Average TimeSynch: a consensus-based protocol for clock synchronization in wireless sensor networks

Luca Schenato [a], Federico Fiorentin [b]

[a] *Department of Information Engineering University of Padova, Padova, Italy*

[b] *Accenture S.p.a, Milan, Italy*

Abstract

This paper describes a new consensus-based protocol, referred to as *Average TimeSync* (ATS), for synchronizing the clocks of a wireless sensor network. This algorithm is based on a cascade of two consensus algorithms, whose main task is to average local information. The proposed algorithm has the advantage of being totally distributed, asynchronous, robust to packet drop and sensor node failure, and it is adaptive to time-varying clock drifts and changes of the communication topology. In particular, a rigorous proof of convergence to global synchronization is provided in the absence of process and measurement noise and of communication delay. Moreover, its effectiveness is shown through a number of experiments performed on a real wireless sensor network.

Key words: Consensus, Time synchronization, drift compensation, networked systems, node failure.

1 Introduction

Recent technological advances in miniaturization and wireless communication are promoting the use of a large number of networked devices for fine-grain ambient monitoring and control. In particular, a special class of these networked systems, known as wireless sensor networks (WSNs), have gained interest and popularity for being self-configuring, rather inexpensive, and useful for a very wide range of possible applications from building climate control to target tracking, from environment monitoring to industrial automation. WSNs are networks of small programmable devices with computational, sensing and memory capabilities that can communicate with their neighbors via a wireless channel. In many of the aforementioned applications, it is essential that the nodes act in a coordinated and synchronized fashion. In particular, many applications require a global clock synchronization, that is all the nodes of the network need to share a common notion of time.

⋆ This work has been partially supported by European Union project FP7-ICT-223866-FeedNetBack, by the Italian CaRiPaRo Foundation project "Wise-Wai". Corresponding author Luca Schenato Ph. +390498277925

Email addresses: schenato@dei.unipd.it (Luca Schenato), f.fiorentin@gmail.com (Federico Fiorentin).

However, global clock synchronization is particularly challenging in the context of wireless sensor networks for several reasons. The first reason is that the nodes cannot communicate directly with each other but they have to do it via multi-hop communication. Therefore, it is not possible to choose a reference node to which all other nodes can be directly synchronized to. Secondly, wireless communication is often unreliable and it is subject to unpredictable packet losses. Finally, wireless sensor networks are made of inexpensive devices that often incur failure, replacement or relocation, thus creating a dynamic communication topology both in terms of communication links and number of nodes. Therefore, many dedicated strategies and protocols have been proposed to address the problem of time synchronization in WSNs, as surveyed in [25][28][8][24].

A common approach to deal with the multi-hop nature of a sensor network is to organize the network into a rooted tree as in the Time-synchronization Protocol for Sensor Networks (TPSN) [10] and in the Flooding Time Synchronization Protocol (FTSP) [16]. Initially one node is elected to be the global clock reference, then a spanning tree rooted at that node is built. Afterwards, each node synchronizes itself with its parent by compensating its offset, i.e., the instantaneous clock difference, and its relative clock drift, i.e., the relative clock speed, using its parent clock readings as the direct reference. This approach suffers from two major limitations. The first limitation arises because if the root node or a non-leaf node dies, then a new root-election or parent-discovery procedure needs to be initiated, thus adding substantial overhead to the code and potentially long periods of network de-synchronization. The second limitation is due to the fact that geographically close nodes might be far in terms of path-length in the constructed tree, and this distance is directly related to the clock synchronization error. This is particularly harmful for many applications such as target tracking or time division medium access (TDMA) scheduling, for which it is really important that clock errors between one node and the others degrade sufficiently smoothly as a function of geographic distance.

Another approach is to divide the network into interconnected single-hop clusters, as suggested in the Reference Broadcast Synchronization (RBS) scheme [6]. In this protocol, within every cluster a reference node is selected to synchronize all the other nodes. The reference nodes of different clusters are synchronized together and act as gateways by converting the synchronized clock readings of one cluster into the consistent local clock readings of another cluster when needed. Like the TPSN, also RBS suffers from large overhead necessary to divide the network into clusters and to elect the reference nodes. Moreover it is somewhat fragile to node failures.

The last approach is to have a fully distributed communication topology where there are no special nodes such as roots or gateways, and all nodes run exactly the same algorithm. This approach has the advantage of being very robust to node failure and new node appearance, but requires specific algorithms for the synchronization since there is no reference node. One example of a completely distributed synchronization strategy is the Reachback Firefly Algorithm (RFA), inspired by the firefly synchronization mechanism proposed in [31]. In this algorithm every node periodically broadcasts a synchronization message and anytime a node hears a message, then it slightly shifts forward the phase of its internal clock which is used to schedule the periodic message broadcasting. Eventually all nodes will advance their phase till they are all synchronized, i.e., they "fire" a message at the same time. This approach however does not compensate for clock drift, therefore the firing period needs to be rather small. Alternatively, in [26] the authors proposed a Distributed Time Synchronization Protocol (DTSC) which is fully distributed and compensates the clock drifts. This protocol is formulated as a distributed gradient descent optimization problem as shown in [12]. Recently, different authors proposed the use of consensus algorithms, i.e., algorithms whose goal is to have all agents of a network to agree upon a common variable, for distributed time syncronization. For example, in [15] the authors propose a consensus-based algorithm to compensate the clock offsets but not the clock drifts, while in [23] the authors studied distributed frequency compensation, i.e., clock drift compensation, for phase locked loops (PLLs) using consensus algorithms. More recently, in [4] a proportional-intergral (PI) consensus-based controller was proposed which consists in a second-order consensus algorithm to compensate both clock offsets and clock drifts.

In this paper, a new synchronization protocol for WSN, named Average TimeSynch (ATS), is proposed based on the cascade of two consensus algorithms where the first consensus synchronizes clock speeds and the second synchronizes the clock offsets. The original contribution of this paper is twofold. The first being that, as compared to other fully distributed algorithms that compensate for both drift and offset [26][27], here a rigorous proof of convergence is provided under the assumptions of absence of process noise, measurement noise, and propagation delay. As compared to the proportional-integral (PI) time-synchronization algorithm [4] which requires a pseudo-synchronous implementation, the ATS is totally asynchronous, thus being resilient to packet losses, and node failure, replacement or relocation. The second main contribution is the presentation of extensive experimental results from a real WSN including a comparison with the FTSP protocol [16], which is considered the de-facto standard for time synchronization in WSN. Moreover, the proposed algorithm is adaptive to slowly time-varying clock drifts and requires minimal memory and computational resources. Preliminary results about this work have appeared in [2],[21] and [20].

The paper is organized as follows. Section 2 introduces some mathematical tools and definitions that will be instrumental for the proof of convergence of the proposed ATS protocol. Section 3 introduces a model for the clock dynamics and formally defines the synchronization objectives, while Section 4 describes the ATS protocol and provides a formal proof of convergence under ideal conditions. Finally, Section 5 describes the experimental apparatus of a typical WSN and presents a set of experiments that test the proposed algorithm and compare it with an alternative protocol available in the literature. Section 6 briefly summarizes the results obtained and proposes potential research directions. To improve readability, the proofs of the theorems are reported in the Appendix section at the end of this paper, unless otherwise stated.

2 Mathematical Preliminaries

This section introduces the necessary mathematical tools to prove convergence of the ATS protocol proposed in the next sections. In particular, some well known results about the positivness of the product of stochastic matrices based on graph properties of their associated graphs are first recalled (Theorem 1), and then the property of a Lyapunov function suitable for stochastic matrices is given (Lemma 2). These two results are finally employed in the main theorem of this section to provide convergence conditions for time-varying systems subject to exponential decaying disturbances (Theorem 3).

Communication in a WSN is modeled as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, N\}$ represents the nodes in the WSN and the edge set $\mathcal{E}$ represents the available directed communication links, i.e., $(i, j) \in \mathcal{E}$ if node j sends information to node i. The symbol $\mathcal{N}_i = \{j \,|\, (i, j)\} \in \mathcal{E}, i \neq j\}$ represents the set of neighbors of i, and $|\mathcal{N}_i|$ its cardinality. A matrix $P \in \mathbb{R}^{N\times N}$ is said to be *stochastic* if $P_{ij} \geq 0$ and $\sum_{j=1}^{N} P_{ij} = 1, \forall i \in \mathcal{N}$, where P_{ij} indicates the $i-j$ entry of matrix P. To simplify notation the previous constraints will be denoted as $P \geq 0$, and $P\mathbf{1} = \mathbf{1}$, where $\mathbf{1} = [1\,1 \cdots 1]^T \in \mathbb{R}^N$. Given a stochastic matrix P its *associated graph* is defined as $\mathcal{G}_P = (\mathcal{N}, \mathcal{E}_P)$ where $(i, j) \in \mathcal{E}_P$ if and only if $P_{ij} > 0$. A stochastic matrix P is said to be *consistent* with a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, denoted as $P \sim \mathcal{G}$, if $\mathcal{G}_P \subseteq \mathcal{G}$, i.e., $\mathcal{E}_P \subseteq \mathcal{E}$. The *union* of two graphs is defined as $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 = (\mathcal{N}, \mathcal{E})$ where $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. The symbol $\mathbb{G}_{sl} = \{\mathcal{G} = (\mathcal{N}, \mathcal{E}) \,|\, (i, i) \in \mathcal{E}, \forall i \in \mathcal{N}\}$ indicates the set of graphs with all self-loops. A graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$ is said to be *strongly connected* if there is a path between any pair of nodes $i, j \in \mathcal{N}$, i.e., there exist $k_1, \ldots k_\ell \in \mathcal{N}$ such that $(i, k_1), (k_1, k_2), \ldots, (k_\ell, j) \in \mathcal{E}$, and it is said to be *complete* if $(i, j) \in \mathcal{E}, \forall i, j \in \mathcal{N}$, i.e., all nodes are directly connected. Note that $\mathcal{G}_P$ is complete if and only if $P > 0$.

From now on it is assumed that the WSN connectivity graph $\mathcal{G}_{WSN} = (\mathcal{N}, \mathcal{E})$ (i) is *undirected*, i.e., $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, (ii) it contains all self loops, i.e., $\mathcal{G} \in \mathbb{G}_{sl}$, and (iii) it is strongly connected. These hypotheses are realistic since the wireless channel is symmetric, each node has access to its own information, and the graph is not disconnected. However, the channel is only *half-duplex*, i.e., two nodes cannot transmit and receive at the same time. As a consequence, the communication protocols that are suggested later in this work, such as the broadcast communication, will give rise to non-symmetric stochastic matrices whose associated graphs are directed.

The previous definitions are instrumental in the next theorem to provide sufficient conditions that guarantee strictly positiveness of products of time-varying stochastic matrices. The positiveness is a useful property to prove convergence of time-varying consensus algorithms. The proof of this theorem and more general conditions can be readily derived from [18] and [3]. Similar results are also available in the context of convergence of Markov Chains [22].

Theorem 1 *Consider the sequence of stochastic matrices* $\{P_k\}_{k=0}^{\infty}$ *such that* $\mathcal{G}_{P_k} \in \mathbb{G}_{sl}$*. If there exist integers* $0 = h_0 < h_1 < \ldots < h_\ell < \ldots$*, where* $h_{\ell+1} - h_\ell < H < \infty$*, such that* $\mathcal{G}_\ell := \cup_{m=h_\ell}^{h_{\ell+1}} \mathcal{G}_{P_m}$ *is strongly connected for all* $\ell = 0, 1, \ldots$*, then there exists a positive integer* K *such that* $Q_\ell = P_{(\ell+1)K-1} \ldots P_{\ell K+1} P_{\ell K} > 0$ *for all* ℓ*.*

It was shown in [18] that the previous condition on the graph sequence $\mathcal{G}_{P_k}$ is also necessary, i.e., it is the weakest condition to have $Q_\ell > 0$. In other words, the theorem states that the communication graph does not need to be connected at all time instants, but only over an arbitrarily long but finite time window.

In order to prove the main theorem of this section, the next technical lemma introduces a Lyapunov function and its property in the context of systems whose dynamics is given by a stochastic matrix:

Lemma 2 *Let* $x \in \mathbb{R}^N$ *and* $P \in \mathbb{R}^{N\times N}$ *a stochastic matrix. Let* $V(x) = \max(x) - \min(x)$*, then*

$$V(Px) \leq (1 - \max_{j=1}^{N} \min_{i=1}^{N} \{P_{ij}\}) V(x)$$

where $\max(x) = \max_{i=1}^{N}\{x_i\}$ *and* $\min(x) = \min_{i=1}^{N}\{x_i\}$*.*

The proof can be found in [32]. It is important to note that $(\max_{j=1}^{N} \min_{i=1}^{N}\{P_{ij}\}) > 0$ if and only if there is at least one column of P whose elements are all positive, i.e., if there is at least one node that is directly connected to all the others.

It is now possible to provide a general theorem for convergence of linear iterative stochastic matrices subject to exponentially decaying disturbances.

Theorem 3 *Let us consider the following linear system*

$$x(k+1) = \big(P(k) + \Delta(k)\big)x(k) + v(k) \qquad (1)$$

where $x(k) \in \mathbb{R}^N$*,* $P(k) \in \mathbb{R}^{N\times N}$ *are stochastic matrices, and* $\Delta(k) \in \mathbb{R}^{N\times N}$ *and* $v(k) \in \mathbb{R}^N$ *are unknown and* $||\Delta(k)||_\infty \leq a\rho^k$*, and* $||v(k)||_\infty \leq a\rho^k$ *for some* $a > 0$ *and* $\rho \in [0, 1)$*. If there exists an integer* K *such that* $Q_\ell = P_{(\ell+1)K-1} \ldots P_{\ell K+1} P_{\ell K} \geq \epsilon > 0$ *for all* $\ell = 0, 1, \ldots$*, then there exists* $\alpha \in \mathbb{R}$ *such that*

$$\lim_{k\to\infty} x(k) = \alpha\mathbf{1}$$

exponentially fast.

The previous theorem states that if the sequence of the consensus matrices $P(k)$ gives rise to a connected graph over an arbitrary but finite time window of length K, even in the presence of both multiplicative and additive but exponentially decaying disturbance, then all nodes will eventually converge to consensus exponentially fast where the consensus parameter α is constant. Consensus subject to multiplicative and additive disturbances has also been addressed in [13], but assuming a special case of Laplacian-based consensus matrices $P(k)$ which are symmetric. As explained at the beginning of this section, the half-duplex nature of the wireless channel leads to non-symmetric consensus matrices, therefore results of [13] cannot be used. The difficulty of dealing with non-symmetric consensus matrices has been well explained in [17]. Another notable work in the context of consensus algorithms driven by external non-vanishing inputs can be found in [14], but those results are applicable only for additive disturbance with identical entries.

It is important to remark that an exponential decaying disturbance is not a necessary condition for convergence to consensus, i.e., $\lim_{k\to\infty} x(k) = \alpha(k)\mathbf{1}$. Indeed, even non vanishing disturbances can lead to consensus, as shown in [14], for example. However, proof of convergence subject to more general disturbances is much more challenging and it is out of the scope of this work.

Implicitly, the theorem also provides an upper bound for the rate of convergence which is given by $\max(\sqrt[K]{1-\epsilon},\rho)$. In practice, the bound $\sqrt[K]{1-\epsilon}$ is very loose since it is based on a worst-case scenario, and the convergence rate is in general much faster. Better convergence rate bounds can be obtained by considering randomized consensus matrices as in [7]. The algorithm proposed in this work is suitable also for randomized communication protocols, but the corresponding mathematical tools to prove convergence need to be adapted. On the other hand, the sufficient conditions stated in the theorem to guarantee convergence are very mild, since no specific order of $P(k)$ is required. This will be particularly useful to prove convergence of the proposed algorithm, since in WSN it is very difficult to enforce a predefined synchronized scheduling sequence of $P(k)$, while it is easy to guarantee the hypotheses of the theorem.

3 Model

This section provides a mathematical model for wireless sensor network clocks. Every node i in a WSN has its own local clock whose first order dynamics is given by:

$$\tau_i(t) = \alpha_i t + \beta_i \tag{2}$$

where τ_i is the local clock reading, α_i is the local clock drift which determines the clock speed, and β_i is the local clock offset. Since the absolute reference time t is not available to the nodes, it is not possible to compute the parameters α_i and β_i. However, it is still possible to obtain indirect information about them by comparing the local clock of one node i with respect to another clock j. In fact, if Eqn. (2) is solved for t, i.e., $t = \frac{\tau_i - \beta_i}{\alpha_i}$ and it is substituted into the same equation for node j,

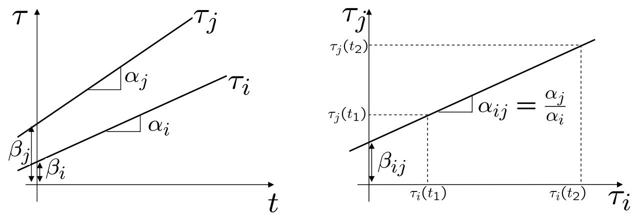

Fig. 1. Clocks dynamics as a function of absolute time t *(left)*, and relative to each other *(right)*.

then it follows:

$$\begin{aligned}\tau_j &= \frac{\alpha_j}{\alpha_i}\tau_i + (\beta_j - \frac{\alpha_j}{\alpha_i}\beta_i)\\ &= \alpha_{ij}\,\tau_i + \beta_{ij}\end{aligned} \tag{3}$$

which is still linear as shown in the right panel of Fig. 1. The goal is to synchronize all the nodes with respect to a *virtual reference clock*, namely:

$$\bar{\tau}(t) = \bar{\alpha}t + \bar{\beta} \tag{4}$$

Every local clock keeps an estimate of the virtual time using a linear function of its own local clock:

$$\hat{\tau}_i(t) = \hat{\alpha}_i\tau_i(t) + \hat{o}_i \tag{5}$$

Our goal is to find $(\hat{\alpha}_i, \hat{o}_i)$ for every node in the WSN such that:

$$\lim_{t\to\infty} \hat{\tau}_i(t) - \bar{\tau}(t) = 0, \qquad i = 1,\ldots,N \tag{6}$$

where N is the total number of nodes in the WSN. Therefore, if the previous expression is satisfied, then all nodes will have a common global reference time given by the virtual reference clock. The previous expression can be rewritten by first substituting Eqn.(2) into Eqn.(5) to get:

$$\hat{\tau}_i(t) = \hat{\alpha}_i\alpha_i t + \hat{\alpha}_i\beta_i + \hat{o}_i \tag{7}$$

Therefore Eqn. (6) is equivalent to

$$\lim_{t\to\infty} \alpha_i\hat{\alpha}_i(t) = \bar{\alpha}, \tag{8}$$

$$\lim_{t\to\infty} \hat{o}_i(t) + \beta_i\hat{\alpha}_i(t) = \bar{\beta}, \quad i = 1,..,N \tag{9}$$

Before moving to the next section which presents how the ATS protocol updates $(\hat{\alpha}_i, \hat{o}_i)$ to satisfy the previous expression, it is important to make a few remarks. The first regards the clock modeling of Eqn.(1). In reality the parameters α_i, β_i are time varying due to ambient conditions or aging, however the ATS protocol is able to track these changes as long as the synchronization period is shorter than the time constants relative to the typical variations of these parameters.

The second point is that the virtual reference clock is a fictitious clock and is not fixed a priori. In fact, the values of its parameters $(\bar{\alpha}, \bar{\beta})$ are not critical [1], since what it is really relevant is that all clocks converge to *one* common virtual reference clock. Indeed, as it will be shown in the next section, the parameters $(\bar{\alpha}, \bar{\beta})$ to which the local clock estimates converge depend on the initial condition and the communication topology of the WSN.

The last remark is that by using the MAC-layer time-stamping [30] available in many sensor network devices, it can be safely assumed that the reading of the local clock $\tau_i(t_1)$ at the transmitting node and the reading of the local clock $\tau_j(t_2)$ at the receiving node are instantaneous, i.e., $t_1 = t_2$ (see Section 5.3 in [27] for a detailed description). If this is not the case, the proposed synchronization protocol cannot be used as it is and needs to be modified to cope with packet delivery delay (see, for example, [28] and [26] for transmission delay compensation).

4 The ATS protocol

The Average TimeSync protocol includes three main parts: the relative drift estimation, the drift compensation, and the offset compensation. Moreover, it is important to specify the communication schedule to guarantee convergence.

4.1 Communication protocol: pseudo-periodic broadcast

Here, a simple deterministic communication protocol is proposed which satisfies conditions of Theorem 1, however many others are possible as long as all nodes transmit sufficiently often, such as the randomized broadcast communication proposed in [7]. Here each node i is assumed to periodically transmit a packet to all its neighbors with a synchronization period equal to T, i.e., the transmission instants t_k^i are defined as $\tau_i(t_\ell^i) = \ell T$ or equivalently

$$t_\ell^i = \frac{\ell T - \beta_i}{\alpha_i} = \ell\, T_i + \bar{\beta}_i \qquad (10)$$

As mentioned above, packets are assumed to be instantaneously received by its neighbors. This protocol is referred to as *pseudo-periodic broadcast* since each node broadcasts its message at every period T based on its own clock, which in reality corresponds to a period T_i. However, since each α_i is slightly different, over time the order of transmissions as well the relative interarrival intervals change, thus the name pseudo-periodic. Let us consider the set of all ordered transmissions of all nodes $\mathbb{T} = \cup_i \cup_\ell \{t_\ell^i\} = \{\bar{t}_0, \bar{t}_1, \ldots\}$, where $\bar{t}_k$ are the ordered events, i.e $\bar{t}_k < \bar{t}_{k+1}$ [2]. Let k_ℓ such that $\bar{t}_{k_\ell} = t_\ell^m$, where $m = \text{argmin}_i \alpha_i = \text{argmax}_i T_i$, i.e., the slowest clock, and without loss of generality it is assumed that $\beta_m = 0$. It should be clear that $t_\ell^m = \ell T/\alpha_{min} = \ell T_{max}$ and $N \leq k_{\ell+1} - k_\ell \leq \lceil \alpha_{max}/\alpha_{min} \rceil N$, where $\alpha_{min} = \min_{i=1}^N \{\alpha_i\}, \alpha_{max} = \max_{i=1}^N \{\alpha_i\}$ and $\lceil \cdot \rceil$ indicates the smallest integer greater or equal than its argument. Also $\forall \ell, \forall j$ there exist integers h, s such that $k_\ell \leq h \leq k_{\ell+1}$ and $\bar{t}_h = t_s^j$, i.e., each node j transmits at least once in the time window of period T_{max} defined by two consecutive transmissions of the slowest clock.

This is indeed only a sufficient condition that satisfies the hypotheses of Theorem 1. However, as mentioned in Section 2, in practice the necessary conditions for asymptotic convergence are that the communication graph is connected and that each node transmits sufficiently often. In fact, occasional packet drops or temporary failures of a node do not affect asymptotic convergence, although they might degrade the speed of convergence.

4.2 Relative Drift Estimation

This part of the protocol is concerned with deriving an algorithm that estimates the relative drift of each clock i with respect its neighbor j. Every node i tries to estimate the relative drifts $\alpha_{ij} = \frac{\alpha_j}{\alpha_i}$ with respect to all its neighbor nodes $j \in \mathcal{N}_i$. This is accomplished by writing the current local time $\tau_j(t_\ell^j)$ of node j into a broadcast packet, then the node i that receives this packet immediately records its own local time $\tau_i(t_\ell^j)$. As discussed in the previous section, we can assume that the readings of the two local clocks are instantaneous since MAC-layer time-stamping is used. Therefore, node i records in its memory the pair $(\tau_{ij}^{old}, \tau_j^{old}) = \big(\tau_i(t_\ell^j), \tau_j(t_\ell^j)\big)$. When a new packet from node j arrives to node i, the same procedure is applied to get the new pair $\big(\tau_i(t_{\ell+1}^j), \tau_j(t_{\ell+1}^j)\big)$, as shown in the right panel of Fig.1. From these two pairs, in principle it is possible to directly compute the relative drift α_{ij}. However, due to unavoidable measurement and quantization errors, the estimate of the values

[1] In practice, $\bar{\alpha}$ should not be too different from the clock speeds. Indeed, it is possible to show that in the absence of external disturbance the ATS would lead to $\bar{\alpha} \in [\min_{i=1}^N\{\alpha_i\}, \max_{i=1}^N\{\alpha_i\}]$, i.e., within the range of the speeds of the network clocks.

[2] The assumption of non-simultaneous events is not critical for the proposed algorithm but is convenient for simplifying the proofs of the following theorems.

α_{ij} is performed via a low-pass filter as follows:

$$\left.\begin{aligned} (\tau_{ij}^{new}, \tau_j^{new}) &= \left(\tau_i(t_\ell^j), \tau_j(t_\ell^j)\right) \\ \eta_{ij}(t^+) &= \rho_\eta\, \eta_{ij}(t) + (1-\rho_\eta) \frac{\tau_j^{new} - \tau_j^{old}}{\tau_{ij}^{new} - \tau_{ij}^{old}} \\ (\tau_{ij}^{old}, \tau_j^{old}) &= (\tau_{ij}^{new}, \tau_j^{new}) \end{aligned}\right\}, t = t_\ell^j \quad (11)$$

$$\eta_{ij}(t) = \eta_{ij}(t^+), \quad t \in (t^+, t_{\ell+1}^j] \quad (12)$$

where $\rho_\eta \in (0,1)$ is a tuning parameter, and t^+ indicates the update. If there is no measurement error and the drift is constant, then the variable η_{ij} converges to the variable α_{ij} as stated in the following theorem:

Theorem 4 *Let us consider the update Eqns (11)-(12) where $0 < \rho_\eta < 1$, the transmission events t_ℓ^i are generated according to the pseudo-periodic broadcast of Eqn. (10), and each τ_i evolves according to Eqn. (2). Then*

$$\lim_{t \to \infty} \eta_{ij}(t) = \alpha_{ij} \quad (13)$$

exponentially fast for any initial condition $\eta_{ij}(0)$.

In practice, the parameter ρ_η is used to tune the trade-off between a fast rate of convergence (ρ_η close to zero) and a high noise immunity (ρ_η close to unity). In fact, filtering is necessary because the quantity $\frac{\tau_j(t_2)-\tau_j(t_1)}{\tau_i(t_2)-\tau_i(t_1)}$ in a real scenario is not constant but it is slowly time-varying and affected by quantization noise. It is important to remark that it is not necessary to perform the update at a fixed frequency, i.e., the packet inter-arrival $t_2 - t_1$ can vary, thus making this algorithm particularly useful for asynchronous and lossy communication. The other important advantage of this algorithm is that it requires little memory. In fact, each node i needs to store only the $|\mathcal{N}_i|$ relative drift estimates η_{ij} and the most recent local clock readings $(\tau_{ij}^{old}, \tau_j^{old})$. Since the size of the set $\mathcal{N}_i$ is in general small even for large networks, this algorithm is also rather scalable.

4.3 Drift Compensation

This part of the algorithm is the core of the Average TimeSync protocol, as it forces all the nodes to converge to a common virtual clock rate, $\bar{\alpha}$, as defined in Eqn. (4). The main idea is to use a distributed consensus algorithm based only on local information exchange. In the consensus algorithms any node keeps its own estimate of a global variable, and it updates its value by averaging it with the estimates of its neighbors (see for example surveys [19],[11]). The algorithm is very simple since every node stores only its own virtual clock drift estimate $\hat{\alpha}_i$, defined in Eqn. (5). As soon as a node i receives a packet from node j at time t_ℓ^j, it updates its estimate $\hat{\alpha}_i$ as follows:

$$\hat{\alpha}_i(t^+) = \rho_v \hat{\alpha}_i(t) + (1-\rho_v)\eta_{ij}(t)\, \hat{\alpha}_j(t), \quad t = t_\ell^j, i \in \mathcal{N}_j \quad (14)$$

where $\hat{\alpha}_j$ is the virtual clock drift estimate of the neighbor node j. The initial condition for the virtual clock drifts of all nodes is set to $\hat{\alpha}_i(0) = 1$. It is now shown that the previous update rule will lead to $\lim_{t\to\infty} \hat{\alpha}_i(t)\alpha_i = \bar{\alpha}$, i.e., all clock estimate $\hat{\tau}_i(t)$ will eventually have the same speed.

Theorem 5 *Consider the drift update equation given by Eqn. (14) with initial condition $\hat{\alpha}_i(0) = 1$ and $0 < \rho_v < 1$, where $\eta_{ij}(t)$ are updated according to Eqns (11)-(12) and t_ℓ^j are defined in Eqn. (10). Then*

$$\lim_{t\to\infty} \hat{\alpha}_i(t)\alpha_i = \bar{\alpha}, \quad \forall i$$

exponentially fast, where $\bar{\alpha} \in \mathbb{R}$.

4.4 Offset compensation

According to the previous analysis, after the drift compensation algorithm is applied, all local virtual clock estimators will eventually have the same drift, i.e., they will run at the same speed. At this point it is only necessary to compensate for possible offset errors. Once again, a consensus algorithm is employed to update the estimated clock offset, previously defined in Eqn. (5), as follows:

$$\hat{o}_i(t^+) = \hat{o}_i(t) + (1-\rho_o)(\hat{\tau}_j(t) - \hat{\tau}_i(t)), \ t = t_\ell^j, i \in \mathcal{N}_j \quad (15)$$

where $\hat{\tau}_j$ and $\hat{\tau}_i$ are computed at the same time instant $t = t_\ell^j$, and $\hat{\tau}_i(t_\ell^i) = \hat{\alpha}_i(t_\ell^i)\tau_i(t_\ell^i) + \hat{o}_i(t_\ell^i)$. Between communication instants, i.e., for $t \neq t_\ell^j$, both $\hat{o}_i(t)$ and $\hat{\alpha}_i(t)$ are kept constant. Informally speaking, each node computes the instantaneous estimated clock difference $\hat{\tau}_j(t) - \hat{\tau}_i(t)$ and tries to update its offset $\hat{o}_i$ in order to reduce this difference. The next theorem shows the convergence of this algorithm:

Theorem 6 *Consider the offset update equation given by Eqn. (15) with initial condition $\hat{o}_i(0) = 0$ and $0 < \rho_o < 1$, where $\hat{\tau}_i$, t_ℓ^j, η_{ij} and $\hat{\alpha}_i$ are defined in Equations (5), (10), (11)-(12), and (14), respectively. Then*

$$\lim_{t\to\infty} \hat{\tau}_i(t) = \hat{\tau}_j(t), \quad \forall i,j \in \mathcal{N}$$

exponentially fast.

It is important to remark that the offset compensation does not need to wait for the drift compensation to synchronize all clock speeds, but it is applied simultaneously, thus providing faster convergence and better performance as shown below in Fig. 7 of Section 5.4.

5 Experimental Results

5.1 Experimental testbed

The ATS protocol has been implemented on a real WSN of 35 Tmote Sky nodes produced by the MoteIv Inc [30]. Each Tmote Sky module has the size of a cards deck and is provided with a 8Mhz 16bit microcontroller MSP430 by Texas Instrument, 10k RAM and 48k Flash in terms of memory, a 250kbps 2.4GHz IEEE 802.15.4 Zigbee-compliant Chipcon Wireless Transceiver CC2420, additional electronics for input-output interfacing, and few sensors. These modules can be powered through a USB port or with a pair of AA batteries, and they can be programmed via TinyOS [29], an operating system specifically designed for WSN to maintain low complexity and low code footprint. The microcontroller MSP430 is provided with a digitally controlled oscillator (DCO) running at $8MHz$ which provides a potential clock resolution of $T_{DCO} = 1/8Mhz = 0.125\mu s$, however it needs to be calibrated using a slower external crystal oscillator (ECO) running at $32768Hz$. Moreover, during idle mode for low power consumption, the DCO is switched off and operations are based on the ECO. Since many important applications run mostly in idle mode, the ECO is used for testing the ATS protocol, therefore the maximal resolution will depend on the ECO resolution which is one oscillation period, called *tick*, where $1\,\text{tick} = 1/32768Hz = 30.5\mu s$. In other words, it is not possible to distinguish synchronization errors that are smaller than $30.5\mu s$ since each local clock $\tau_i(t)$ is given by an integer counter that is incremented by one unit at every ECO cycle. An important feature of the radio chip CC2420 is the so called *MAC-layer time-stamping*, which allows each node to read the local clock at the beginning of the transmission or reception of the first bit, namely the Start Frame Delimiter (SFD), of a message. This mechanism strongly reduces potential unpredictable delays between the readings of the transmitting and receiving node, as also explained in [27]. Although a mismatch between transmission and reception times still exists due to the operating system and the detection of the SFD, it has been experimentally observed to be negligible as compared to the ECO resolution, therefore communication delay can be safely neglected, which is a major assumption of the proposed ATS protocol.

In order to test the ATS protocol, a 7x5 grid was built for a total of 35 nodes as shown in Figure 2. Since most nodes were all in communication range of each other, they were forced to communicate only with close neighbors, i.e., messages received from distant nodes were neglected. Such a topology has a diameter of 10 hops, i.e., the worst-case minimum distance in terms of communication steps between two nodes, as for example the bottom-left node and the top-right node. Each node was running the same ATS protocol, i.e., there was no base station or predefined reference node. The protocol parameters were set to $\rho_o = \rho_v = 0.5$ and $\rho_\eta = 0.2$. All nodes were polled by an additional external node every 5 seconds, i.e., they were asked to report the value of their estimated time $\hat{\tau}_i(t)$ at the same time instant t to evaluate the instantaneous clock synchronization errors. The nodes adopted the pseudo-periodic communication scheme described in Section 4.1 for different synchronization periods. An average packet loss around $5-10\%$ was observed probably due to packet collision. The remainder of this section is dedicated to presenting the results of the ATS protocol under different scenarios.

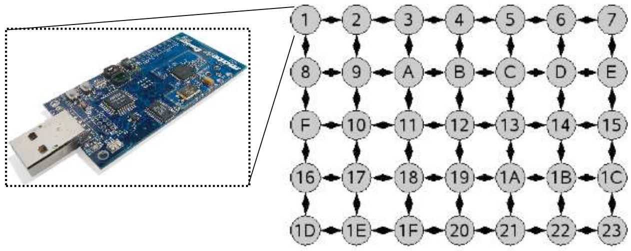

Fig. 2. Wireless sensor network communication topology of 35 Tmote Sky nodes including a close-up of the Tmote Sky device.

5.2 Dynamic topology

This experiment, shown in Fig. 3, was intended to study the robustness properties of the ATS protocol subject to node failure and node replacement, as well as the performance in terms of convergence speed and steady state synchronization error. The synchronization period was set to $30s$ which is sufficiently large to exhibit the effects of different clock speeds. The experiment was run for about 2.5 hours and presents 4 different regions of operation indicated by the letters A,B,C,D which model potential node failure or the addition of new nodes. In Region A all nodes are turned on simultaneously with random initial conditions of their local clocks. After about 120 polling cycles, corresponding to $120 \cdot 5s = 10min$ and about $120/6 = 20$ packets sent per node, the synchronization error between any two nodes is included between $\pm 10\,ticks$, i.e., the maximum error is smaller that $20ticks = 600\mu s$, i.e., well below one millisecond. At the beginning of Region B, about 40% of the nodes chosen at random in the grid are switched off and then switched on at different random times. Once a node is switched on, it starts updating its estimated time $\hat{\tau}_i(t)$ using the ATS protocol but does not transmit any message for the first three synchronization periods to avoid to inject large disturbances into the already synchronized network, and then it starts transmitting and receiving messages equally. The plot in Fig. 3 clearly shows that the nodes get synchronized as soon as they are turned on without perturbing the overall network behavior. At the beginning Region C, about 20% of the nodes turned off their radio, i.e., they stopped updating their parameters

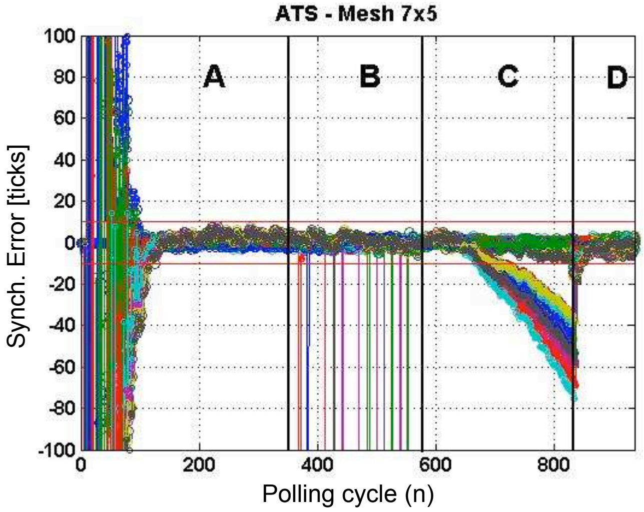

Fig. 3. Synchronization error $\hat{\tau}_i - \hat{\tau}_j$ as a function of time for the 7x5 WSN grid. Polling period is $5s$ and synchronization period is $T = 30s$. Region A: all nodes are on. Region B: 40% of the nodes are turned off and then turned on at random times. Region C: 20% of the nodes turned off their radio. Region D: the nodes turned on again the radio.

$\eta_{ij}, \hat{\alpha}_i, \hat{o}_i$, so their estimated time $\hat{\tau}_i$ started drifting away from the rest of the synchronized grid due to different internal clock speeds. At the beginning of Region D, their radios are turned on again and after a short transient the nodes quickly synchronize again.

5.3 *Comparison between ATS and FTSP*

This experiment compared the performance of the proposed ATS protocol with the FTSP by [16], for which there is a freely available implementation for TinyOS in [9]. The FTSP is considered the de-facto standard for time synchronization in WSN since it has been shown to be resilient to dynamic changes in the communication topology and to compensate different clock drifts, therefore many newly proposed algorithms are compared against it.

Fig. 4 shows the performance obtained under the same conditions for a 3x3 WSN grid with synchronization period $T = 60s$, which indicates a slightly better performance of the ATS protocol and the absence of big sporadic errors as compared to the FTSP.

5.4 *Effect of node distance and synchronization period*

These sets of experiments were designed to explore the performance of ATS protocol as a function of relative distance of two nodes in terms of the number of communication hops, and as a function of the synchronization period. In Fig. 5 it has been displayed the average synchronization error at steady state for the 7x5 WSN grid

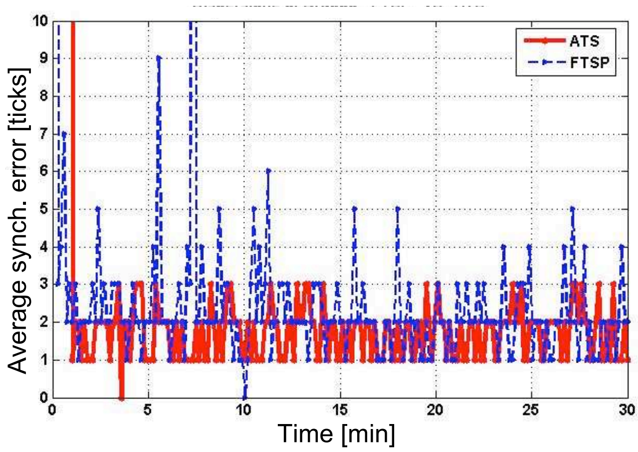

Fig. 4. Performance comparison between the ATS protocol and the FTSP by [16]: maximum synchronization error $\max_{i,j} |\hat{\tau}_i - \hat{\tau}_j|$ as a function of time between any two nodes for a 3x3 WSN grid with synchronization period $T = 60s$.

relative to the node in position (1,1) with synchronization period of $T = 30s$. The figure clearly shows that the synchronization error gradually increases as a function of the hop distance and that the average error between single-hop distance nodes is smaller than 1 tick, i.e., close to the limit of the clock resolution. Interestingly, although the synchronization error increases with hop-distance, the synchronization error between adjacent nodes is only weakly affected by network size, thus making ATS protocol particularly suitable for TDMA communication scheduling in large networks. This observation is consistent with recent results on performance scaling for grids and planar networks [5][1].

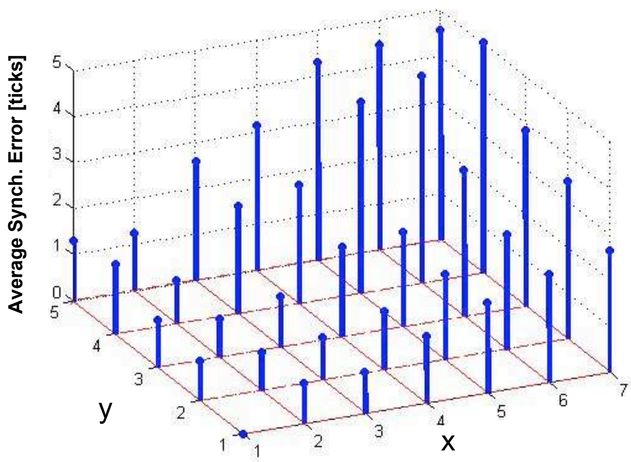

Fig. 5. Average synchronization error of each node from node $i = 1$ as a function of grid location for the 7x5 WSN with synchronization period $T = 30s$.

Fig. 6 shows the averagrecte steady state synchronization error among all nodes measured in a 3x3 WSN as a function of different synchronization periods ranging from $T = 7s$ to $T = 14\,min$. Obviously, performance degrades for longer synchronization periods, however it exhibits a remarkable linear dependence, thus being very useful for predicting the synchronization error as a function of the synchronization period.

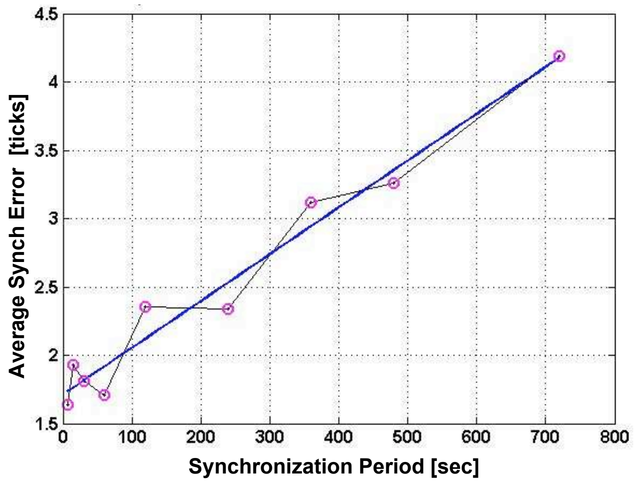

Fig. 6. Average synchronization error between any two nodes in a 3x3 WSN grid as function of the synchronization period from $T = 7s$ to $T = 14\,min$. The straight line represents the best interpolating line.

Finally, Fig. 7 shows the synchronization error for a 3x3 WSN grid for a long synchronization period $T = 4\,min$. It is evident how after every synchronization cycle the clock offsets are almost completely compensated, but the different clock drifts tend to make the clocks diverge between two synchronization cycle. However, the drift compensation part of the ATS protocol slowly learns these different clock speeds and eventually totally compensate them after 6 synchronization cycles.

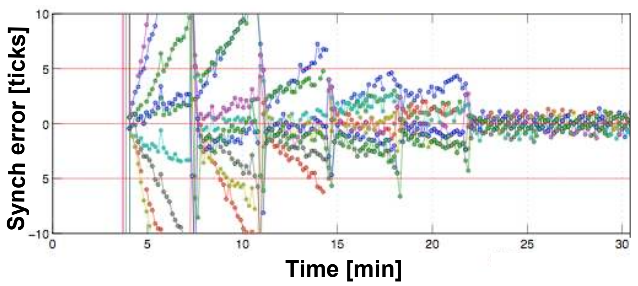

Fig. 7. Synchronization error $\hat{\tau}_i - \hat{\tau}_j$ as a function of time for the 3x3 WSN grid. Polling period is $5s$ and synchronization period is $T = 4\,min$.

6 Conclusions and future work

This paper presented a new synchronization algorithm for WSN, the Average TimeSync protocol, which is based on the cascade of two consensus algorithms whose main idea is to average local information to achieve a global agreement on a specific quantity of interest. The proposed algorithm is fully distributed, asynchronous, includes drift compensation and is computationally light. Moreover, it is robust to dynamic network topologies due, for example, to node failure or replacement. Finally, a thorough set of experiments was presented to show the good performance of the proposed protocol also in realistic scenarios. Future work include the problem of adapting the ATS algorithm for TDMA applications with controlled scheduling of sleeping nodes for very low-power consumption.

Acknowledgements

The authors would like to thank Alessio Basso and Giovanni Gamba for implementing and testing the preliminary versions of ATS protocol on the Tmote Sky nodes, and Sandro Zampieri for his useful discussions.

Appendix

Proof of Theorem 3. The first step is to show that $x(k)$ is bounded, i.e., $||x(k)||_\infty \leq M$ for some $M > 0$. In the following, the infinity norm for vectors and the induced infinite norm for matrices are used, since they are particularly suitable for stochastic matrices, therefore, unless differently stated, the simplified notation $||\cdot|| = ||\cdot||_\infty$ is adopted. In fact, if P is stochastic, then $||P|| = 1$. Moreover:

$$\begin{aligned}||x(k+1)|| &= ||(P(k) + \Delta(k))x(k) + v(k)|| \\ &\leq ||P(k)x(k)|| + ||\Delta(k)x(k)|| + ||v(k)|| \\ &\leq ||P(k)||\,||x(k)|| + ||\Delta(k)||\,||x(k)|| + ||v(k)|| \\ &\leq (1 + a\rho^k)||x(k)|| + a\rho^k \\ &\leq \gamma_{k,0}||x(0)|| + \sum_{m=0}^{k-1} \gamma_{k-1,m} a\rho^m + a\rho^k \qquad (16)\end{aligned}$$

where $\gamma_{k,m} = (1 + a\rho^k)(1 + a\rho^{k-1}) \cdots (1 + a\rho^m)$ for $k \geq m$. The last inequality follows by induction from the solution of the linear time-varying system $z(k+1) = (1 + a\rho^k)z(k) + a\rho^k$ where we used $z(0) = ||x(0)||$, and the fact that $||x(k)|| \leq z(k)$. Now note that

$$\begin{aligned}1 \leq \gamma_{k,m} \leq \gamma_{k,0} &= e^{\log(\gamma_{k,0})} = e^{\sum_{m=0}^{k} \log(1+a\rho^m)} \\ &\leq e^{\sum_{m=0}^{k} a\rho^m} \leq e^{\sum_{m=0}^{\infty} a\rho^m} = e^{a\frac{1}{1-\rho}} = \bar{\gamma}\end{aligned}$$

where it was used the positive monotonicity of the exponential function and the property $\log(1+a) \leq a, \forall a \geq 0$. Using this fact into Eqn. (16) above, it follows:

$$\begin{aligned}||x(k+1)|| &\leq \bar{\gamma}||x(0)|| + \textstyle\sum_{m=0}^{k-1} \bar{\gamma} a\rho^m + \bar{\gamma} a\rho^k \\ &\leq \bar{\gamma}||x(0)|| + \textstyle\sum_{m=0}^{\infty} \bar{\gamma} a\rho^m \leq \bar{\gamma}||x(0)|| + \bar{\gamma} a\frac{1}{1-\rho} = M\end{aligned}$$

which implies that $||x(k)||$ is bounded for all k.

Consider now the function $V(x) = \max(x) - \min(x)$ as defined in Lemma 2. This function will be used as Lyapunov function to prove convergence of the state to a consensus. This function is nonnegative and has the property that $V(x) = 0$ if and only if $x = \alpha\mathbf{1}$ for some $\alpha \in \mathbb{R}$. Moreover, if P is stochastic and $P \geq \epsilon > 0$, then $V(Px) \leq (1-\epsilon)V(x)$ according to Lemma 2.

Next we prove that under the hypotheses of the theorem $\lim_{k\to\infty} V(x(k)) = 0$ exponentially fast. Let $w(k) = \Delta(k)x(k) + v(k)$ and $Q(k+h,h) = P(k+h)\cdots P(h+1)P(h)$, then Eqn. (1) can be written as

$$x(k+1) = P(k)x(k) + w(k). \tag{17}$$

and more generally

$$\begin{aligned} x(k+h+1) &= Q(k+h,k)x(k) + Q(k+h,k+1)w(k) + \ldots \\ &\quad \ldots + Q(k+h,h+k)w(k+h-1) + w(k+h) \\ &= Q(k+h,k)x(k) + \tilde{w}(k+h,k) \end{aligned}$$

Since $||x(k)|| < M$, then $||w_k|| \leq ||\Delta(k)||||x(k)|| + ||v(k)|| \leq a(M+1)\rho^k$. Also note that $Q(k+h,k)$ is still a stochastic matrix being the product of stochastic matrices, therefore $||Q(k+h,k)|| = 1$, from which follows that

$$\begin{aligned} ||\tilde{w}(k+h,k)|| &\leq ||Q(h+k,k+1)||||w(k)|| + \ldots + ||w(k+h)|| \\ &\leq a(M+1)\rho^k\Big(\sum_{\ell=0}^{h}\rho^\ell\Big) \leq \frac{a(M+1)}{1-\rho}\rho^k \end{aligned}$$

If there exists K such that $\lim_{\ell\to\infty} x(K\ell) = \alpha\mathbf{1}$ exponentially, then the previous inequalities implies that that $\lim_{k\to\infty}(x(k+h+1) - x(k)) = 0$ exponentially fast for all $0 \leq h \leq K$. Therefore the study of the convergence can be limited to the subsequence

$$x\big((\ell+1)K\big) = Q\big((\ell+1)K-1,\ell K\big)x(\ell K) + \tilde{w}\big((\ell+1)K-1,\ell K\big)$$

To simplify the notation, define $x_\ell = x(\ell K)$, $\tilde{w}_\ell = \tilde{w}\big((\ell+1)K-1,\ell K\big)$ and $Q_\ell = Q\big((\ell+1)K-1,\ell K\big)$. Note that by hypothesis $Q_\ell \geq \epsilon > 0$, and that $||\tilde{w}_\ell|| \leq \frac{a(M+1)}{1-\rho}\rho^{\ell K} \leq b\rho^\ell$ by previous analysis. Now it is possible to study the evolution of the sequence $V(x_\ell)$:

$$\begin{aligned} V(x_{\ell+1}) &= \max(x_{\ell+1}) - \min(x_{\ell+1}) \\ &= \max(Q_\ell x_\ell + \tilde{w}_\ell) - \min(Q_\ell x_\ell + \tilde{w}_\ell) \\ &\leq \max(Q_\ell x_\ell) + \max(\tilde{w}_\ell) - \min(Q_\ell x_\ell) - \min(\tilde{w}_\ell) \\ &\leq (1-\epsilon)V(x_\ell) + 2b\rho^\ell \end{aligned}$$

where we used the fact that $V(Q_\ell x_\ell) \leq (1-\epsilon)V(x_\ell)$, and that $\max(x) \leq ||x||$ and $\min(x) \geq -||x||$. Let us define $z_{\ell+1} = (1-\epsilon)z_\ell + 2b\rho^\ell$ with initial condition $z_0 = V(x_0)$, then by induction it follows that $V(x_\ell) \leq z_\ell, \forall \ell$. Using standard linear system theory it follows that $\lim_{\ell\to\infty} z_\ell = 0$ exponentially fast since $\epsilon \in (0,1)$ and $\rho \in [0,1)$. From this follows that also $\lim_{\ell\to\infty} V(x_\ell) = 0$. From the considerations above, it follows that $\lim_{k\to\infty} V(x(k)) = 0$ which implies that $x(k) \stackrel{k\to\infty}{\longrightarrow} A \stackrel{def}{=} \{c\mathbf{1} : c \in \mathbb{R}\}$ exponentially fast. If we define $\alpha(k) \stackrel{def}{=} \mathrm{argmin}_{\alpha\in\mathbb{R}}||x(k) - \alpha\mathbf{1}||$ and $u(k) \stackrel{def}{=} x(k) - \alpha(k)\mathbf{1}$, then the previous condition can be restated as $\lim_{k\to\infty} u(k) = 0$ exponentially fast.

Next we show that $\lim_{k\to\infty} \alpha(k) = \alpha$. According to the argument above we can write $x(k) = \alpha(k)\mathbf{1} + u(k)$ where $||u(k)|| \leq c\lambda^k$ for some $c > 0$ and $\lambda \in (0,1)$. Therefore, by substituting it into Eqn. (17), it follows that:

$$\begin{aligned} x(k+1) &= P(k)(\alpha(k)\mathbf{1} + u(k)) + w(k) \\ &= \alpha(k)\mathbf{1} + P(k)u(k) + w(k) \\ x(k+1) &= \alpha(k+1)\mathbf{1} + u(k+1) \end{aligned}$$

from which, by rearranging the different terms, follows that

$$\begin{aligned} &|\alpha(k+1) - \alpha(k)| = ||(\alpha(k+1) - \alpha(k))\mathbf{1}|| \\ &\quad = ||P(k)u(k) + w(k) - u(k+1)|| \\ &\quad \leq ||u(k)|| + ||w(k)|| + ||u(k+1)|| \leq 2c\lambda^k + aM\rho^k \end{aligned}$$

Therefore $|\alpha(k+1) - \alpha(k)|$ satisfies the Cauchy's convergence test, which implies that $\lim_{k\to\infty} \alpha(k) = \alpha$ exponentially fast. Consequently, this implies that $\lim_{k\to\infty} x(k) = \alpha(k)\mathbf{1} + u(k) = \alpha\mathbf{1}$ exponentially fast. This concludes the proof of the theorem. □

Proof of Theorem 4. Note first that from Eqn. (2) it follows that $\frac{\tau_j(t_2)-\tau_j(t_1)}{\tau_i(t_2)-\tau_i(t_1)} = \alpha_{ij}$ for all $t_2 > t_1$. Therefore, we have that

$$\eta_{ij}(t) = \rho_\eta^\ell \eta(0) + \sum_{h=0}^{\ell-1}\rho_\eta^h(1-\rho_\eta)\alpha_{ij} = \rho_\eta^\ell\eta(0) + \alpha_{ij}(1-\rho_\eta^\ell)$$

where $\ell = \lfloor (t-\bar{\beta}_j)/T_j \rfloor$. Since $0 < \rho_\eta < 1$, then $\lim_{t\to\infty} \eta_{ij}(t) = \lim_{\ell\to\infty} \rho_\eta^\ell \eta(0) + \alpha_{ij}(1-\rho_\eta^\ell) = \alpha_{ij}$, and the convergence is exponential. □

Proof of Theorem 5. Consider the new variable $x_i(t) = \alpha_i\hat{\alpha}_i(t)$. By multiplying both sides of Eqn. (14) by α_i, and by adding and subtracting the term $(1-\rho_v)\hat{\alpha}_j(t)\alpha_j$ on the right hand side, it follows that:

$$x_i(t^+) = \rho_v x_i(t) + (1-\rho_v)x_j(t) + (1-\rho_v)\left(\frac{\alpha_i\eta_{ij}(t)}{\alpha_j} - 1\right)x_j(t)$$

which can be written in vector form as

$$x(t^+) = (P(t) + \Delta(t))x(t)$$

where $x = [x_1\ x_2\ \cdots\ x_N]^T$. The matrix $\Delta(t)$ converges to zero exponentially since $\lim_{t\to\infty} \alpha_i \eta_{ij}(t) - \alpha_j = 0$ exponentially fast according to Theorem 4. The matrix $P(t) = P(t_\ell^j) = \bar{P}^j$ is a stochastic matrix whose associated graph $\mathcal{G}_{\bar{P}^j} \in \mathcal{G}_{sl}$ has self-loops and $(i,j) \in \mathcal{E}_{\bar{P}^j}, \forall i \in \mathcal{N}_j$, i.e., it includes all outgoing links of the transmitting node j. According to the pseudo-periodic communication protocol defined above $x(t)$ is constant except for time instants $\bar{t}_k$ defined by the ordered transmission instants t_ℓ^j, therefore it is possible to consider the discrete time systems $x(k+1) = (P(k) + \Delta(k))x(k)$, where with a little abuse of notation $k = \bar{t}_k$. Let us define

$$Q_\ell = P(k_{\ell+1} - 1) \cdots P(k_\ell + 1)P(k_\ell)$$

where $k_\ell = t_\ell^m$, i.e., the transmission instants of the slowest clock m. Since by construction $t_{\ell+1}^m - t_\ell^m = T_{max} \geq T_i, \forall i$, it means that for each $j \in \mathcal{N}$ there exists k such that $k_\ell \leq k < k_{\ell+1}$ and $P(k) = \bar{P}^j$, i.e., each node transmits at least once within two transmissions of the slowest node m. This implies that $\mathcal{G}_{Q_\ell} \subseteq \cup_{k=k_\ell}^{k_{\ell+1}} \mathcal{G}_{P(k)} \subseteq \cup_{j\in\mathcal{N}} \mathcal{G}_{\bar{P}^j} = \mathcal{G}_{WSN}$, i.e., $\mathcal{G}_{Q_\ell}$ are all strongly connected. Therefore, the sequence $\{P(k)\}_{k=0}^\infty$ satisfies the conditions of Theorem 1 and consequently the linear system $x(k+1) = (P(k) + \Delta(k))x(k)$ satisfies the conditions of Theorem 3. Therefore, $\lim_{t\to\infty} x(t) = \bar{\alpha}\mathbf{1}$ exponentially fast, thus concluding the proof. □

Proof of Theorem 6. The proof follows along the same lines of Theorem 5. Define $x_i(t) = \hat{o}_i(t) + \hat{\alpha}_i(t)\beta_i$. By substituting x_i and Eqn. (7) into Eqn. (15), it follows that:

$$\begin{aligned} x_i(t^+) &= x_i(t) + (1-\rho_o)\big(\alpha_j\hat{\alpha}_j(t)t + x_j(t) - \alpha_i\hat{\alpha}_i(t)t + \\ &\quad + x_i(t)) - \beta_i(\hat{\alpha}_i(t) - \hat{\alpha}_i(t^+)) \\ &= \rho_o x_i(t) + (1-\rho_o)x_j(t) - \beta_i\big(\hat{\alpha}_i(t) - \hat{\alpha}_i(t^+)\big) + \\ &\quad + (1-\rho_o)\big(\alpha_j\hat{\alpha}_j(t) - \alpha_i\hat{\alpha}_i(t)\big)t \end{aligned}$$

which can be written in vector form as

$$x(t^+) = P(t)x(t) + v(t)$$

where $P(t) = P(t_\ell^j) = \bar{P}^j$ is a stochastic matrix which includes all outgoing links of node j, and $v(t)$ is an exponentially decreasing vector since $|\hat{\alpha}_i(t^+) - \hat{\alpha}_i(t)| \leq |\hat{\alpha}_i(t^+) - \bar{\alpha}/\alpha_i| + |\bar{\alpha}/\alpha_i - \hat{\alpha}_i(t)| \to 0$ and $|\alpha_j\hat{\alpha}_j(t) - \alpha_i\hat{\alpha}_i(t)|t \leq |\alpha_j\hat{\alpha}_j(t) - \bar{\alpha}|t + |\bar{\alpha} - \alpha_i\hat{\alpha}_i(t)|t \to 0$ exponentially fast as $t, t^+ \to \infty$ according to Theorem 5. Therefore, using the same arguments of Theorem 5 relative to the discrete time system $x(k+1) = P(k)x(k) + v(k)$ where $k = \bar{t}_k$ and Theorem 3 it follows that $\lim_{t\to\infty} x(t) = \lim_{k\to\infty} x(k) = \bar{\beta}\mathbf{1}$, or equivalently that $\lim_{t\to\infty} \hat{o}_i + \beta_i\hat{\alpha}_i(t) = \bar{\beta}$, exponentially fast. The final claim of the theorem can be obtained by observing that $|\hat{\tau}_i(t) - \hat{\tau}_j(t)| \leq |\hat{\tau}_i(t) - \bar{\tau}(t)| + |\bar{\tau}(t) - \hat{\tau}_j(t)|$ and $|\hat{\tau}_i(t) - \bar{\tau}(t)| \leq |(\alpha_i\hat{\alpha}_i(t) - \bar{\alpha})|t + |\hat{o}_i + \beta_i\hat{\alpha}_i(t) - \bar{\beta}| \to 0$ exponentially fast for $t \to \infty$ by Theorem 5. □

References

[1] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson. Coherence in large-scale networks: dimension dependent limitations of local feedback. *IEEE Trans. Automat. Control (to appear)*.

[2] A. Basso. Time synchronization in wireless sensor networks (in Italian). Master's thesis, University of Padova, Department of Information Engineering, April 2006. n. 364/05.

[3] M. Cao, A.S. Morse, and B.D.O. Anderson. Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.

[4] R. Carli, A. Chiuso, S. Zampieri, and L. Schenato. A PI consensus controller for networked clocks synchronization. In *IFAC World Congress on Automatic Control (IFAC 08)*, 2008.

[5] R. Carli, F. Garin, and S. Zampieri. Quadratic indices for the analysis of consensus algorithms. In *Proceedings of Information Theory and Applications Workshop*, pages 96–104, August 2009.

[6] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of Symposium on Operating systems design and implementation (OSDI'02)*, pages 147–163, 2002.

[7] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *Selected Areas in Communications, IEEE Journal on*, 26(4):634–649, May 2008.

[8] Ya. R. Faizulkhakov. Time synchronization methods for wireless sensor networks: A survey. *Programming and Computing Software*, 33(4):214–226, 2007.

[9] FTSP. *TinyOS repository*, 2004.

[10] S. Ganeriwal, R. Kumar, and M. Srivastava. Timingsync protocol for sensor networks. In *Proceedings of SenSys'03*, 2003.

[11] F. Garin and L. Schenato. *Networked Control Systems*, chapter A Survey on distributed estimation and control applications using linear consensus algorithms (to appear). Springer Lecture Notes in Control and Information Sciences. Springer.

[12] A. Giridhar and P. R. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. In *IEEE Conference on Decision and Control (CDC'06)*, San Diego, December 2006.

[13] S. Kar and J.M.F. Moura. Distributed consensus

algorithms in sensor networks with communication channel noise and random link failures. In *41st Asilomar Conference on Signals, Systems and Computers*, 2007.

[14] F. Knorn, R. Stanojevic, M. Corless, and R. Shorten. A framework for decentralised feedbackconnectivity control with application to sensor networks. *International Journal of Control*, 82(11):20952114, 2009.

[15] Q. Li and D. Rus. Global clock synchronization in sensor networks. *IEEE Transactions on Computers*, 55(2):214–226, 2006.

[16] M. Maròti, B. Kusy, G. Simon, and À. Ldeczi. The flooding time synchronization protocol. In *Proceedings of international conference on Embedded networked sensor systems (SenSys'04)*, pages 39–49, 2004.

[17] L. Moreau. Stability of continuous-time distributed consensus algorithms. In *Proceedings of IEEE Conference on Decision and Control (CDC'04)*, volume 4, pages 3998– 4003, December 2004.

[18] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169– 182, 2005.

[19] R. Olfati Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in multi-agent networked systems. *Proceedings of IEEE*, 95(1):215–233, January 2007.

[20] L. Schenato and F. Fiorentin. Average TimeSync: a consensus-based protocol for time synchronization in wireless sensor networks. In *IFAC Workshop on Estimation and Control of Networked Systems*, 2009.

[21] L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *IEEE Conference on Decision and Control (CDC 07)*, New Orleans, December 2007.

[22] E. Seneta. *Non-negative Matrices and Markov Chains*. John Wiley & Sons, Inc., Springer, 2006.

[23] O. Simeone and U. Spagnolini. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP Journal on Wireless Communications and Networking*, 2007.

[24] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Processing Mag.*, 25(5):81–97, September 2008.

[25] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: A survey. *IEEE Network*, 18:45–50, 2004.

[26] R. Solis, V. Borkar, and P. R. Kumar. A new distributed time synchronization protocol for multihop wireless networks. In *IEEE Conference on Decision and Control (CDC'06)*, pages 2734–2739, San Diego, December 2006.

[27] P. Sommer and R. Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 37–48, 2009.

[28] B. Sundararaman, U. Buyand, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281 – 323, 2005.

[29] TinyOS. *TinyOS website*, 2002.

[30] TmoteSky. *Tmote sky data sheet*. Moteiv inc., 2004.

[31] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of SenSys'05*, 2005.

[32] J. C. Willems. Lyapunov functions for diagonally dominant systems. *Automatica*, 12(5):519–523, 1976.