Proceedings of the First IFAC Workshop on Estimation and Control of Networked Systems
September 24-26, 2009, Venice, Italy

Average TimeSync: a consensus-based protocol for time synchronization in wireless sensor networks

L. Schenato F. Fiorentin

University of Padova, Italy (e-mail: {schenato,fiorentin}@dei.unipd.it).

Abstract: This paper describes a novel consensus-based protocol, referred as *Average TimeSync* (ATS), for synchronizing a wireless sensor network. This algorithm is based on a cascade of two consensus algorithms, whose main idea is averaging local information. The proposed algorithm has the advantage to be totally distributed, asynchronous, robust to packet drop and sensor node failure, and it is adaptive to clock drifts and changes on the communication topology. In particular, we provide a rigorous proof of convergence to global synchronization in the absence of process noise, measurement noise and communication delay, and we show its effectiveness through a number of experiments performed on a wireless sensor network.

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

1. INTRODUCTION

Recent technological advances in miniaturization and wireless communication is promoting the use of a large number of networked devices for fine-grain ambient monitoring and control. In this context, it is essential that the devices 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 refer to a common notion of time.

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. Secondly, the 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. As a consequence, many dedicated strategies and protocols have been already proposed to address the problem of time synchronization in WSNs Simeone et al. (2008).

One natural approach to deal with the multi-hop nature of a sensor network is to organize the network in a rooted tree as in the Time-synchronization Protocol for Sensor Networks (TPSN) proposed by Ganeriwal et al. (2003) and in the Flooding Time Synchronization Protocol (FTSP, Maròti et al. (2004). Initially one node is elected to be the global clock reference, then a spanning tree rooted at that node is builded, and each node synchronizes itself with its parent by compensating its offset, i.e. the instantaneous clock difference, and its relative clock skew, i.e. the relative clock speed, using its parent clock readings as the direct reference.

Another approach to the same problem is to divide the network in interconnected single-hop clusters, as suggested in the Reference Broadcast Synchronization (RBS) scheme by Elson et al. (2002). 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 local clocks of one cluster into local clocks of another cluster when needed.

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. One example of a completely distributed synchronization strategy is the Reachback Firefly Algorithm (RFA), inspired by firefly synchronization mechanism suggested by Werner-Allen et al. (2005). In this algorithm every node periodically broadcasts a synchronization message and anytime they hear a message they advance by a small quantity the phase of their internal clock that schedules 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 skew, therefore the firing period needs to be rather small. Solis et al. (2006) proposed a Distributed Time Synchronization Protocol (DTSC) which is fully distributed and compensates also for clock skews. This protocol is formulated as a distributed gradient descent optimization problem as shown by Giridhar and Kumar (2006). 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, Simeone and Spagnolini (2007) studied distributed frequency compensation, i.e. clock skew compensation, for phase locked loops (PLLs) using consensus algorithms, while

⋆ This work is sponsored by EU PF7 Project "FeedNetBack".

Carli et al. (2008) proposed a proportional-intergrative (PI) consensus-based controller to compensate both clock offsets and clock skews.

The contribution of this paper is to propose and analyze a novel time synchronization protocol for WSN, named Average TimeSynch (ATS), which builds upon our previous work Schenato and Gamba (2007) and can compensate both different clock skews and clock offsets. As compared to Schenato and Gamba (2007), here we provide a rigorous proof of convergence of the proposed protocol under the assumptions of absence of process noise, measurement noise, and propagation delay. We show extensive experimental results from a real WSN including a comparison with FTSP Maròti et al. (2004), which is considered the de-facto standard for time synchronization in WSN. As compared to the PI time-synchronization algorithm Carli et al. (2008) which requires a pseudo-synchronous implementation, the Average Time-synch is totally asynchronous, thus being resilient to packet losses, and node failure, replacement or relocation. Moreover, the proposed algorithms is adaptive to slow clock skew drifts and requires minimal memory and computational resources.

In the interest of space, all the proofs of the following theorems can be found in the longer version of this work in Fiorentin and Schenato (2009).

2. MATHEMATICAL PRELIMINARIES

In this section, we provide the necessary mathematical tools to prove convergence of the ATS protocol proposed in the next sections. In particular, we provide convergence conditions for time-varying systems subject to exponential decaying disturbances.

We model a WSN as a 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 communication links, i.e. $(i, j) \in \mathcal{E}$ if node j can communicate with node i. We define with $\mathcal{N}_i = \{j \,|\, (i, j) \in \mathcal{E}, i \neq j$ the set of neighbors of i. A matrix $P \in \mathbb{R}^{N\times N}$ is said *stochastic* if $P_{ij} \geq 0$ and $\sum_j P_{ij} = 1, \forall i \in \mathcal{N}$, 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 $\mathbf{1} = [1\,1\,\ldots\,1]^T \in \mathbb{R}^N$. A matrix is said doubly stochastic if it is stochastic and $\mathbf{1}^T P = \mathbf{1}^T$. Given a stochastic matrix P we associate a graph $\mathcal{G}_P = (\mathcal{N}, \mathcal{E}_P)$ 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 $G = (\mathcal{N}, \mathcal{E})$, 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 $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. We indicate with $\mathbb{G}_{sl} = \{\mathcal{G} = (\mathcal{N}, \mathcal{E}) \,|\, (i, i) \in \mathcal{E}, \forall i \in \mathcal{N}\}$ the set of graphs with all self-loops. A graph $G = (\mathcal{N}, \mathcal{E})$ is said to be *strongly connected* if there a path from each node pair $i, j \in \mathcal{N}$, 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 *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 we assume that the WSN connectivity graph $\mathcal{G}_{WSN} = (\mathcal{N}, \mathcal{E})$ is undirected, i.e. $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, it contains all self loops, i.e. $\mathcal{G} \in \mathbb{G}_{sl}$, and 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. We now give an important theorem which provides sufficient conditions to guarantee the convergence of time-varying consensus algorithms. The proof of this theorem and more general conditions for time-varying stochastic matrices can be found in Moreau (2005).

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 Moreau (2005) 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 any time instant, but only over an arbitrarily long but finite time window.

Before stating the main theorem, we need to introduce a technical lemma that it is needed in the proof.

Lemma 2. Let $x \in \mathbb{R}^N$ and $P \in \mathbb{R} \times \mathbb{N}$ a stochastic matrix. Let $V(x) = \max(x) - \min(x)$, then we have $V(Px) \leq (1 - \max_j \min_i P_{ij})V(x)$.

The proof can be found in Seneta (2006). Note that $\max_j \min_i P_{ij} > 0$ only if there is at least one column whose elements are all positive.

We now 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

$$x(k+1) = \big(P(k) + \Delta(k)\big)x(k) + v(k) \tag{1}$$

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 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 consensus sequence $P(k)$ give rise to a connected graph over an arbitrary but finite time window of length H, even in the presence of multiplicative but exponentially decaying disturbance, eventually all nodes will converge to a consensus exponentially fast. Consensus subject to multiplicative and additive disturbances has also been addressed in Kar and Moura (2007), but assuming a special case of consensus matrices $P(k)$ arising from the laplacian of the communication graph, while here $P(k)$ are generic stochastic matrices.

3. MODELING

In this section, we provide a mathematical modeling 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 skew 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 measuring the local clock of one node i with respect to another clock j. In fact, if we solve Eqn. (2) for t, i.e. $t = \frac{\tau_i - \beta_i}{\alpha_i}$ and we substitute it into the same equation for node j we get:

$$\tau_j = \frac{\alpha_j}{\alpha_i}\tau_i + (\beta_j - \frac{\alpha_j}{\alpha_i}\beta_i) = \alpha_{ij}\,\tau_i + \beta_{ij}$$

which is still linear. We want to synchronize all the nodes with respect to a *virtual reference clock*, namely:

$$\bar{\tau}(t) = \bar{\alpha}t + \bar{\beta} \tag{3}$$

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

$$\hat{\tau}_i = \hat{\alpha}_i\tau_i + \hat{o}_i \tag{4}$$

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

$$\lim_{t\to\infty} \hat{\tau}_i(t) = \bar{\tau}(t), \qquad i = 1, \ldots, N \tag{5}$$

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 clock time. The previous expression can be rewritten by first substituting Eqn.(2) into Eqn.(4) to get:

$$\hat{\tau}_i(t) = \hat{\alpha}_i\alpha_i t + \hat{\alpha}_i\beta_i + \hat{o}_i \tag{6}$$

therefore Eqn. (5) is equivalent to:

$$\lim_{t\to\infty} \alpha_i\hat{\alpha}_i(t) = \bar{\alpha}, \tag{7}$$

$$\lim_{t\to\infty} \hat{o}_i(t) + \beta_i\hat{\alpha}_i(t) = \bar{\beta}, \quad i = 1, .., N \tag{8}$$

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 remark few points. The first regards the clock modeling of Eqn.(2). In reality the parameters $\alpha_i(t), \beta_i(t)$ are time varying due to ambient conditions or aging, therefore the updating period of the synchronization protocol should be shorter than the variations of these parameters.

The second point is that the virtual reference clock is a fictitious clock and it not fixed a priori. In fact, the values of its parameters $(\bar{\alpha}, \bar{\beta})$ are not important, 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 MAC-layer time-stamping TmoteSky (2004), as shown in the next sections, we can safely assume that the reading of the local clock $\tau_i(t_1)$, packet transmission and reading of the local clock $\tau_j(t_2)$ is instantaneous, i.e. $t_1 = t_2$. If this not the case, our synchronization protocol cannot be used as it is and needs to be modified to cope with packet delivery delay.

4. THE ATS PROTOCOL

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

4.1 Communication protocol: pseudo-periodic broadcast

Here, we present a simple deterministic communication protocol which satisfies conditions of Theorem 1, however many other are possible as long as all nodes transmit sufficiently often, such as the randomized broadcast communication proposed by Fagnani and Zampieri (2008). We assume that each node i periodically transmits a packet to all its neighbors with a synchronization period equal to T, i.e. the transmission instants t^i_k are defined as $\tau_i(t^i_\ell) = \ell T$ or equivalently

$$t^i_\ell = \frac{\ell T - \beta_i}{\alpha_i} = \ell\, T_i + \bar{\beta}_i \tag{9}$$

As mentioned above, we assume that packets are instantaneously received by its neighbors. We refer to this protocol as *pseudo-periodic broadcast* since each node broadcasts its message at every period T based on its own clock. However, since each α_i is slightly different, over time the order of nodes transmissions as well the relative interarrival intervals change, thus the name pseudo-periodic. Let us consider the ordered set of all transmissions of all nodes $\mathbb{T} = \cup_i \cup_\ell \{t^i_\ell\} = \{\bar{t}_0, \bar{t}_1, \ldots\}$, where $\bar{t}_k$ are the ordered events, i.e $\bar{t}_k < \bar{t}_{k+1}$. Let k_ℓ such that $\bar{t}_{k_\ell} = t^m_\ell$, where $m = \text{argmin}_i\alpha_i = \text{argmax}_i T_i$, i.e. the slowest clock, and without loss of generality we assume that $\beta_m = 0$. It should be clear that $t^m_\ell = \ell T/\alpha_{min} = \ell T_{max}$ and $N \leq k_{\ell+1} - k_\ell \leq \lceil \alpha_{max}/\alpha_{min} \rceil N$, where $\alpha_{min} = \min_i \alpha_i, \alpha_{max} = \max_i \alpha_i$ 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 $\bar{t}_h = t^j_s$, 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.

4.2 Relative Skew Estimation

This part of the protocol is concerned with deriving an algorithm to estimate for each clock i the relative skew with respect its neighbors j. Let $\mathcal{N}_i$ the set of nodes that can directly transmit packets to node i. Every node i tries to estimate the relative skews $\alpha_{ij} = \frac{\alpha_j}{\alpha_i}$ with respect to its neighbor nodes $j \in \mathcal{N}_i$. This is accomplished by writing the current local time $\tau_j(t^j_\ell)$ of node j into a broadcast packet, then the node i that receives this packet immediately records its own local time $\tau_i(t^j_\ell)$. As discussed in the previous section, we can assume that the readings of the two local clocks is instantaneous since we are using MAC-layer time-stamping. Therefore, node i records in its memory the pair $(\tau^{old}_{ij}, \tau^{old}_j) = \big(\tau_i(t^j_\ell), \tau_j(t^j_\ell)\big)$. When a new packet from node j arrives to node i, the same procedure is applied to get the new pair $\big(\tau_i(t^j_{\ell+1}), \tau_j(t^j_{\ell+1})\big)$, as shown in Fig.**??**(right), and the estimate of the relative skew α_{ij} is performed as follows:

$$\left.\begin{aligned}(\tau^{new}_{ij}, \tau^{new}_j) &= \big(\tau_i(t^j_\ell), \tau_j(t^j_\ell)\big)\\ \eta_{ij}(t^+) &= \rho_\eta\,\eta_{ij}(t) + (1-\rho_\eta)\frac{\tau^{new}_j - \tau^{old}_j}{\tau^{new}_{ij} - \tau^{old}_{ij}}\\ (\tau^{old}_{ij}, \tau^{old}_j) &= (\tau^{new}_{ij}, \tau^{new}_j)\end{aligned}\right\}, t = t^j_\ell \tag{10}$$

$$\eta_{ij}(t) = \eta_{ij}(t^+), \qquad t \in (t^+, t^j_{\ell+1}] \tag{11}$$

where $\rho_\eta \in (0, 1)$ is a tuning parameter, and t^+ indicates the update. If there is no measurement error and the skew 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 Equations (10)-(11) where $0 < \rho_\eta < 1$, the transmission events t_ℓ^i are generated according to the pseudo-periodic broadcast of Eqn. (9), and each τ_i evolves according to Eqn. (2). Then

$$\lim_{t\to\infty} \eta_{ij}(t) = \alpha_{ij} \tag{12}$$

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

In practice, Equations (10)-(11) act a low pass filter where 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 skew estimates η_{ij} and the most recent local clock readings $(\tau_{ij}^{old}, \tau_j^{old})$. Since the size of $\mathcal{N}_i$ is in general small even for large networks, this algorithm is also rather scalable.

4.3 Skew 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. (3). The main idea is to use a distributed consensus algorithm based only on local information exchange. In 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, as described in the survey by Olfati-Saber (2007). In practice, every node bootstraps each other till all of them converge to a common value, i.e. till they agree upon a global value. The algorithm is very simple, in fact every node stores its own virtual clock skew estimate $\hat{\alpha}_i$, defined in Eqn. (4). 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:

$$\hat{\alpha}_i(t^+) = \rho_v \hat{\alpha}_i(t) + (1-\rho_v)\eta_{ij}(t)\,\hat{\alpha}_j(t), \ \ t = t_\ell^j, i \in \mathcal{N}_j \tag{13}$$

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

Theorem 5. Consider the skew update equation given by Equation (13) with initial condition $\hat{\alpha}_i(0) = 1$ and $0 < \rho_v < 1$, where $\eta_{ij}(t)$ are updated according to Equations (10)-(11) and t_ℓ^j is defined in Eqn. (9). 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 skew compensation algorithm is applied, the virtual clock estimators have all the same skew, i.e. they run at the same speed. At this point it is only necessary to compensate for possible offset errors. Once again, we adopt a consensus algorithm to update the estimated clock offset, previously defined in Eqn. (4), as follows:

$$\hat{o}_i(t^+) = \hat{o}_i(t) + (1-\rho_o)(\hat{\tau}_j(t) - \hat{\tau}_i(t)), \ \ t = t_\ell^j, i \in \mathcal{N}_j \tag{14}$$

where $\hat{\tau}_j$ and $\hat{\tau}_i$ are computed at the same time instant $t = t_\ell^j$, and $\hat{\alpha}_i(t)$ is kept constant for $t \neq t_\ell^j$. i.e. when the node i does not receive any message from one of its neighbors. Informally speaking, each node compute the instantaneous estimated clock difference $\hat{\tau}_j(t) - \hat{\tau}_i(t)$ and try to update its offset $\hat{o}_i$ in order to reduce the difference. The next theorem shows the convergence of this algorithm:

Theorem 6. Consider the offset update equation given by Equation (14) 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 (4), (9), (10)-(11), and (13), respectively. Then

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

exponentially fast.

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 (see TmoteSky (2004)).

Since we are interested in applications that run mostly in idle mode, we adopt the external crystal oscillator (ECO) running at $32768Hz$ for testing our 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 order to test our ATS protocol, we built a 7x5 grid for a total of 35 nodes. Since most nodes were all in communication range of each other, we forced them 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 maximum distance in terms of communication steps necessary to transmit a message from one node to another. 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 error. The nodes adopted the pseudo-periodic communication scheme described above for different synchronization periods. We observed an average packet loss around $5 - 10\%$ probably due to packet collision. In the following we present the results of the ATS protocol under different scenarios.

5.2 Dynamic topology

In this experiment, shown in Fig. 1, we study 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 replacement of new nodes. In Region A all nodes are

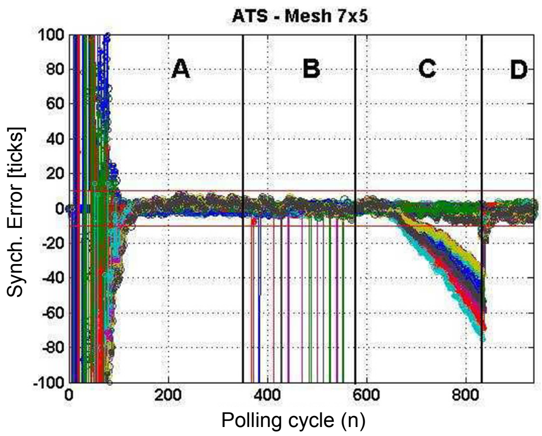

Fig. 1. Synchronization error $\hat{\tau}_i - \hat{\tau}_j$ as a function of time for the 7x5 WSN grid. Polling period is $5s$ and synchronization period is $T = 30s$. Region A: all nodes are on. Region B: 40% of the nodes are turned off and then turned on at random times. Region C: 20% of the nodes turned off their radio. Region D: the nodes turned on again the radio.

turned on simultaneously with random initial conditions of their local clocks. After about 120 polling cycles, corresponding to $120 \cdot 5s = 1h\,40min$ or $120/6 = 20$ synchronization updates, 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 picked 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. 1 clearly shows that the nodes get synchronized as soon as they are turned on without perturbing the overall performance. At the beginning Region C, about 20% of the nodes turned off their radio, i.e. they stopped updating their parameters $\eta_{ij}, \hat{\alpha}_i, \hat{o}_i$, 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

In this experiment we compared the performance of our proposed ATS protocol with the FTSP by Maròti et al. (2004), for which there is a freely available implementation for TinyOS in FTSP (2004). The FTSP is considered the the de-facto standard for time synchronization in WSN since it has been shown to resilient to dynamic changes in the communication topology and to compensate different clock skews, therefore many newly proposed algorithms are compared against it. Fig. 2 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 our ATS protocol and the absence of big sporadic errors as compared to the FTSP.

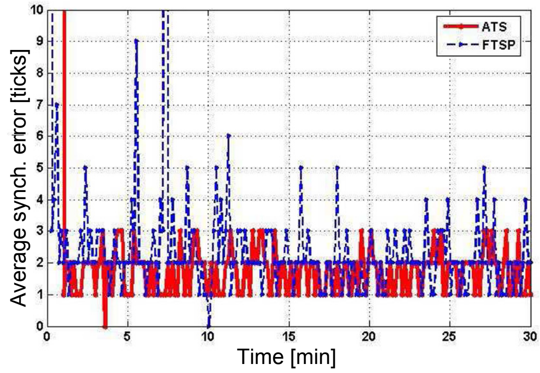

Fig. 2. Performance comparison between the ATS protocol and the FTSP by Maròti et al. (2004): maximum synchronization error $\max_{i,j} |\hat{\tau}_i - \hat{\tau}_j|$ as a function of time between any two nodes for a 3x3 WSN grid with synchronization period $T = 60s$.

5.4 Effect of node distance and synchronization period

In these sets of experiment, we explore the performance of ATS protocol as a function of relative distance in terms of communication hops between two nodes, and as a function of the synchronization period. In Fig. 3 it has been displayed the average synchronization error at steady state for the 7x5 WSN grid relative to the 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, we observed that 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.

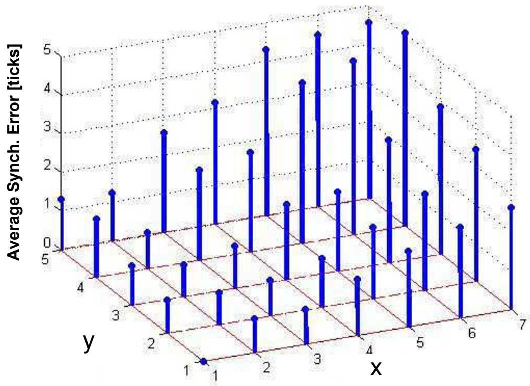

Fig. 3. Average synchronization error of each node from node $i = 1$ as a function of grid location for the 7x5 WSN with synchronization period $T = 30s$.

In Fig. 4 we show the average steady state synchronization error among all nodes measured in a 3x3 WSN as a function of different of synchronization periods ranging from $T = 7s$ to $T = 14\,min$. Obviously, performance degrades for longer synchronization period, however it exhibits a remarkable linear dependence, thus being very easy to predict synchronization error as a function of synchronization period. Finally, in Fig. 5 we show the synchronization

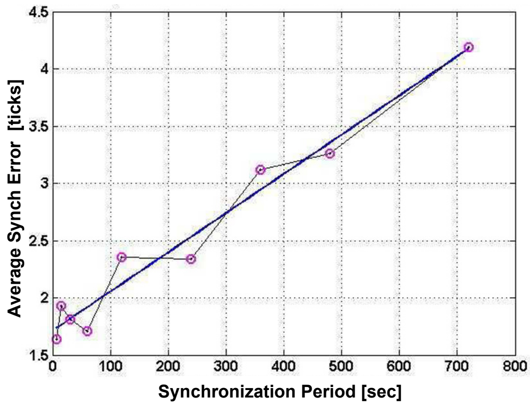

Fig. 4. Average synchronization error between any two nodes in a 3x3 WSN grid as function of the synchronization period from $T = 7s$ to $T = 14\,min$. The rect represents the best interpolating line.

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 is almost completely compensated, but the different clock skews tend to make the clocks diverge between two synchronization cycle. However, the skew compensation part of the ATS protocol slowly learns these different clock speeds and eventually totally compensate them after 6 synchronization cycles.

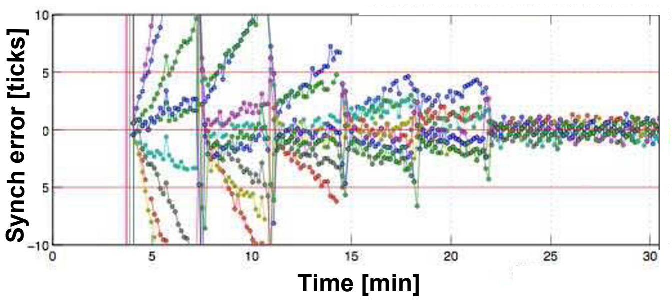

Fig. 5. Synchronization error $\hat{\tau}_i - \hat{\tau}_j$ as a function of time for the 3x3 WSN grid. Polling period is $5s$ and synchronization period is $T = 4\,min$.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented a novel synchronization algorithm for WSN, the Average TimeSync protocol. The proposed algorithm is fully distributed, asynchronous, includes skew compensation and is computationally lite. Moreover, it is robust to dynamic network topologies due, for example, to node failure or replacement. Future work includes a theoretical analysis for a scenario with include process and measurement error in order to estimate not only rate of convergence but also steady state error.

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.

REFERENCES

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). A PI consensus controller for networked clocks synchronization. *Submitted to IEEE Transactions on Automatic Control*.

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

Fagnani, F. and Zampieri, S. (2008). Randomized consensus algorithms over large scale networks. *IEEE Trans. on Selected areas in communication*, 26(4), 634–649.

Fiorentin, F. and Schenato, L. (2009). Average TimeSync: a consensus-based protocol for time synchronization in wireless sensor networks. Technical report, Univerity of Padova. [Online] Available at `http://paduaresearch.cab.unipd.it/`.

FTSP (2004). *TinyOS repository*. URL `http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.xcontrib/ethz/snpk/tos/lib/net/TimeSync/`.

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

Giridhar, A. and Kumar, P.R. (2006). Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4), 98–107.

Kar, S. and Moura, J. (2007). Distributed consensus algorithms in sensor networks with communication channel noise and random link failures. In *41st Asilomar Conference on Signals, Systems and Computers*.

Maròti, M., Kusy, B., Simon, G., and Àkos Ldeczi (2004). The flooding time synchronization protocol. In *Proceedings of SenSys'04*, 39–49.

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

Olfati-Saber, R. (2007). Algebraic connectivity ratio of ramanujan graphs. In *Proceedings of the 2007 American Control Conference*.

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

Seneta, E. (2006). *Non-negative Matrices and Markov Chains*. John Wiley & Sons, Inc., Springer.

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

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

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

TmoteSky (2004). *Tmote sky data sheet*. Moteiv inc. URL `http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf`.

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