(2)}(p) = \inf\{\delta \in \mathbb{N}^+ \mid 1 - 2^{-\delta+1} \geqslant p\} = 1 + \lceil -\log_2(1-p) \rceil.$$

This simple example shows why it is not appropriate to use the parameter $D(p)$ simply defined by:

$$D(p) \stackrel{\text{def}}{=} \inf\{\delta \in \mathbb{N}^+ \mid \Pr(D_{\mathbb{G}} \leqslant \delta) \geqslant p\},$$

where $D_{\mathbb{G}} = \inf\{\delta \in \mathbb{N}^+ \mid \forall i \in [n], \forall t \in \mathbb{N}, \mathbb{G} \in \Gamma_i^{t,\delta}\}$ is the *dynamic diameter* of the dynamic graph $\mathbb{G}$ [11]. Indeed, for each node $i \in \{1, 2\}$, we have:

$$\Pr(D_{\mathbb{G}} \leqslant \delta) \leqslant \Pr\left(\bigcap_{\ell=0}^{\infty} \Gamma_i^{\ell\delta,\delta}\right) = \prod_{\ell=0}^{\infty} \Pr(\Gamma_i^{\ell\delta,\delta}) = 0,$$

and thus $D(p)$ is infinite if p is positive. In other words, the dynamic diameter of almost all dynamic graphs is infinite in this example, while $\hat{D}^{(1)}(p)$ is finite when $p < 1$.

We now state some general properties on the probabilistic diameters $\hat{D}^{(1)}(p)$ and $\hat{D}^{(2)}(p)$.

▶ **Lemma 1.** *For any memoryless probability measure and all real numbers $p \in \left[\frac{1}{2}, 1\right]$, if $\hat{D}^{(1)}(p)$ is finite, then $\hat{D}^{(2)}(p)$ is finite and $\hat{D}^{(2)}(p) \leqslant 2\hat{D}^{(1)}(p)$.*

Proof. Because of the self-loops, the digraphs $\mathbb{G}(t+1 : t+\delta)$ and $\mathbb{G}(t+\delta+1 : t+2\delta)$ are both subgraphs of $\mathbb{G}(t+1 : t+2\delta)$, and hence $\Gamma_i^{t,\delta} \cup \Gamma_i^{t+\delta,\delta} \subseteq \Gamma_i^{t,2\delta}$. It follows that:

$$\begin{aligned}\Pr\left(\Gamma_i^{t,2\hat{D}^{(1)}(p)} \cap \Gamma_j^{t,2\hat{D}^{(1)}(p)}\right) &\geqslant 1 - \Pr\left(\overline{\Gamma_i^{t,2\hat{D}^{(1)}(p)}}\right) - \Pr\left(\overline{\Gamma_j^{t,2\hat{D}^{(1)}(p)}}\right) \\ &\geqslant 1 - \Pr\left(\overline{\Gamma_i^{t,\hat{D}^{(1)}(p)}} \cap \overline{\Gamma_i^{t+\hat{D}^{(1)}(p),\hat{D}^{(1)}(p)}}\right) \\ &\quad - \Pr\left(\overline{\Gamma_j^{t,\hat{D}^{(1)}(p)}} \cap \overline{\Gamma_j^{t+\hat{D}^{(1)}(p),\hat{D}^{(1)}(p)}}\right) \\ &\geqslant 1 - 2(1-p)^2.\end{aligned}$$

The second inequality holds because of the above-proved inclusion, and the third one because of the memoryless assumption. If $p \in \left[\frac{1}{2}, 1\right]$, then $1-2(1-p)^2 \geqslant p$ and $\hat{D}^{(2)}(p) \leqslant 2\hat{D}^{(1)}(p)$. ◀

Using a similar proof, it is possible to show that, for all $p \in [\frac{1}{2}, 1]$, for all ℓ_1 and $\ell_2 \leqslant \ell_1$ such that $\ell_1 + \ell_2 \leqslant n$, if $\hat{D}^{(\ell_1)}(p)$ and $\hat{D}^{(\ell_2)}(p)$ are finite, then $\hat{D}^{(\ell_1+\ell_2)}(p)$ is finite and $\hat{D}^{(\ell_1+\ell_2)}(p) \leqslant 2\hat{D}^{(\ell_1)}(p)$. Therefore, by induction on $\ell \leqslant n$, if $\hat{D}^{(1)}(p)$ is finite, then all $\hat{D}^{(\ell)}(p)$ are finite and $\hat{D}^{(\ell)}(p) \leqslant 2^{\lceil \log_2 \ell \rceil}\hat{D}^{(1)}(p)$. Finally, we prove the following finiteness result for $\hat{D}^{(1)}(p)$.

▶ **Lemma 2.** *For any memoryless probability measure, if $\hat{D}^{(1)}(p_0)$ is finite for some $p_0 \in (0, 1]$, then $\hat{D}^{(1)}(p)$ is finite for all real numbers $p \in [0, 1)$.*

Proof. For every node i, for every integers $t \geqslant 0$ and $\ell > 0$, we have

$$\begin{aligned}\Pr\left(\Gamma_i^{t,\ell\hat{D}^{(1)}(p_0)}\right) &\geqslant \Pr\left(\bigcup_{h=0}^{\ell-1} \Gamma_i^{t+h\hat{D}^{(1)}(p_0),\hat{D}^{(1)}(p_0)}\right)\\ &= 1 - \prod_{h=0}^{\ell-1} \Pr\left(\overline{\Gamma_i^{t+h\hat{D}^{(1)}(p_0),\hat{D}^{(1)}(p_0)}}\right)\\ &\geqslant 1 - (1-p_0)^{\ell}.\end{aligned}$$

The first inequality holds because of the self-loops, as explained in the proof of Lemma 1. The second one is due to the memoryless assumption on the Pr probability measure.

If p_0 is positive, then $\lim_{\ell\to\infty} 1 - (1-p_0)^{\ell} = 1$. Thus, for any real number p less than one, there exists some integer ℓ_0 such that $\Pr(\Gamma_i^{t,\ell_0\hat{D}^{(1)}(p_0)}) \geqslant p$, which implies that $\hat{D}^{(1)}(p)$ is finite and $\hat{D}^{(1)}(p) \leqslant \ell_0\hat{D}^{(1)}(p)$. ◀

Since all $\hat{D}^{(\ell)}$ are non-decreasing, Lemmas 1 and 2 imply that if $\hat{D}^{(1)}(p_0)$ is finite for some $p_0 \in (0, 1]$, then all probabilistic diameters are finite for all $p \in [0, 1)$, in which case the network $(\mathcal{G}_n, \Pr_n)$ is said to be *strongly connected with high probability*. This notion is linked to the notion of dynamic diameter. Assume that a dynamic graph $\mathbb{G}$ has a finite dynamic diameter $D_{\mathbb{G}}$. Let Pr be the probability measure such that $\Pr(\{\mathbb{G}\}) = 1$, then for all real numbers $p \in (0, 1]$,

$$D_{\mathbb{G}} = \hat{D}^{(1)}(p) = \hat{D}^{(2)}(p) = \cdots = \hat{D}^{(n)}(p).$$

3 The *SAP* algorithm

We present the self-stabilizing *SAP* algorithm designed in [12] for the mod P-synchronization problem in dynamic networks with a finite dynamic diameter, and recall its basic properties.

3.1 Description of the algorithm

A typical approach to solve the mod P-synchronization problem consists in the following algorithm: at each round, each node sends its own variable $C_i \in \{0, \ldots, P-1\}$ and applies the following update rule:

$$C_i \leftarrow \left[\min_{j \in \mathrm{In}_i} C_j + 1\right]_P,$$

where In_i denotes the current set of i's incoming neighbors, and $[c]_P$ is the remainder of the Euclidean division of c by P. Unfortunately, this naive algorithm does not work[2] when

[2] see Theorem 4.13 in [1].

$\hat{D}^{(1)}(p)$ is too large compared to the period P. To overcome this problem, the SAP algorithm uses self-adaptive periods and the basic fact that for any positive integer M, we have

$$[\,[\,c\,]_{PM}\,]_P = [\,c\,]_P.$$

More precisely, each node i uses two *integer* variables M_i and C_i, and computes the clock value C_i not modulo P, but rather modulo the time-varying period PM_i. The variable M_i is used as a guess to find a large enough multiple of P so to make the clocks eventually stabilized. Until synchronization, the variables M_i increase so that the shortest period PM_i eventually becomes large enough compared to the largest clock value in the network. In the rest of this paper, $\mathcal{S}_t$ denotes the set of executions in which the system is synchronized in round t. Once all clocks are congruent modulo P, they remain congruent forever, meaning that $\mathcal{S}_t \subseteq \mathcal{S}_{t+1}$. The update rule for M_i is parametrized by a function $g : \mathbb{N} \to \mathbb{N}$. The corresponding algorithm is denoted SAP_g, and its code is given below. Line 5 in the pseudo-code implies that $C_i(t) < PM_i(t)$, and for the sake of simplicity, we assume that this inequality also holds initially, that is, $C_i(0) < PM_i(0)$.

Algorithm 1 Pseudo-code of node i in the SAP_g algorithm

Variables:
1: $C_i \in \mathbb{N}$;
2: $M_i \in \mathbb{N}^+$;
In each round do:
3: send $\langle C_i, M_i \rangle$ to all
4: receive $\langle C_{j_1}, M_{j_1} \rangle, \langle C_{j_2}, M_{j_2} \rangle, \ldots$ from the set In_i of incoming neighbours
5: $C_i \leftarrow \left[\min_{j \in \mathrm{In}_i} C_j + 1\right]_{PM_i}$
6: $M_i \leftarrow \max_{j \in \mathrm{In}_i} M_j$
7: **if** $C_j \not\equiv_P C_{j'}$ for some $j, j' \in \mathrm{In}_i$ **then**
8: $\quad M_i \leftarrow g(M_i)$
9: **end if**

In the rest of the paper, the function g is supposed to be a non-decreasing and *inflationary* function, i.e., $x < g(x)$ for every positive integer x. Therefore, each M_i variable is non-decreasing. If ℓ is a positive integer, g^ℓ denotes the ℓ-th iterate of g. For every positive real number x, we let

$$g^*(x) \stackrel{\text{def}}{=} \inf\{\ell \in \mathbb{N}^+ \mid g^\ell(1) \geqslant x\}.$$

Since g is inflationary, $g^*(x)$ is finite for all integers x, and $g^*(x) \leqslant x$. The algorithm is parametrized by such a function g, and the corresponding algorithm will be denoted SAP_g.

3.2 Properties of *SAP*'s executions

Let us consider an execution ϵ of the SAP_g algorithm over a network of size n, with the dynamic graph $\mathbb{G}$. We start with three basic properties of ϵ which directly come from the pseudo-code.

▶ **Lemma 3.** *If (i,j) is an arc in $\mathbb{G}(s:t)$, then $C_j(t) \leqslant C_i(s-1) + t - s + 1$.*

▶ **Lemma 4.** *If (i,j) is an arc in $\mathbb{G}(s:t)$, then one of the following statements is true:*
1. $C_j(t) \equiv_P C_i(s-1) + t - s + 1$;
2. $M_j(t) \geqslant g(M_i(s-1))$.

▶ **Lemma 5.** *Let d be a positive integer. If $C_i(t) + d \leqslant PM_i(t)$ holds for all nodes i, then all the clocks C_i are greater than 0 in the round interval $[t+1, t+d-1]$.*

For the probabilistic correctness proof of SAP_g, we will use another property of its executions, stated in the lemma below, which is a refinement of an analogous property established in the deterministic case under the condition of a finite dynamic diameter [12]. The proof is given in the Appendix.

▶ **Lemma 6.** *Let d be any positive integer, and k be a node such that $C_k(t) = \min_{j\in[n]} C_j(t)$. If the execution ϵ belongs to $\Gamma_k^{t,d}$ and all the clocks C_i are greater than 0 in the round interval $[t+1, t+d-1]$, then the network is synchronized in round $t+d$.*

4 Probabilistic correctness of *SAP*

Our approach for the correctness proof of the SAP_g algorithm relies on a fundamental probabilistic hyperproperty relating the adaptive mechanism for the periods in SAP_g to the order one probabilistic diameter of the network.

We fix some integer n, some real $p \in (0,1)$ and some initial state $q \in \mathcal{Q}^n$ of SAP_g. We consider a memoryless probability measure Pr on $(\mathcal{G}_n, \Sigma_n)$, and so on the set $\mathcal{E}_q$ of SAP_g's executions starting in q. We assume that the probabilistic network $(\mathcal{G}_n, \Pr_n)$ is strongly connected w.h.p., and we let

$$t_0 \stackrel{\text{def}}{=} \hat{D}^{(2)}(p) \left\lceil \frac{\log\left((1-p)^{-1}\right)}{p} \left(\sqrt{g^*\Big(\frac{2\hat{D}^{(1)}(p)}{P}\Big)} + \sqrt{2} \right)^2 \right\rceil \tag{1}$$

which is finite since g is inflationary. For all positive integers t, we consider the random variable $M(t) \stackrel{\text{def}}{=} \min_{i\in[n]} M_i(t)$.

▶ **Lemma 7.** *For every real number $p \in (0,1)$, we have*

$$\Pr\left(\Big(M(t_0) \geqslant \frac{2\hat{D}^{(1)}(p)}{P} \Big) \cup \mathcal{S}_{t_0} \right) \geqslant p. \tag{2}$$

Proof. For ease of notation, we let $\bar{g} = g^*\left(\frac{2\hat{D}^{(1)}(p)}{P}\right)$ and $\ell_0 = t_0/\hat{D}^{(2)}(p)$. In the first part of the proof, we construct a family of independent random variables B_t that all follow a Bernoulli distribution. In each execution in $\mathcal{E}_q$ that is not synchronized at round t, there exist two nodes $i_1(t)$ and $i_2(t)$ such that

$$C_{i_1(t)}(t) \not\equiv_P C_{i_2(t)}(t). \tag{3}$$

Then i_1 and i_2 can be viewed as two random variables that map any execution of SAP_g to a sequence of type $\mathbb{N} \to [n]$. Let B_t be the random variable equal to 1 on $\Gamma_{i_1(t)}^{t,\hat{D}^{(2)}(p)} \cap \Gamma_{i_2(t)}^{t,\hat{D}^{(2)}(p)}$ and equal to 0, otherwise. By definition of $\hat{D}^{(2)}(p)$, each B_t follows a Bernoulli distribution whose parameter $\Pr(B_t = 1)$ is greater than or equal to p. Since Pr is memoryless, the random variable

$$B \stackrel{\text{def}}{=} \sum_{\ell=0}^{\ell_0-1} B_{\ell\hat{D}^{(2)}(p)}$$

is a sum of independent Bernoulli variables.

We now show that in all executions in $\mathcal{E}_q$ that are not synchronized in round $\ell_0\hat{D}^{(2)}(p)$, it holds that

$$M(\ell_0\hat{D}^{(2)}(p)) \geqslant g^B(1). \tag{4}$$

For that, we fix such an execution and prove by induction on ℓ_0 that Eq. (4) holds in this execution.

1. Base case: $\ell_0 = 0$. Then we have $B = 0$, and so $M(\ell_0\hat{D}^{(2)}(p)) \geqslant 1 = g^B(1)$ as needed.
2. Inductive case: Assume that

   $$M(\ell_0\hat{D}^{(2)}(p)) \geqslant g^{\sum_{t=0}^{\ell_0-1} B_{t\hat{D}^{(2)}(p)}}(1)$$

   holds for for some $\ell_0 \in \mathbb{N}$ and that the system is not synchronized in round $(\ell_0+1)\hat{D}^{(2)}(p)$. Then, the nodes $i_1 = i_1(\ell_0\hat{D}^{(2)}(p))$ and $i_2 = i_2(\ell_0\hat{D}^{(2)}(p))$ satisfy Eq. (3). Therefore, for every node i, there exists some $x \in \{1,2\}$ such that

   $$C_i((\ell_0+1)\hat{D}^{(2)}(p)) \not\equiv_P C_{i_x}(\ell_0\hat{D}^{(2)}(p)) + \hat{D}^{(2)}(p). \tag{5}$$

   If $B_{\ell_0\hat{D}^{(2)}(p)} = 0$, then the inductive case immediately follows. Otherwise, $B_{\ell_0\hat{D}^{(2)}(p)} = 1$, and so the digraph $\mathbb{G}(\ell_0\hat{D}^{(2)}(p)+1 : (\ell_0+1)\hat{D}^{(2)}(p))$ contains all the arcs of the form (i_1,i) and (i_2,i). Then for every node i, it holds that

   $$M_i((\ell_0+1)\hat{D}^{(2)}(p)) \geqslant g(M_{i_x}(\ell_0\hat{D}^{(2)}(p))) \geqslant g(M(\ell_0\hat{D}^{(2)}(p))). \tag{6}$$

   The first inequality holds by Lemma 4 and Eq. (5), and the second one because g is non-decreasing. Using the induction hypothesis, we get

   $$M((\ell_0+1)\hat{D}^{(2)}(p)) \geqslant g\left(g^{\sum_{t=0}^{\ell_0-1} B_{t\hat{D}^{(2)}(p)}}(1)\right) = g^{\sum_{t=0}^{\ell_0} B_{t\hat{D}^{(2)}(p)}}(1).$$

We now let $x_0 = \frac{1}{p}\left(\bar{g} + \log(1-p)^{-1} + \sqrt{2\bar{g}\log(1-p)^{-1} + \log^2(1-p)^{-1}}\right)$, and easily check that

$$\ell_0 \geqslant \frac{1}{p}(\sqrt{\bar{g}} + \sqrt{2\log(1-p)^{-1}})^2 \geqslant \frac{1}{p}(\bar{g} + 2\log(1-p)^{-1} + 2\sqrt{2\bar{g}\log(1-p)^{-1}}) \geqslant x_0 > 0. \tag{7}$$

Moreover, x_0 satisfies

$$-\frac{x_0 p}{2}\left(1 - \frac{\bar{g}}{x_0 p}\right)^2 = \log(1-p). \tag{8}$$

We obtain

$$\begin{aligned}
\Pr\left(\left(M(t_0) \geqslant \frac{2\hat{D}^{(1)}(p)}{P}\right) \cup \mathcal{S}_{t_0}\right) &\geqslant \Pr\left(B \geqslant g^*\left(\frac{2\hat{D}^{(1)}(p)}{P}\right)\right) \\
&\geqslant 1 - \Pr\left(B \leqslant \frac{\bar{g}}{x_0 p}\,\mathbb{E}(B)\right) \\
&\geqslant 1 - e^{-\frac{\mathbb{E}(B)}{2}\left(1-\frac{\bar{g}}{x_0 p}\right)^2} \\
&\geqslant 1 - e^{-\frac{x_0 p}{2}\left(1-\frac{\bar{g}}{x_0 p}\right)^2} \\
&= p.
\end{aligned}$$

The first inequalities comes from Eq. (4). The second and the fourth inequalities hold because, by definition of B and Eq. (7), we have $\mathbb{E}(B) \geqslant \ell_0 p \geqslant x_0 p$. The third inequality is a Chernoff bound [15] applied to B, which is a sum of independent Bernoulli variables. The last equality is by Eq. (8). ◀

Combined with the basic properties of the SAP_g's executions stated in the previous section, Lemma 7 allows us to show our main result:

▶ **Theorem 8.** *If g is a non-decreasing and inflationary function, then the SAP_g algorithm solves the mod P-synchronization problem in any probabilistic network that is strongly connected w.h.p. More precisely, for all $p \in (0,1)$, nodes synchronize within*

$$\hat{D}^{(2)}(p)\left\lceil \frac{\log\left((1-p)^{-1}\right)}{p}\left(\sqrt{g^*\left(\frac{2\hat{D}^{(1)}(p)}{P}\right)}+\sqrt{2}\right)^2\right\rceil + 3\hat{D}^{(1)}(p)$$

rounds with probability p^4, if $\hat{D}^{(1)}(p)$ and $\hat{D}^{(2)}(p)$ denote the order one and two probabilistic diameters of the network.

Proof. For ease of notation, we let $\hat{D}^{(1)} = \hat{D}^{(1)}(p)$. We first define four random variables:

1. Let i_0 be any node satisfying $C_{i_0}(t_0) = \min_{i\in[n]} C_i(t_0)$.
2. Let t_1 be the smallest integer greater than or equal to t_0, such that at least one node holds a clock equal to 0 in round t_1 if it exists, or is equal to infinity otherwise.
3. If t_1 is finite, let i_1 be any node such that $C_{i_1}(t_1) = 0$. Otherwise, let i_1 be an arbitrary node.
4. If t_1 is finite, let i_2 be any node satisfying $C_{i_2}(t_1+\hat{D}^{(1)}) = \min_{i\in[n]} C_i(t_1+\hat{D}^{(1)})$. Otherwise, let i_2 be an arbitrary node.

Then we define the events E and E' as

$$E \stackrel{\text{def}}{=} \{\epsilon \in \mathcal{G}_n \mid t_1 < t_0 + \hat{D}^{(1)}\} \qquad \text{and} \qquad E' \stackrel{\text{def}}{=} \{\epsilon \in \mathcal{G}_n \mid M(t_0) \geqslant \tfrac{2\hat{D}^{(1)}}{P}\}.$$

By Lemma 6, we have $\Gamma_{i_0}^{t_0,\hat{D}^{(1)}} \cap \overline{E} \subseteq \mathcal{S}_{t_0+\hat{D}^{(1)}}$. Since $\mathcal{S}_{t_0+\hat{D}^{(1)}} \subseteq \mathcal{S}_{t_0+3\hat{D}^{(1)}}$, it follows that

$$\Pr\left(\mathcal{S}_{t_0+3\hat{D}^{(1)}} \mid \Gamma_{i_0}^{t_0,\hat{D}^{(1)}} \cap \overline{E} \cap (E' \cup \mathcal{S}_{t_0})\right) = 1. \tag{9}$$

In any execution belonging to E, t_1 is finite and $C_{i_1}(t_1) = 0$. Therefore, in any execution in $E' \cap E \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}}$, every variable M_i satisfies

$$PM_i(t_1+\hat{D}^{(1)}) \geqslant PM(t_1+\hat{D}^{(1)}) \geqslant PM(t_0) \geqslant 2\hat{D}^{(1)} = C_{i_1}(t_1)+2\hat{D}^{(1)} \geqslant C_i(t_1+\hat{D}^{(1)})+\hat{D}^{(1)}.$$

The first and third inequalities above are by definition of M and E', respectively. The second one holds because M is non-decreasing, and the last one comes from Lemma 3 and the fact that the execution is in $\Gamma_{i_1}^{t_1,\hat{D}^{(1)}}$. Lemma 5 then applies, and Lemma 6 shows that

$$E' \cap E \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}} \cap \Gamma_{i_2}^{t_1+\hat{D}^{(1)},\hat{D}^{(1)}} \subseteq \mathcal{S}_{t_1+2\hat{D}^{(1)}}.$$

Since the random variable t_1 is greater than t_0, we get $\mathcal{S}_{t_0} \subseteq \mathcal{S}_{t_1+2\hat{D}^{(1)}}$, and so

$$\Pr\left(\mathcal{S}_{t_1+2\hat{D}^{(1)}} \mid (E' \cup \mathcal{S}_{t_0}) \cap E \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}} \cap \Gamma_{i_2}^{t_1+\hat{D}^{(1)},2\hat{D}^{(1)}} \cap \Gamma_{i_0}^{t_0,\hat{D}^{(1)}}\right) = 1. \tag{10}$$

We are now in position to bound the probability $\Pr(\mathcal{S}_{t_0+3\hat{D}^{(1)}})$ from below. For the sake of readability, the conditional probability given the event $(E' \cup \mathcal{S}_{t_0}) \cap \Gamma_{i_0}^{t_0,\hat{D}^{(1)}}$ is now denoted $\Pr'$. Then we have

$$\begin{aligned}
\Pr(\mathcal{S}_{t_0+3\hat{D}^{(1)}}) &\geqslant \Pr(\mathcal{S}_{t_0+3\hat{D}^{(1)}} \cap \Gamma_{i_0}^{t_0,\hat{D}^{(1)}} \cap (E' \cup \mathcal{S}_{t_0})) \\
&= \Pr'(\mathcal{S}_{t_0+3\hat{D}^{(1)}}) \times \Pr(\Gamma_{i_0}^{t_0,\hat{D}^{(1)}} \mid E' \cup \mathcal{S}_{t_0}) \times \Pr(E' \cup \mathcal{S}_{t_0}) \\
&\geqslant p^2 \Pr'(\mathcal{S}_{t_0+3\hat{D}^{(1)}}) \\
&= p^2 \Pr'(\mathcal{S}_{t_0+3\hat{D}^{(1)}} \mid E) \Pr'(E) + p^2 \Pr'(\mathcal{S}_{t_0+3\hat{D}^{(1)}} \mid \overline{E}) \Pr'(\overline{E}) \\
&\geqslant p^2 \Pr'(\mathcal{S}_{t_1+2\hat{D}^{(1)}} \mid E) \Pr'(E) + p^2 \Pr'(\overline{E}) \\
&\geqslant p^2 \Pr'(\mathcal{S}_{t_1+2\hat{D}^{(1)}} \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}} \cap \Gamma_{i_2}^{t_1+\hat{D}^{(1)},\hat{D}^{(1)}} \mid E) \Pr'(E) + p^2 \Pr'(\overline{E}) \\
&\geqslant p^2 \Pr'(\mathcal{S}_{t_1+2\hat{D}^{(1)}} \mid E \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}} \cap \Gamma_{i_2}^{t_1+\hat{D}^{(1)},\hat{D}^{(1)}}) \\
&\quad \times \Pr'(\Gamma_{i_2}^{t_1+\hat{D}^{(1)},\hat{D}^{(1)}} \mid E \cap \Gamma_{i_1}^{t_1,\hat{D}^{(1)}}) \\
&\quad \times \Pr'(\Gamma_{i_1}^{t_1,\hat{D}^{(1)}} \mid E) \Pr'(E) + p^2 \Pr'(\overline{E}) \\
&\geqslant p^4 \Pr'(E) + p^2 \Pr'(\overline{E}) \\
&\geqslant p^4
\end{aligned}$$

Lemma 7 and the fact that Pr is memoryless are used in the second inequality. The third inequality is based on Eq. (9) and the fact that, in any execution in E, it holds that $\mathcal{S}_{t_1+2\hat{D}^{(1)}} \subseteq \mathcal{S}_{t_0+3\hat{D}^{(1)}}$. Sixth inequality relies on Eq. (10) and the fact that Pr is memoryless. ◀

As in the deterministic analysis of SAP_g, the above bound on its stabilization time provides an upper bound on its space complexity, namely each node uses at most

$$\log_2 P + 2\log_2\left(g^{t_0+3\hat{D}^{(1)}(p)}(\overline{M}_0)\right)$$

bits with probability p^4, if t_0 is defined by Eq. (1) and $\overline{M}_0 = \max_{i\in[n]} M_i(0)$. The time bound and the space bound thus depend respectively on the functions g^* and g, leading to a time-space trade-off for choosing g: the faster g grows, the lower the synchronization time is, and the higher its space complexity is.

5 The *SAP* algorithm in the Application to push-based models

The previous section demonstrates that the notion of probabilistic diameter is a powerful tool for the probabilistic analysis of distributed algorithms. This section applies our general result in Theorem 8 to a probablistic communication model, called the PUSH model [26]. This popular model in rumor spreading consists in the following: Given a base network G of size n, in each round t, each node randomly selects one of its outgoing neighbors in G with equal probability to send its round t message. This communication strategy yields a probablity measure on $\mathcal{G}_n$ which is clearly memoryless.

The PUSH model has been extensively studied with various base networks. In this section, we use two seminal works for this model: First, Feige et al. [24] introduced the notion of *almost sure rumor coverage time*, denoted $\mathcal{T}(G)$, and provided a general upper bound on this parameter. Interestingly, $\mathcal{T}(G)$ is equal to our probabilistic order one diameter for the specific value $p = 1 - 1/n$, namely $\hat{D}^{(1)}(1 - 1/n)$. Second, Pittel [33] refined this general bound on $\mathcal{T}(G)$ in the particular case where G is fully-connected.

5.1 The push model in a general symmetric network

In the PUSH communication model, Feige et al. [24] showed that a rumor reaches the n nodes of a symmetric and connected network within $12n\log_2 n$ rounds with probability $1-1/n$. In other words, the order one probabilistic diameter satisfies:

$$\hat{D}^{(1)}(1-1/n) \leqslant 12n\log_2 n.$$

Using Lemma 1 and the inequalities $(1-1/n)^4 \geqslant 1-4/n$ and $g^* \geqslant 1$, Theorem 8 then yields the following result.

▶ **Corollary 9.** *Let g be any non-decreasing and inflationary function. For the PUSH model in a general symmetric connected network with n nodes, the SAP_g algorithm achieves mod P-synchronization within $324\,n\,(\log_2 n)^2 g^*(24\,P^{-1}\,n\log_2 n)$ rounds with probability $1-\frac{4}{n}$.*

In the context of Corollary 9, Table 1 provides the probabilistic time and space complexities of SAP_g for two different choices of g, namely $g = x \mapsto x+1$ and $g = x \mapsto 2x$ (recall the notation $\overline{M}_0 = \max_{i\in[n]} M_i(0)$). It illustrates the general space-time trade-off that we have just pointed out, at the end of Section 4.

g	stabilization time	space complexity
$g = x \mapsto x+1$	$O\left(n^2\log^3 n\right)$	$O\left(\log\left(\overline{M}_0+n\right)\right)$
$g = x \mapsto 2x$	$O\left(n\log^3 n\right)$	$O\left(\log\overline{M}_0 + n\log^3 n\right)$

Table 1 The SAP_g algorithm for the PUSH model in a symmetric network of size n.

5.2 The push model in fully-connected networks

In the particular case of fully-connected networks, Frieze and Grimmett [26] improved the above upper bound on the time complexity of rumor spreading in the PUSH model. Their bound as well as its refinement by Pittel [33] are asymptotic in the sense that they hold for sufficiently large networks. This is why our new bound on the stabilization time of the SAP_g algorithm in the particular case of a fully-connected network will be proved to hold only in sufficiently large networks.

Let us briefly recall the main result in [33]: Pittel first defines the random variable S_n as

$$S_n(\mathbb{G}) \stackrel{\text{def}}{=} \inf\{\delta\in\mathbb{N} \mid \mathbb{G}\in\Gamma_{i_0}^{0,\delta}\},$$

where $\mathbb{G}\in\mathcal{G}_n$, and i_0 is a fixed node. Since the network is fully-connected, all nodes play the same role and S_n does not actually depend on the choice of the origin node i_0.

▶ **Theorem 10** (Theorem 1 in [33]). *If $\omega:\mathbb{N}\to\mathbb{N}$ tends to infinity, then*

$$\lim_{n\to\infty} \Pr_n(|S_n - \log n - \log_2 n| \leqslant \omega(n)) = 1.$$

It follows that for every $p\in[0,1)$ and every such function ω, there exists a positive integer $N_p(\omega)$ such that for all integers $n \geqslant N_p(\omega)$, it holds that

$$\Pr_n\big(S_n - \log n - \log_2 n \leqslant \omega(n)\big) \geqslant \Pr_n\big(|S_n - \log n - \log_2 n| \leqslant \omega(n)\big) \geqslant p. \tag{11}$$

Here, log denotes the natural logarithm. In the PUSH model, all random variables $\mathbb{G}(t)$ are identically distributed. Hence, for all nodes i, and all non-negative integers t and δ, we have $\Pr(\Gamma_i^{t,\delta}) = \Pr(\Gamma_{i_0}^{0,\delta})$, and thus

$$\hat{D}^{(1)}(p) = \inf\left\{\delta \in \mathbb{N} \mid \Pr(\Gamma_{i_0}^{0,\delta}) \geqslant p\right\}.$$

Denoting by N_p the integer $N_p(\log)$ defined in Eq. (11), we obtain that for all integers $n \geqslant N_p$,

$$\hat{D}^{(1)}(p) \leqslant \log_2 n + 2\log n. \tag{12}$$

▶ **Corollary 11.** *Let g be a non-decreasing inflationary function. For any real number $p \in [\frac{1}{2}, 1)$ and any integer $n \geqslant N_p$, the SAP_g algorithm achieves mod P-synchronization within $81\,\log(1-p)^{-1}\,(\log_2 n)\,g^*\left(6\,P^{-1}\,\log_2 n\right)$ rounds with probability p^4 in the fully-connected graph of size n and the communication PUSH model.*

As a complement to Eq. (12), we now compute the value of $\hat{D}^{(1)}(p)$ in small fully-connected networks, and thus obtain an approximation of N_p. For that, we fix a node $i_0 \in [n]$ and an integer $t_0 \in \mathbb{N}$, and we define the random variable $R_n(t)$ by

$$R_n(t) = \left|\{j \in [n] \mid (i_0, j) \text{ is an arc of } \mathbb{G}(t_0+1 : t_0+t)\}\right|.$$

Observe that for the PUSH model in a fully-connected graph, the probability distribution of $R_n(t)$ does not depend on the choices of i_0 and t_0. Moreover, the probability $\Pr_n$ is perfectly described by the sequence of the random variables $R_n(1), R_n(2), \ldots$

▶ **Lemma 12.** *Let $a, b \in \{0, \ldots, n\}$. If $a \leqslant b \leqslant 2a$, then*

$$\Pr_n(R_n(t+1) = b \mid R_n(t) = a) = \frac{1}{n^a} \sum_{\ell=b-a}^{a} \binom{a}{\ell}\binom{n-a}{a-\ell}\left\{\begin{matrix}\ell\\ b-a\end{matrix}\right\} a^{a-\ell}(b-a)!$$

where $\left\{\begin{matrix}a\\ b\end{matrix}\right\}$ is the Stirling number of the second kind. Otherwise, $\Pr_n(R_n(t+1)=b \mid R_n(t) = a)$ is null.

Proof. We denote by A and B the two sets of nodes that are the targets of an arc whose source is i_0 in the digraphs $\mathbb{G}(1:t)$ and $\mathbb{G}(1:t+1)$, respectively. Thus a node j belongs to B if and only if there exists an arc from A to j in $\mathbb{G}(t+1)$.

In round $t+1$, each node in A picks one node uniformly, among all nodes. Then the total number of draws is n^a. Since each draw is equiprobable, we only have to count the number of favorable draws, that is, the draws such that $|B| = b$. Let ℓ_0 be the number of nodes in A that pick a node in $[n] \setminus A$ in round $t+1$; we have

$$\Pr_n(|B| = b \mid |A| = a) = \sum_{\ell=0}^{a} \Pr_n(|B| = b \cap \ell_0 = \ell \mid |A| = a).$$

We now fix some $\ell_0 \in \{0, \cdots, a\}$, and sample ℓ_0 nodes among the a nodes in A. For that, there are $\binom{a}{\ell_0}$ possibilities. Moreover, we partition the set $[n] \setminus A$ into two parts: $B \setminus A$, of size $b-a$ and $[n] \setminus B$. The number of possible partitionings is $\binom{n-a}{b-a}$. Then there are exactly $\left\{\begin{matrix}\ell_0\\ b-a\end{matrix}\right\}(b-a)!$ surjective mappings from the previously chosen set of ℓ_0 nodes in A into the set $B \setminus A$ [36]. Finally, $a - \ell_0$ nodes in A pick a node belonging to A in round $t+1$. There are $a^{a-\ell_0}$ possibilities. Gathering all mentioned terms, and removing terms in which $\left\{\begin{matrix}\ell_0\\ b-a\end{matrix}\right\} = 0$, we obtain the final expression of the lemma. ◀

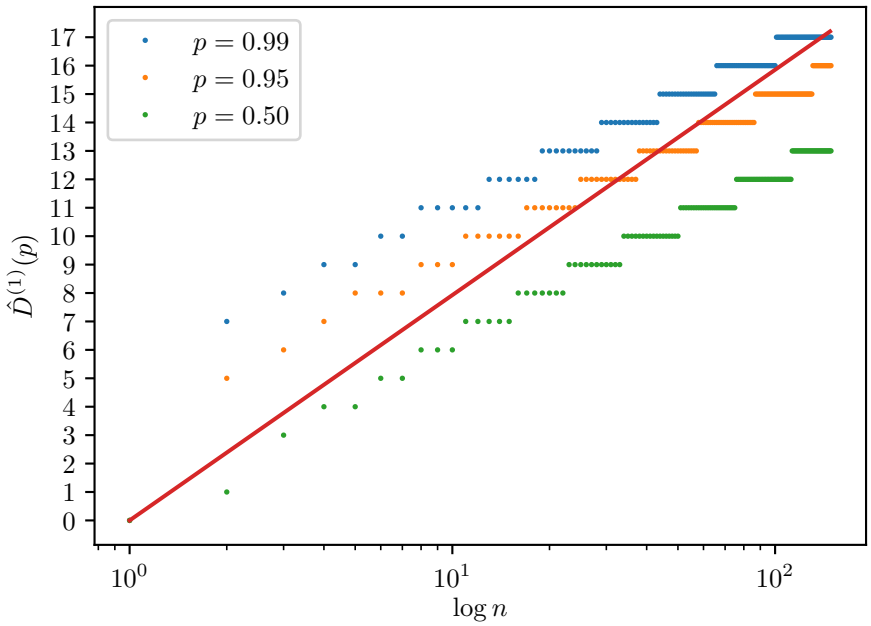

Figure 2 Some values of $\hat{D}^{(1)}(p)$ in the PUSH model, in a fully-connected network of size n.

Interestingly, a different expression of $\Pr_n(|B| = b \mid |A| = a)$ was used by Pittel in [33]. Lemma 12 shows that $(R_n(t))_{t\geqslant 1}$ is a Markov process, and provides an effective and efficient way for computing the probability distribution of each random variable $R_n(t)$ as well as the value of $\hat{D}^{(1)}(p)$. Figure 2 reports our results: The straight line represents the theoretical bound provided by Eq. (12) as a function of the logarithm of the network size $\log n$. For each $p \in \{0.5,\ 0.95,\ 0.99\}$, the values of $\hat{D}^{(1)}(p)$ provided by Lemma 12 are denoted by dots.

Figure 2 yields an estimation of N_p: Choosing $p = 0.5$, all the values of $\hat{D}^{(1)}(0.5)$ that we have computed are smaller that the bound provided by Eq. (12). This suggests that Corollary 11 holds for all n, that is, $N_{0.5} = 1$. Similarly, Figure 2 suggests that $N_{0.95} = 59$. By contrast, an estimate for $N_{0.99}$ would require to compute $\hat{D}^{(1)}(0.99)$ for larger values of n.

6 Concluding Remarks

This paper provides a general solution to the mod P-synchronization problem in general probabilistic communication models. Our proof is based on two basic assumptions on the probabilistic network, making it applicable to a broad range of models. The case of the PUSH model for a general symmetric network that we have examined at the end of the paper, provides an example of probabilistic networks for which no clock synchronization algorithms have been yet devised.

Our paper extends the findings of [12], which proved the correctness of SAP_g in a wide class of dynamic networks, including networks that have an infinite dynamic diameter. The probabilistic study developed in this paper significantly enlarges the scope of correctness for SAP_g, and demonstrates the versatility of this algorithm.

References

1 Karine Altisen, Stéphane Devismes, Swan Dubois, and Franck Petit. Introduction to distributed self-stabilizing algorithms. *Synthesis Lectures on Distributed Computing Theory*, 8(1):1–165, 2019.

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

3 Anish Arora, Shlomi Dolev, and Mohamed G. Gouda. Maintaining digital clocks in step. *Parallel Processing Letters*, 1:11–18, 1991.

4 Paul Bastide, George Giakkoupis, and Hayk Saribekyan. Self-stabilizing clock synchronization with 1-bit messages. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA, 2021*, pages 2154–2173, 2021.

5 Michael Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols. In *Proceedings of the Second Symposium on Principles of Distributed Computing*, pages 27–30, 1983.

6 Philip. A. Bernstein, Vassos Hadzilacos, and Nathan Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

7 Henrik Björklund, Sven Sandberg, and Sergei Vorobyov. Memoryless determinacy of parity and mean payoff games: a simple proof. *Theoretical Computer Science*, 310(1-3):365–378, 2004.

8 Lucas Boczkowski, Amos Korman, and Emanuele Natale. Minimizing message size in stochastic communication patterns: fast self-stabilizing protocols with 3 bits. *Distributed Comput.*, 32(3):173–191, 2019.

9 Paolo Boldi and Sebastiano Vigna. Universal dynamic synchronous self-stabilization. *Distributed Computing*, 15(3):137–153, 2002.

10 Christian Boulinier, Franck Petit, and Vincent Villain. Synchronous vs. asynchronous unison. *Algorithmica*, 51(1):61–80, 2008.

11 Bernadette Charron-Bost and Shlomo Moran. The firing squad problem revisited. *Theoretical Computer Science*, 793:100–112, 2019.

12 Bernadette Charron-Bost and Louis Penet de Monterno. Self-Stabilizing Clock Synchronization in Dynamic Networks. In *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

13 Bernadette Charron-Bost and Louis Penet de Monterno. Impossibility of self-stabilizing synchronization with bounded memory. ., 2024.

14 Bernadette Charron-Bost and André Schiper. The Heard-Of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009.

15 Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

16 Michael R Clarkson and Fred B Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010.

17 Alejandro Cornejo and Fabian Kuhn. Deploying wireless networks with beeps. In *24th International Symposium on Distributed Computing, DISC 2010*, volume 6343 of *Lecture Notes on Computer Science*, pages 148–162. Springer, 2010.

18 Shlomi Dolev. Possible and impossible self-stabilizing digital clock synchronization in general graphs. *Real Time Syst.*, 12(1):95–107, 1997.

19 Shlomi Dolev and Jennifer L. Welch. Self-stabilizing clock synchronization in the presence of byzantine faults. *J. ACM*, 51(5):780–799, 2004.

20 Cynthia Dwork, Nancy A. Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, 1988.

21 Shimon Even and Sergio Rajsbaum. Unison, canon, and sluggish clocks in networks controlled by a synchronizer. *Mathematical systems theory*, 28(5):421–435, 1995.

22 Shimon Even and Sergio Rajsbaum. Unison, canon, and sluggish clocks in networks controlled by a synchronizer. *Math. Syst. Theory*, 28(5):421–435, 1995.

23 Rui Fan and Nancy Lynch. Gradient clock synchronization. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 320–327, 2004.

24 Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1(4):447–460, 1990.

25 Michael Feldmann, Ardalan Khazraei, and Christian Scheideler. Time- and space-optimal discrete clock synchronization in the beeping model. In *32nd ACM Symposium on Parallelism in Algorithms and Architectures, SPAA*, pages 223–233. ACM, 2020.

26 Alan M Frieze and Geoffrey R Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, 1985.

27 Mohamed Gouda and Ted Herman. Stabilizing unison. *Inf. Process. Lett.*, 35(4):171–175, 1990.

28 Ted Herman and Sukumar Ghosh. Stabilizing phase-clocks. *Information Processing Letters*, 54(5):259–265, 1995.

29 Ali Jadbabaie. Natural algorithms in a networked world: technical perspective. *Commun. ACM*, 55(12):100, 2012.

30 Ronald Kempe, Joseph Y. Dobra, and Moshe Y. Gehrke. Gossip-based computation of aggregate information. In *Proceeding of the 44th IEEE Symposium on Foundations of Computer Science, FOCS*, pages 482–491, Cambridge, MA, USA, 2003.

31 Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

32 Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. Tight bounds for clock synchronization. *Journal of the ACM (JACM)*, 57(2):1–42, 2010.

33 Boris Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, 1987.

34 TK Srikanth and Sam Toueg. Optimal clock synchronization. *Journal of the ACM (JACM)*, 34(3):626–645, 1987.

35 Steven H. Strogatz. From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143(1-4):1–20, 2000.

36 Horst Wegner. Stirling numbers of the second kind and bonferroni's inequalities. *Elemente der Mathematik*, 60(3):124–129, 2005.

A Extra proofs

First, we state the following lemma, which is a reformulation of Lemma 2 in [12]. We fix any execution ϵ of SAP_g.

▶ **Lemma 13.** *If the clock value of the agent i is greater than 0 at round t, then it is equal to*

$$C_i(t) = 1 + \min_{j \in \mathrm{In}_i(t)} C_j(t-1).$$

▶ **Lemma 5.** *Let d be a positive integer. If $C_i(t) + d \leqslant PM_i(t)$ holds for all nodes i, then all the clocks C_i are greater than 0 in the round interval $[t+1, t+d-1]$.*

Proof. Let i be any node, and let $\ell \in [d-1]$. We have

$$1 + \min_{j \in \mathrm{In}_i(t+\ell)} C_j(t+\ell-1) \leqslant 1 + C_i(t+\ell-1) \leqslant \ell + C_i(t) < PM_i(t) \leqslant PM_i(t+\ell-1).$$

The first inequality is due to the self-loop at node i in $\mathbb{G}(t+\ell)$, the second one is a consequence of the self-loop and Lemma 3, the third inequality is the assumption of the lemma. The fourth one comes from the fact that M_i is non-decreasing. It follows from line 5 that $C_i(t+\ell) \neq 0$. ◀

▶ **Lemma 6.** *Let d be any positive integer, and k be a node such that $C_k(t) = \min_{j \in [n]} C_j(t)$. If the execution ϵ belongs to $\Gamma_k^{t,d}$ and all the clocks C_i are greater than 0 in the round interval $[t+1, t+d-1]$, then the network is synchronized in round $t+d$.*

Proof. We fix an execution ϵ and a positive integer d. First, we prove by induction on $\ell \in [d-1]$ that

$$\forall i \in [n], \quad C_i(t+\ell) = \ell + \min_{j \in \mathrm{In}_i(t+1:t+\ell)} C_j(t). \tag{13}$$

1. The base case $\ell = 1$ is an immediate consequence of Lemma 13.
2. Inductive step: let us assume that Eq. (13) holds for some ℓ with $1 \leqslant \ell < d-1$. For every node i in $[n]$, we have

$$\begin{aligned} C_i(t+\ell+1) &= 1 + \min_{j \in \mathrm{In}_i(t+\ell+1)} C_j(t+\ell) \\ &= 1 + \ell + \min_{j \in \mathrm{In}_i(t+\ell+1)} \left(\min_{j' \in \mathrm{In}_i(t+1:t+\ell)} C_{j'}(t) \right) \\ &= 1 + \ell + \min_{j \in \mathrm{In}_i(t+1:t+\ell+1)} C_j(t). \end{aligned}$$

   The first equality is a direct consequence of Lemma 13, the second one is by inductive hypothesis, and the third one is due to the fact that $\mathbb{G}(t+1 : t+\ell+1) = \mathbb{G}(t+1 : t+\ell) \circ \mathbb{G}(t+\ell+1)$.

This completes the proof of Eq (13) for every integer $\ell \in [d-1]$.

Then for each node i, we get

$$\begin{aligned} C_i(t+d) &= \left[1 + \min_{j \in \mathrm{In}_i(t+d)} C_j(t+d-1) \right]_{PM_i(t+d-1)} \\ &= \left[d + \min_{j \in \mathrm{In}_i(t+1:t+d)} C_j(t) \right]_{PM_i(t+d-1)} \\ &= \left[d + C_k(t) \right]_{PM_i(t+d-1)}. \end{aligned}$$

The second equality comes from a reasoning similar to the inductive case above, using Eq (13) at round $t + d - 1$. It follows that all the counters $C_i(t + d)$ are equal modulo P, i.e., the system is synchronized in round $t + d$. ◀