A random broadcast consensus synchronization algorithm for large scale wireless mesh networks

Nicolas Gresset, Jonathan Letessier
Mitsubishi Electric R&D Centre Europe, France

***Abstract*—In wireless sensor networks, a robust synchronization is an enabler for optimizing performance, for example allowing powerful radio resource management and energy saving. In this paper, we combine a consensus algorithm, already proven efficient for the synchronization of wireless sensor networks, with a random broadcast for fully distributing its implementation. Unfortunately, when the distances between the wireless devices increase, the accumulation of non-negligible propagation delays resulting from the consensus algorithm introduce a drift in addition to the clock skew and offset of each node. We introduce a monitoring and broadcasting of the drift correction by a reference node that does not participate to the consensus algorithm, and show this allows for both solving the clock drift and aligning all nodes to the reference node's clock.**

I. Introduction

Wireless sensor networks (WSNs) have been extensively studied in order to bring easy, robust, and cost efficient deployments of telecommunication systems. The WSN synchronization techniques have been investigated in [1], [2], [3] in the aim of reducing the data collection latency or reduce the energy consumption of the telecommunication infrastructure. Several distributed synchronization protocols have been addressed for low-scale sensor networks, with almost no propagation delay between the nodes. For example, pulse-coupled oscillators [4] have been studied in a slotted-ALOHA scenario [5] and relatively small networks with low propagation delays.

The graph theory is a powerful analytical tool to design distributed algorithms for synchronizing networks including large number of nodes. Recently, the interest has grown on the consensus algorithms [6] that provides convergence of several processes to the same value in a distributed fashion. The synchronization is one of the applications of the consensus algorithm among many others [7]. The consensus algorithms has been deeply analyzed from an algebraic point of view [8][9] and for complex random topologies [10]. In [11], authors use MAC-layer time-stamping so as to quickly have access to the nodes time and periodically resynchronize the network.

In this paper, we consider a WSN where the sensors are placed inside or outside houses, receive commands from a concentrator/coordinator and send back statistics. The WSN topology is evolving and nodes can move, appear or disappear. The distributed synchronization protocol proposed in this paper for half-duplex devices combines random broadcast with consensus between the nodes. Furthermore, a coordinator monitors the system and broadcasts a drift correction through the WSN.

In Section II, we present the distributed protocol and the system model and parameters. In Section III, we make an asymptotic analysis of the random system behavior. In Section IV, we propose a correction of the global system clock drift induced by the propagation delays and the nodes local clock drift by applying an estimation/correction from the coordinator. Simulation results are shown in VI in order to illustrate the theoretical results presented in this paper.

II. System model and parameters

We consider a WSN comprising N nodes having half-duplex transmission and reception capabilities. The wireless mesh network connectivity is represented by a graph $\mathcal{G}$, where $\mathbf{C}$ is the $N \times N$ connectivity matrix of the graph. The graph has one connected component, i.e., a path in the graph exists between any two nodes, and particularly between the coordinator and any other node.

In that case, we propose a distributed synchronization algorithm based on a random broadcast timing exchange, i.e., at each iteration or step, each node equiprobably selects at random its transmission or reception state. When in transmission state, the node broadcasts a synchronization sequence. When in reception state, the node receives a superimposition of the synchronization sequences from the transmitting neighboring nodes, and can estimate the mis-synchronization with each node by using correlation-based detection techniques. Thus, at a given step n, each node either transmit or receive timing information to/from its neighboring nodes, and the random graph $\mathcal{G}_n$ is direct and independent from one step to the other.

Let $\mathbf{\Lambda}_n$ be the $N \times 1$ indicator vector of nodes transmitting at step n. If the k-th node is transmitting, $\mathbf{\Lambda}_n(k) = 1$, and $\mathbf{\Lambda}_n(k) = 0$ otherwise. The instantaneous connectivity matrix of the directed graph $\mathcal{G}_n$ at step n is denoted $\boldsymbol{\Omega_n}$ and is equal to

$$\boldsymbol{\Omega_n} = \mathbf{C} \odot \left[(\mathbf{1} - \mathbf{\Lambda}_n)\mathbf{\Lambda}_n^T\right], \quad (1)$$

where $\odot$ is the element by element matrix multiplication operator and $\mathbf{1}$ is a all-ones vector of length N.

A consensus algorithm is then applied. At each step n, the k-th node updates its synchronization time $t_{k,n}$ according to an averaging of the relative synchronization time received from other nodes, as follows:

$$t_{k,n} = t_{k,n-1} + T + \varepsilon_k + \beta_{k,n} \sum_j \mathbf{\Omega}_n(k,j) \times (t_{j,n-1} - t_{k,n-1} + \tau_{j,k} + \varepsilon_j - \varepsilon_k) - \xi_{k,n} \quad (2)$$

where $\tau_{j,k}$ is the propagation delay between the k-th node and the j-th node, T is the time period between two synchronization steps. The k-th node oscillator being not perfect, a time clock drift ε_k occurs between two synchronization steps with respect to an absolute perfect reference clock. The parameters $\beta_{k,n}$ will be determined in the following in order to ensure convergence to the system. Finally, a correction factor $\xi_{k,n}$ is applied, and will be discussed in section IV.

We define $\mathbf{\Phi}_n$ as a vector which k-th entry is the mis-synchronization $\mathbf{\Phi}_n(k) = t_{k,n} - t_{0,n}$ at step n between the k-th node and an absolute reference clock $t_{0,n}$. The whole system model can be rewritten in a matrix form as

$$\mathbf{\Phi}_n = \mathbf{M}_n(\mathbf{\Phi}_{n-1} + \boldsymbol{\varepsilon}) + \boldsymbol{\beta}_n \mathcal{D}(\mathbf{\Omega}_n \boldsymbol{\tau}) - \boldsymbol{\xi}_n, \tag{3}$$

where $\mathbf{M}_n = \mathbf{I} - \boldsymbol{\beta}_n \mathbf{L}_n$, $\boldsymbol{\beta}_n = \mathrm{diag}(\beta_{1,n}, \ldots, \beta_{N,n})$, and $\mathcal{D}(\mathbf{X})$ creates a vector with the diagonal elements of $\mathbf{X}$. By definition, the matrix $\mathbf{L}_n$ is the Laplacian matrix of the graph at step n, i.e., the subtraction between the degree and connectivity matrices:

$$\mathbf{L}_n = \mathrm{diag}(|\mathbf{\Omega}_{1,n}|, \ldots, |\mathbf{\Omega}_{N,n}|) - \mathbf{\Omega}_n, \tag{4}$$

where $|\mathbf{\Omega}_{j,n}| = \sum_{i=1}^{N} \mathbf{\Omega}_n(j,i)$ is the degree of the j-th node at step n, i.e., the number of neighbors from which it receives a synchronization signal at step n.

By solving the recurrence in (3), one obtain

$$\mathbf{\Phi}_n = \mathbf{a}_n \mathbf{\Phi}_0 + \mathbf{b}_n \boldsymbol{\varepsilon} + \mathbf{c}_n - \hat{\boldsymbol{\xi}}_n \tag{5}$$

where

$$\begin{aligned}
\mathbf{a}_n &= \prod_{i=1}^{n} \mathbf{M}_i && (6)\\
\mathbf{b}_n &= \sum_{j=1}^{n} \prod_{l=j}^{n} \mathbf{M}_l && (7)\\
\mathbf{c}_n &= \sum_{j=1}^{n-1} \prod_{l=j+1}^{n} \mathbf{M}_l \boldsymbol{\beta}_j \mathcal{D}(\mathbf{\Omega}_j \boldsymbol{\tau}) + \boldsymbol{\beta}_n \mathcal{D}(\mathbf{\Omega}_n \boldsymbol{\tau}) && (8)\\
\hat{\boldsymbol{\xi}}_n &= \sum_{j=1}^{n-1} \prod_{l=j+1}^{n} \mathbf{M}_l \boldsymbol{\xi}_j + \boldsymbol{\xi}_n && (9)
\end{aligned}$$

We first consider that $\boldsymbol{\beta}_n = \beta \mathbf{I}$, where β is a global correction factor applied to all nodes. In next sections, we perform a statistical and asymptotic analysis of the system in order to define β for convergence and $\hat{\boldsymbol{\xi}}_n$ for drift correction. Another strategy for the $\boldsymbol{\beta}_n$ correction that further improve the system performance will be presented in section V.

III. Analysis of the random broadcast-based average consensus algorithm

In this section, we derive the asymptotic expression of (6), (7), (8) in order to derive the asymptotic behavior of (5).

A. Asymptotic expression of $\mathbf{a}_n$

First, the i.i.d properties of the random matrices $\mathbf{M}_i$ give

$$\mathbf{a}_n = \prod_{i=1}^{n} \mathbf{M}_i \sim E[\mathbf{M}_i]^n \quad (\text{as } n \to +\infty). \tag{10}$$

From (1), $E\left[(\mathbf{1} - \mathbf{\Lambda}_i)\mathbf{\Lambda}_i^T\right] = \mathbf{1}\mathbf{1}^T/4$ and $E\left[|\mathbf{\Omega}_{i,n}|\right] = |\mathbf{C}_i|/4$, we observe that $E[\mathbf{L}_i] = \frac{1}{4}\mathbf{L}$ where $\mathbf{L}$ is the Laplacian matrix of the graph $\mathcal{G}$ defined and diagonalized as:

$$\mathbf{L} = \mathrm{diag}(|\mathbf{C}_1|, \ldots, |\mathbf{C}_N|) - \mathbf{C} = \mathbf{Q}\mathbf{\Delta}\mathbf{Q}^{-1} \tag{11}$$

where $|\mathbf{C}_j| = \sum_{i=1}^{N} \mathbf{C}(j,i)$ is the number of neighbors of the j-th node,and $\mathbf{\Delta} = \mathrm{diag}(\delta_1, \ldots, \delta_N)$. As a result, the Laplacian matrix has one null eigenvalue $\delta_1 = 0$ associated to the eigenvector $\mathbf{1}/\sqrt{N}$. When the graph has one connected component, the other eigenvalues $\delta_i > 0$ are strictly positive.

Then, (10) leads to

$$\mathbf{a}_n \sim \mathbf{Q}\left(\mathbf{I} - \frac{\beta}{4}\mathbf{\Delta}\right)^n \mathbf{Q}^{-1} \quad (\text{as } n \to +\infty). \tag{12}$$

In order to ensure a convergence of the system with no delay and clock drift, one must satisfy

$$\forall i > 2, \left|1 - \frac{\beta\delta_i}{4}\right| < 1 \tag{13}$$

which is for example achieved as in [12] by setting $0 < \beta < 4/\max(|\mathbf{C}_i|) < 8/\max(\delta_i)$. The higher the value of β, the higher the convergence speed of the system. As a result, $E[\mathbf{M}_i]^n$ asymptotically has one non-null eigenvalue equal to one, and associated to the null eigenvalue of $\mathbf{L}$ and its eigenvector $\mathbf{1}/\sqrt{N}$. Finally, the asymptotic expression of $\mathbf{a}_n$ is

$$\mathbf{a}_n \sim \frac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{14}$$

B. Asymptotic expression of $\mathbf{b}_n$

Equivalently, in order to derive the asymptotic expression of (5), we consider the asymptotic expectation of (7):

$$\begin{aligned}
\mathbf{b}_n &\sim E\left[\sum_{j=1}^{n} \prod_{l=j}^{n} \mathbf{M}_l\right] && (15)\\
&= \mathbf{Q}\sum_{j=1}^{n}\left(\mathbf{I} - \frac{\beta}{4}\mathbf{\Delta}\right)^{n-j+1} \mathbf{Q}^{-1} && (16)\\
&= \mathbf{Q}\mathrm{diag}\left(\lambda_{1,n}, \ldots, \lambda_{N,n}\right)\mathbf{Q}^{-1} && (17)
\end{aligned}$$

where

$$\begin{cases} \delta_i = 0 \quad \Rightarrow \quad \lambda_{i,n} = n, \\ \delta_i > 0 \quad \Rightarrow \quad \lambda_{i,n} = \frac{1-(1-\beta\delta_i/4)^{n+1}}{\beta\delta_i/4} - 1. \end{cases} \tag{18}$$

Thus, if the convergence condition (13) is satisfied, one obtain

$$\begin{aligned}
\mathbf{b}_n \sim &\quad \mathbf{Q}\mathrm{diag}\left(n, \tfrac{4}{\beta\delta_2} - 1, \ldots, \tfrac{4}{\beta\delta_n} - 1\right)\mathbf{Q}^{-1} && (19)\\
= &\quad \tfrac{n+1}{N}\mathbf{1}\mathbf{1}^T + \mathbf{V}\mathrm{diag}\left(\tfrac{4}{\beta\delta_2}, \ldots, \tfrac{4}{\beta\delta_n}\right)\mathbf{V}^T - \mathbf{I} && (20)
\end{aligned}$$

where, from (11), we denote $\mathbf{V}$ as the $N \times (N-1)$ matrix of the $N-1$ eigenvectors of $\mathbf{L}$ associated to the eigenvalues $\delta_2, \ldots, \delta_N$, such that $\mathbf{V}\mathbf{V}^T = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$.

C. Asymptotic expression of $\mathbf{c}_n$

Finally, by using the independence relationship

$$\forall i \neq j, \quad E\left[\mathbf{M}_i \mathcal{D}(\mathbf{\Omega}_j \boldsymbol{\tau})\right] = E\left[\mathbf{M}_i\right] E\left[\mathcal{D}(\mathbf{\Omega}_j \boldsymbol{\tau})\right] \tag{21}$$

and $E\left[\mathcal{D}(\mathbf{\Omega}_j \boldsymbol{\tau})\right] = \frac{1}{4}\mathcal{D}(\mathbf{C}\boldsymbol{\tau})$, one obtain

$$\begin{aligned} \mathbf{c}_n &\sim E\left[\sum_{j=1}^{n-1} \prod_{l=j+1}^{n} \mathbf{M}_l \mathcal{D}(\mathbf{\Omega}_j \boldsymbol{\tau}) + \mathcal{D}(\mathbf{\Omega}_n \boldsymbol{\tau})\right] & (22) \\ &= \frac{1}{4}\left(\mathbf{Q}\sum_{j=0}^{n-1}\left(\mathbf{I} - \frac{\beta}{4}\mathbf{\Delta}\right)^j \mathbf{Q}^{-1}\right)\mathcal{D}(\mathbf{C}\boldsymbol{\tau}) & (23) \\ &= \frac{\beta}{4}\mathbf{Q}\mathrm{diag}\left(\alpha_{1,n}, \dots, \alpha_{N,n}\right)\mathbf{Q}^{-1}\mathcal{D}(\mathbf{C}\boldsymbol{\tau}) & (24) \end{aligned}$$

where

$$\begin{cases} \delta_i = 0 & \Rightarrow \quad \alpha_{i,n} = n, \\ \delta_i > 0 & \Rightarrow \quad \alpha_{i,n} = \frac{1-(1-\beta\delta_i/4)^n}{\beta\delta_i/4}. \end{cases} \tag{25}$$

Thus, if the convergence condition (13) is satisfied, one obtain

$$\begin{aligned} \mathbf{c}_n &\sim \mathbf{Q}\mathrm{diag}\left(\frac{n}{4}, \frac{1}{\beta\delta_2}, \dots, \frac{1}{\beta\delta_n}\right)\mathbf{Q}^{-1}\mathcal{D}(\mathbf{C}\boldsymbol{\tau}) \\ &= \frac{n}{4N}\mathbf{1}\mathbf{1}^T\mathcal{D}(\mathbf{C}\boldsymbol{\tau}) \\ &\quad + \mathbf{V}\mathrm{diag}\left(\frac{1}{\beta\delta_2}, \dots, \frac{1}{\beta\delta_n}\right)\mathbf{V}^T\mathcal{D}(\mathbf{C}\boldsymbol{\tau}). \end{aligned} \tag{26}$$

D. Asymptotic expression of $\mathbf{\Phi}_n$

By using the asymptotic results (14), (20) and (26), (5) becomes

$$\mathbf{\Phi}_n \sim n\boldsymbol{\mu} + \boldsymbol{\eta} - \hat{\boldsymbol{\xi}}_n, \tag{27}$$

where

$$\boldsymbol{\mu} = \frac{\mathbf{1}\mathbf{1}^T}{N}\left(\boldsymbol{\varepsilon} + \frac{1}{4}\mathcal{D}(\mathbf{C}\boldsymbol{\tau})\right) \tag{28}$$

and

$$\begin{aligned} \boldsymbol{\eta} &= \frac{\mathbf{1}\mathbf{1}^T}{N}\left(\mathbf{\Phi}_0 + \boldsymbol{\varepsilon}\right) - \boldsymbol{\varepsilon} \\ &\quad + \mathbf{V}\mathrm{diag}\left(\frac{1}{\delta_2}, \dots, \frac{1}{\delta_n}\right)\mathbf{V}^T\left(4\boldsymbol{\varepsilon}/\beta + \mathcal{D}(\mathbf{C}\boldsymbol{\tau})\right) \end{aligned} \tag{29}$$

and where $\mathbf{1}\mathbf{1}^T\mathbf{\Phi}_0/N$ is a vector filled with the average of initial states $\Phi_{j,0}$, $\mathbf{1}\mathbf{1}^T\boldsymbol{\varepsilon}/N$ is a vector filled with the average clock drift of the nodes, and $\mathbf{1}\mathbf{1}^T\mathcal{D}(\mathbf{C}\boldsymbol{\tau})/N$ is a vector filled with the average propagation delay between neighbors.

By setting the correction factor $\hat{\boldsymbol{\xi}}_n = \mathbf{0}$, several observations can be made from (27). First, the constant part $\boldsymbol{\eta}$ of the asymptotic expression (27) shows a mis-synchronization between nodes, which mainly depends on the cumulative delays between each node and its neighbors, and on the clock drift being most of the time negligible with respect to the propagation delays, except if the synchronization period T is large. Then, we observe that all the nodes experience a same clock drift, resulting from all nodes clock drifts and all the propagation delays in the system. Indeed, coefficients of $\boldsymbol{\mu}$ are equal to the sum of the average mean of the entries of $\boldsymbol{\varepsilon}$ and $\frac{1}{4}\mathcal{D}(\mathbf{C}\boldsymbol{\tau})$. This implies that one can use the same clock drift correction for all nodes, i.e., $\boldsymbol{\xi}_n = \mathbf{1}\xi_n$. As a remark, without any propagation delay, and by assuming that $E[\varepsilon_j] = 0$, no clock drift correction is needed and the proposed synchronization protocol allows for a cooperative correction of the drift for cheap devices with low-precision oscillators.

IV. System drift correction by a reference node

In this section, we consider the correction of the system linear drift expressed as $n\boldsymbol{\mu}$ in (27). We observe that the expression (9) of $\hat{\boldsymbol{\xi}}_n$ is similar to the expression (8) of $\mathbf{c}_n$. One key assumption allowing the derivation of $\mathbf{c}_n$ is the independence relationship (21) resulting from the random broadcast protocol. Equation (9) points out that the drift correction should be independent of all random connectivity matrices realizations, which cannot be achieved by using the observation of one or more nodes participating to the consensus algorithm. Thus, in order to reach independence between the correction factor and the random connectivity variables, we propose to use the node 0, which is for example a coordinator node, in order to evaluate the system drift and correct it. By definition, we set for all step n, $\Phi_{0,n} = 0$ and $\varepsilon_0 = 0$. The node 0 has a $1 \times N$ random connectivity vector $\hat{\mathbf{\Omega}}_n$ with the N nodes of the graph. It applies at each iteration the computation of the updated synchronization time from the consensus (2), makes the difference with its reference synchronization time, and estimates the system drift

$$\xi_n = \beta \sum_j \hat{\mathbf{\Omega}}_n(j)\left(\Phi_{j,n-1} + \tau_{j,0} + \varepsilon_j\right). \tag{30}$$

As a remark, the coordinator does not update its synchronization time as it is a reference for the absolute time and clock drift. Then, the coordinator broadcasts ξ_n through the wireless mesh network, which implies

$$\boldsymbol{\xi}_n = \beta\mathbf{1}\hat{\mathbf{\Omega}}_n\left(\mathbf{\Phi}_{n-1} + \boldsymbol{\tau}_0 + \boldsymbol{\varepsilon}\right) \tag{31}$$

where $\boldsymbol{\tau}_0 = [\tau_{1,0}, \dots, \tau_{N,0}]^T$.

Let us note that, any random matrix $\mathbf{L}_i$ being a Laplacian Matrix, $\mathbf{L}_i\mathbf{1} = [0 \dots 0]^T$, which involves that $\mathbf{M}_i\mathbf{1} = \mathbf{1}$ and $\forall(i,j), \mathbf{M}_i\boldsymbol{\xi}_j = \boldsymbol{\xi}_j$.

Thus, (9) becomes

$$\hat{\boldsymbol{\xi}}_n = \sum_{j=1}^{n}\boldsymbol{\xi}_j = \sum_{i=1}^{n-1}\prod_{j=i+1}^{n}(\mathbf{I} - \beta\mathbf{1}\hat{\mathbf{\Omega}}_j)\beta\mathbf{1}\hat{\mathbf{\Omega}}_i\left(\mathbf{\Phi}_{i-1} + \boldsymbol{\tau}_0 + \boldsymbol{\varepsilon}\right) \tag{32}$$

The term $\sum_{j=1}^{n}\boldsymbol{\xi}_j$ shows that the same result can be obtained by broadcasting a long term filtered version of ξ_n, instead of all ξ_n. This allows for achieving a trade off between the broadcast period (and overhead), and the adaptive algorithm speed.

By using the statistical independence of $\hat{\mathbf{\Omega}}_j$ random vectors, $E[\hat{\mathbf{\Omega}}_j] = \hat{\boldsymbol{\theta}}$ where $2\hat{\boldsymbol{\theta}}$ is the long term connectivity vector between the coordinator and its $|\hat{\boldsymbol{\theta}}|$ neighbors, and $(\mathbf{I} - \beta\mathbf{1}\hat{\boldsymbol{\theta}})^{n-i}\beta\mathbf{1}\hat{\boldsymbol{\theta}} = (1 - \beta|\hat{\boldsymbol{\theta}}|)^{n-i}\beta\mathbf{1}\hat{\boldsymbol{\theta}}$, one obtain

$$\hat{\boldsymbol{\xi}}_n \sim \mathbf{1}\sum_{i=1}^{n}(1 - \beta|\hat{\boldsymbol{\theta}}|)^{n-i}\beta\hat{\boldsymbol{\theta}}\left(\mathbf{\Phi}_{i-1} + \boldsymbol{\tau}_0 + \boldsymbol{\varepsilon}\right). \tag{33}$$

By substituting (33) in the asymptotic linear form $n\boldsymbol{\mu} + \boldsymbol{\eta}$ of $\boldsymbol{\Phi}_n$, and by using the asymptotic arithmetic-geometric progression expression, we obtain:

$$\boldsymbol{\Phi}_n \sim n(\mathbf{I}-\mathbf{1}\hat{\boldsymbol{\theta}}/|\hat{\boldsymbol{\theta}}|)\boldsymbol{\mu}+\boldsymbol{\eta}+\mathbf{1}\frac{\hat{\boldsymbol{\theta}}}{|\hat{\boldsymbol{\theta}}|}\left(\frac{\boldsymbol{\mu}}{\beta|\hat{\boldsymbol{\theta}}|}-(\boldsymbol{\eta}+\boldsymbol{\tau}_0+\boldsymbol{\varepsilon})\right). \tag{34}$$

Finally, by using the fact that $\hat{\boldsymbol{\theta}}\mathbf{1}/|\hat{\boldsymbol{\theta}}| = 1$, and by using (28) and $(\mathbf{I}-\mathbf{1}\hat{\boldsymbol{\theta}}/|\hat{\boldsymbol{\theta}}|)\boldsymbol{\mu} = \mathbf{0}$, we observe that the asymptotic behavior of $\boldsymbol{\Phi}_n$ has no drift anymore. Furthermore, the asymptotic limit is equal to:

$$\boldsymbol{\Phi}_n \sim \frac{1}{|\hat{\boldsymbol{\theta}}|}\frac{\mathbf{1}\mathbf{1}^T}{4N}\mathcal{D}(\mathbf{C}\boldsymbol{\tau}) - \left(\mathbf{I} - \frac{1}{\beta|\hat{\boldsymbol{\theta}}|}\frac{\mathbf{1}\mathbf{1}^T}{N}\right)\boldsymbol{\varepsilon} - \mathbf{1}\frac{\hat{\boldsymbol{\theta}}}{|\hat{\boldsymbol{\theta}}|}\boldsymbol{\tau}_0$$
$$+\left(\mathbf{I}-\mathbf{1}\frac{\hat{\boldsymbol{\theta}}}{|\hat{\boldsymbol{\theta}}|}\right)\times\mathbf{V}\mathrm{diag}\left(\frac{1}{\delta_2},\ldots,\frac{1}{\delta_n}\right)\mathbf{V}^T\left(\frac{4\boldsymbol{\varepsilon}}{\beta}+\mathcal{D}(\mathbf{C}\boldsymbol{\tau})\right) \tag{35}$$

which does not depend on the initial nodes states $\boldsymbol{\Phi}_0$.

This leads to the following important and non-intuitive result: by not participating to the consensus algorithm and broadcasting the drift correction applied at each node, the coordinator make the nodes converge around its reference time clock $\Phi_{0,n} = 0$.

V. Averaging the Timing Difference

In this section, we introduce another strategy for the choice of the correction factors $\boldsymbol{\beta}$. On top of to the global correction factor applied in previous sections, we introduce a local correction at each node in order to compute the average timing difference of transmitting nodes, i.e., by setting $\boldsymbol{\beta}_n = \beta\mathrm{diag}(\{f(|\boldsymbol{\Omega}_{i,n})|\})$, where $\forall x > 0, f(x) = x^{-1}$ and $f(0) = 0$. In this case,

$$E\left[\boldsymbol{\beta}_i\mathbf{L}_i\right] = \frac{\beta}{4}\mathrm{diag}\left(\left\{\sum_{j=1}^{|\mathbf{C}_k|}\frac{2^{-|\mathbf{C}_k|}}{j}\binom{|\mathbf{C}_k)|}{j}\right\}\right)\mathbf{L} \leq \frac{\beta}{4}\mathbf{L}. \tag{36}$$

Thus, any β satisfying the convergence condition (13) also ensures convergence for this scenario.

A thorough analysis of this scenario is difficult and out of the scope of this paper. However, one can make the following observation: the real-time averaging of the timing difference changes the eigenvalues δ_i in (35) and results in more equals values exhibiting better performance. The analysis is left for further studies and the performance improvement will be illustrated in the simulation results provided in section VI.

VI. Simulation Results

In this section, we compare the proposed algorithm to a multi-hop based protocol which corresponds for each node to a synchronization to the average of the sequences received at a first synchronization step, and a retransmission during the next step. Since both algorithm have no clock drift with respect to the coordinator clock, we only compare relative timing difference. The wireless mesh network comprises N nodes uniformly placed at random in a circular area around the coordinator. In an initial step, we assume that a power control algorithm ensures a connectivity of each node to the system. For $N = 1000$ and a 1km-radius deployment, up to 30 hops are required for communication from the coordinator to the WSN edge nodes. In this paper, we consider that all the nodes are transmitting with the same power. A micro cell path loss model [13] provides the propagation attenuation between houses in a suburban environment. An edge of the graph $\mathcal{G}$ exists between two nodes if the SNR on the associated wireless link is above a given threshold.

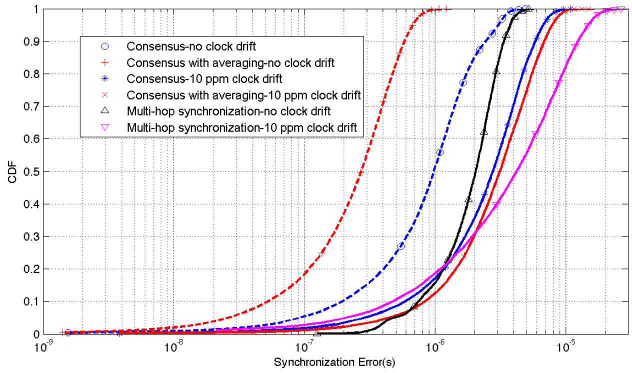

Fig. 1. Cumulative density function of the synchronization error to the coordinator for the multi-hop and the consensus algorithms with no clock drift at the nodes or a maximal clock drift of 10ppm with a frame period of 100ms. Average number of neighbors=17.

In Fig. 1, $N = 1000$ nodes are uniformly placed at random in a $1\mathrm{km}^2$ circular area around the coordinator. A transmit power of 0 dBm for the synchronization sequence allows for achieving an average connectivity of 17 neighbors per node, which are in average distant of $35m$ and corresponds to an average delay of $0.12\mu s$ in free space.

The cumulative density functions of the synchronization error to the coordinator for the multi-hop synchronization algorithm and the random-broadcast based consensus algorithms with a drift correction by the coordinator are compared. We consider a random initialization for the random-broadcast based consensus algorithm, which corresponds to a case where the node turns on at any moment and runs the algorithm. By doing so, the synchronization is implemented in a fully distributed fashion for the nodes, and their initial clock timings have no impact on the system asymptotic behavior. We first assume that the nodes oscillator have no clock drift. The coordinator only monitors the system behavior and broadcasts the measured drift ξ_n resulting from the propagation delays accumulation. We observe that, by using the consensus algorithms, the nodes are better synchronized to the coordinator which provides the reference clock only by sending the clock drift correction. The consensus with averaging brings additional performance improvement in that case. We now assume that the synchronization step periodicity is $T = 100ms$ and that the oscillators precision is uniformly distributed with an upper bound of $10ppm$, which result in an inter-frame de-

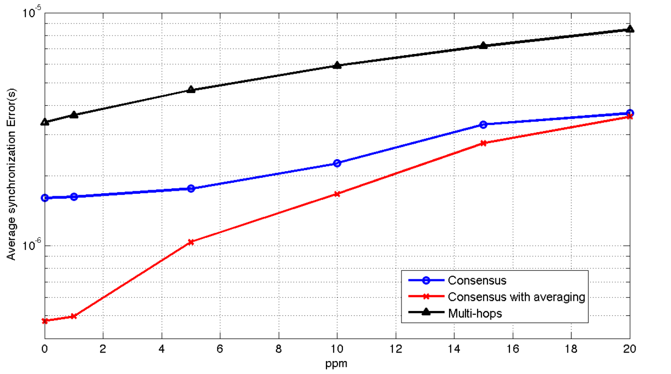

Fig. 2. Average synchronization error to the coordinator, with a varying maximal clock drift at each node and no propagation delay. Number of nodes=200.

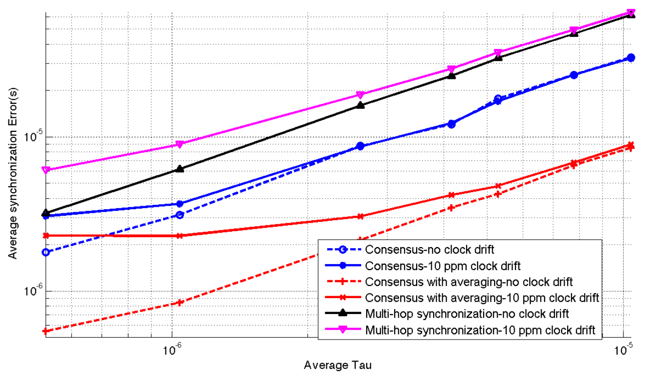

Fig. 4. Average synchronization error to the coordinator, with a density of nodes expressed in average delay to the neighboring nodes. Number of nodes=300.

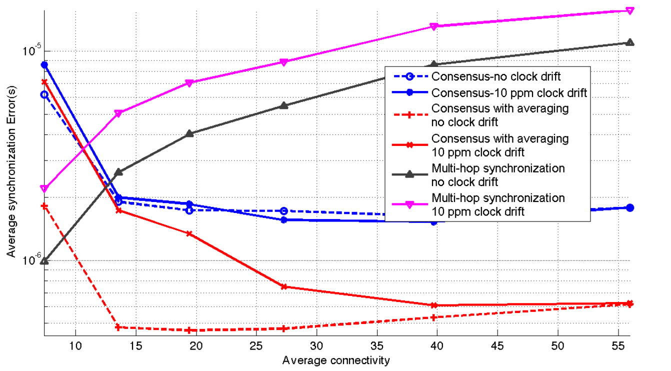

Fig. 3. Average synchronization error to the coordinator, with a varying connectivity (sequence boost). Number of nodes=300.

synchronization range of $[-1 \quad 1]\mu s$ at each node due to the clock drift. This scenario illustrates a lower gain due to the dominating effect of the clock drifts.

This behavior is confirmed by Fig. 2, where the algorithms performance is drawn as a function of the nodes maximal clock drift for a system with no propagation delay. The consensus algorithm with averaging of the timing difference outperforms other algorithm for low clock drift values.

In Fig. 3, the synchronization sequence is boosted so as to make the average connectivity of each node vary. In Fig. 4, the relative position between nodes is scaled so as to make the average delay between the neighboring nodes vary. In both cases, the consensus method with an averaging of the timing difference outperforms the multi-hop synchronization and the consensus algorithm with a global correction factor only. We remark that the performance of the consensus algorithm degrades when the network connectivity decreases, which is a well known behavior of the consensus algorithm. This indicates that a parallel power control algorithm must be performed so as to equalize the nodes average connectivity. This will be addressed in future work.

VII. Conclusions

We have presented a random broadcast consensus synchronization algorithm for wireless mesh networks. In a fully distributed implementation, the non-negligible delay propagation and clock drift cause a global system clock drift. We introduce a correction of this drift by a coordinator that is not involved in the cooperative synchronization process, that furthermore align all nodes synchronization times to the coordinator reference clock.

References

[1] A. Agreev, “Time synchronization and energy efficiency in wireless sensor networks,” Ph.D. dissertation, University of Trento, Italia, 2010.
[2] B. M. Sadler and A. Swami, “Synchronization in sensor networks: an overview,” in *Proc. of Military Communications Conference*, October 2006.
[3] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks (Elsevier)*, vol. 3, pp. 281–323, 2005.
[4] R. Mirollo and S. Strogatz, “Synchronization of pulse-coupled biological oscillators,” in *SIAM J. Appl. Math*, vol. 50, no. 6, December 1990, pp. 1645–1662.
[5] A. Tyrrell and G. Auer, “Imposing a reference timing onto firefly synchronization in wireless networks,” in *Proceedings of Vehicular Technology Conference*, April 2007, pp. 222–226.
[6] R. Olfati-Saber and R. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.
[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, p. 215, January 2007.
[8] F. Xiao, L. Wang, H. Shi, and A. Wang, “Algebraic characterizations of consensus problems for networked dynamic systems,” in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, June 2005, pp. 622–627.
[9] Z. Li, Y. Jia, J. Du, and F. Yu, “Algebraic criteria for consensus problem of networked systems with continuous-time dynamics,” in *American Control Conference*, June 2008, pp. 4394–4399.
[10] F. Xiao and L. Wang, “Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays,” *IEEE Trans. on Automatic Control*, vol. 53, no. 8, Sept. 2008.
[11] L. Schenato and F. Fiorentin, “Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks,” *Automatica, Elsevier*, vol. 47, pp. 1878–1886, 2011.
[12] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” in *Proceedings of IEEE Conference on Decision and Control*, Hawaii, December 2003, pp. 4997–5002.
[13] 3GPP TR 36.814 V9.0.0, “Further advancements for eutra physical layer aspects,” *3GPP*, pp. 29–30, March 2010.