A randomized linear algorithm for clock synchronization in multi-agent systems

Saverio Bolognani, Ruggero Carli, Enrico Lovisari, and Sandro Zampieri

***Abstract*—A broad family of randomized clock synchronization protocols based on a second order consensus algorithm is proposed. Under mild conditions on the graph connectivity, it is proved that the parameters of the algorithm can always be tuned in such a way that the clock synchronization is achieved in the probabilistic mean–square sense. This family of algorithms contains, as particular cases, several known approaches which range from distributed asynchronous to hierarchical synchronous protocols. This is illustrated by specializing the algorithm for the well-known broadcast and gossip scenarios in wireless communications, and for the standard hierarchical protocol used in the context of wired communications in data networks. In these cases, we show how the feasible range for the algorithm parameters can be explicitly computed. Finally, the performance of this strategy is validated by actual implementation in a real testbed and by numerical simulations.**

I. Introduction

Multi-agents systems, i.e. systems composed of a large number of interconnected agents, have received a tremendous amount of attention in the last years. This is mainly motivated by the many emerging applications where these systems are applied, e.g., sensor networks for environmental monitoring, mobile robotic networks for vehicle tracking and mapping, camera networks for surveillance, just to mention few.

With respect to classical applications, the tight constraints in terms of bandwidth and communication delays, the need for robustness with respect to agent failures or malicious attacks, and the ineffectiveness of classical system theoretical tools in terms of complexity and computational burden, make communication and control much harder in multi-agent systems.

In almost all applications, one of the most critical issue is time synchronization among agents. In some applications, synchronization requirements are rather basic. For example, in any sensor network, different devices have to provide their measurements with proper time–stamping for subsequent data fusion and processing. In other scenarios higher precision is needed, and achieving synchronization can be very challenging. For instance, there are applications where the collected data need to be interpreted according to fast dynamical models for the system (e.g. in distributed detection and localization of moving targets), other applications where precise time synchronization might be required to perform specific measurements on the system (e.g. synchronized voltage phasor measurement in electric power networks, and certain distance measurements based on time-of-flight), and some other network ancillary services that rely on correct time sync (e.g. TDMA communication, where the use of a shared communication channel is regulated by precise slotting of the access times).

In addition to fulfill particular application requirements, solutions for large scale systems are also required to be robust and scalable. Robustness is in general meant with the respect to partial failure of the system, communication faults, node appearance and disappearance, and also malicious attacks. Instead scalable solutions are those solutions whose performance is independent from the system size; reconfiguration should require minimal effort every time a new node enters or leaves the network, or if two networks merge.

Clock synchronization in multi-agent systems has been a lively topic in the last years. In [1], [2], [3] the authors propose a family of time synchronization algorithms based on the construction of a hierarchical coordination tree. However, this solution is not scalable and not robust as meant above, and, thus, it is poorly suited for the above mentioned applications. Indeed, maintaining such an architecture may be unbearable in many scenarios, and these solutions exhibit little robustness against the failure of any node which is not a leaf of the tree. Other algorithms available in the literature try to circumvent the main drawbacks of tree-based solutions by constructing different architectures, like clusters of nodes, each one headed by an elected master node [4], [5]. Master nodes then synchronize among them, at a higher level of coordination. Unless the communication architecture is specifically designed, there are no guarantees that master nodes can communicate more reliably over the longer distances of the higher layer.

On the other side, distributed (leaderless) protocols aim to guarantee scalability and robustness by avoiding a clear hierarchy among the agents. Existing algorithms in this sense include [6] and [7], which however suffer from specific drawbacks: the algorithm proposed in [6], inspired by the fireflies integrate-and-fire synchronization mechanism, can compensate for different clock offsets but not for different clock skews; on the other hand, the algorithm proposed in [7] compensates for the clock skews but not for the time offsets. Fully distributed protocols that can compensate for both clock skews and offsets have been proposed in [8], [9], [10], [11], [12]. For these algorithms, convergence has been proved by the authors, under reasonable assumptions. The main weakness of these solutions resides in their highly nonlinear dynamic behavior, which prevents the analysis of their robustness with respect to data losses in the communication, quantization noise, communica-

S. Bolognani is with the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland (e-mail: bsaverio@ethz.ch). E. Lovisari is with the Univ. Grenoble Alpes and CNRS, Gipsa-Lab, Grenoble, France (e-mail: enrico.lovisari@gipsa-lab.fr). R. Carli and S. Zampieri are with the University of Padova, Italy. (e-mail: {carlirug|zampi}@dei.unipd.it).

tion errors, and unmodeled dynamics of the clocks.

Recent literature has also cast the clock synchronization problem into an estimation and tracking framework, deriving algorithms based on classical Kalman filter methods and message passing techniques, and capable of identifying the clock parameters in a fully distributed fashion (e.g. [13], [14]). These strategies usually reconstruct offset and skew of each clock and then perform reverse reconstruction to estimate the absolute time. A different approach, adopted in the present paper, is to steer the network to synchronization via a control protocol aiming to stabilize the relative error among pairs of clocks. The idea has originally appeared in the literature in the form of a linear proportional-integral(PI)-like distributed control algorithm, reminiscent of the very popular linear consensus algorithm [15], [16]. The performance and the robustness of this strategy have been thoroughly analyzed via numerical simulations.

The first order counterpart of this algorithm (where clock skews are therefore not corrected) has been the focus of several works appeared in literature, driven by the widespread interest in consensus algorithms. The two main differences between first and second order consensus methods are the following. 1) In first order consensus, at each update step, variables remain within the convex hull of the previous step's variables. Because such a convex hull never expands, the consensus error remains bounded, and the control objective is to make this hull shrink to zero. On the contrary, in second order consensus algorithms the system is inherently unstable, and the control has to counterbalance such instability while steering the agents one towards the other. 2) In first order consensus, the state of the system is directly observed, and therefore the study of the statistics of the synchronization error is straightforward. In second order consensus, part of the states are not directly observable, and computing the evolution of the error dynamics is much more involved. In addition, only the time reading has to reach consensus, while the second component of the state, the compensation for the different skews, will not. For these reasons, providing a formal proof of the convergence of the algorithm in [15], [16] has been proven to be a difficult task, except for some special cases where either the communication graph was restricted to some special families, or some assumptions were made on the asynchronous activation of the nodes.

In the present paper, we adopt a unifying approach and we study a very general randomized synchronization communication protocol coupled with a PI control at each agent. Our main result, Theorem 1, shows that, under very mild conditions on the graph connectivity, it is always possible to properly tune the PI control so that the network achieves clock synchronization. Within the studied family of protocols fall many clock synchronization strategies (deterministic and randomized, hierarchical and distributed, synchronous and asynchronous, broadcast and point-to-point) that have been proposed in the literature for specific applications, where different communication technologies are available. In order to exemplify the broad applicability of our results, we chose three different scenarios:

- Broadcast communication in wireless networks
- Gossip communication in wireless networks
- Hierarchical communication in wired networks.

In each of these scenarios we verify that the general convergence result presented in Theorem 1 applies, and by further analysis we provide a precise rule to tune the algorithm so that convergence is guaranteed. Finally, we validate the algorithm performance in the broadcast scenario on an experimental testbed, while the gossip and the hierarchical scenarios are illustrated via numerical simulations.

The paper is organized as follows: in Section II we introduce a mathematical model for the individual clocks and their dynamic behavior, while in Section III we propose a prototype of synchronization algorithm. The proposed algorithm is extremely general and flexible, and can be specialized for basically any communication strategy among the clocks. Section IV studies the convergence of the proposed algorithm, providing an algebraic characterization that is easily verified in some notable cases. Indeed, in Section V we apply our results to the three cases of broadcast, gossip and hierarchical communication protocols. Section VI concludes the paper. In order to increase readability, all the proofs have been collected in the Appendix.

A. Mathematical preliminaries and notation

The symbols $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{C}$ denote the set of natural, real, non-negative real, and complex numbers, respectively. For vectors and matrices, x^* and M^* denote complex conjugation (transposition if real vector/matrices).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph (or digraph), where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes and $\mathcal{E}$ is the set of edges, i.e., $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $(i, j) \in \mathcal{E}$ if there is an edge going from node i to node j. In our context, the edge (i, j) models the fact that node j can receive information from node i. By $\mathcal{N}_i^{\text{out}}$ we denote the set of out-neighbors of node i, i.e. $\mathcal{N}_i^{\text{out}} = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$, while by $\mathcal{N}_i^{\text{in}}$ we denote the set of in-neighbors of node i, i.e. $\mathcal{N}_i^{\text{in}} = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$. The graph $\mathcal{G}$ is undirected if $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. Note that for an undirected graph we have $\mathcal{N}_i^{\text{in}} = \mathcal{N}_i^{\text{out}}$. In this case, we do not distinguish between in-neighbors and out-neighbors and we simply use the notation $\mathcal{N}_i$ to denote the set of neighbors of i. A graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ is said to be a subgraph of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $\mathcal{E}' \subseteq \mathcal{E}$. A directed path in a digraph is an ordered sequence of nodes such that any ordered pair of nodes appearing consecutively in the sequence is an edge of the digraph. A digraph $\mathcal{G}$ is strongly connected if for any pair of vertices (i, j) there exists a directed path connecting i to j. A cycle in a digraph is a directed path that starts and ends at the same node and that contains no repeated node except for the initial and the final node. A digraph is acyclic if it contains no cycle. A directed tree is an acyclic directed graph with the following property: there exists a node, called the root, such that any other vertex of the directed graph can be reached by one and only one directed path starting at the root. In a directed tree, every in-neighbor of a node is called a parent and every out-neighbor is called a child. A directed spanning tree of a digraph is a spanning subgraph that is a directed tree.

Given a matrix $M \in \mathbb{R}^{N \times N}$, we define the associated graph $\mathcal{G}_M = (\mathcal{V}, \mathcal{E}_M)$ by taking N nodes and putting an edge (j, i)

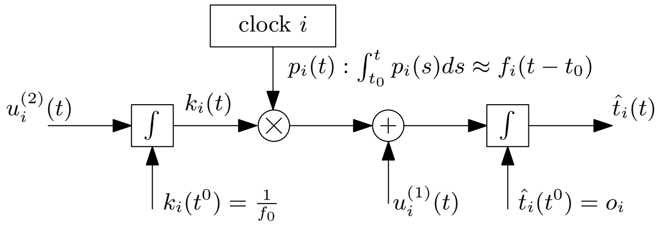

Fig. 1. A schematic representation of the adopted clock model. Control signals $u_1(t)$ and $u_2(t)$ are impulsive signals that are nonzero at update times only. Counter signal $p_i(t)$ is an sequence of pulses at frequency f_i. Initial conditions are set to $k_i(t^0) = \frac{1}{f_0}$ and $\hat{t}_i(t^0) = o_i$.

in $\mathcal{E}_M$ if $M_{ij} \neq 0$. *Vice versa*, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, \ldots, N\}$, the matrix M is compatible with $\mathcal{G}$ if $\mathcal{E}_M \subseteq \mathcal{E}$. With the symbol $\mathbf{1}_N$, or $\mathbf{1}$ if the dimension is well understood, we denote the N-dimensional all-one vector. A weighted Laplacian $L \in \mathbb{R}^{N \times N}$ of the graph $\mathcal{G}$ is a matrix compatible with $\mathcal{G}$ such that $L_{ij} \geq 0$ if $i \neq j$, and $L\mathbf{1} = 0$. Given the vector $v \in \mathbb{R}^N$, by $\operatorname{diag}\{v\}$ we denote the diagonal matrix having the components of v as diagonal elements. Finally, $\mathbb{E}$ denotes mathematical expectation.

II. Clock model

We assume that each agent i in a multi-agent systems (for example, each sensor in a wireless sensor network) is equipped with an individual clock, i.e., a system capable of returning an estimate $\hat{t}_i$, called *time reading*, of the real time t.

In order to derive a mathematical model for each clock, we consider a typical implementation, which is represented in Figure 1. Each clock i has an on-board oscillator which generates a sequence of pulses, called *ticks*, and described in mathematical terms as an impulsive signal $p_i(t)$, at a frequency f_i. The frequency f_i is unknown, and deviates from the known nominal frequency f_0 mainly because of electronic components' tolerance. The time estimator is given by a counter. This counter is initialized at a time t_i^o to a value o_i and every tick increases this counter by a value k_i. The value of the counter at time t is then formally described as

$$\hat{t}_i(t) = \int_{t_i^o}^{t} k_i p_i(s) ds + o_i \tag{1}$$

Equation (1) models the behavior of a single uncontrolled clock for which the values k_i and o_i are fixed. We notice that we can interpret the value k_i as the *oscillator frequency correction* of clock i, and a reasonable choice is to set $k_i = \frac{1}{f_0}$. In fact, notice that $\int_{t_i^o}^{t} p_i(s)ds = \lfloor f_i(t - t_i^o) \rfloor$, where $\lfloor a \rfloor$ denotes the largest integer smaller than or equal to a. If $f_i(t - t_i^o) \gg 1$, then $\lfloor f_i(t - t_i^o) \rfloor \approx f_i(t - t_i^o)$, so that (1) yields

$$\hat{t}_i = k_i f_i (t - t_i^o) + o_i = \frac{f_i}{f_0}(t - t_i^o) + o_i \, .$$

In the ideal case $f_i = f_0$ and $o_i = t_i^o$, the clock obtains the the correct estimate $\hat{t}_i = t$, for all $t \geq 0$, while in general the estimate is a linear affine function of time with slope $\frac{f_i}{f_0} \neq 1$ and with initial offset $o_i - t_i^o$. Since no information on the absolute time is available, that is the best achievable by an isolated clock.

In case the clock is an agent in a network and is able to communicate with other agents, it has the capability to instantaneously modify both k_i and $\hat{t}_i$ at the instant T_{up} at which it receives information from its neighbors. In particular, the controlled counterpart of (1) is

$$\begin{cases} \hat{t}_i(t) = \int_{t_i^o}^{t} \left(k_i(s) p_i(s) + u_i^{(1)}(s) \right) \, ds + o_i \\ k_i(t) = \int_{t_i^o}^{t} u_i^{(2)}(s) \, ds + k_i(t_i^o) \end{cases} \tag{2}$$

where $u_i^{(1)}(t)$ and $u_i^{(2)}(t)$ are impulsive signals that are nonzero only at update times. Therefore, between two consecutive update times T_{up} and T'_{up} the clock receives no information and thus the value k_i is kept fixed and the counter $\hat{t}_i$ evolves according to (1). Thus, for $t \in (T_{\text{up}}, T'_{\text{up}}]$, the same argument as before establishes that

$$\begin{cases} \hat{t}_i(t) = \hat{t}_i(T_{\text{up}}^+) + k_i(T_{\text{up}}^+) f_i (t - T_{\text{up}}) \\ k_i(t) = k_i(T_{\text{up}}^+) \end{cases} \tag{3}$$

where the corrections $u_i^{(1)}(T_{\text{up}})$ and $u_i^{(2)}(T_{\text{up}})$ act as follows

$$\begin{cases} \hat{t}_i(T_{\text{up}}^+) = \hat{t}_i(T_{\text{up}}) + u_i^{(1)}(T_{\text{up}}) \\ k_i(T_{\text{up}}^+) = k_i(T_{\text{up}}) + u_i^{(2)}(T_{\text{up}}) \end{cases} .$$

Assume we have a network of N agents, labeled 1 through N, each equipped with a time estimator as described in the previous section. The goal of this paper is to study how a proper design of the control signals $u_i^{(1)}$ and $u_i^{(2)}$, for $i = 1, \ldots, N$, can yield synchronization in networks of clocks as defined in the following statement.

Problem. *In a network of N agents, each equipped with a clock giving a time estimator $\hat{t}_i(t), i = 1, \ldots, N$, of the absolute time t, design an algorithm so that* synchronization *of the clocks is achieved, in the sense that there exist $\gamma \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$ such that*

$$\left| \hat{t}_i(t) - (\gamma t + \beta) \right| \xrightarrow{t \to \infty} 0, \quad \forall i = 1, \ldots, N. \tag{4}$$

III. Clock synchronization protocol prototype

In this section we propose a very general prototype of clock synchronization algorithm, denoted RandSync, which corresponds to a specific choice of the control signals $u_i^{(1)}$ and $u_i^{(2)}$, and which encompasses many different communication strategies between the agents.

Assume that at a random time T_{up} a group of agents wake up and communicate each other. Each agent i belonging to this group performs the following operations.

1) Agent i sends its time reading to some agents of this group of activated agents;
2) Agent i receives the time readings from some of the agents of this group of activated agents;
3) Agent i uses the received time readings $\hat{t}_j(T_{\text{up}})$ by updating the states $\hat{t}_i$ and k_i as follows

$$\begin{cases} \hat{t}_i(T_{\text{up}}^+) = \hat{t}_i(T_{\text{up}}) + \sum_j E_{ij}(T_{\text{up}}) \left[\hat{t}_j(T_{\text{up}}) - \hat{t}_i(T_{\text{up}}) \right] \\ k_i(T_{\text{up}}^+) = k_i(T_{\text{up}}) + \alpha \sum_j E_{ij}(T_{\text{up}}) \left[\hat{t}_j(T_{\text{up}}) - \hat{t}_i(T_{\text{up}}) \right] \end{cases} \tag{5}$$

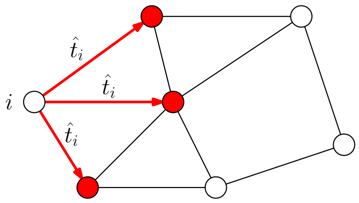

Fig. 2. **Broadcast wireless communication in RANDSYNC-BCAST:** the graph represents the admissible communication between pairs of wireless nodes. When node i activates, it sends its current time reading to neighbor nodes (thick red arrows). The neighbor nodes (in red) update their state according to the algorithm.

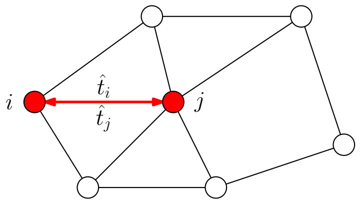

Fig. 3. **Gossip wireless communication in RANDSYNC-GOSSIP:** the graph represents the admissible communication between pairs of wireless nodes. When node i activates, it exchanges current time readings with one of its neighbors (thick red arrow). Both node i and the neighbor node (in red) update their state according to the algorithm.

where $\alpha > 0$, and T_{up}^+ denotes the time instant immediately following T_{up}.

We assume the coefficients $E_{ij}(T_{\text{up}})$ satisfy the following property.

Assumption 1. *For every update time T_{up} we have that $E_{ij}(T_{up}) \geq 0$ for any i, j, and*

$$\sum_{j \neq i} E_{ij}(T_{up}) < 1.$$

Observe that the coefficients E_{ij} will be typically different at different activation times, as they depend on the family of nodes that are activated.

Next, to clarify the broad applicability of the proposed algorithm, we briefly present three examples corresponding to different communication protocols which are employed in wireless and wired sensor networks. Further details for these scenarios will be provided in Section V.

Example 1. Broadcast communication *(*RANDSYNC-BCAST*). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a given digraph, where $\mathcal{V} = \{1, \dots, N\}$ is the set of agents, and an edge $(i, j) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the possibility for agent j to receive information from agent i. The* RANDSYNC-BCAST *protocol is described as follows. At each random time T_{up} there is only one node, say i, which performs the transmission step. Specifically, node i broadcasts its time reading $\hat{t}_i(T_{up})$ to all its out-neighbors (see Figure 2), which update their states $\hat{t}_j$ and k_j according to*

$$\begin{cases} \hat{t}_j(T_{up}^+) = \hat{t}_j(T_{up}) + q\left[\hat{t}_i(T_{up}) - \hat{t}_j(T_{up})\right] \\ k_j(T_{up}^+) = k_j(T_{up}) + \alpha q\left[\hat{t}_i(T_{up}) - \hat{t}_j(T_{up})\right], \end{cases} \quad \forall\, j \in \mathcal{N}_i^{out} \tag{6}$$

where $0 < q < 1$. The RANDSYNC-BCAST *protocol therefore corresponds to the specific implementation of* (5) *in which $E_{ji}(T_{up}) = q$ for all $j \in \mathcal{N}_i^{out}$, and zero otherwise ($i$ being the node that transmits at time T_{up}).* ■

Example 2. Gossip communication *(*RANDSYNC-GOSSIP*). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a given undirected graph, where again $\mathcal{V} = \{1, \dots, N\}$ is the set of agents, and the set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ models the admissible bidirectional communication among the agents. The* RANDSYNC-GOSSIP *protocol is described as follows. At each random time T_{up} a node, say i, activates. Node i selects randomly one of its neighbors, say j, $j \in \mathcal{N}^{(i)}$. The two nodes i, j send each other their time readings $\hat{t}_i(T_{up})$ and $\hat{t}_j(T_{up})$ (see Figure 3). Node i and node j update their states according to*

$$\begin{cases} \hat{t}_i(T_{up}^+) = \hat{t}_i(T_{up}) + q\left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right] \\ k_i(T_{up}^+) = k_i(T_{up}) + \alpha q\left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right] \end{cases} \tag{7}$$

and

$$\begin{cases} \hat{t}_j(T_{up}^+) = \hat{t}_j(T_{up}) + q\left[\hat{t}_i(T_{up}) - \hat{t}_j(T_{up})\right] \\ k_j(T_{up}^+) = k_j(T_{up}) + \alpha q\left[\hat{t}_i(T_{up}) - \hat{t}_j(T_{up})\right] \end{cases} \tag{8}$$

where again $0 < q < 1$. The RANDSYNC-GOSSIP *protocol therefore corresponds to the specific implementation of* (5) *in which $E_{ij}(T_{up}) = E_{ji}(T_{up}) = q$, and zero otherwise ($i$ and j being the two nodes that communicate at time T_{up}).* ■

Example 3. Hierarchical communication *(*RANDSYNC-TREE*). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a given undirected graph, where again $\mathcal{V} = \{1, \dots, N\}$ is the set of agents, and the set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ models the admissible bidirectional communication among the agents. In this scenario, the communication graph $\mathcal{G}$ is an undirected tree. The Hierarchical Communication protocol is described as follows. At each random time T_{up} there is only one node, say i, which requests a time reading from its parent in the tree, say j. Once the parent node j receives a request from an agent i, it responds with the time reading $\hat{t}_j$ (see Figure 4). Node i then computes the difference $\hat{t}_j - \hat{t}_i$, and updates its own states according to*

$$\begin{cases} \hat{t}_i(T_{up}^+) = \hat{t}_i(T_{up}) + \left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right] = \hat{t}_j(T_{up}) \\ k_i(T_{up}^+) = k_i(T_{up}) + \alpha\left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right], \end{cases} \tag{9}$$

The RANDSYNC-TREE *protocol therefore corresponds to the specific implementation of* (5) *in which $E_{ij}(T_{up}) = 1$, and zero otherwise (i being the node that requests the time reading from its parent at time T_{up}).*[1] ■

A convenient way to describe the proposed algorithm consists in employing a vector representation of the states $\hat{t}_i(t)$ and $k_i(t)$, $i \in \mathcal{V}$. Let $\hat{t}(t)$ and $k(t)$ be the vectors $[\hat{t}_1(t) \dots \hat{t}_N(t)]^*$ and $[k_1(t) \dots k_N(t)]^*$, respectively. Let moreover $E(T_{\text{up}})$ be a matrix whose off-diagonal entries are $E_{ij}(T_{\text{up}})$, while the diagonal elements $E_{hh}(T_{\text{up}})$ are

$$E_{hh}(T_{\text{up}}) = -\sum_{\ell \neq h} E_{h\ell}(T_{\text{up}}).$$

[1] Notice that, in this specific case, Assumption 1 is not strictly satisfied. The analysis in Section V shows how the proposed approach can be technically extended to the case in which $\sum_{j \neq i} E_{ij}(T_{\text{up}}) \leq 1$.

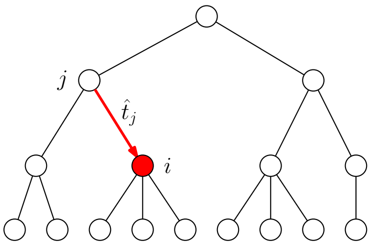

Fig. 4. **Hierarchical communication in wired networks in RANDSYNC-TREE:** the graph represents the architecture of the communication network between nodes. When node i activates, it requests the current time reading from its parent node in the tree (thick red arrow). Once the time reading is received, node i updates its state according to the algorithm.

Using this notation, at the activation time $T_{\rm up}$, the following update occurs

$$\begin{cases}\hat{t}(T_{\rm up}^+) = \hat{t}(T_{\rm up}) + E(T_{\rm up})\hat{t}(T_{\rm up})\\ k(T_{\rm up}^+) = k(T_{\rm up}) + \alpha E(T_{\rm up})\hat{t}(T_{\rm up}).\end{cases} \tag{10}$$

If $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the *graph of the admissible communications*, then the matrices $E(T_{\rm up})$ are weighted Laplacians of subgraphs of $\mathcal{G}$. In fact, the off-diagonal entries of any $E(T_{\rm up})$ are non-negative, and $E(T_{\rm up}(h))\mathbf{1} = 0$.

Finally, given a specific clock synchronization protocol, we define by $\mathcal{M}$ the set of all possible matrices E that result from the allowed combinations of such decisions (nodes, subset of neighbors), so that $E(T_{\rm up}) \in \mathcal{M}$ at any update time $T_{\rm up}$.

The prototype that we described is extremely general, and by choosing the set of matrices $\mathcal{M}$, it can be specialized in order to meet the specifications of many common communication architectures, for example the broadcast wireless communication and the gossip wireless communication which we have introduced in the previous examples, and also hierarchical wired architectures. These notable examples will be presented in Section V, where the behavior of the proposed algorithm will be illustrated via experiments, simulations, and further theoretical analysis.

Remark. *Observe that the update $\hat{t}_i(T_{up}^+) = \hat{t}_i(T_{up}) + \sum_j E_{ij}(T_{up})\left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right]$ can be rewritten as $\hat{t}_i(T_{up}^+) = \left(1 - \sum_j E_{ij}(T_{up})\right)\hat{t}_i(T_{up}) + \sum_j E_{ij}(T_{up})\hat{t}_j(T_{up})$ which represents a convex combination of $\hat{t}_i(T_{up})$ and of the time readings $\hat{t}_j(T_{up})$ received by node i at time T_{up}. This implies that*

$$\max_{(h,k)\in\mathcal{V}\times\mathcal{V}} |\hat{t}_h(T_{up}^+) - \hat{t}_k(T_{up}^+)| \leq \max_{(h,k)\in\mathcal{V}\times\mathcal{V}} |\hat{t}_h(T_{up}) - \hat{t}_k(T_{up})|.$$

In other words the updating steps on the variable $\hat{t}$ resemble the iterations of the standard consensus algorithms. Clearly, if the frequencies f_i, $i \in \mathcal{V}$, are different from each other, acting only on the variable $\hat{t}$ does not allow, in general, to reach asymptotic synchronization. For this reason, in the proposed algorithm, the correction $\sum_j E_{ij}(T_{up})\left[\hat{t}_j(T_{up}) - \hat{t}_i(T_{up})\right]$, suitably scaled by the parameter α, is applied also to the oscillator frequency corrections with the aim of reaching asymptotic synchronization on the quantities $k_i f_i$, $i \in \mathcal{V}$. In this sense we can say that the algorithm we introduced above is based on a second order consensus algorithm.

Remark. *Observe that in the proposed approach, we are assuming either negligible or perfectly predictable communication delays. If this is not true, and if bidirectional communication is available, then the agents can implement some more complex routines in order to estimate their true time difference, like the ones adopted in the Network Time Protocol [17] and in the Precision Time Protocol [18], and use such computed time differences in the update equations* (5). *In the convergence results in Section IV, and in the technical analysis in the Appendix, we will prove that the synchronization error dynamics is described by a stable linear system. If one interprets the effect of uncompensated delays as an additive disturbance to such system, then standard results ensure that the synchronization error remains bounded.*

IV. CONVERGENCE RESULTS

In this section we state our main convergence result for the general clock synchronization algorithm presented in Section III.

We define as $\{T_{\rm up}(h)\}_{h\in\mathbb{N}}$ the ordered sequence of all time instants in which the algorithms is executed, i.e., the sequence obtained by ordering the activation times of all the agents. A specific clock synchronization protocol is therefore fully characterized by:

- a family $\mathcal{M}$ of weighted Laplacians of subgraphs of the graph of the admissible communications $\mathcal{G}$,
- a positive tuning parameter $\alpha \in \mathbb{R}$,
- a random process describing the update times $\{T_{\rm up}(h)\}_{h\in\mathbb{N}}$,
- a random sequence $\{E(h)\}_{h\in\mathbb{N}}$ that specifies which update matrix $E(h) \in \mathcal{M}$ is chosen at the h-th update time $T_{\rm up}(h)$. [2]

We make the following assumptions about the update times $\{T_{\rm up}(h)\}$ and the sequence $\{E(h)\}$.

Assumption 2. *Let us denote by $\delta_{up}(h) = T_{up}(h+1) - T_{up}(h)$ the inter-time between consecutive update times. We assume that $\{\delta_{up}(h)\}_{h\in\mathbb{N}}$ is an i.i.d. process, with moments*

$$\begin{cases}\mathbb{E}\left[\delta_{\rm up}(h)\right] = \mu\\ \mathbb{E}\left[\delta_{\rm up}^2(h)\right] = \sigma^2\end{cases} \quad \forall h \in \mathbb{N}$$

Assumption 3. *For each update time $T_{up}(h)$ a matrix $E(h)$ is independently picked from the family of matrices $\mathcal{M}$ according to a probability distribution $p_{\mathcal{M}}$. With no loss of generality we can assume that each Laplacian in $\mathcal{M}$ can be picked with nonzero probability.*

In other words the previous assumption says that $E(h)$ is an independent and identically distributed, matrix valued, stochastic process. The family $\mathcal{M}$ of weighted Laplacians, the probability distribution $p_{\mathcal{M}}$, and the first and second order moments of the inter-times $\delta_{\rm up}(h)$, are characteristic of the specific implementation of the algorithm. We will show in Section V how Assumptions 2 and 3 are practically verified in many scenarios.

[2] With a slight abuse of notation, we used the symbol $E(h)$ to denote the update matrix at time $T_{\rm up}(h)$, i.e. $E(T_{\rm up}(h))$.

Since switching rule and update instants are random processes, we will be interested in the following notion of synchronization, which solves, in the mean-square sense, the Problem expressed in (4).

Definition (Mean-square synchronization)**.** *We say that switching system with updates* (10) *at times* $\{T_{up}(h)\}_{h\in\mathbb{N}}$*, and autonomous evolution* (3) *between consecutive updates, achieves mean-square synchronization if there exist* $\gamma \in \mathbb{R}^+$ *and* $\beta \in \mathbb{R}$ *such that*

$$\mathbb{E}\left[\left(\hat{t}_i(t) - (\gamma t + \beta)\right)^2\right] \overset{t\to\infty}{\longrightarrow} 0. \tag{11}$$

for all $i \in \mathcal{V}$*.*

It is convenient, for the analysis of the algorithm behavior, to consider the discrete time system which is obtained by sampling the clocks at the update times $\{T_{\rm up}(h)\}_{h\in\mathbb{N}}$. Denote by

$$x(h) = \big[\hat{t}_1(T_{\rm up}(h)) \quad \dots \quad \hat{t}_N(T_{\rm up}(h)) \qquad f_0 k_1(T_{\rm up}(h)) \quad \dots \quad f_0 k_N(T_{\rm up}(h))\big]^*$$

the vector of time readings and oscillator frequency corrections multiplied by the nominal frequency at the update times. By using (10), together with the model (3) for the autonomous evolution between consecutive updates, we obtain the following linear update for $x(h)$

$$x(h+1) = \begin{bmatrix} I + E(h) + \alpha f_0 \delta_{\rm up}(h) D E(h) & \delta_{\rm up}(h) D \\ \alpha f_0 E(h) & I \end{bmatrix} x(h) \tag{12}$$

where

$$D := \operatorname{diag}\{d_1, \dots, d_N\}, \quad d_i := f_i / f_0 > 0$$

contains the ratios between the individual oscillator frequencies f_i and their nominal value f_0. Given the sampling strategy that we introduced, we consider the following notion of mean-square synchronization for the sampled system.

Definition (Mean-square synchronization of the sampled system)**.** *We say that system* (12) *achieves* mean-square synchronization *if there exist* $\gamma \in \mathbb{R}^+$ *and* $\beta \in \mathbb{R}$ *such that*

$$\mathbb{E}[(\hat{t}_i(T_{\rm up}(h)) - (\gamma T_{up}(h) + \beta))^2] \overset{h\to\infty}{\longrightarrow} 0. \tag{13}$$

for all $i \in \mathcal{V}$*.*

The following Proposition shows that mean-square synchronization of the sampled systems is a necessary and sufficient condition for mean-square synchronization of the original switched system.

Proposition 1. *Consider the original switching system with updates* (10) *at times* $\{T_{up}(h)\}_{h\in\mathbb{N}}$*, and autonomous evolution* (3) *between consecutive updates. Consider then its sampled version* (12)*, and let Assumption 2 hold true. Then the former achieves mean-square synchronization if and only if so does the latter.*

Notice that, if we define

$$\bar{E} := \mathbb{E}[E(h)] = \sum_{E\in\mathcal{M}} E \; p_{\mathcal{M}}(E) \tag{14}$$

where $p_{\mathcal{M}}(E)$ is the probability of picking $E \in \mathcal{M}$ as defined in Assumption 3, then we have that

$$\mathcal{G}_{\bar{E}} = \bigcup_{E\in\mathcal{M}} \mathcal{G}_E \subseteq \mathcal{G}\,.$$

We can now state the following result, which gives sufficient conditions for mean-square synchronization to take place. To do so we first introduce the following notation. Let $w \in \mathbb{R}^N$ be the unique vector such that $w^*\bar{E} = 0$ and $w^*\mathbf{1} = 1$ and let $V \in \mathbb{R}^{N\times(N-1)}$ be a full column rank matrix such that $w^*V = 0$. Then we have that $\begin{bmatrix}1 & V\end{bmatrix}^{-1}$ is invertible and

$$\begin{bmatrix}1 & V\end{bmatrix}^{-1} = \begin{bmatrix} w^* \\ W^* \end{bmatrix}$$

for some $W \in \mathbb{R}^{N\times(N-1)}$.

Theorem 1. *Consider the clock synchronization protocol presented in section III, and let Assumptions 1, 2, and 3 hold true. Additionally assume the following properties:*

1) *The graph* $\mathcal{G}_{\bar{E}}$ *contains a directed spanning tree.*
2) *The matrix* W^*DV *has all eigenvalues with positive real part, where the matrices* $V, W \in \mathbb{R}^{N\times(N-1)}$ *are defined as above from the vector* $w \in \mathbb{R}^N$*.*

Then there exists a value $\bar{\alpha} > 0$ *such that, for any* $\alpha \in (0, \bar{\alpha})$*, mean-square synchronization is achieved.*

Sketch of the proof. The proof is postponed to the Appendix, and shows how mean-square synchronization can be analyzed by studying the stability of a matrix $M(\alpha, D)$. The eigenvalues of such matrix $M(\alpha, D)$ are a continuous function of both the parameter α and the clock frequencies contained in D. Via a matrix perturbation argument based on the tools developed in [19], we prove that, under the conditions of Theorem 1, $M(\alpha, D)$ is stable for α sufficiently small.

Observe that, in the statement of Theorem 1, the second condition does not depend on the specific choice of V. In fact, assume that $V_1, V_2 \in \mathbb{R}^{N\times(N-1)}$ are full column rank matrices such that $w^*V_1 = w^*V_2 = 0$. Notice that the columns of both V_1 and V_2 form a basis of orthonormal vectors of $\operatorname{span}\{w\}^\perp$ and so there exists an invertible matrix $T \in \mathbb{R}^{N-1\times N-1}$ such that $V_2 = V_1 T$. Since we have that $W_2^* = T^{-1}W_1^*$, we can conclude that $W_2^*DV_2 = T^{-1}(W_1^*DV_1)T$ an so $W_1^*DV_1$ and $W_2^*DV_2$ have the same eigenvalues.

Condition 2) may seem a bit involved and difficult to check. However, in many practical cases the condition is immediately satisfied, and does not need to be checked explicitly, as shown hereafter.

Corollary 1. *Consider the clock synchronization protocol presented in section III, and let the graph of admissible communication* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *be strongly connected. Let Assumptions 1, 2, and 3 hold. If* $\bar{E} = \bar{E}^*$*, then there exists a value* $\bar{\alpha} > 0$ *such that, for any* $\alpha \in (0, \bar{\alpha})$*, mean-square synchronization is achieved.*

The result of Corollary 1 is remarkable, since, provided that $\bar{E} = \bar{E}^*$, it ensures that existence of $\bar{\alpha} > 0$ for any matrix $D > 0$, i.e. for any difference between the frequencies of the clocks. Condition $\bar{E} = \bar{E}^*$ is clearly much simpler than condition 2) of Theorem 1, and it is for example satisfied in the following very common scenarios, as the examples in Section V will show:

- in the case of highly regular graphs like complete graphs or Cayley graphs [20], with a proper choice for the weights of the matrices $E \in \mathcal{M}$;
- in case of symmetric protocols like symmetric gossip [21], since at every iteration two nodes perform the exact same update, so that all the elements of $\mathcal{M}$ are symmetric matrices, regardless of the communication graph.

Remark. *If the clocks have the same frequency, i.e. $f_i = \bar{f} \in \mathbb{R}^+$, for all $i \in \mathcal{V}$, then condition* 2) *of Theorem 1 is automatically verified, for any (possibly asymmetric) communication protocol. Because mean-square synchronization of the system depends continuously on the clock frequencies, as shown in the proof of Theorem 1, then Condition* 2) *remains automatically verified if the clock frequencies f_i are sufficiently similar.*

Moreover, any value of α that achieves synchronization in the case of equal clock frequencies will also achieve synchronization when clock frequencies f_i are sufficiently similar.

For certain specific scenarios, it is possible to compute or to estimate the value of the upper bound $\bar{\alpha}$ which ensures synchronization. This is the approach followed by the authors in [22], where they considered a very similar model for the clocks and a specialization of the algorithm based on asymmetric gossip communication. We perform a similar analysis in the next section, for some other specific scenarios. In some cases, such analysis is only possible assuming that all clock frequencies are identical. As discussed in the remark, and illustrated in the next section, the upper bound $\bar{\alpha}$ computed under this assumption, and evaluated assuming that all frequencies are equal to the nominal frequency, is still meaningful when small perturbations of the clock frequencies are expected.

V. Examples

In this section, we study three specialized settings of the general RandSync algorithm presented in Section III, for which we also provide some possible practical applications. For each of them we illustrate how the convergence results can be used and how the proposed algorithm behaves. In particular we consider implementations of the algorithm based on

- **broadcast communication** (RandSync-Bcast), for which we can compute the threshold $\bar{\alpha}$ analytically (for a complete communication graph), and for which we validate the algorithm on a real experimental testbed;
- **gossip communication** (RandSync-Gossip), for which, similarly to RandSync-Bcast algorithm, we can compute the threshold $\bar{\alpha}$ analytically (for a complete communication graph), and for which we illustrate the effectiveness of the algorithm via numerical simulations on random geometric graphs;
- **hierarchical communication** (RandSync-Tree), for which we compute the threshold $\bar{\alpha}$ analytically, and we validate such bound via simulations.

A. RandSync-Bcast *in wireless sensor networks*

Consider a network of wireless agents. Each agent i activates according to a independent Poisson processes with intensity λ (i.e. with exponentially distributed waiting times between subsequent activations). As described in Example 1, when an agent i activates, it broadcasts its time reading $\hat{t}_i$ to all its neighbors in the communication graph $\mathcal{G}$ (see Figure 2) which update their states according to (6). The update matrix $E^{(i)}$, corresponding to the activation of node i, is then

$$E^{(i)} = q \sum_{j \in \mathcal{N}_i} (e_j e_i^* - e_j e_j^*) = \left[\begin{array}{ccc} & & \\ +q & -q & \\ +q & & -q \end{array}\right] \begin{array}{l} \leftarrow \text{row } i \\ \leftarrow \text{row } j_1 \\ \leftarrow \text{row } j_2 \end{array}$$

where $e_i \in \mathbb{R}^N$ is the i-th vector of the canonical base and where j_1, j_2 are two neighbors of i. The strategy is asymmetric, since only the agents that receive the time reading from node i update their state, while agent i holds its state constant. Therefore node i does not need to gather any information from its neighbors, and the communication happens only in one direction (as indicated by the red thick arrows in Figure 2).

This protocol is very easy to implement, because of the inherent broadcast nature of the wireless channel. As no response is required from the nodes, the transmitting node i does not need to identify itself or its neighbors, and media access control is extremely simple.

Notice that, since the transmission times of each agent are independent one each other, then $\{T_{\text{up}}(h)\}_{h \in \mathbb{N}}$ are the samples of a Poisson process of intensity $N\lambda$, with inter-time moments

$$\begin{cases} \mu = \mathbb{E}\left[\delta_{\text{up}}(h)\right] = \frac{1}{\lambda N}, \\ \sigma^2 = \mathbb{E}\left[\delta_{\text{up}}^2(h)\right] = \frac{2}{N^2 \lambda^2} \end{cases}$$

Assumptions 2 and 3 are therefore verified. For the specific case in which the communication graph is the complete graph, and therefore all pairs of nodes can communicate, it is easy to check that $\bar{E}$ is a symmetric matrix and equals to

$$\bar{E} = -q(I - \frac{1}{N}\mathbf{1}\mathbf{1}^*).$$

Corollary 1 therefore applies, and thus there exists a positive constant $\bar{\alpha}$ such that mean-square convergence of the algorithm RandSync-Bcast is guaranteed for any value of the parameter α smaller than $\bar{\alpha}$. In the specific case of a complete graph and when the clocks are all equal, we can also explicitly determine the maximum value $\bar{\alpha}$, as the following proposition states.

Proposition 2. *Consider a network of clocks with a complete graph of admissible communications $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume that the clocks are all equal with $f_i = f_0$. Assume that they activate according to Poisson processes with the same intensities, and that they implement the* RandSync-Bcast *synchronization protocol. Then mean-square synchronization is ensured if*

$$0 < \alpha < \frac{\lambda N (2 - q)}{f_0}.$$

As discussed at the end of Section IV, the interval for α derived in Proposition 2 remains meaningful if the true clock frequencies (which are unknown) are sufficiently close to the nominal frequency.

Nominal frequency	f_0	32768 Hz
Frequency deviation	$\mathrm{stdev}(f_i)$	±0.15 Hz
Relative frequency deviation	$\frac{\mathrm{stdev}(f_i)}{f_0}$	4.6 ppm

TABLE I
CLOCK PARAMETERS IN THE EXPERIMENTAL SETUP

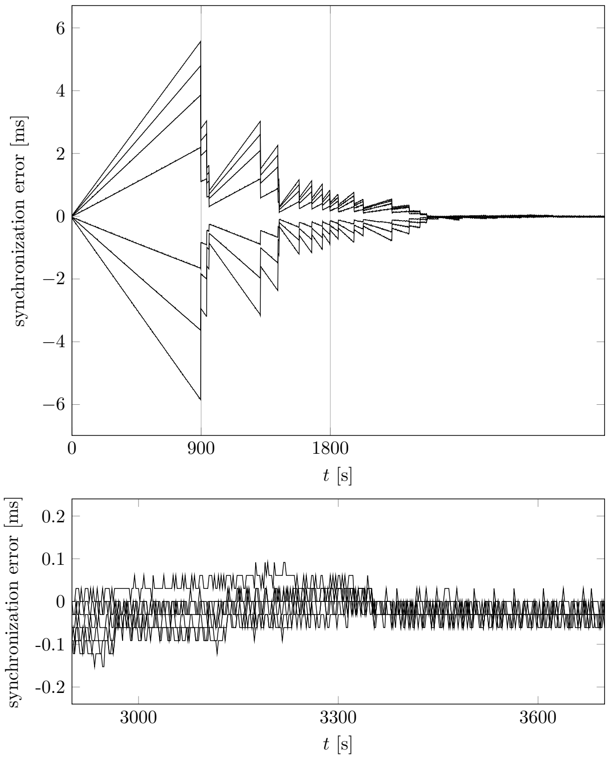

Fig. 5. Time difference between 7 among 20 nodes, compared to a reference node, over a time span of more than one hour. The vertical line at 900 s indicates the time instant when the RANDSYNC-BCAST synchronization algorithm has been activated with $\alpha = 0$ and the vertical line at 1800 s indicates the time instant when the skew correction algorithm ($\alpha = 9.8 \times 10^{-4}$) has been activated. The steady state behavior is enlarged in the bottom panel.

The proposed synchronization algorithm has been implemented and validated on the FIT IoT-LAB/SensLAB testbed [23], [24]. SensLAB is a very large scale open wireless sensor network platform which allows fast prototyping of distributed algorithms via a convenient infrastructure for code deployment and node monitoring. Each node of the testbed is based on a low power MSP430-based platform [25] (equipped with off-the-shelf 16-bit micro-controllers) with a IEEE 802.15.4 radio interface.

The on-board 32 KHz quartz oscillator has been used for driving the time counter. This choice corresponds to a granularity of 30 μs. Skew corrections have been implemented by periodically increasing (or decreasing) the time counter by 1 tick. By adjusting the period of this correction, proper skew correction can be achieved by the algorithm.

Wireless communication has been implemented according to a CSMA (Carrier Sense Multiple Access) protocol, in which however re-transmission and back-off strategies have been disabled for the specific time synchronization packets in order to avoid random transmission delays. Packets are timestamped before being forwarded to the radio interface transmission buffer and as soon as they are received. This way, a constant deterministic delay has been observed, and has been compensated in the algorithm. Notice that this deterministic delay can be easily estimated by the nodes by exchanging few packets any by comparing the timestamp differences.

For this experimental validation, 20 nodes have been used among the 256 nodes available in the testbed. The communication graph between these nodes resulted to be a complete graph, with limited packet losses. The source code used in these experiments is available online for review [26].

Each node triggered its own broadcast transmission with random, exponentially distributed, waiting times, with an average waiting time of 2048 s (about 34 min). The energy consumption and the communication burden for each node is therefore extremely small.

The experiment consisted in three phases (see Figure 5).

- In the first 900 s (15 min), no synchronization algorithm has been implemented. This stage allowed us to evaluate the clock skew differences among the nodes, which can be as large as 13 ppm.
- At time equals to 900 s, the proposed time synchronization algorithm has been activated with $\alpha = 0$ and $q = 1/2$. This way, the time synchronization error has been corrected by adjusting the clock offsets at each algorithm execution, but leaving the clock skews untouched. It is clear that this approach can keep the time synchronization error bounded, but cannot drive it to zero. Experimental results show that, with the adopted intensity for the triggering process, the error seems to remain in the order of magnitude of 1 ms.
- Finally, at time equals to 1800 s (30 min), the parameter α of the synchronization algorithm has been set to $2^{-9}/f_0 \approx 1.9 \times 10^{-3}/f_0$, which is smaller than the threshold $\bar{\alpha}$ computed according to Proposition 2. It is clear that in the following 15 minutes the algorithm is capable of correcting the skew differences among the nodes, driving the synchronization error to zero. After this transient, the clocks of the different nodes result to be synchronized up to the time granularity of the devices, as the bottom panel of Figure 5 shows.

It is interesting to notice that the performance of RANDSYNC-BCAST are comparable, in terms of steady state accuracy of the time synchronization, with the experimental results presented in [9], where two other methods based on broadcast wireless communication have been considered and implemented: the *Average TimeSynch* (ATS) algorithm [9] and the *Flooding Time Synchronization Protocol* (FTSP) [2]. The testbed considered in [9] is extremely similar to the one presented in this paper. However, the computational and communication burden of RANDSYNC-BCAST is remarkably smaller than both ATS and FTSP. Indeed, in the implementations reported in [9], nodes transmit their time readings to other nodes every minute, while in our experimental validation the expected waiting time between subsequent communications of the same node is about 34 minutes. Some extra comments about this comparison are provided in [27], where the proposed algorithm is shown to be more robust against disturbances, compared to ATS, and much simpler to implement.

B. RandSync-Gossip *in wireless sensor networks*

We consider again a wireless network, as in the previous example. In this case, however, we adopt a gossip (peer-to-peer) communication strategy, as described in Example 2. According to such strategy, when node i activates (according to its own independent Poisson process), it sends its time reading to node j, one node randomly chosen among its neighbors and itself (see Figure 3). When node j receives the message $\hat{t}_i$, it computes the difference $\hat{t}_i - \hat{t}_j$, and sends it back to node i. Then, nodes i and j update their states according to (7) and (8). Both Assumptions 2 and 3 are easily verified. In this case the strategy is symmetric: node i and node j, after a gossip-like exchange of information, execute a symmetric update of their parameters. The update matrix corresponding to the update of nodes i and j is then

$$E^{(ij)} = -q(e_i - e_j)(e_i - e_j)^* = \begin{bmatrix} -q & q \\ q & -q \end{bmatrix} \begin{array}{l} \leftarrow \text{row } i \\ \leftarrow \text{row } j \end{array}$$

Notice that, as the matrix $E^{(ij)}$ is symmetric for any pair i, j, the matrix $\bar{E}$ is necessarily symmetric (for any underlying communication graph $\mathcal{G}$, and for any probability distribution $p_{\mathcal{M}}$). Corollary 1 then applies, and the algorithm RandSync-Gossip is guaranteed to achieve mean-square synchronization if the parameter α is smaller than some positive threshold $\bar{\alpha}$. Again, in the case of a complete graph and when the clocks are all equal, we can determine the maximum value of $\bar{\alpha}$. To state our result we need an additional assumption related to the initial condition of the algorithm. Let

$$\Sigma(h) = \mathbb{E}\left[x(h)x(h)^*\right] := \begin{bmatrix} \Sigma_{11}(h) & \Sigma_{12}(h) \\ \Sigma_{21}(h) & \Sigma_{22}(h) \end{bmatrix}$$

where $\Sigma_{rs}(h) \in \mathbb{R}^{N \times N}$, $r, s = 1, 2$. We assume the following property.

Assumption 4. *The matrices $\Sigma_{rs}(0)$, $r, s = 1, 2$, have the following structure*

$$\Sigma_{rs}(0) = \alpha_{rs} I + \beta_{rs} \mathbf{1}\mathbf{1}^*,$$

where α_{rs}, β_{rs}, $r, s = 1, 2$ are non-negative real numbers.

This structure, which we shall prove to be positively invariant for the system, is easily implementable. As an example, the clocks could initiate the $\hat{t}_i$'s to independent random variables, and the k_i's to some common deterministic value, such as the reasonable $k_i(0) = 1/f_0$ for all $i \in \mathcal{V}$.

We have the following result.

Proposition 3. *Consider a network of clocks with a complete graph of admissible communications $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume that the clocks are all equal with $f_i = 1$. Assume that they activate according to Poisson processes with the same intensities, and that they implement the* RandSync-Gossip *synchronization protocol. Then mean-square synchronization is ensured if*

$$0 < \alpha < \frac{\lambda}{f_0}.$$

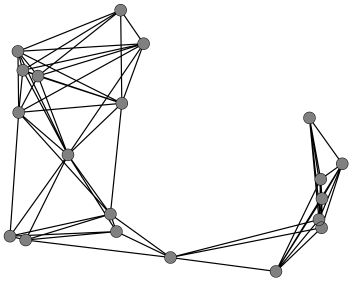

Fig. 6. The undirected random geometric graph with $N = 20$ nodes considered for the numerical simulation of RandSync-Gossip.

As discussed at the end of Section IV, the interval for α derived in Proposition 3 remains meaningful if the true clock frequencies (which are unknown) are sufficiently close to the nominal frequency.

The proposed algorithm has been simulated for a network of $N = 20$ clocks communicating through the edge of a random geometric graph. The graph is built as follows: The coordinates of the nodes are picked randomly in $[0, 1]^2$, and two nodes communicate if their Euclidean distance is less than $r = 3 \log N/N$, a choice that is known to guarantee connectedness with high probability. One realization of the resulting graph is illustrated in Figure 6.

We set $f_0 = 1$ to be the nominal frequencies of the clocks, while the true frequency of each clock, say the i-th, is given by $f_i = f_0 d_i$ for $d_i = \max\{0.1, 1 + \epsilon_i\}$, $\epsilon_i \sim \mathcal{N}(0, 1)$. This results in 20 different clock frequencies ranging from $f_{\text{min}} = 0.1 f_0$ to $f_{\text{max}} = 2.53 f_0$, hence with differences up to 150% of the nominal frequency. We also set for simplicity $\lambda = 1$, and following Proposition 3, we chose $\alpha = 0.9 \frac{\lambda}{f_0} = 0.9$ and $q = 1/2$.

We run 20 simulations. For each simulation, we picked randomly the initial offset for any agent i as $\hat{t}_i(0) \sim \mathcal{N}(0, 1)$, while we set $k_i(0) = 1$, i.e., all clocks start with the same estimate of estimate of their frequency. The results on these simulations are illustrated in Figure 7, where we show a typical trajectory of the time readings and the error decay for the 20 simulations. For the latter, in particular, for each of the 20 simulations we plot in log-scale the process

$$e(t) = ||\Omega \hat{t}(t)||_2$$

where $\Omega = I_N - \frac{1}{N}\mathbf{1}_N \mathbf{1}_N^T$ is a projector onto the orthogonal to $\mathbf{1}_N$. As one can see, clocks synchronize to the same ramp-shaped function essentially exponentially in time.

C. RandSync-Tree *in wired data networks*

As a third example, we consider a typical wired data network divided into nested subnetworks. The nodes belonging to each subnetwork can communicate with the root node of their subnetwork (the *router*), and each router is a node of a larger,

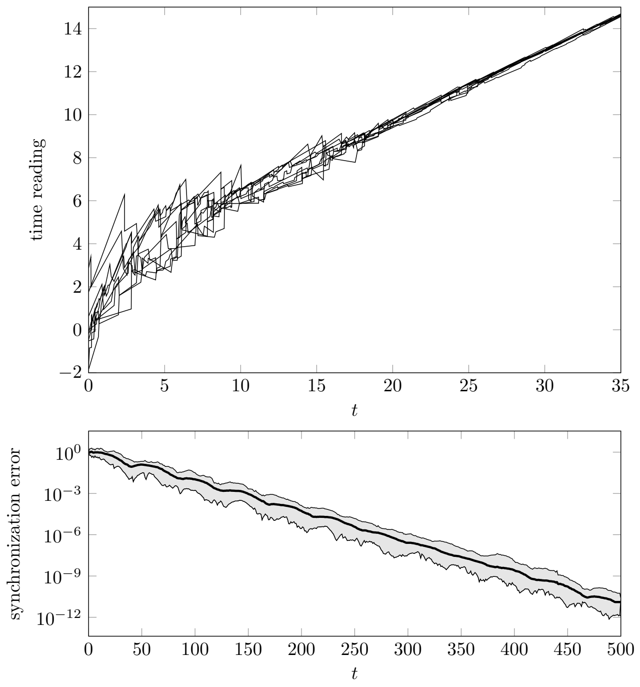

Fig. 7. Results of the numerical simulations of RandSync-Gossip. On the left, a typical trajectory of the time readings, asymptotically approaching the same ramp-shaped function. On the right, the normalized error $e(t) = ||\Omega\hat{t}(t)||_2/||\Omega\hat{t}(0)||_2$ in log-scale (minimum, maximum, and average value at every time, for 20 simulations).

higher subnetwork, recursively. We assume that each router is provided with a time server, and agents can request the time reading of the router of their subnetwork, as described in Example 3. Similar tree-like communication architectures have been exploited in numerous clock synchronization algorithms, and such hierarchical structure (which is inherent in most wired data networks) is also adopted in the standard Network Time Protocol (NTP) which is currently used over the Internet [17].

Once a router j receives a request from an agent i in its subnetwork, it responds with the time reading $\hat{t}_j$ (see Figure 4). Node i then computes the difference $\hat{t}_j - \hat{t}_i$, and updates its own states according to (9). Such update result in an asymmetric update matrix

$$E^{(i)} = e_i(e_j^* - e_i^*) = \left[\begin{array}{cc} -1 & +1 \\ & \end{array}\right] \begin{array}{l} \leftarrow \text{row } i \\ \leftarrow \text{row } j \end{array}$$

Notice that the row sum of the off-diagonal elements of E is not strictly smaller than 1, as the algorithm would require. It is however possible to show that the results of Section IV hold also in this case, and, in general, in all the cases in which $\sum_{j \neq i} E_{ij} \leq 1, \forall i \in \mathcal{V}$, provided the graph $\mathcal{G}$ of the admissible communications is acyclic. Again it would follow from Theorem 1, the existence of an $\bar{\alpha} > 0$ such that, if $\alpha < \bar{\alpha}$ then the RandSync-Tree algorithm reaches the mean-square synchronization. However in this hierarchical architecture, it is possible to provide a more refined and complete analysis which is stated in the following Proposition. We assume that all the nodes, except for the root, activate according to independent Poisson processes with the same intensity λ. Furthermore, without loss of generality, we assume that the root is node 1. Then we have the following result.

Proposition 4. *Consider a network of clocks whose graph of admissible communications $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed tree. Assume that all the nodes, except for the root (node 1), activate according to independent Poisson processes with the same intensity λ, they retrieve the time reading from their parent node, and they update their clock parameters according to the* RandSync-Tree *algorithm described in* (9). *Let $f_{max} = \max\{f_i \,|\, i \in V \setminus \{1\}\}$. Then mean-square synchronization is ensured if*

$$0 < \alpha < \frac{\lambda}{f_{max}}. \tag{15}$$

We simulated the RandSync-Tree algorithm on a network with $N = 21$ nodes. The communication graph is a (k, l) regular tree in which every node (except the leaves) has $k = 4$ children, and the depth of the tree is $l = 2$.

We set once again $f_0 = 1$ to be the nominal frequencies of the clocks, while the true frequency of each clock, say the i-th, is given by $f_i = f_0 d_i$ for $d_i = 1 + \epsilon_i$, $\epsilon_i \sim \mathcal{U}[-.3, .3]$, thus $f_{\max} \leq 1.3$. We also set for simplicity $\lambda = 1$, and we pick the initial conditions of the states of the clocks according to $\hat{t}_i(0) \in \mathcal{N}(0, 25)$ and $k_i(0) \in \mathcal{U}(0, 10)$, for all $i \in \mathcal{V} \setminus \{1\}$.

With these parameters, Proposition 4 ensures synchronization if the parameter α is smaller than $\bar{\alpha} = 1/1.3 \approx 0.77$.

We consider three cases, with different values for α. In particular, we choose $\alpha = 0.1$, $\alpha = 0.5$ and $\alpha = 1.5$. The simulated synchronization errors of the agents with respect to the root of the tree are shown in Figure 8. As expected, the system achieves synchronization to the root, for both $\alpha = 0.1$ and $\alpha = 0.5$. The parameter α can be interpreted as a control gain for the class of synchronization algorithms presented in this paper. As such, a fundamental tradeoff emerges between smoother but slower convergence (lower values of α) and faster convergence with larger transients (higher values of α). Depending on the application, a different tradeoff may be sought. When $\alpha = 1.5$, the system does not synchronize in the mean-square sense, but an interesting phenomenon occurs, which resembles what has been already observed in the case of noisy consensus with packet drop and is related to the so called Levy flights [28]. Indeed, even if clocks tend to drift apart, synchronization is achieved abruptly at seemingly random times. Intuitively, this happens because for any $\varepsilon > 0$ there exist (with non-zero probability) a time t' such that $|\hat{t}_i(t') - \hat{t}_1(t')| < \varepsilon$, for all $i > 1$, and this is related to fast sequences of activations of the nodes from the root to the leaves of the tree. Such an effect is not described by the proposed mean-square analysis of the system, and is related to the multiplicative effect of the random matrices $E(h)$. Therefore, even if the system does not achieve mean square synchronization, there exist instants at which it is synchronized with arbitrary precision, these times possibly being very far away in the absolute time. Of course, such a notion of synchronization is useless for real-world application.

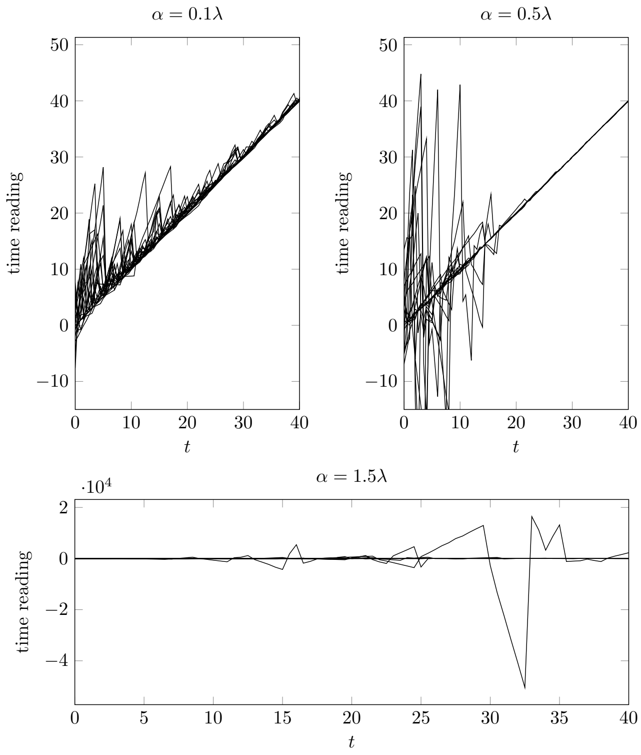

Fig. 8. Simulated clock synchronization errors, for the case of the hierarchical protocol RANDSYNC-TREE, for different values of α. In the third panel, for $\alpha = 1.5\lambda$, only the few largest trajectories can be seen, due to scaling.

VI. CONCLUSIONS

This paper studies randomized clock synchronization PI-controllers in multi-agent networks, establishing the existence of tuning parameters for which the network achieves synchronization in the mean-square sense. Several instances of randomized algorithms are presented and discussed in light of the theoretical results, and their effectiveness is studied in real and simulation testbeds. Future research directions include and are not limited to a more thorough characterization of the interval in which the tuning parameter can lie to ensure synchronization in the mean-square sense, in particular with respect to the topology of the network and well-known geometrical and graph-theoretical parameters, such as the dimension in which the network is embedded and the second largest eigenvalue of the average communication graph.

REFERENCES

[1] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timingsync protocol for sensor networks," in *Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2003.

[2] M. Maròti, B. Kusy, G. Simon, and Àkos Lèdeczi, "The flooding time synchronization protocol," in *Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2004.

[3] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu, "Tiny-sync: tight time synchronization for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, pp. 1–34, 2007.

[4] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Symp. on Operating Systems Design and Implementation (OSDI'02)*, 2002.

[5] K. L. Noh, E. Serpedin, and K. Qaraqe, "A new approach for time sycnhronization in wireless sensor networks: Pairwise broadcast synchronization," *IEEE Trans. on Wireless Communications*, vol. 7, no. 9, pp. 3318–3322, 2008.

[6] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, 2005.

[7] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP J. on Wireless Sensor Networks*, 2007.

[8] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *IEEE Conf. on Decision and Control (CDC)*, 2006.

[9] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[10] P. Sommer and R. Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," in *Intl. Symp. on Information Processing in Sensor Networks (IPSN)*, 2009.

[11] K. S. Yildirim and A. Kantarci, "External gradient time synchronization in wireless sensor networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 633–641, 2014.

[12] M. Leng and Y. C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Trans. on Signal Processing*, vol. 59, no. 11, pp. 1504–1514, 2011.

[13] B. Luo and Y. C. Wu, "Distributed clock parameters tracking in wireless sensor network," *IEEE Trans. on Wireless Communications*, vol. 12, no. 12, pp. 6464–6475, 2013.

[14] J. Du and Y. C. Wu, "Distributed clock skew and offset estimation in wireless sensor networks: asynchronous algorithm and convergence analysis," *IEEE Trans. on Wireless Communications*, vol. 12, no. 11, pp. 5908–5917, 2013.

[15] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Optimal synchronization for networks of noisy double integrators," *IEEE Trans. on Automatic Control*, vol. 56, no. 5, pp. 1146–1152, 2011.

[16] R. Carli and S. Zampieri, "Network clock synchronization based on the second order linear consensus algorithm," *IEEE Trans. on Automatic Control*, vol. 59, no. 2, pp. 409–422, 2014.

[17] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[18] R. L. Scheiterer, C. Na, D. Obradovic, and G. Steindl, "Synchronization performance of the precision time protocol in industrial automation networks," *IEEE Trans. on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1849–1857, 2009.

[19] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.

[20] L. Babai, "Spectra of cayley graphs," *J. of Combinatorial Theory, Series B.*, pp. 27:180–189, 1979.

[21] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2508 – 2530, 2006.

[22] R. Carli, E. D'Elia, and S. Zampieri., "A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks," in *IEEE Conf. on Decision and Control (CDC)*, 2011.

[23] "FIT IoT-LAB." [Online]. Available: http://www.iot-lab.info

[24] C. Burin des Roziers, G. Chelius, T. Ducrocq, E. Fleury, A. Fraboulet, A. Gallais, N. Mitton, T. Noël, and J. Vandaele, "Using SensLAB as a first class scientific tool for large scale wireless sensor network experiments," in *NETWORKING 2011*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6640, pp. 147–159.

[25] Texas Instruments. MSP430 technical documentstion. [Online]. Available: http://www.ti.com/msp430

[26] S. Bolognani, "Randsync source code," 2014. [Online]. Available: https://github.com/saveriob/RandSync

[27] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri, "A randomized linear algorithm for clock synchronization in multi-agent systems," in *IEEE Conf. on Decision and Control (CDC)*, 2012.

[28] J. Wang and N. Elia, "Distributed averaging under constraints on information exchange: Emergence of lévy flights," *IEEE Trans. on Automatic Control*, vol. 57, no. 10, pp. 2435–2449, 2012.

[29] F. Gantmacher, *The Theory of Matrices*. Chelsea Publishing Company, 1959.

[30] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. on Selected Areas in Communications*, vol. 26, pp. 634–649, 2008.

Appendix

A. Proof of Proposition 1

Since (12) is a sampled version of the original switched system, one direction of the equivalence is obvious. On the other hand, notice first that, from (12), mean-square synchronization of the sampled systems also implies that

$$\mathbb{E}\left[(f_i k_i(T_{\text{up}}(h)) - \gamma)^2\right] \overset{h\to\infty}{\longrightarrow} 0.$$

Let $t \in (T_{\text{up}}(h), T_{\text{up}}(h+1))$ and observe that

$$\begin{aligned}\hat{t}_i(t) - (\gamma t + \beta) = {} & \hat{t}_i(T_{\text{up}}(h)^+) - (\gamma T_{\text{up}}(h) + \beta) \\ & + \left(f_i k_i(T_{\text{up}}(h)^+) - \gamma\right)(t - T_{\text{up}}(h)).\end{aligned}$$

It follows that

$$\begin{aligned}&\mathbb{E}\left[\left(\hat{t}_i(t) - (\gamma t + \beta)\right)^2\right] \\ &\quad = \mathbb{E}\left[\left(\hat{t}_i(T_{\text{up}}(h)^+) - (\gamma T_{\text{up}}(h) + \beta)\right.\right. \\ &\qquad \left.\left. + \left(f_i k_i(T_{\text{up}}(h)^+) - \gamma\right)(t - T_{\text{up}}(h))\right)^2\right] \\ &\quad \leq \mathbb{E}\left[\left(\hat{t}_i(T_{\text{up}}(h)^+) - (\gamma T_{\text{up}}(h) + \beta)\right)^2\right] \\ &\qquad + \mathbb{E}\left[\left(f_i k_i(T_{\text{up}}(h)^+) - \gamma\right)^2\right]\mathbb{E}\left[(t - T_{\text{up}}(h)))^2\right] \\ &\qquad + 2\sqrt{\mathbb{E}\left[\left(\hat{t}_i(T_{\text{up}}(h)^+) - (\gamma T_{\text{up}}(h) + \beta)\right)^2\right]} \times \\ &\qquad \times \sqrt{\mathbb{E}\left[(f_i k_i(T_{\text{up}}(h)^+) - \gamma)^2\right]\mathbb{E}\left[(t - T_{\text{up}}(h)))^2\right]}\end{aligned}$$

where the inequality has been obtained by applying the Cauchy-Schwarz inequality. Since by hypothesis we have that $\mathbb{E}\left[\left(\hat{t}_i(T_{\text{up}}(h)^+) - (\gamma T_{\text{up}}(h) + \beta)\right)^2\right] \overset{h\to\infty}{\longrightarrow} 0$, $\mathbb{E}\left[(f_i k_i(T_{\text{up}}(h)^+) - \gamma)^2\right] \overset{h\to\infty}{\longrightarrow} 0$ and $\mathbb{E}\left[(t - T_{\text{up}}(h)))^2\right] \leq \sigma^2$ we can conclude that also

$$\mathbb{E}\left[\left(\hat{t}_i(t) - (\gamma t + \beta)\right)^2\right] \overset{h\to\infty}{\longrightarrow} 0.$$

B. Proof of Theorem 1

Let $x'(h), x''(h) \in \mathbb{R}^N$ be such that $x(h) = \begin{bmatrix} x'(h) \\ x''(h) \end{bmatrix}$. Let moreover $w \in \mathbb{R}^N$ be such that $w^*\bar{E} = 0$ and $w^*\mathbf{1} = 1$. If $V \in \mathbb{R}^{N\times(N-1)}$ is a full column rank matrix such that $w^*V = 0$, then we have that $[\mathbf{1} \quad V]$ is invertible and that $\begin{bmatrix}\mathbf{1} & V\end{bmatrix}^{-1}\bar{E}\begin{bmatrix}\mathbf{1} & V\end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{E}\end{bmatrix}$ and that $\begin{bmatrix}\mathbf{1} & V\end{bmatrix}^{-1} = \begin{bmatrix} w^* \\ W^*\end{bmatrix}$ for some $W \in \mathbb{R}^{N\times(N-1)}$. Therefore we have that $W^*\mathbf{1} = 0$, $W^*V = I_{N-1}$, $VW^* + \mathbf{1}w^* = I_N$ and $\tilde{E} = W^*\bar{E}V$. It is convenient to introduce the quantities $y(h), z(h) \in \mathbb{R}^{N-1}$ defined as

$$y(h) := W^*x'(h) \qquad z(h) := W^*Dx''(h). \tag{16}$$

In words, $y(h)$ is an error vector which is zero only if $x'(h)$ belongs to $\text{span}\{\mathbf{1}\}$, namely if the clocks are synchronized. Analogously, $z(h)$ is zero only if $x''(h)$ belongs to $\text{span}\{D^{-1}\mathbf{1}\}$, namely if all the clocks increase their time readings with the same slope, i.e., $d_1k_1(h) = \cdots = d_Nk_N(h)$.

Since we are interested in mean-square synchronization to a ramp, we introduce the mean trajectory $m(h) = \mathbb{E}[x(h)]$ and the matrix

$$\Sigma(h) = \mathbb{E}\left[\begin{bmatrix} y(h) \\ z(h)\end{bmatrix}\begin{bmatrix} y(h)^* & z(h)^*\end{bmatrix}\right] = \begin{bmatrix}\Sigma_{yy}(h) & \Sigma_{yz}(h) \\ \Sigma_{yz}(h)^* & \Sigma_{zz}(h)\end{bmatrix}$$

where $\Sigma_{yy}(h) := \mathbb{E}[y(h)y(h)^*]$, $\Sigma_{yz}(h) := \mathbb{E}[y(h)z(h)^*]$ and $\Sigma_{zz}(h) := \mathbb{E}[z(h)z(h)^*]$.

From (12), by statistical independence of the samples of the processes $\{E(h)\}_{h\in\mathbb{N}}$ and $\{\delta_{\text{up}}(h)\}_{h\in\mathbb{N}}$, we obtain the linear evolution

$$m(h+1) = \begin{bmatrix} I + \bar{E} + \alpha f_0 \mu D\bar{E} & \mu D \\ \alpha f_0 \bar{E} & I\end{bmatrix} m(h) = \bar{M}m(h)\,. \tag{17}$$

Let $v_1 = \begin{bmatrix}\mathbf{1} \\ 0\end{bmatrix}$ and $v_2 = \begin{bmatrix} 0 \\ \frac{1}{\mu}D^{-1}\mathbf{1}\end{bmatrix}$. It is straightforward to verify that $(\bar{M} - I)v_1 = 0$, $(\bar{M} - I)v_2 = v_1$, and that $(\bar{M} - I)v = v_2$ has no solution, namely, v_1 and v_2 are the eigenvector and the generalized eigenvector, respectively, of the matrix $\bar{M}$ associated with the eigenvalue 1, and that the multiplicity of 1 is exactly 2. If the other eigenvalues of $\bar{M}$ are strictly inside the unit circle, then $m(h) \to \begin{bmatrix}\beta\mathbf{1} + \gamma h\mathbf{1} \\ \gamma\frac{1}{\mu}D^{-1}\mathbf{1}\end{bmatrix}$ for some $\gamma \in \mathbb{R}$ and $\beta \in \mathbb{R}$. In particular, $\mathbb{E}[x'(h)] \to \beta\mathbf{1} + \gamma h\mathbf{1}$. Notice that by Frobenius-Perron theorem w^* is a vector whose entries are non negative. Moreover, using (17), direct computation shows that $\begin{bmatrix} 0 & w^*\end{bmatrix} m(h) = \begin{bmatrix} 0 & w^*\end{bmatrix} m(0)$ for all $h \geq 0$, and thus in particular $\gamma\frac{1}{\mu}w^*D^{-1}\mathbf{1} = w^*\mathbb{E}x''(0)$. Under the assumption that all the clock initialize their value k_i to a strictly positive value, this proves that $\gamma > 0$.

The network achieves thus mean-square synchronization to a ramp if $\bar{M}$ has only stable eigenvalues, except 1 with multiplicity 2, and if $\Sigma(h) \overset{h\to\infty}{\longrightarrow} 0$. Notice now that

$$\begin{bmatrix} w^* \\ W^*\end{bmatrix} E(h)\begin{bmatrix}\mathbf{1} & V\end{bmatrix} = \begin{bmatrix} 0 & * \\ 0 & \tilde{E}(h)\end{bmatrix}$$

$$\begin{bmatrix} w^* \\ W^*\end{bmatrix} DE(h)\begin{bmatrix}\mathbf{1} & V\end{bmatrix} = \begin{bmatrix} 0 & * \\ 0 & \tilde{F}(h)\end{bmatrix}$$

The latter and the definitions (16) in the discrete time system (12) yield the update law of the signals $y(h)$ and $z(h)$

$$\begin{bmatrix} y(h+1) \\ z(h+1)\end{bmatrix} = \begin{bmatrix} I + \tilde{E}(h) + \alpha f_0 \delta_{\text{up}}(h)\tilde{F}(h) & \delta_{\text{up}}(h)I \\ \alpha f_0 \tilde{F}(h) & I\end{bmatrix}\begin{bmatrix} y(h) \\ z(h)\end{bmatrix}$$

where $\tilde{E}(h) = W^*E(h)V$ and $\tilde{F}(h) = W^*DE(h)V$.

We start with the analysis in the second moment. By definition of $\Sigma(h)$ and by statistical independence of the samples of the processes $\{E(h)\}_{h\in\mathbb{N}}$ and $\{\delta_{\text{up}}(h)\}_{h\in\mathbb{N}}$, we obtain the linear time–invariant system

$$\Sigma(h+1) = \mathbb{E}[A(h)\Sigma(h)A(h)^*] \tag{18}$$

where

$$\begin{aligned} A(h) &:= \begin{bmatrix} A_{11}(h) & A_{12}(h) \\ A_{21}(h) & A_{22}(h)\end{bmatrix} \\ &= \begin{bmatrix} I + \tilde{E}(h) + \alpha f_0 \delta_{\text{up}}(h)\tilde{F}(h) & \delta_{\text{up}}(h)I \\ \alpha f_0 \tilde{F}(h) & I\end{bmatrix}\end{aligned}$$

Define now $Y(h) := \text{vec}(\Sigma_{yy}(h))$, $W(h) := \text{vec}(\Sigma_{yz}(h))$, $W'(h) := \text{vec}(\Sigma_{yz}(h)^*)$ and $Z(h) := \text{vec}(\Sigma_{zz}(h))$, where vec means the vectorization operator, namely the operator which maps a matrix into a column vector by stacking the columns of the matrix. Notice that $\Sigma(h) \overset{h\to\infty}{\longrightarrow} 0$ is equivalent to $Y(h), W(h), W'(h), Z(h) \overset{h\to\infty}{\longrightarrow} 0$, hence we study the evolution of these vectors.

Using the standard property that $\mathrm{vec}(ABC) = (C^* \otimes A)\mathrm{vec}(B)$ it is possible to show that

$$\begin{aligned} &Y(h+1)\\ &\quad = \mathbb{E}\big[A_{11}(h)\otimes A_{11}(h)\big]Y(h) + \mathbb{E}\big[A_{12}(h)\otimes A_{11}(h)\big]W(h)\\ &\qquad + \mathbb{E}\big[A_{11}(h)\otimes A_{12}(h)\big]W'(h) + \mathbb{E}\big[A_{12}(h)\otimes A_{12}(h)\big]Z(h). \end{aligned}$$

Through similar computations we obtain

$$\begin{bmatrix} Y(h+1)\\ W(h+1)\\ W'(h+1)\\ Z(h+1)\end{bmatrix} = M \begin{bmatrix} Y(h)\\ W(h)\\ W'(h)\\ Z(h)\end{bmatrix} \tag{19}$$

where (here $A_{ij} = A_{ij}(h)$ for sake of notation)

$$M = \mathbb{E}\begin{bmatrix} A_{11}\otimes A_{11} & A_{11}\otimes A_{12} & A_{12}\otimes A_{11} & A_{12}\otimes A_{12}\\ A_{11}\otimes A_{21} & A_{11}\otimes A_{22} & A_{12}\otimes A_{21} & A_{12}\otimes A_{22}\\ A_{21}\otimes A_{11} & A_{21}\otimes A_{12} & A_{22}\otimes A_{11} & A_{22}\otimes A_{12}\\ A_{21}\otimes A_{21} & A_{21}\otimes A_{22} & A_{22}\otimes A_{21} & A_{22}\otimes A_{22}\end{bmatrix}$$

In what follows we will need to determine only some of the blocks of M. We first notice that M can be written in the form

$$M(\alpha) = M_0 + \alpha M_1 + \alpha^2 M_2$$

where

$$M_0 = \begin{bmatrix} \mathbb{E}[(I+\tilde{E}(h))\otimes(I+\tilde{E}(h))] & * & * & *\\ 0 & (I+\tilde{E})\otimes I & 0 & \mu I\otimes I\\ 0 & 0 & I\otimes(I+\tilde{E}) & \mu I\otimes I\\ 0 & 0 & 0 & I\otimes I\end{bmatrix}$$

and

$$M_1 = f_0\begin{bmatrix} * & * & * & *\\ * & * & * & *\\ * & * & * & *\\ 0 & \tilde{F}\otimes I & I\otimes\tilde{F} & 0\end{bmatrix}$$

while the computation of the matrix M_2 is not needed. Now we have to understand under what conditions $M(\alpha)$, is a stable matrix, namely if its eigenvalues are strictly inside the unit circle. We start with the following lemma describes the stability of M_0. Recall that an eigenvalue of a matrix is called semi–simple if its algebraic and geometric multiplicities coincide.

Lemma A.1. *Assume that $\mathcal{G}_{\bar{E}}$ contains a directed spanning tree and that Assumption 1 holds true. Then the matrix M_0 has $(N-1)^2$ eigenvalues 1 and the all other eigenvalues are stable, namely, strictly inside the unit circle. Moreover the eigenvalue 1 is semi–simple.*

Proof. From the upper-block-triangular structure of M_0, it is clear that the eigenvalues of M_0 coincide with the eigenvalues of its diagonal blocks. The last block has eigenvalues equal to 1. We will check that the other blocks have stable eigenvalues. Notice that $I\otimes I + \tilde{E}\otimes I = (I + \mathbb{E}[\tilde{E}(h)])\otimes I = W^*(I+\bar{E})V\otimes I$ and so its eigenvalues coincide with the eigenvalues of $W^*(I+\bar{E})V$. Notice now that $I+\bar{E}$ is a stochastic matrix and that the graph associated with it is strongly connected. Observe moreover that Assumption 1 implies that diagonal elements of $I+\bar{E}$ are non-zero. We can conclude [29] that $I+\bar{E}$ is primitive, which implies that it has one eigenvalue 1 with algebraic multiplicity one and the remaining which are strictly inside the unit circle. Now, $I+\bar{E}$ and $W^*(I+\bar{E})V$ have the same eigenvalues except the eigenvalue 1. Indeed, if $W^*(I+\bar{E})Vv = \nu v$ and $v\neq 0$, then multiplying by V on the left, and using $VW^* = I - \mathbf{1}w^*$, $w^*\bar{E} = 0$, and $w^*V = 0$, we obtain $(I+\bar{E})(Vv) = \nu(Vv)$. Since the columns of V are linearly independent, we correctly obtain that to any eigenvector of $W^*(I+\bar{E})V$ it corresponds an eigenvector of $(I+\bar{E})$ with the same eigenvalue, except $\mathbf{1}$. Indeed, if by contradiction $Vv = \gamma\mathbf{1}$, multiplication by W^* on the left immediately yields $\gamma\mathbf{1} = 0$, which is only possible if $v = 0$, contradicting $v\neq 0$. This allows to conclude that $W^*(I+\bar{E})V$ has eigenvalues strictly inside the unit circle. The matrix $I\otimes I + I\otimes\tilde{E}$ can be proved to be stable in the same way. Finally consider the first diagonal block. Notice that $\mathbb{E}[(I+\tilde{E}(h))\otimes(I+\tilde{E}(h))] = (W^*\otimes W^*)\mathbb{E}[(I+E(h))\otimes(I+E(h))](V\otimes V)$. Following Proposition 4.3 in [30] we have that $\mathbb{E}[(I+E(h))\otimes(I+E(h))]$ is a stochastic matrix and that the graph associated contains a directed spanning tree. Moreover it can be seen that Assumption 1 implies that diagonal elements of this matrix are non-zero. We can conclude also this matrix is primitive, which implies that it has one eigenvalue 1 with algebraic multiplicity one and the remaining which are strictly inside the unit circle. Similarly to the previous case, the eigenvalues of $(W^*\otimes W^*)\mathbb{E}[(I+E(h))\otimes(I+E(h))](V\otimes V)$ are a subset of those of $\mathbb{E}[(I+E(h))\otimes(I+E(h))]$ and do not include the eigenvalue 1, so we can conclude that $(W^*\otimes W^*)\mathbb{E}[(I+E(h))\otimes(I+E(h))](V\otimes V)$ has eigenvalues strictly inside the unit circle. Finally observe that, since $[0\ 0\ 0\ I]M_0 = [0\ 0\ 0\ I]$, we can conclude that the geometric multiplicity of the eigenvalue 1 is $(N-1)^2$. which implies that this eigenvalue is semi–simple. □

In order to prove Theorem 1, we need to use of the following perturbation result, taken from [19, Lemma 7].

Theorem A.1. *Let $M(\alpha)\in\mathbb{R}^{N\times N}$ be a matrix dependent on the parameter α in a sufficiently smooth way so that the first derivative $\dot{M}(\alpha)$ exists at $\alpha = 0$[3]. Assume that λ is a semi–simple eigenvalue of $M(0)$ having geometric multiplicity m. Let $r_1,\dots,r_m$ and $l_1,\dots,l_m$ be right and left eigenvectors relative to the eigenvalue λ such that, if $\mathcal{R} = \begin{bmatrix} r_1 & \dots & r_m\end{bmatrix}$ and $\mathcal{L} = \begin{bmatrix} l_1 & \dots & l_m\end{bmatrix}^*$ then $\mathcal{L}\mathcal{R} = I_m$, where I_m is the $m\times m$ identity. Then the derivatives of $\lambda(\alpha)$ w.r.t. α, at $\alpha = 0$, exist, and they are given by the eigenvalues of the matrix $\mathcal{L}M'\mathcal{R}$ where $M' := \dot{M}(\alpha)|_{\alpha=0}$.*

This theorem allows us to study the eigenvalues of the matrix $M(\alpha)$ for small positive α, and infer its stability properties. Setting $\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & \tilde{E}^{-1}\otimes\tilde{E}^{-1}\end{bmatrix}$ and $\mathcal{R} = \begin{bmatrix} * & -\mu I\otimes\tilde{E}^T & -\mu\tilde{E}^T\otimes I & \tilde{E}^T\otimes\tilde{E}^T\end{bmatrix}^T$ it can be shown that $\mathcal{L}M_0 = \mathcal{L}$, $M_0\mathcal{R} = \mathcal{R}$, $\mathcal{L}\mathcal{R} = I_{N-1}$ Thus the columns of $\mathcal{R}$ and the rows of $\mathcal{L}$ are, respectively, the $(N-1)^2$ right and left eigenvectors of M_0 associated with the $(N-1)^2$

[3] With the symbol $\dot{M}$ we mean the element-wise derivative of the matrix valued function $M(\alpha)$

eigenvalues in 1 of M_0, which turn out to be semi-simple. We are now in the position to prove our result.

We can then proceed to prove Theorem 1. By assumption $\mathcal{G}$ is connected, thus the previously defined matrices exist and the eigenvalues in 1 are semi–simple. Notice that

$$\mathcal{L}\dot{M}(\alpha)|_{\alpha=0}\mathcal{R} = \mathcal{L}M_1\mathcal{R} = -f_0\mu^2(I \otimes \tilde{F}\tilde{E}^{-1} + \tilde{F}\tilde{E}^{-1} \otimes I)$$

Observe now that the eigenvalues of $I \otimes \tilde{F}\tilde{E}^{-1} + \tilde{F}\tilde{E}^{-1} \otimes I$ coincides with the complex numbers which can be obtained by summing two arbitrary pairs of eigenvalues of $\tilde{F}\tilde{E}^{-1}$. Observe moreover that $\tilde{F}\tilde{E}^{-1} = W^*DV$ Since by the hypothesis of the theorem we have that W^*DV has eigenvalues with positive real part and since $f_0 > 0$, we can thus apply Theorem A.1 to conclude that the derivatives of all the eigenvalues in 1 of $M(\alpha)$ are complex numbers with negative real part and so, using continuity arguments, we have that for $\alpha > 0$ small enough $M(\alpha)$ is stable. This immediately implies that there exists α' such that if $\alpha \in (0, \alpha')$ then $M(\alpha)$ is a stable matrix, which in turn implies $\Sigma(h) \overset{t\to\infty}{\longrightarrow} 0$.

We conclude the proof with the analysis in the first moment, i.e., we study the eigenvalues of $\bar{M}$. Notice that by construction the eigenvalues of

$$\begin{aligned}\tilde{M} &= \begin{bmatrix} W^* & 0 \\ 0 & W^*D \end{bmatrix} \bar{M} \begin{bmatrix} V & 0 \\ 0 & D^{-1}V \end{bmatrix} \\ &= \begin{bmatrix} I + \tilde{E} + \alpha f_0 \mu W^* DV\tilde{E} & \mu D \\ \alpha f_0 W^* DV \tilde{E} & I \end{bmatrix}\end{aligned}$$

are the same as $\bar{M}$, except the eigenvalue in 1. Also notice that $\tilde{m}(h+1) = \tilde{M}\tilde{m}(h)$, where $\tilde{m}(h) = \begin{bmatrix} y(h) \\ z(h) \end{bmatrix}$. Similarly to what has been done for the second moment, notice that $\tilde{M} = \tilde{M}_0 + \alpha\tilde{M}_1$. It is straightforward to see that $\tilde{M}_0$ has $N-1$ simple eigenvalues in 1 and that the corresponding left eigenvectors are the rows of $\mathcal{L} = \begin{bmatrix} 0 & I \end{bmatrix}$ and the right eigenvectors are the columns of $\mathcal{R} = \begin{bmatrix} -\frac{1}{\mu}\tilde{E}^{-1} \\ I \end{bmatrix}$. We obtain $\mathcal{L}\tilde{M}_1\mathcal{R} = -\frac{f_0}{\mu}W^*DV$, so if the conditions in the theorem are satisfied, all the eigenvalues of $\tilde{M}$ are stable for α small enough, and thus $\bar{M}$ has all stable eigenvalues except one eigenvalue in 1 with multiplicity 2.

To conclude, the analysis in the first and second moments implies that there exist $\bar{\alpha}$ such that if $\alpha \in (0, \bar{\alpha})$ then the mean–square synchronization is achieved.

C. *Proof of Corollary 1*

If $\bar{E} = \bar{E}^*$ is symmetric, then $w = \frac{1}{N}\mathbf{1}$ and we can choose $W = V$. The criterion is thus satisfied, because $W^*DV = V^*DV$ is positive definite.

D. *Proof of Proposition 2*

First recall that, since we assume that the transmission times are the samples of a Poisson process of intensity $N\lambda$, we have $\mu = 1/\lambda N$ and $\sigma^2 = 2/\lambda^2 N^2$. Thanks to the structure of the update matrices in the asymmetric broadcast protocol, it is now easy to see that $E^{(i)} = -q(I - \mathbf{1}e_i^*)$ from which we get the symmetric matrix $\bar{E} = -q(I - \frac{1}{N}\mathbf{1}\mathbf{1}^*)$. Therefore, $w = \frac{1}{N}\mathbf{1}$ and we can take $W = V$, $V^*V = I_{N-1}$ and $V^*\mathbf{1} = 0$, obtaining $\tilde{E} = V^*\bar{E}V = -qI_{N-1}$. Consequently, employing the Kronecker product we can express the matrix M that describes the evolution of the error variance (see (19) in the proof of Theorem 1) as $M(\alpha) = M'(\alpha) \otimes I_{(N-1)^2}$, for a certain matrix $M'(\alpha)$. Letting $\beta := \alpha f_0 \mu q$ and performing a change of basis via the transformation matrix $\begin{bmatrix} \mu^2 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & \mu & \mu & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, it can be shown that the eigenvalues of $M'(\alpha)$ are $q \in (0, 1)$, stable, and those of

$$\tilde{M}(\alpha) = \begin{bmatrix} (1-q)^2 - 2\beta(1-q) + 2\beta^2 & 2(1-q-2\beta) & 2 \\ \beta^2 - 2\beta(1-q) & 1-q-2\beta & 1 \\ \beta^2 & -2\beta & 1 \end{bmatrix}$$

It is a matter of tedious but straightforward computation to show that the characteristic polynomial of $\tilde{M}(\alpha)$ is

$$\begin{aligned}&\phi(z) = z^3 + z^2(-(1-q)^2 - 2 + q + 2\beta(2-q-\beta)) + \\ &z((1-q)^3 + (1-q)^2 + 1 - q - 2\beta(1-q)(2-q)) - (1-q)^3\end{aligned}$$

By applying the bilinear transformation $z = \frac{1+s}{1-s}$ we obtain the polynomial

$$\begin{aligned}&\psi(s) = Q_3 s^3 + Q_2 s^2 + Q_1 s + Q_0 \\ &Q_3 = 2(1-q)^3 + 2(1-q)^2 + 4 - 2q - 2\beta((2-q)^2 - \beta) \\ &Q_2 = -4(1-q)^3 + 4 - 2\beta(2q - q^2 - \beta) \\ &Q_1 = 2(1-q)^3 - 2(1-q)^2 + 2q - 2\beta(-(2-q)2 + \beta) \\ &Q_0 = -2\beta(-1 + (1-q)^2 + \beta)\end{aligned}$$

By the Routh-Hurwitz criterion, for stability it is enough that all the coefficients of this polynomial are positive and that $Q_1Q_2 - Q_3Q_0 > 0$. One can prove that

- The terms Q_3 and Q_2 are always positive;
- The term Q_1 is positive for $\beta \in (0, (2-q)^2)$;
- The term Q_0 is positive for $\beta \in (0, q(2-q))$;
- The term $Q_1Q_2 - Q_3Q_0$ is positive when $\beta > 0$.

We can argue that we have stability if $\beta \in (0, q(2-q))$, or equivalently $\alpha f_0 \in (0, \lambda N(2-q))$.

E. *Proof of Proposition 3*

It can be seen, by expanding the expression (12) for the evolution of the covariance matrix $\Sigma(h)$, that under Assumption 4 we have that $\Sigma_{rs}(h) = \gamma_{rs}(h)\Omega$ for $r, s = 1, 2$ and for all $h \geq 1$, where $\Omega = I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ and where $\gamma_{rs}(h)$ are non-negative real numbers. The terms γ_{rs}, $r, s = 1, 2$, can be expressed recursively and result to be

$$\begin{aligned}&\gamma_{11}(h+1) \\ &\quad = \left(1 + (1 + 2\alpha f_0\mu + \alpha^2 f_0^2\sigma^2)\frac{4q^2}{N} - 2(1 + \alpha f_0\mu)\frac{2q}{N}\right)\gamma_{11}(h) \\ &\qquad + \left(2\mu - 2(\mu + \alpha f_0\sigma^2)\frac{2q}{N}\right)\gamma_{12}(h) + \sigma^2\gamma_{22}(h)\end{aligned}$$

$$\begin{aligned}\gamma_{12}(h+1) &= \left(\alpha f_0\left(-\frac{2q}{N}\right) + \alpha f_0(1 + \alpha f_0\mu)\left(\frac{4q^2}{N}\right)\right)\gamma_{11}(h) \\ &\quad + \left(1 - (1 + 2\alpha f_0\mu)\frac{2q}{N}\right)\gamma_{12}(h) + \mu\gamma_{22}(h)\end{aligned}$$

$$\gamma_{22}(h+1) = \alpha^2 f_0^2\frac{4q^2}{N}\gamma_{11}(h) + 2\alpha f_0\left(-\frac{2q}{N}\right)\gamma_{12}(h) + \gamma_{22}(h)$$

Exploiting the relation $\sigma^2 = 2\mu^2$ and by performing a change of basis with the transformation matrix $\begin{bmatrix} \mu^2 & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 1 \end{bmatrix}$, it can be shown that to establish mean square synchronization is enough to study the stability of the matrix

$$\begin{aligned} &\bar{M}(\alpha) \\ &= \begin{bmatrix} \begin{array}{c} 1 - \frac{4q}{N}(1+\alpha f_0 \mu) \\ +\frac{4q^2}{N}(1 + 2\alpha f_0 \mu + 2\alpha^2 f_0^2 \mu^2) \end{array} & 2 - \frac{4q}{N}(1+2\alpha f_0\mu) & 2 \\ -\frac{2q}{N}\alpha f_0 \mu + \frac{4q^2}{N}\alpha f_0\mu(1+\alpha f_0\mu) & 1 - \frac{2q}{N}(1+2\alpha f_0 \mu) & 1 \\ \frac{4q^2}{N}\alpha^2 f_0^2 \mu^2 & -\frac{4q}{N}\alpha f_0 \mu & 1 \end{bmatrix} \end{aligned}$$

Now let $q = 1/2$ an recall that $\mu = 1/(N\lambda)$. Then, through straightforward computations it is possible show that the characteristic polynomial of $\bar{M}(\alpha)$ is

$$\begin{aligned} \phi(z) = \frac{1}{N^3}\Bigg(& N^3 z^3 + \left(2\frac{\alpha f_0}{\lambda}N - \left(\frac{\alpha f_0}{\lambda}\right)^2 - 3N^3 + 2N^2 \right) z^2 \\ & + \left(2\frac{\alpha f_0}{\lambda} - 2\frac{\alpha f_0}{\lambda}N + 3N^3 - 4N^2 + N \right) z - N^3 + 2N^2 - N \Bigg) \end{aligned}$$

Applying the bilinear transformation yields the polynomial

$$\begin{aligned} \Psi(s) &= \left(Q_3 s^3 + Q_2 s^2 + Q_1 s + Q_0\right)/N^3 \\ Q_3 &= 8N^3 - 8N^2 - 4N\alpha f_0/\lambda + 2N + 2\left(\alpha f_0/\lambda\right)^2 + 2\alpha f_0/\lambda \\ Q_2 &= 8N^2 - 4N + 2\left(\alpha f_0/\lambda\right)^2 - 2\alpha f_0/\lambda \\ Q_1 &= 4N\alpha f_0/\lambda + 2N - 2\left(\alpha f_0/\lambda\right)^2 - \alpha f_0/\lambda \\ Q_0 &= -2\left(\alpha f_0/\lambda\right)^2 + 2\alpha f_0/\lambda \end{aligned}$$

By the Routh-Hurwitz criterion, for stability it is enough that $Q_3 > 0$, $Q_2 > 0$, $Q_0 > 0$ and $Q_1 Q_2 - Q_3 Q_0 > 0$. One can prove that

- The term Q_0 is positive if $0 < \frac{\alpha f_0}{\lambda} < 1$;
- The term Q_3 and Q_2 are positive if $0 < \frac{\alpha f_0}{\lambda} < 1$ and $N \geq 2$;
- The term $Q_1 Q_2 - Q_3 Q_0$ is positive if $\frac{\alpha f_0}{\lambda} > 0$.

Hence it follows that for $N \geq 2$, the mean-square stability is guaranteed if $\alpha < \frac{\lambda}{f_0}$.

F. Proof of Proposition 4

Assume that node 1 is the root of the tree. As it receives no information from the other nodes, we can assume that its time estimate evolves as $\hat{t}_1(t) = \frac{f_1}{f_0} t$ and its frequency correction as $k_1(t) = \frac{1}{f_0}$. Consider node i and its parent j. Assume that node i receives information from node j at the time instants $T_{\text{up}}(h)$, $h = 0, 1, 2, \ldots$. Assume moreover that node j reaches mean-square synchronization with node 1, i.e.

$$\hat{t}_j(h) = \hat{t}_1(h) + \eta_j(h),$$

where we abuse the notation and write $\hat{t}_j(h)$ instead of $\hat{t}_j(T_{\text{up}}(h))$, etc. Here $\mathbb{E}\left[\eta_j^2(h)\right]$ goes to 0 as $h \to \infty$. According to RANDSYNC-TREE algorithm, we can write that

$$\begin{cases} \hat{t}_i(h+1) = \hat{t}_j(h) + f_i \delta_{\text{up}}(h) k_i(h+1) \\ k_i(h+1) = k_i(T_{\text{up}}(h)) + \alpha\left(\hat{t}_j(h) - \hat{t}_i(h)\right) \end{cases}$$

where $\delta_{\text{up}}(h) := T_{\text{up}}(h+1) - T_{\text{up}}(h)$. Let $z_i(t) := f_i\, k_i(t) - f_1\, k_1(t) = f_i\, k_i(t) - \frac{f_1}{f_0}$.

The second equation can be rewritten as

$$k_i(h+1) = k_i(h) + \alpha\left(\hat{t}_1(h) + \eta_j(h) - \hat{t}_i(h)\right) \tag{20}$$

Let $z_i(t) := f_i\, k_i(t) - f_1\, k_1(t) = f_i\, k_i(t) - \frac{f_1}{f_0}$. Then

$$z_i(h+1) = f_i k_i(h) - \frac{f_1}{f_0} + \alpha f_i\left(\hat{t}_1(h) + \eta_j(h) - \hat{t}_i(h)\right). \tag{21}$$

Consider $\hat{t}_1(h) - \hat{t}_i(h)$. Straightforward manipulations yield

$$\hat{t}_1(h) - \hat{t}_i(h) = -\eta_j(h-1) - \delta_{\text{up}}(h-1) z_i(h).$$

Plugging the last expression into (21) we get

$$\begin{aligned} z_i(T_{\text{up}}(h+1)) = {} & \left(1 - \alpha f_i \delta_{\text{up}}(h-1)\right) z_i(T_{\text{up}}(h)) \\ & + \alpha f_i \left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right) \end{aligned} \tag{22}$$

Let us define $P_i(h)^2 := \mathbb{E}\left[z_i^2(T_{\text{up}}(h))\right]$ and $W_j(h)^2 := \mathbb{E}\left[\left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right)^2\right]$. Then

$$\begin{aligned} P_i(h+1)^2 = {} & \left(1 - 2\frac{\alpha f_i}{\lambda} + 2\frac{\alpha^2 f_i^2}{\lambda^2}\right) P_i(h)^2 \\ & + \alpha^2 f_i^2 \mathbb{E}\left[\left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right)^2\right] \\ & + 2\left(\alpha f_i - \frac{\alpha^2 f_i^2}{\lambda}\right) \times \\ & \quad \times \mathbb{E}\left[z_i(T_{\text{up}}(h))\,\left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right)\right] \\ \leq {} & \left(1 - 2\frac{\alpha f_i}{\lambda} + 2\frac{\alpha^2 f_i^2}{\lambda^2}\right) P_i(h)^2 \\ & + \alpha^2 f_i^2 \mathbb{E}\left[\left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right)^2\right] + \\ & + 2\left(\alpha f_i - \frac{\alpha^2 f_i^2}{\lambda}\right) \sqrt{\mathbb{E}\left[z_i^2(T_{\text{up}}(h))\right]} \times \\ & \quad \times \sqrt{\mathbb{E}\left[\left(\eta_j\left(T_{\text{up}}(h)\right) - \eta_j\left(T_{\text{up}}(h-1)\right)\right)^2\right]} \\ = {} & \left[\left(1 - \frac{\alpha f_i}{\lambda}\right) P_i(h) + \alpha f_i W_j(h)\right]^2 + \left(\frac{\alpha f_i}{\lambda}\right)^2 P_i(h)^2 \\ = {} & \left\| \begin{bmatrix} 1 - \alpha f_i/\lambda \\ \alpha f_i/\lambda \end{bmatrix} P_i(h) + \begin{bmatrix} \alpha f_i W_j(h) \\ 0 \end{bmatrix} \right\|^2 \\ \leq {} & \left(\left\| \begin{bmatrix} 1 - \alpha f_i/\lambda \\ \alpha f_i/\lambda \end{bmatrix} P_i(h) \right\| + \left\| \begin{bmatrix} \alpha f_i W_j(h) \\ 0 \end{bmatrix} \right\| \right)^2 \end{aligned}$$

where we used the Cauchy-Schwarz inequality and the triangular inequality. We can argue that

$$\begin{aligned} P_i(h+1) &\leq \left\| \begin{bmatrix} 1 - \alpha f_i/\lambda \\ \alpha f_i/\lambda \end{bmatrix} P_i(h) \right\| + \left\| \begin{bmatrix} \alpha f_i W_j(h) \\ 0 \end{bmatrix} \right\| \\ &= \sqrt{1 - 2\frac{\alpha f_i}{\lambda} + 2\frac{\alpha^2 f_i^2}{\lambda^2}} P_i(h) + \alpha f_i W_j(h) \end{aligned}$$

and consequently, since $W_j(h)$ converges to zero, we have that, $P_i(h)$ converges to zero if $|1 - 2\frac{\alpha f_i}{\lambda} + 2\frac{\alpha^2 f_i^2}{\lambda^2}| < 1$, namely, if $0 < \alpha < \frac{\lambda}{f_i}$. Finally, since

$$\hat{t}_i(h+1) - \hat{t}_1(h+1) = \eta_j(h) + \delta_{\text{up}}(h) z_i(h+1).$$

and since both $\mathbb{E}\left[\eta_j^2(h)\right]$ and $\mathbb{E}\left[z_i^2(h+1)\right]$ converge to zero, we can conclude that the node i reaches mean-square synchronization with node 1.