



HAL
open science

TAYLORS: A fasT, scAlable Yet Lightweight prOtocol for Resilient Synchronization of WSN

Benoît Perroux, Jérôme Ermont, Katia Jaffrès-Runser

► **To cite this version:**

Benoît Perroux, Jérôme Ermont, Katia Jaffrès-Runser. TAYLORS: A fasT, scAlable Yet Lightweight prOtocol for Resilient Synchronization of WSN. 2025. hal-04933542

HAL Id: hal-04933542

<https://hal.science/hal-04933542v1>

Preprint submitted on 6 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TAYLORS: A fast, scAlable Yet Lightweight prOtolocol for Resilient Synchronization of WSN

Benoît PERROUX
IRIT, Université de Toulouse,
CNRS, Toulouse INP, UT3, Alliantech
Toulouse, France
benoit.perroux@irit.fr

Jérôme ERMONT
IRIT, Université de Toulouse,
CNRS, Toulouse INP, UT3
Toulouse, France
jerome.ermont@irit.fr

Katia JAFFRÈS-RUNSER
IRIT, Université de Toulouse,
CNRS, Toulouse INP, UT3
Toulouse, France
katia.jaffres-runser@irit.fr

Abstract—As of today, wireless sensor networks (WSNs) are regularly deployed to monitor physical phenomena. This is made possible by the collection of time-stamped data building on synchronized sensor clocks. Some applications benefit from a temporary deployment of the network. This is for instance the case for acoustics calibration of a concert hall using connected microphones. In this case, the clock synchronization protocol must converge fast and in a bounded time. In this paper, we propose TAYLORS, a distributed and thus robust synchronization protocol for WSN that offers a fast bounded convergence and that keeps a stable clock even in dynamic settings. TAYLORS combines both max-consensus at initialization and average-consensus over the long term. This work provides as well a novel mathematical model to derive a worst-case bound on convergence that is leveraged to motivate the design of TAYLORS. Experimental results show that the proposed solution is simple, fast and suitable for very basic micro-controllers.

Index Terms—WSN, synchronization, distributed consensus.

I. INTRODUCTION

The deployment versatility of wireless sensor networks (WSN) has paved the way for rolling out various monitoring applications in the last decade in various sectors (industry, health, agriculture, etc.). Monitoring relies on the collection of time series of multi-variate data. As such, a core WSN service is data timestamping which generally relies on the time synchronization of sensor’s oscillators. A large palette of synchronization protocols has been developed throughout the years, tailored for various application contexts and leveraging different wireless technologies. Main design objectives encompass synchronization accuracy, energy consumption, scalability and robustness to topology changes or message losses.

This paper focuses on adding the objective of fast and *bounded* deployment time of the synchronization service. This is motivated by industrial scenarios where the WSN is rolled-out for a short duration (in the order of hours) and requires an on-the-spot and autonomous organization of the network to collect data. This is for instance useful for the acoustic calibration of a concert hall using a network of microphones or for the temporary measurement of vibrations of a vehicle or an airplane structure. The point is to provide an upper bound on the time the synchronization service needs to achieve the desired synchronization accuracy and to start data collection.

In a nutshell, two typical designs of synchronization protocols exist: hierarchical or distributed ones. Hierarchical proto-

cols use an active tree topology to distribute a global clock information that nodes use to compensate for their offset and/or their skew. They can be seen as the wireless counterpart of wired protocols such as NTP [1]. For WSN, typical protocols are TPSN [2] or PulseSync [3]. Hierarchical protocols offer high accuracy and fast diffusion of synchronization data once the tree is built. However, in a dynamic network, with nodes regularly joining or leaving the network due to mobility or perhaps evolving wireless connectivity, robustness issues arise that necessitate more advanced topology control mechanisms.

The distributed design of wireless synchronization protocols, on the other side, offers inherent resilience to connectivity changes and avoids the single point of failure of hierarchical protocols. Indeed, distributed protocols execute the same message-passing algorithm with the aim of converging to a common clock through consensus. No node is of such importance that its departure or move causes large perturbations. However, the drawback of distributed consensus protocols is the time needed to reach an initial agreement that might be long and non-predictable for certain topologies. If such limitations can be overlooked, an on-the-spot deployment of a robust distributed monitoring service becomes possible.

In this work, we focus on the large-scale deployment of very small sensor nodes in a space such as a hangar or a vast concert hall: up to a hundred nodes are deployed with no prior assumption on the topology. These nodes, equipped with sensors such as microphones, have to timestamp their measurements. The target environment is dynamic: channel condition varies rapidly due to the presence of spectators for instance. Considering the large scale deployment and the need for robustness, using a distributed synchronization protocol is essential. This protocol has to offer a bounded and preferably fast convergence time (in the order of minutes), using basic algorithmic operations that lightweight sensors can perform.

This paper presents two contributions: *i*) the mathematical derivation of a bound on the convergence time of classical distributed synchronization protocol based on average consensus and *ii*) a novel protocol, TAYLORS, that combines a unique fast converging max consensus initialization step with a long-term robust average consensus synchronization. Note that the consensus protocols we consider are asynchronous. In the remainder of this paper, we review main distributed

consensus synchronization protocols in Section II. Before leveraging Perron-Frobenius random matrix-theory to derive an upper bound on the convergence time of one of the simplest asynchronous average consensus protocol in Section IV, we illustrate the difficulty of calculating such a bound in Section III. Section V presents TAYLORS, that offers a small deterministic convergence time while still being simple, robust and stable. Finally, Section VI concludes this work.

II. DISTRIBUTED SYNCHRONIZATION PROTOCOLS

A. Related works

Several distributed synchronization protocols have been proposed in the past decade. While each one of them has its own particularities, from great simplicity to fast convergence or increased long-term stability, they all share some key features. First, they offer robustness since they are highly adaptive to topology modifications and message losses. Second, they only leverage local information to reach a global clock consensus shared by all the nodes. Third, the absolute value of the final clock has little importance, important is the global agreement.

In [4], the authors propose a simple and lightweight protocol, SISP. This protocol is designed for the smallest wireless sensors with little memory, it performs a basic asynchronous average consensus from clock values periodically broadcast by each node. A detailed description of SISP is given in Section II-B. The ATS [5] synchronization protocol improves this average consensus mechanism by adding a fully distributed drift compensation. This allows for a more stable clock and thus less frequent clock broadcasts, with the drawback of a more complex algorithm in terms of computation and memory footprint. In [6], the authors build on the performance of the ATS algorithm to propose a multi-hop version named MACTS. With the observation that the more connected the topology, the faster the convergence, they make the nodes fast-relay the messages, creating virtual links. This mechanism is only used in transitory state, as it reduces to a simple ATS mechanism when the convergence is detected. Later, MTS [7] protocol opts to use a max consensus algorithm instead of an average consensus like previous ones: nodes converge towards the clock speed and offset of the fastest-running one. Convergence is definitely faster with max consensus since a unique flooding of clock values in a mesh topology is enough. However, keeping this approach for long-term synchronization is risky since arrival of a fast-running node induces an abrupt consensus clock variation which is detrimental to the correct timestamping of data. The TSMA [8] mitigates this issue by using both a max-consensus approach similar to MTS to the drift compensation, and an average-consensus approach like SISP to the offset correction. But convergence speed of offset compensation is then similar to SISP or ATS.

More complex solutions exist that build on gradient descent (GTSP [9]) or control theory (TSMPID [10]), or simply provide pure syntonization (RFA [11]). They are thus out of the scope of this work. Next, we detail SISP to illustrate how a basic average consensus algorithm works and leverage it to derive an appropriate mathematical model to bound

its convergence time. SISP is as well expanded to derive TAYLORS thanks to its simplicity and capacity of reaching tens of micro-second accuracy [12].

B. The SISP protocol

The SImple Synchronization Protocol (SISP) was introduced in [4]. Each node calls algorithm Algorithm 1 regularly.

Algorithm 1 SISP Procedure, called regularly

```

1:  $LCLK \leftarrow LCLK + 1$ 
2:  $SCLK \leftarrow SCLK + 1$ 
3: if  $LCLK \bmod T == 0$  then
4:   broadcast( $SCLK$ )
5: else if msgReceived( $RCLK$ ) then
6:    $SCLK \leftarrow (SCLK + RCLK)/2$ 
7: end if

```

Two software counters, acting as clocks, are used in the algorithm, the Local Clock (LCLK) and the Shared Clock (SCLK). While the former is intrinsic to each node and tracks the time since system startup, the latter aims at reaching a common value among all nodes of the network. After system startup, the `sisp()` procedure is called repeatedly using native system interruptions triggered at the same pace for all nodes of the network. Each node broadcasts its current value of SCLK with a predefined period T . Upon reception of a SYNC message (SYNC), the node computes the average between its own SCLK and the received value, called the Received Clock (RCLK). By successively averaging the clock values from neighboring nodes, the network reaches consensus.

Note that there is no assumption on the start time of `sisp()` calls: the network is asynchronous. Moreover, simple sensors constrain clocks to be defined as integers.

III. PROBLEM STATEMENT

This section presents, using experiments and simulations, the observations our mathematical model has to capture in order to derive a bound on the convergence time of SISP.

A *clock tick* is the duration between two increments of the SCLK. It is, by construction, the maximum provable synchronization accuracy. A *synchronization round* is defined as the period T , where each node sends one and only one SYNC. Experiments are conducted in the FIT IoT-LAB testbed [13] using an implementation of SISP on Qorvo DWM1001 Ultra-Wide Band (UWB) boards. In our implementation, a clock tick is set to 10ms and $T = 100$, corresponding to a round of 1s. Simulations are conducted with a simple in-house simulator representing the evolution of the shared clock of nodes at the end of each round. In both experiments and simulation, the local clock counters LCLK are initialised with non-zero values at system start. An experiment is reproduced 30 times with the same initial LCLK values. The variability observed in our results stems from the asynchronicity of the global network coming from the asynchronous start time of each node. The convergence is observed once the increment of the SCLK is equal to a constant value two consecutive rounds in a row.

a) *Impact of network topology*: A wireless network topology has a strong impact on the performance of synchronization protocols in general. The more connected, the faster the diffusion of clock information is. Using experiments, we illustrate the convergence speed of SISP on two extreme topologies: *i*) a full mesh topology, and *ii*) a line topology. For each topology, we test a 5 and a 10-node network. We observe the following:

- Impact of topology: convergence of full mesh is faster than line topology. For 5 nodes, mesh convergence lasts 2.5 ± 0.21 rounds while line topology lasts 21.1 ± 11.7 .
- Impact of network size: the average convergence reduces from 2.5 ± 0.21 to 2.1 ± 0.25 rounds for the mesh topology. On the contrary, for the line topology, convergence grows exponentially: average is going from 21.1 ± 11.7 to 127.7 ± 46.8 , with minimum (resp. maximum) values going from 21 (resp. 58) to 49 (resp. 211) rounds.

b) *Impact of the emission sequence*: The asynchronous initialization of the protocol induces a randomization of the sequence of SYNC message emissions in a round. As such, at each network restart, a new emission sequence SYNC messages is observed. Figure 1 shows how the emission sequence impacts convergence on a 4-node line topology considering two different emission sequences. Convergence time is greatly impacted: it almost doubles from 13 to 24 rounds, and SCLK differ with value 6200 (left) or 6500 (right) after 25 rounds.

As a conclusion, the mathematical model of the convergence has to capture the impact of topology and emission sequence.

IV. CONVERGENCE STUDY OF SISP

A. Related Works

Several synchronization protocols of Section II are presented alongside convergence in finite time proofs, with a common assumption being the need for the network to be connected for at least an arbitrarily long but finite duration [5]. However, in the context of an on-the-spot deployment of sensor nodes, we require a bound on the convergence time to plan the start time of data collection. Studying convergence time of general distributed consensus algorithms builds around two

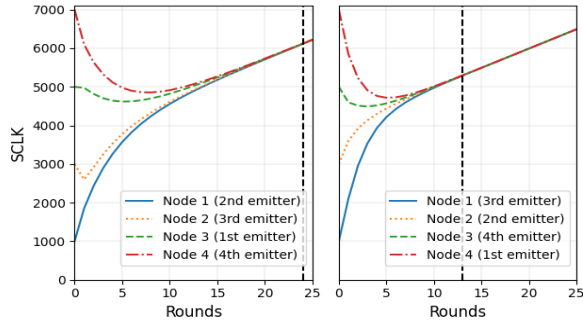


Fig. 1. Impact of two emission sequences for a 4-node line topology. Node i communicates only with nodes $i - 1$ and $i + 1$. On the left, convergence is reached after 24 rounds, while on the right it takes 13 rounds.

main methods: Lyapunov-based theory and Perron-Frobenius theorem in stochastic matrix theory.

Lyapunov-based methods are widely used to prove the stability and asymptotic convergence or divergence rate of dynamical systems. They leverage the Laplacian of a graph to characterize information flow, and depending on the algebraic connectivity of the graph, deduce the convergence rate [14]. The rate computed offers a loose bound on convergence time, but doesn't capture well the impact of fine-grained parameters such as emission sequence of messages for instance.

Perron-Frobenius theorem applied to stochastic power matrix theory can be leveraged to extract convergence properties of Markov chains [15]. The method revolves around the definition of a stochastic matrix representing the transition between successive states of a system [16], [17]. The Perron-Frobenius theorem links the second-largest eigenvalue of this transition matrix to the convergence rate of the system. The smaller this eigenvalue is, the faster the convergence. Among the protocols cited in Section II, only TSMPID [10] leverages stochastic matrix theory to prove convergence in finite time.

In this paper, we show that it is possible to express the impact of an emission sequence in a stochastic transition matrix representing the evolution of the shared clocks counters in an arbitrary topology. Using the second largest eigenvalue, we can compare the convergence speed of emission sequences and extract the worst-case emission sequence. Then, its convergence time can be calculated by simulation.

B. Stochastic Model

For a given node ν_i , at time t , $L_i(t)$ is the local clock (LCLK) and $S_i(t)$ its vision of the shared clock (SCLK). The vector $S(t)$ lists SCLK for all nodes at time t . For each node ν_i , its drift α_i is such as $L_i(t + \tau) = L_i(t) + \alpha_i \tau$. The vector $\vec{\alpha}$ lists all the individual drifts. In the absence of drift $\vec{\alpha} = \mathbb{1}$. We suppose $\vec{\alpha}$ constant for the duration of the experiment even though drift typically varies over time due to factors such as aging or temperature. Furthermore, we assume a constant and thus negligible communication delay, and no message losses. In order to study the asynchronous evolution of $S(t)$ with SISP protocol, we model an asynchronous instance of SISP as a synchronous protocol where the emission sequence within the round is fixed. It is the case where clock drifts are constant. In each round, numbered with $k > 0$, each node emits a SYNC once following a fixed sequence. Next, we express $S(t_{k+1})$ as a function of $S(t_k)$ to exhibit a stochastic transition matrix.

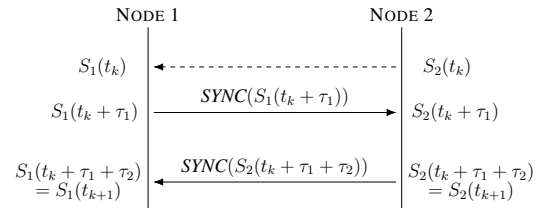


Fig. 2. Synchronization round of SISP with 2 nodes.

1) *Example with 2 nodes:* Fig. 2 depicts the evolution of the SCLK of 2 nodes in a round starting from $S(t_k)$. We have:

$$S_1(t_k + \tau_1) = S_1(t_k) + \alpha_1 \tau_1 \quad (1)$$

When node 2 receives the SYNC of 1, S_2 is:

$$\begin{aligned} S_2(t_k + \tau_1) &= [S_1(t_k) + \alpha_1 \tau_1 + (S_2(t_k) + \alpha_2 \tau_1)]/2 \\ &= \frac{S_1(t_k) + S_2(t_k)}{2} + \frac{\alpha_1 + \alpha_2}{2} \tau_1 \end{aligned} \quad (2)$$

Once node 2 has sent a SYNC, the round ends with S_2 :

$$\begin{aligned} S_2(t_{k+1}) &= S_2(t_k + \tau_1) + \alpha_2 \tau_2 \\ &= \frac{S_1(t_k) + S_2(t_k)}{2} + \frac{\alpha_1 + \alpha_2}{2} \tau_1 + \alpha_2 \tau_2 \end{aligned} \quad (3)$$

and S_1 :

$$\begin{aligned} S_1(t_{k+1}) &= S_1(t_k + \tau_1 + \tau_2) \\ &= ((S_1(t_k + \tau_1) + \alpha_1 \tau_2) + (S_2(t_{k+1}))) / 2 \\ &= \frac{3S_1(t_k) + S_2(t_k)}{4} + \frac{3\alpha_1 + \alpha_2}{4} \tau_1 + \frac{\alpha_1 + \alpha_2}{2} \tau_2 \end{aligned} \quad (4)$$

The evolution of clocks can be expressed in a matrix form:

$$S(t_{k+1}) = \begin{pmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \end{pmatrix} S(t_k) + \begin{pmatrix} 3/4 & 1/4 \\ 1/2 & 1/2 \end{pmatrix} \vec{\alpha} \tau_1 + \begin{pmatrix} 1/2 & 1/2 \\ 0 & 1 \end{pmatrix} \vec{\alpha} \tau_2 \quad (5)$$

2) *Generalization:* We define the matrices E_i corresponding to the effect of the emission of a message by node ν_i on the clocks in the network.

$$E_i = (e_{i_n,p})_{1 \leq n \leq N, 1 \leq p \leq N} \quad (6)$$

$$\text{with } e_{i_n,p} = \begin{cases} 1 & \text{if } \nu_n \notin \mathcal{N}(\nu_i) \ \& \ p = n \\ 0.5 & \text{if } \nu_n \in \mathcal{N}(\nu_i) \ \& \ (p = n \ || \ p = i) \\ 0 & \text{otherwise} \end{cases}$$

and $\mathcal{N}(\nu_i)$ the set of neighbors of node ν_i . These matrices express that any node that does not receive the SYNC message, including the emitter itself, will keep its clock value, while a node receiving the SYNC will compute the average between the received value and its own clock. With this notation, we represent the vector S after the emission of a node as:

$$S(t_k + \tau_i) = E_i \cdot (S(t_k) + \tau_i \vec{\alpha}) \quad (7)$$

By iterating in a round composed of a given emission sequence by two nodes a then b , we have:

$$\begin{aligned} S(t_{k+1}) &= E_b (E_a \cdot (S(t_k) + \tau_a \vec{\alpha}) + \tau_b \vec{\alpha}) \\ &= E_b E_a \cdot S(t_k) + [E_b E_a \cdot \tau_a + E_b \cdot \tau_b] \cdot \vec{\alpha} \end{aligned} \quad (8)$$

We define $P_{a,b}$ as:

$$P_{a,b} = E_b \cdot E_a \quad (9)$$

Finally, we can express the values of SCLK at round $k + 1$ with regard to the values at the previous round:

$$S(t_{k+1}) = P_\sigma \cdot S(t_k) + \left[\sum_{i=1}^N \left(\tau_{\sigma_i} \prod_{j=N}^i E_{\sigma_j} \right) \right] \cdot \vec{\alpha} \quad (10)$$

with $\sigma = \{\sigma_1, \dots, \sigma_N\}$ the ordered set corresponding to the emission sequence. By applying (10) to the example previously presented, we again find the result shown in (5). Furthermore, with the second term of the sum not depending on k , we pose:

$$\delta = \left[\sum_{i=1}^N \left(\tau_{\sigma_i} \prod_{j=N}^i E_{\sigma_j} \right) \right] \cdot \vec{\alpha} \quad (11)$$

and rewrite (10) as:

$$S(t_{k+1}) = P \cdot S(t_k) + \delta \quad (12)$$

We finally express the values of the Shared Clock of each node at round k as a function of their initial values:

$$S(t_k) = P^k \cdot S(t_0) + \sum_{i=0}^{k-1} P^i \delta \quad (13)$$

C. *No drift case*

If drift is neglect, $\vec{\alpha} = \mathbb{1}$ and δ is:

$$\delta = \left[\sum_{i=1}^N \left(\tau_{\sigma_i} \prod_{j=N}^i E_{\sigma_j} \right) \right] \cdot \mathbb{1} = \sum_{i=1}^N \tau_{\sigma_i} \cdot \mathbb{1} = (t_{k+1} - t_k) \cdot \mathbb{1} \quad (14)$$

Following this, we can rewrite (13) as:

$$S(t_k) = P^k \cdot S(t_0) + (t_k - t_0) \cdot \mathbb{1} \quad (15)$$

Extensive simulations comparing our model with and without drifts show that drifts have a negligible impact on convergence time compared the other parameters such as topology and emission sequence. It mostly impacts the consensus clock value. Thus, next results are given without drifts.

D. *Convergence analysis*

From the previously defined stochastic model, we first observe that P is a product of several stochastic matrices E_i , and is thus stochastic as well. If the matrix is irreducible, Perron-Frobenius theorem states that P^k converges towards a stationary distribution noted Π_P at a rate depending on the second-largest eigenvalue of the transition matrix P , noted λ_2 . Irreducibility of P is obtained with Theorem 1. As we focus on connected topologies without packet loss, the premise Theorem 1 is always verified here. As such, $\exists k_{conv} \ \forall n \geq k_{conv}, P^n = \Pi_P$. For $n \geq k_{conv}$, we can rewrite (13) at round n as:

$$S(t_n) = \Pi_P \cdot S(t_0) + \sum_{i=0}^{k_{conv}-1} P^i \delta + (n - k_{conv}) \Pi_P \cdot \delta \quad (16)$$

Finally, we have:

$$S(t_{n+1}) - S(t_n) = \Pi_P \cdot \delta = cste * \mathbb{1} \quad (17)$$

The difference between the clocks at each round stays constant over time, and convergence is reached.

Theorem 1. *If the graph of the network is connected, the matrix P is irreducible.*

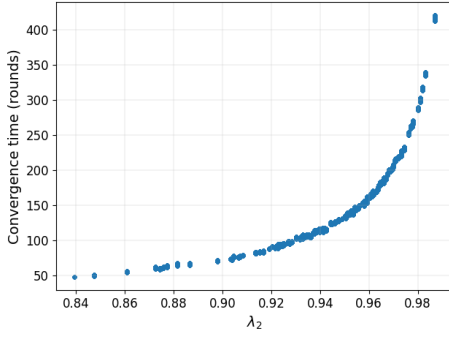


Fig. 3. Relationship between λ_2 and the convergence time for a 10-node linear network.

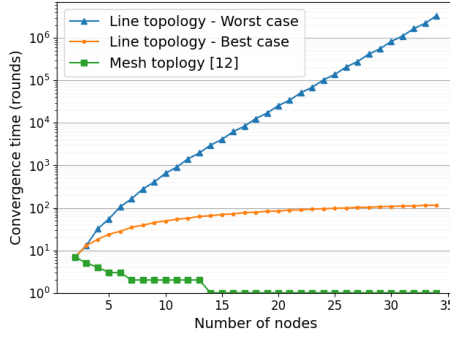


Fig. 4. Convergence of SISF for line and full-mesh topologies. Initial clocks are in $\{0, 10\ 000\}$ ticks.

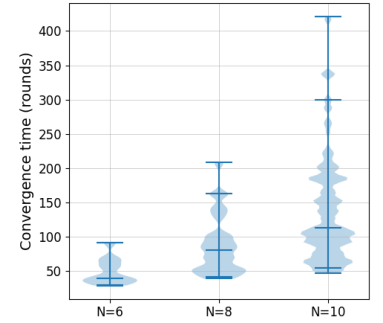


Fig. 5. Distribution of convergence time for 6, 8 and 10-node linear networks.

Proof. Let the graph $G(V, E)$ representing the network be connected. Suppose P reducible. By definition, if P is reducible it can be conjugated into an upper triangular block form by a permutation Q :

$$QPQ^{-1} = \begin{pmatrix} A & B \\ 0 & D \end{pmatrix}$$

i.e. $\exists I \subset \{0, 1, \dots, N-1\}$ such as $\forall i \in I, \forall j \notin I, P_{i,j} = 0$.

$G(V, E)$ is connected so $\forall i \in I, \forall j \notin I$ there is a path containing two nodes $i' \in I$ and $j' \notin I$ such as $\{i', j'\} \in E$. By the definition of E_i in (6), $e_{i',j'} = 0.5$. By generalizing the definition of E_i , we have $E_i = (M_i + \frac{1}{2}I)$ with M_i nonnegative.

$$\begin{aligned} P_{i',j'} &= (E_\alpha \dots E_{i'} \dots E_\omega)_{i',j'} \\ &= \left(\left(M_\alpha + \frac{1}{2}I \right) \dots E_{i'} \dots \left(M_\omega + \frac{1}{2}I \right) \right)_{i',j'} \\ &\geq \left(\frac{1}{2}I \right)^{N-1} e_{i',j'} = \frac{1}{2^N} > 0 \end{aligned}$$

It contradicts the fact that P is reducible, so P is irreducible. \square

E. Worst-case convergence time

Fig. 3 plots the relation between the second eigenvalue λ_2 of P and the rate of convergence. Points represent emission sequences of a 10-node linear topology obtained by enumerating them for all orders, computing λ_2 and deriving convergence time by simulation. Numerical results confirm the relation between λ_2 and convergence speed. Thus, since computing λ_2 of a sequence is done with linear time complexity, it is possible for all orders to get the best and worst-case emission sequence for an arbitrary topology. Worst and best-case convergence times for an arbitrary network of N nodes are shown in Fig. 4. Worst-case is computed for the line topology and best-case for the full-mesh topology according to [12] with $\frac{1}{N} \log_2[\max_{i,j}(S_i(0) - S_j(0))]$.

In the line topology, the difference between best and worst-case is reduced in small networks, but as N grows, the worst-case increases exponentially, reaching a million rounds for a line of 32 nodes. Setting the time before data collection to

this worst-case convergence time is clearly not reasonable. As shown in the distribution of convergence times given in Fig. 5, the worst-case is relatively rare. But still, rolling out SISF or ATS without improving the protocol convergence speed algorithmically provides worst-case bounds that are not practical. In the following, we propose TAYLORS whose initialisation stage provides a convergence bound in $\mathcal{O}(N)$.

V. TAYLORS

TAYLORS is based on SISF and keeps its main characteristics (simplicity, robustness and accuracy). The proposed design improves the initial convergence to allow for fast and bounded on-the-spot deployment as well as the long-term stability by better adapting to the dynamicity of the environment.

A. Fast consensus

The first modification of SISF aims at bounding the initial convergence stage by changing average-consensus with max-consensus whose convergence speed is in the order of the network diameter (i.e. the largest path between any two nodes in the network). But different from MTS [7] and TSMA [8], TAYLORS switches from max- to average-consensus once consensus is reached in order to offer a long-term stable shared clock. Indeed, as pointed out in [10] and confirmed in our simulations, max-consensus induces more aggressive clock jumps and instability than average-consensus when sensors have really different clock drifts.

B. Stability awareness

Once synchronization is reached, a distributed protocol is left vulnerable to the arrival of non-synchronized nodes whose local clock is likely to shift the shared clock back or forth. To avoid this situation, we introduce a stability indicator in TAYLORS. This stability indicator, common in distributed consensus algorithms[8], is a way for a given node to assess if it has reached consensus with its neighbors or not. TAYLORS uses a bit flag to stay as lightweight as possible. Any new node arriving checks the flag and participates in the consensus if and only if the flag is not set. This flag can also be leveraged to detect the end of the initial convergence and start data collection. The flowchart is given in Fig. 6.

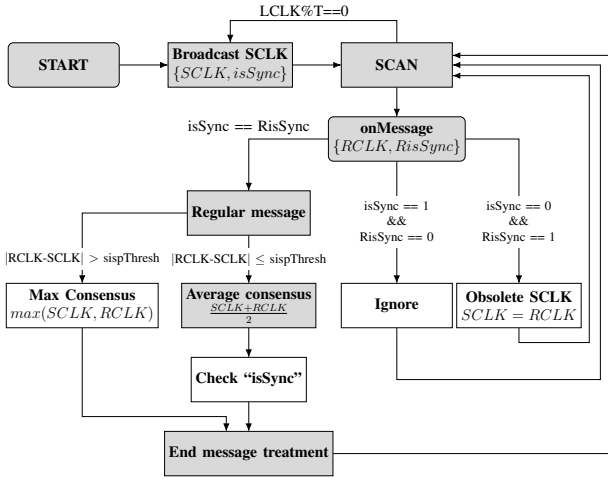


Fig. 6. TAYLORS flowchart. Original SISP states are colored in gray.

TABLE I
MEAN CONVERGENCE TIME OF SISP AND TAYLORS (ROUNDS)

		SISP	TAYLORS
Mesh	5 nodes	2.5	1.5
	10 nodes	2.1	1.34
Line	5 nodes	21.1	2.4
	10 nodes	121.7	6.1

C. Experimental results

We compare the performances of TAYLORS with SISP for both line and full-mesh topologies with the setup of Section III. Results of Tab. I only compare convergence times since final accuracy is, by construction, the same for both protocols. While results for the full mesh topology offer a small betterment, it is in the context of a line topology that the results are really visible. A line composed of 10 nodes converges on average 20 times faster with TAYLORS than with SISP. This is in line with the exponential decrease of convergence speed between SISP and TAYLORS. We checked as well that TAYLORS improves long-term stability. In the case of two originally independent clusters joining in a single network, the reconfiguration time of the synchronization is in the same order of magnitude than an initial convergence of the new large network. Our algorithm proposes a flooding of the largest shared clock among the two clusters into the other, with a reconfiguration time yet again bounded linearly by the diameter of the clusters.

VI. CONCLUSION & FUTURE WORKS

In this paper, we tackle the problem of deploying a lightweight synchronization protocol for large-scale sensor networks in a bounded and possibly fast time. Building on related works, we derive first a stochastic model of the convergence of local clocks to a shared clock to extract a precise bound on the convergence of average-consensus protocols. Our analysis shows that worst-case average-consensus bounds are too large to be leveraged practically. Thus, we propose TAYLORS, a protocol which combines the benefits of max

consensus at initialization and average consensus over the long term to offer a fast initial synchronization, long-term stability and a fast reconfiguration when two independent clusters join in a single network. Future works will enhance the convergence analysis method to account for the presence of non-deterministic communication delays and packet loss using additive and multiplicative disturbances [5] and by assimilating packet loss to a switching topology [5], [14]. Another issue we plan to tackle is related to the implementation of the protocol on small micro-controllers which can only use integer computations. Rounding errors then lead to non-linearity and adds complexity. Further work can consider the method proposed by [18] to tackle this issue.

REFERENCES

- [1] J. Martin, J. Burbank, W. Kasch, and P. D. L. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, Jun. 2010.
- [2] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *IEEE SenSys*, 2003, pp. 138–149.
- [3] C. Lenzen, P. Sommer, and R. Wattenhofer, "PulseSync: An Efficient and Scalable Clock Synchronization Protocol," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 717–727, 2015.
- [4] A. van den Bossche, T. Val, and R. Dalce, "SISP: A lightweight synchronization protocol for Wireless Sensor Networks," in *ETFA2011*, 2011, pp. 1–4.
- [5] L. Schenato and F. Fiorentin, "Average TimeSync: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.
- [6] F. Shi, X. Tuo, L. Ran, Z. Ren, and S. X. Yang, "Fast Convergence Time Synchronization in Wireless Sensor Networks Based on Average Consensus," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1120–1129, 2020.
- [7] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time Synchronization in WSNs: A Maximum-Value-Based Consensus Approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, 2014.
- [8] Z. Dengchang, A. Zhulin, and X. Yongjun, "Time Synchronization in Wireless Sensor Networks Using Max and Average Consensus Protocol," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, p. 192128, Mar. 2013.
- [9] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *IEEE IPSN*, Apr. 2009, p. 37–48.
- [10] Y. C. Koo, M. N. Mahyuddin, and M. N. A. Wahab, "Novel Control Theoretic Consensus-Based Time Synchronization Algorithm for WSN in Industrial Applications: Convergence Analysis and Performance Characterization," *IEEE Sensors J.*, vol. 23, no. 4, p. 4159–4175, Feb. 2023.
- [11] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *SenSys '05*, New York, NY, USA, Nov. 2005, p. 142–153.
- [12] O. Hotescu, K. Jaffrès-Runser, A. van den Bossche, and T. Val, "Synchronizing Tiny Sensors with SISP: A Convergence Study," in *ACM MSWIM*, 2017, pp. 279–287.
- [13] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed," in *IEEE WF-IoT*, Milan, Italy, Dec. 2015.
- [14] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference.*, vol. 2, 2003, pp. 951–956.
- [15] E. Seneta, *Non-negative matrices and Markov chains*. Springer Science, 2006.
- [16] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, p. 988–1001, Jun. 2003.
- [17] D. Deplano, M. Franceschelli, and A. Giua, "A nonlinear Perron–Frobenius approach for stability and consensus of discrete-time multi-agent systems," *Automatica*, vol. 118, p. 109025, Aug. 2020.
- [18] Y. Mocquard, B. Sericola, F. Robin, and E. Anceaume, "Stochastic Analysis of Average Based Distributed Algorithms," *Journal of Applied Probability*, vol. 58, no. 2, p. 394, Jun. 2021.