Distributed Clock Synchronization for Wireless Sensor Networks Using Belief Propagation

Mei Leng and Yik-Chung Wu

***Abstract*—In this paper, we study the global clock synchronization problem for wireless sensor networks. Based on belief propagation, we propose a fully distributed algorithm which has low overhead and can achieve scalable synchronization. It is also shown analytically that the proposed algorithm always converges for strongly connected networks. Simulation results show that the proposed algorithm achieves better accuracy than consensus algorithms. Furthermore, the belief obtained at each sensor provides an accurate prediction on the algorithm's performance in terms of MSE.**

***Index Terms*—Belief propagation, fully distributed, global clock synchronization, wireless sensor network (WSN).**

I. Introduction

WIRELESS sensor network (WSN), emerged as an important research area in recent years, consists of many small-scale miniature devices (known as sensor nodes) capable of onboard sensing, computing, and communications. WSNs are used in industrial and commercial applications to monitor data that would be difficult or inconvenient to monitor using wired equipment. These applications include habitat monitoring, controlling industrial machines and home appliances, object tracking, and event detection, etc. [1], [2]. Most of these applications require collaborative execution of a distributed task amongst a large set of synchronized sensor nodes. Furthermore, data fusion, power management and transmission scheduling require all the nodes running on a common time frame. However, every individual sensor in a WSN has its own clock. Different clocks will drift from each other with time due to many factors, such as imperfection of oscillators and environmental changes. This makes clock synchronization between different nodes an indispensable piece of infrastructure [3], [4].

A. The Clock Synchronization Problem

Considering a network with $M+1$ sensors $\{S_0, S_1, \ldots, S_M\}$, these sensors are randomly distributed in the field and can be self-organized into a network by establishing connections between neighboring sensors that are in each other's communication range. An example of a network with 10 sensors is shown in Fig. 1(a), where each circle represents a sensor and each edge represents the ability to transmit and receive packets between the corresponding pair of sensors.

In the problem of clock synchronization, each sensor S_i has a clock which gives clock reading $c_i(t)$ at real time t. The first-order model for the function $c_i(t)$ is

$$c_i(t) \triangleq t + \theta_i \tag{1}$$

where θ_i represents the clock offset of S_i. This first-order clock model has been widely used in the literature (see [3], [4], [7], [8], [10], [13], [14], [16], and references therein), and in order to achieve global clock synchronization, the clock offsets of all $M+1$ sensors (i.e., $\{\theta_i\}_{i=0}^{M}$) should be estimated and compensated. Without loss of generality, suppose S_0 is the reference node with $\theta_0 = 0$, the task of global clock synchronization is then to synchronize the other M sensors $\{S_i\}_{i=1}^{M}$ to the reference node S_0, that is, to recover clock offsets $\{\theta_i\}_{i=1}^{M}$.

B. Related Works

A wide variety of clock synchronization protocols have been proposed in the literature. Depending on how sensors are networked, we can put existing protocols into three categories: tree-structure-based, cluster-structure-based, and fully distributed. In tree-structure-based protocols, one sensor is elected as the reference node, then a spanning tree rooted at this node is constructed, and each sensor synchronizes with its parent in the tree [35], as shown in Fig. 1(b). One of the most representative protocols in this class is time-synchronization protocol for sensor network (TPSN) [4], where each sensor obtains its clock offset with respect to the root by adding message delays along its unique path to the root. Other similar schemes are flooding time synchronization protocol (FTSP) [7], lightweight tree-based synchronization (LTS) [8], and tiny-sync [9]. This approach suffers from two main limitations. The first limitation is that it requires high overhead to maintain the tree structure and it is sensitive to root failure. The second limitation is that two sensors which are close to each other geographically may belong to different levels which are far in terms of tree distance from the root, for example S_8 and S_9 in Fig. 1(b). Since clock errors are directly related to the tree distance, such structures can be harmful in applications where clocks are required to degrade smoothly as a function of geographic distance, such as object tracking [15].

On the other hand, in cluster-structure-based protocols, sensors are grouped into clusters according to their geographical locations. Generally, several reference nodes are selected first, then sensors in the listening distance of the same reference node are grouped into one cluster and are synchronized by the reference node. Sensors belonging to multiple neighborhoods act as gateways to convert local clocks of one cluster into that of

Manuscript received November 15, 2010; revised March 09, 2011; accepted July 11, 2011. Date of publication July 25, 2011; date of current version October 12, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yao-Win Peter Hong. This work was supported in part by the Hong Kong University Seed Funding Programme under Project 201011159014.

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: meileng@eee.hku.hk; ycwu@eee.hku.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2162832

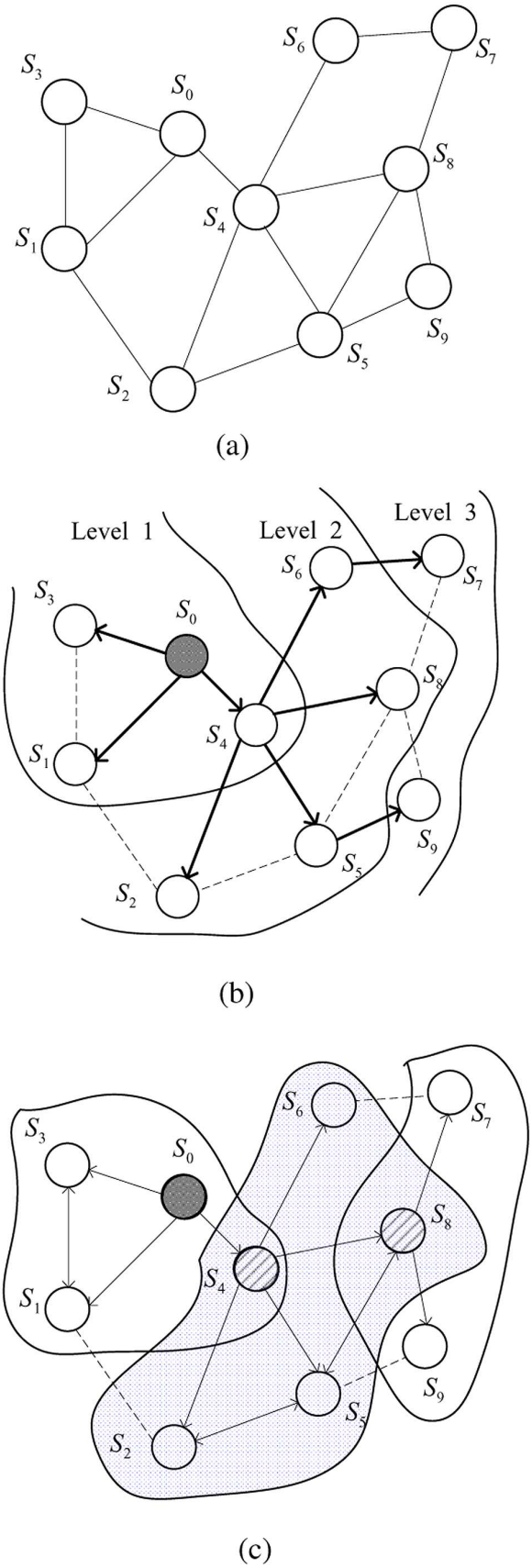


Fig. 1. Network topology for different synchronization schemes. (a) A network with 10 sensors $\{S_i\}_{i=0}^{9}$. (b) Tree-structure-based synchronization, where S_0 is the root node. (c) Cluster-structure-based synchronization, where S_0 is the reference node, S_4 and S_8 are gateway nodes.

another cluster, as shown in Fig. 1(c). Typical protocol in this class include reference broadcast synchronization (RBS) [10], pairwise broadcast synchronization (PBS) [11], and hierarchy referencing time synchronization (HRTS) [12]. Unfortunately, these protocols suffer from similar disadvantages to TPSN that they require substantial overhead to group sensors into clusters and are sensitive to reference nodes failures [36].

Finally, in fully distributed protocols, no global structure needs to be maintained and there is no special node such as root or gateway. With all sensors running exactly the same algorithm and communicating with nodes in their neighborhoods only, these protocols are robust to dynamic topology changes and are highly scalable. Existing protocols in this category can be further divided into two classes: physical-layer-based synchronization and packet-based synchronization. In the former scheme, based on the fact that clocks are generated by oscillators, only pulses emitted by oscillators are exchanged at the physical layer, and sensors are synchronized to transmit and receive at the same rate. Existing algorithms of this class are presented in [17]–[19]. Since oscillators cannot be adjusted, clocks are usually left unsynchronized in this scheme. Therefore, this scheme has limitations in application where timing information is required. On the other hand, in the latter scheme, time messages between two sensors are exchanged in the form of packets, and sensors synchronize with each other using timing information extracted from these packets. Several distributed algorithms have been proposed in [13]–[15]. However, these algorithms are based on the average consensus principle, where clock offsets are adjusted to an average common value, which may result in abrupt changes in local clocks and hence discontinuity in local times. For example, when a sensor with maximum clock offset goes down, the average of remaining clock offsets then reduces, and the common clock will head backwards after resynchronization. This can lead to serious faults, such as a node missing important deadlines or recording the same event multiple times. Moreover, message delays are not considered in these algorithms. As shown in [16], when message delays exist, the steady state in such consensus protocols will deviate from the average value and hence sensors cannot be synchronized with high accuracies. Since our objective is to provide global clock synchronization, we adopt the packet-based synchronization scheme in this paper and propose a fully distributed algorithm which overcomes aforementioned limitations of existing average consensus algorithms.

The rest paper is organized as follows. The cooperative clock synchronization algorithm is proposed in Section II. Convergence analysis is presented in Section III. In Section IV, simulations results are presented to demonstrate the superior performances of the proposed algorithm in terms of accuracy. Finally, conclusions are drawn in Section V.

II. Distributed Clock Synchronization Using Belief Propagation

A. System Model

Before discussing global clock synchronization across the whole network, we first consider two-way timing message exchange between two sensors S_i and S_j, which is shown in Fig. 2.

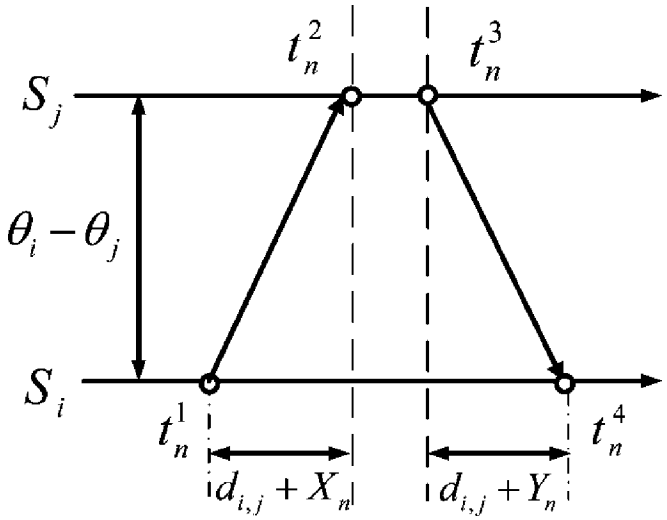


Fig. 2. The n^{th} round of two-way time-stamps exchange between two nodes i and j.

In the first round of message exchange, S_i sends a synchronization message to S_j at real time t_1^1 with its clock reading $c_i(t_1^1)$ embedded in the message. On reception of that message at real time t_1^2, S_j records its corresponding clock reading $c_j(t_1^2)$, and replies S_i at real time t_1^3. The replied message contains both $c_j(t_1^2)$ and $c_j(t_1^3)$. Then S_i records the reception time of S_j's reply as $c_i(t_1^4)$. The next round message exchange is the same as the first round except that S_i also embed time-stamp $c_i(t_1^4)$ in its synchronization message to S_j. After N rounds of message exchange, S_i informs S_j about $c_i(t_N^4)$ with one more message, and then both nodes obtain a set of time stamps $\{c_i(t_n^1), c_j(t_n^2), c_j(t_n^3), c_i(t_n^4)\}_{n=1}^N$. With $c_i(t) = t + \theta_i$, the above procedure can be modeled as

$$c_j(t_n^2) - \theta_j = c_i(t_n^1) - \theta_i + d_{i,j} + X_n \tag{2}$$

$$c_j(t_n^3) - \theta_j = c_i(t_n^4) - \theta_i - d_{i,j} - Y_n \tag{3}$$

where θ_i and θ_j represent the clock offsets at S_i and S_j, respectively; $d_{i,j}$ stands for the fixed portion of message delay between S_i and S_j; and X_n and Y_n are variable portions of the message delay. Since X_n and Y_n are due to numerous independent random processes, we can assume that X_n and Y_n are independent and identically distributed (i.i.d.) Gaussian random variables, and this assumption was experimentally verified in [10].

Based on (2) and (3), pairwise clock synchronization has been extensively studied in [11], [20]–[22]. However, extending these algorithms to network wide requires construction of spanning tree, and the disadvantages have been discussed in the previous section. In the following, we will develop a fully distributed algorithm which estimates the clock offsets using the Bayesian framework. First, denote $T_{\{i,j\},n} \triangleq c_j(t_n^2) + c_j(t_n^3) - c_i(t_n^1) - c_i(t_n^4)$ and $Z_n \triangleq X_n - Y_n$. By observing that the uplink and downlink undergo the same amount of fixed delay, we can rewrite the original model by adding (2) to (3), and obtain

$$T_{\{i,j\},n} = 2\theta_j - 2\theta_i + Z_n, \quad \text{for } n = 1, \ldots, N \tag{4}$$

where $\{Z_n\}_{n=1}^N$ are i.i.d. Gaussian random variables with zero mean and variance σ^2. Stacking the observations in a vector as $\mathbf{T}_{i,j} \triangleq [T_{\{i,j\},1}, \ldots, T_{\{i,j\},N}]$, we have the following likelihood function:

$$\begin{aligned} &p(\mathbf{T}_{i,j}|\theta_i, \theta_j) \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp\left\{-\frac{1}{2\sigma^2}\left\|\mathbf{T}_{i,j} - 2(\theta_j - \theta_i)\mathbf{1}\right\|^2\right\} \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp\left\{-\frac{1}{2}\left(\frac{\sigma^2}{N}\right)^{-1}\left[2(\theta_i - \theta_j) + \frac{1}{N}\mathbf{1}^T\mathbf{T}_{i,j}\right]^2\right\} \end{aligned} \tag{5}$$

where $\mathbf{1}$ is a vector with length N and elements 1, and the superscript $\{\cdot\}^T$ represents the transpose operation. Denote the prior distributions of θ_i and θ_j as $p(\theta_i)$ and $p(\theta_j)$, respectively, the joint posterior distribution $p(\theta_i, \theta_j|\mathbf{T}_{i,j})$ is given by

$$p(\theta_i, \theta_j|\mathbf{T}_{i,j}) \propto p(\mathbf{T}_{i,j}|\theta_i, \theta_j)p(\theta_i)p(\theta_j). \tag{6}$$

To make inferences on θ_i and θ_j, we need to obtain their respective marginal distributions, which are denoted as $g_i(\theta_i)$ and $g_j(\theta_j)$. Mathematically

$$\begin{aligned} g_i(\theta_i) &= \int p(\theta_i, \theta_j|\mathbf{T}_{i,j})\, \mathrm{d}\theta_j \\ &\propto \int p(\mathbf{T}_{i,j}|\theta_i, \theta_j)p(\theta_j)p(\theta_i)\, \mathrm{d}\theta_j \end{aligned} \tag{7}$$

and the clock offset θ_i can be estimated by maximizing $g_i(\theta_i)$, which is equivalent to maximizing the posterior distribution $p(\theta_i|\mathbf{T}_{i,j})$ and hence achieves the optimal solution in Bayesian sense. Similar derivation can be obtained for θ_j.

Extending the above idea to a network with M sensors, we need to find the joint posterior distribution $p(\theta_1, \ldots, \theta_M|\{\mathbf{T}_{i.j}\}_{i=1,\ldots,M,j\in\mathcal{B}_i})$, where $\mathcal{B}_i$ denotes the indices of neighboring sensors of S_i. Then the marginal distribution $g_i(\theta_i)$ is obtained by integrating out all other variables, that is

$$\begin{aligned} &g_i(\theta_i) \\ &= \int_{\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_M} \cdots \int p(\theta_1, \ldots, \theta_M|\{\mathbf{T}_{i.j}\}_{i=1,\ldots,M,j\in\mathcal{B}_i}) \\ &\qquad \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_{i-1}\, \mathrm{d}\theta_{i+1} \ldots \mathrm{d}\theta_M. \end{aligned} \tag{8}$$

Apparently, the joint distribution depends on interactions among all the variables and is complicated. Even when the joint distribution is obtained in closed-form, it is almost impossible to find the marginal distributions $\{g_i(\theta_i)\}_{i=1}^M$ by brute-force integration.

In order to provide a computationally efficient algorithm to calculate the marginal distributions, we observe that the state of clock offset at any sensor depends directly only on its neighboring sensors whose number is usually far less than the total number of variables. To explore such conditional independence structure, we make use of belief propagation in the following and propose an efficient algorithm which computes a set of marginal distributions from the joint posterior distribution without full integration in (8).

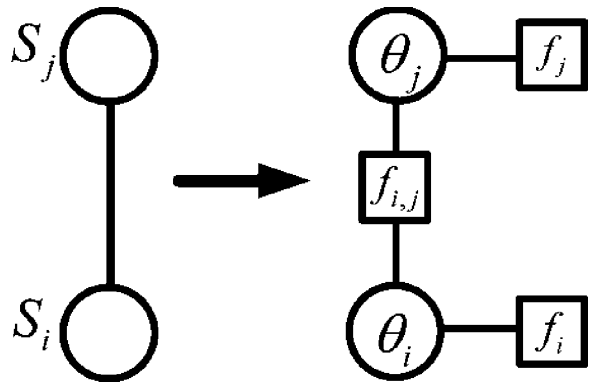


Fig. 3. The factor graph for two sensors S_i and S_j in pairwise clock synchronization. The functions represented by factor nodes f_i, f_j, and $f_{i,j}$ are $p(\theta_i)$, $p(\theta_j)$, and $p(\mathbf{T}_{i,j}|\theta_i, \theta_j)$, respectively.

B. Cooperative Clock Synchronization Using Belief Propagation (BP)

BP works on graphical models, and one of the most widely used graphical models, factor graph, is used in this paper. Let us consider the factor graph for pairwise clock synchronization first. The two random variables θ_i and θ_j are represented by variables nodes (circles), and they are connected to factor nodes (squares) which represent prior information and relationships between variable nodes. This is shown in Fig. 3, in which the factor node $f_{i,j}$ represents the likelihood function $p(\mathbf{T}_{i,j}|\theta_i, \theta_j)$ and is connected to both θ_i and θ_j. On the other hand, the factor node f_i represents the prior distribution of θ_i, i.e., $p(\theta_i)$, and is connected only to the variable node θ_i. This can be easily extended to a network with multiple sensors, since any two neighboring sensors establish a connection between each other in the same way as they do during pairwise clock synchronization. Therefore, for the network in Fig. 1(a), we can build a factor graph as shown in Fig. 4. Notice that the function represented by factor nodes $f_{i,j}$ and $f_{j,i}$ is the same likelihood function, either notation can be used.

With this factor graph, BP calculates the marginal distribution at every variable θ_i based on local message-passing. Specifically, it involves two kinds of messages:

- $b_i(\theta_i)$: belief of its own state at the variable node θ_i, which equals to product of all the incoming messages from neighboring factor nodes. That is

$$b_i^{(l+1)}(\theta_i) = m_{f_i \to \theta_i}^{(l)} \prod_{j \in \mathcal{B}_i} m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i) \tag{9}$$

  where $\mathcal{B}_i$ is the set of indices of S_i's neighboring sensors.
- $m_{x \to \theta_i}(\theta_i)$: message from the factor node x to the variable node θ_i. For $x = f_{i,j}$, it indicates $f_{i,j}$'s belief on θ_i's state, resulting from interactions between θ_i and other variable nodes connected to $f_{i,j}$. The message is defined as [26]

$$m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i) = \int p(\mathbf{T}_{i,j}|\theta_i, \theta_j) b_j^{(l)}(\theta_j)\, \mathrm{d}\theta_j. \tag{10}$$

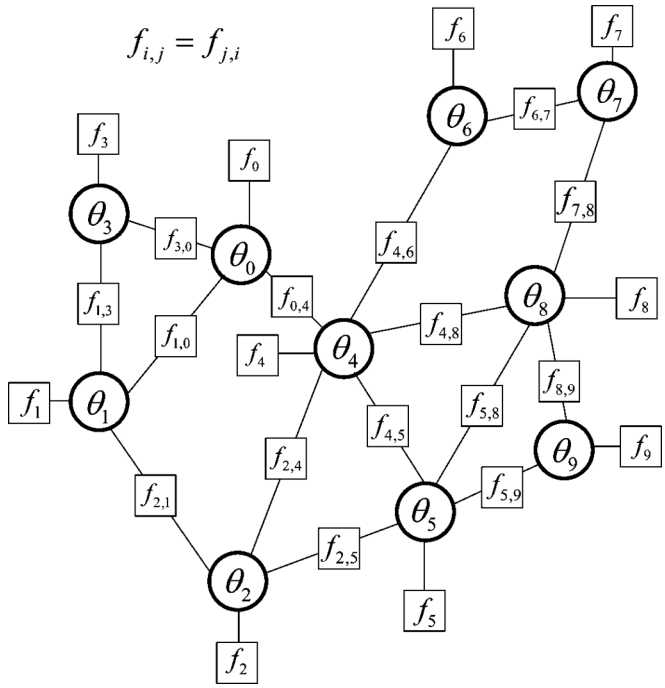


Fig. 4. The factor graph for the network in Fig. 1(a), where each variable node represent one clock offset. The factor nodes f_i represent the prior distribution $p(\theta_i)$ for clock offset θ_i at the i^{th} sensor S_i, and the factor nodes $f_{i,j}$ represent the likelihood function $p(\mathbf{T}_{i,j}|\theta_i, \theta_j)$ established by S_i and S_j through two-way message exchange.

For the message from the factor node f_i to the variable node θ_i (i.e., $x = f_i$), we have

$$\begin{aligned} m_{f_i \to \theta_i}^{(l)} &= p(\theta_i) \int b_j^{(l)}(\theta_j)\, \mathrm{d}\theta_j, \\ &= p(\theta_i) \end{aligned} \tag{11}$$

where (11) is due to the fact that beliefs are probability functions.

In the BP procedure, beliefs and messages are iteratively updated at variable nodes and factor nodes, respectively. For the global clock synchronization problem, denoting the belief of θ_i at iteration l as $b_i^{(l)}(\theta_i)$ and the message from $f_{i,j}$ to θ_i at iteration l as $m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i)$, each iteration is carried out in three steps:

1) Every variable node θ_i broadcast its current belief $b_i^{(l)}(\theta_i)$ to neighboring factor nodes $\{f_{i,j}\}_{j \in \mathcal{B}_i}$;
2) Acting like intermediate processors, the factor node $f_{i,j}$ calculates $m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i)$ with (10) based on the belief $b_j^{(l)}(\theta_j)$ it receives from θ_j, and then transmits this message to θ_i. Notice that from (11), $m_{f_i \to \theta_i}$ does not depend on information from other nodes and thus is fixed during iterations;
3) After collecting all the messages $\{m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i)\}_{j \in \mathcal{B}_i}$ from neighboring factor nodes, the variable node θ_i updates its belief with (9), and obtain $b_i^{(l+1)}(\theta_i)$.

After convergence, the belief at each variable node corresponds to the marginal distribution of that variable exactly (when the underlying topology is loop free) or approximately (when the

underlying topology has loops) [24], [25]. Therefore, the estimations obtained by maximizing these marginal distributions are optimal in Bayesian sense. Moreover, since each sensor runs exactly the same algorithm and communicates only with its neighbors, this procedure is fully distributed and computationally efficient.

C. Computation of Beliefs and Messages

In the following, exact expressions of the beliefs and messages at different nodes will be derived. First, let us focus on the beliefs before any message is exchanged. As introduced in Section I, a reference node S_0 with $\theta_0 = 0$ is assigned in the network and all other sensors $\{S_i\}_{i=1}^M$ synchronize their clocks to S_0. Mathematically, this means that the belief of θ_0 is deterministic and given by $b_0^{(l)}(\theta_0) = \delta(\theta_0, 0)$ for $l = 0, 1, 2, \ldots$, where $\delta(\theta_0, 0)$ is the Dirac delta function. On the other hand, for the rest of M sensors, it is reasonable to set their initial beliefs as their prior distributions, i.e., $b_i^{(0)}(\theta_i) = p(\theta_i)$. However, in practice, it is difficult to obtain a prior distribution on the clock offset. It can be assumed that θ_i is uniformly distributed in the range $[-\infty, +\infty]$. Notice that the uniform distribution can be equally represented as a Gaussian distribution with zero mean and infinite variance $p(\theta_i) \sim \mathcal{N}(\theta_i; 0, +\infty)$. Therefore, we have $b_i^{(0)}(\theta_i) \sim \mathcal{N}(\theta_i; 0, +\infty)$ for $i = 1, \ldots, M$.

At the first iteration, the message from the factor node $f_{i,j}$ to a neighboring variable node θ_i is defined in (10). If S_j is the reference node (for example from $f_{1,0}$ to θ_1 in Fig. 4), the message is given by

$$\begin{aligned} m_{f_{i,0} \to \theta_i}^{(0)}(\theta_i) &= \int p(\mathbf{T}_{i,0}|\theta_i, \theta_0)\delta(\theta_0, 0)\, \mathrm{d}\theta_0 \\ &\propto \exp\left\{-\frac{1}{2}\left(\frac{\sigma^2}{N}\right)^{-1}\left(2\theta_i + \frac{1}{N}\mathbf{1}^T\mathbf{T}_{i,0}\right)^2\right\}. \end{aligned} \quad (12)$$

It can be seen that the message in (12) is in the form of Gaussian distribution. By denoting its mean as $\nu_{0 \to i}^{(0)} \triangleq -\frac{1}{2N}\mathbf{1}^T\mathbf{T}_{i,0}$ and variance as $C_{0 \to i}^{(0)} \triangleq \frac{\sigma^2}{4N}$, we have

$$m_{f_{i,0} \to \theta_i}^{(0)}(\theta_i) \sim \mathcal{N}(\theta_i; \nu_{0 \to i}^{(0)}, C_{0 \to i}^{(0)}). \quad (13)$$

On the other hand, if S_j is not the reference node (for example from $f_{2,1}$ to θ_1 in Fig. 4), the message is given by

$$\begin{aligned} m_{f_{i,j} \to \theta_i}^{(0)}(\theta_i) &= \int p(\mathbf{T}_{i,j}|\theta_j, \theta_i) b_j^{(0)}(\theta_j)\, \mathrm{d}\theta_j \\ &\propto \int p(\mathbf{T}_{i,j}|\theta_j, \theta_i)\, \mathrm{d}\theta_j \quad (14) \\ &\propto 1 \quad (15) \end{aligned}$$

where in (14) we used $b_j^{(0)}(\theta_j) \sim \mathcal{N}(\theta_j; 0, +\infty)$ and (15) is due to the fact that the result from integration in (14) does not depend on θ_i. Therefore, from (9), (13), and (15), we can write the updated belief $b_i^{(1)}(\theta_i)$ at S_i in the Gaussian form, that is

$$b_i^{(1)}(\theta_i) \sim \mathcal{N}(\theta_i; \mu_i^{(1)}, P_i^{(1)}) \quad (16)$$

where $\mu_i^{(1)} = \nu_{0 \to i}^{(0)}$ and $P_i^{(1)} = C_{0 \to i}^{(0)}$ if S_i is directly connected to the reference node, and $\mu_i^{(1)} = 0$ and $P_i^{(1)} = +\infty$ otherwise.

At the next iteration, variable nodes broadcast their new beliefs to neighboring factor nodes and this results in new messages at the intermediate factor nodes. Specifically, with belief $b_i^{(1)}(\theta_i)$ at θ_i, the new message from $f_{k,i}$ to a neighboring node θ_k, $m_{f_{k,i} \to \theta_k}^{(1)}(\theta_k)$, can be obtained as

$$\begin{aligned} &m_{f_{k,i} \to \theta_k}^{(1)}(\theta_k) \\ &= \int p(\mathbf{T}_{k,i}|\theta_k, \theta_i) b_i^{(1)}(\theta_i)\, \mathrm{d}\theta_i \\ &\propto \int \exp\left\{-\frac{1}{2}\left(\frac{\sigma^2}{N}\right)^{-1}\left(2\theta_k - 2\theta_i + \frac{1}{N}\mathbf{1}^T\mathbf{T}_{k,i}\right)^2\right\} \\ &\quad \times \exp\left\{-\frac{1}{2}\left[P_i^{(1)}\right]^{-1}\left[\theta_i - \mu_i^{(1)}\right]^2\right\}\, \mathrm{d}\theta_i \\ &\propto \exp\left\{-\frac{1}{2}\left[\frac{\sigma^2}{4N} + P_i^{(1)}\right]^{-1}\left[\theta_k - \mu_i^{(1)} + \frac{1}{2N}\mathbf{1}^T\mathbf{T}_{k,i}\right]^2\right\}. \end{aligned} \quad (17)$$

It can be seen that the message in (17) is also in the Gaussian form. By denoting its mean as $\nu_{i \to k}^{(1)} = \mu_i^{(1)} - \frac{1}{2N}\mathbf{1}^T\mathbf{T}_{k,i}$ and variance as $C_{i \to k}^{(1)} = \frac{\sigma^2}{4N} + P_i^{(1)}$, we have

$$m_{f_{k,i} \to \theta_k}^{(1)}(\theta_k) \sim \mathcal{N}(\theta_k; \nu_{i \to k}^{(1)}, C_{i \to k}^{(1)}). \quad (18)$$

After collecting all the incoming messages from neighboring factor nodes, variable node θ_k updates its belief $b_k^{(1)}(\theta_k)$ by using (9). Since both the prior distribution and the incoming messages have the form of Gaussian, the updated belief of θ_k must be Gaussian distributed.

In general, it can be seen that all the messages and beliefs during the iterative procedure are in Gaussian forms. Denoting

$$b_j^{(l)}(\theta_j) \sim \mathcal{N}(\theta_j; \mu_j^{(l)}, P_j^{(l)}),\ \forall j \in \{1, \ldots, M\} \quad (19)$$

and with similar calculations in (17), we can always obtain the message

$$m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i) \sim \mathcal{N}(\theta_i; \nu_{j \to i}^{(l)}, C_{j \to i}^{(l)}) \quad (20)$$

with

$$C_{j \to i}^{(l)} = \frac{\sigma^2}{4N} + P_j^{(l)} \quad (21)$$

and

$$\nu_{j \to i}^{(l)} = \mu_j^{(l)} - \frac{1}{2N}\mathbf{1}^T\mathbf{T}_{i,j}. \quad (22)$$

The message in (20) represents the general form for all the messages circulating around the network. To see this, we notice that the belief at the reference node S_0, i.e., $\delta(\theta_0, 0)$, can be equally

represented in the Gaussian form with $P_0^{(l)} = 0$ and $\mu_0^{(l)} = 0$. Putting this into (21) and (22), it can be seen that (20) reduces to (13) and it represents the message from S_0 to its neighboring sensors. Moreover, for other sensors, their beliefs are initially in Gaussian forms with infinite variance, that is, $P_j^{(0)} = +\infty$ for $j \in \{1, \ldots, M\}$, hence (20) reduces to a constant and is consistent with (15).

After collecting incoming messages from neighboring factor nodes, the belief of θ_i is updated by using (9), that is

$$b_i^{(l+1)}(\theta_i) = p(\theta_i) \prod_{j \in \mathcal{B}_i} m_{f_{i,j} \to \theta_i}^{(l)}(\theta_i). \tag{23}$$

Substituting $p(\theta_i) \sim \mathcal{N}(\theta_i; 0, +\infty)$ and (20) into (23), the variance of $b_i^{(l+1)}(\theta_i)$ is obtained as

$$\begin{aligned} \left[P_i^{(l+1)}\right]^{-1} &= (+\infty)^{-1} + \sum_{j \in \mathcal{B}_i} \left[C_{j \to i}^{(l)}\right]^{-1} \\ &= \sum_{j \in \mathcal{B}_i} \left[\frac{\sigma^2}{4N} + P_j^{(l)}\right]^{-1} \end{aligned} \tag{24}$$

and the mean of $b_i^{(l+1)}(\theta_i)$ is obtained as

$$\begin{aligned} \mu_i^{(l+1)} &= P_i^{(l+1)} \left\{ (+\infty)^{-1} \times 0 + \sum_{j \in \mathcal{B}_i} \left[C_{j \to i}^{(l)}\right]^{-1} \nu_{j \to i}^{(l)} \right\}, \\ &= P_i^{(l+1)} \sum_{j \in \mathcal{B}_i} \left[C_{j \to i}^{(l)}\right]^{-1} \nu_{j \to i}^{(l)}, \qquad (25) \\ &= P_i^{(l+1)} \sum_{j \in \mathcal{B}_i} \left[\frac{\sigma^2}{4N} + P_j^{(l)}\right]^{-1} \left[\mu_j^{(l)} - \frac{1}{2N} \mathbf{1}^T \mathbf{T}_{i,j}\right] \qquad (26) \end{aligned}$$

where from (25) to (26), we have used (21) and (22). Notice that an equivalent form of (25) is

$$\mu_i^{(l+1)} = \frac{\sum_{j \in \mathcal{B}_i} \left[C_{j \to i}^{(l)}\right]^{-1} \nu_{j \to i}^{(l)}}{\sum_{j \in \mathcal{B}_i} \left[C_{j \to i}^{(l)}\right]^{-1}} \tag{27}$$

it can be seen from (27) that each variable θ_i updates its mean of belief as weighted average of expectations from neighboring factor nodes, and expectations with small variance will contribute more to θ_i's update, and hence uncertainties due to noises and random delays are reduced. In contrast, although algorithms based on the average consensus principle have similar forms as in (27), the averaging coefficients are usually fixed to be constant.

At the end of iteration l, sensor S_i can estimate its clock offset by maximizing the belief $b_i^{(l)}(\theta_i)$ with respect to θ_i. Since $b_i^{(l)}(\theta_i)$ is Gaussian, the optimal estimation for θ_i at iteration l is given by $\hat{\theta}_i^{(l)} = \mu_i^{(l)}$. This iterative procedure is formally given in Algorithm 1. Notice that factor nodes are imaginary for derivation of algorithm only. In practice, the update is performed at individual sensor directly.

Algorithm 1: Distributed clock synchronization using belief propagation

1. **Initialization:**
2. Set the belief at the reference node S_0 as $b_0(\theta_0) \sim \mathcal{N}(\theta_0; \mu_0, P_0)$, where $\mu_0 = 0$ and $P_0 = 0$.
3. Set the belief at S_i as $b_i^{(0)}(\theta_i) \sim \mathcal{N}(\theta_i; \mu_i^{(0)}, P_i^{(0)})$ for $i = 1, \ldots, M$, where $\mu_i^{(0)} = 0$ and $P_i^{(0)} = +\infty$.
4. **Iteration until convergence**:
5. **for** the l^{th} iteration **do**
6. **sensors** S_i with $i = 1 : M$ **in parallel**
7. broadcast the current belief $b_i^{(l-1)}(\theta_i)$ to neighboring sensors;
8. receive $b_j^{(l-1)}(\theta_j)$ from its neighboring sensor S_j, where $j \in \mathcal{B}_i$ and $j \neq i$;
9. update its belief as $b_i^{(l)}(\theta_i) \sim \mathcal{N}(\theta_i; \mu_i^{(l)}, P_i^{(l)})$, where
$$\left[P_i^{(l)}\right]^{-1} = \sum_{j \in \mathcal{B}_i} \left[\frac{\sigma^2}{4N} + P_j^{(l-1)}\right]^{-1}$$
and
$$\mu_i^{(l)} = P_i^{(l)} \sum_{j \in \mathcal{B}_i} \left[\frac{\sigma^2}{4N} + P_j^{(l-1)}\right]^{-1} \left[\mu_j^{(l-1)} - \frac{1}{2N} \mathbf{1}^T \mathbf{T}_{i,j}\right].$$
10. estimate its clock offset as $\hat{\theta}_i^{(l)} = \mu_i^{(l)}$.
11. **end parallel**

12. **end for**

Remark 1: For a network with only two sensors, say S_i and S_j, it can be seen that $b_i^{(0)}(\theta_i) = p(\theta_i) m_{f_{i,j} \to \theta_i}^{(0)}(\theta_i) = p(\theta_i) \int p(\mathbf{T}_{i,j} | \theta_i, \theta_j) p(\theta_j) \, \mathrm{d}\theta_j$, which equals to the marginal distribution $g_i(\theta_i)$ in (7). Therefore, the algorithm based on belief propagation includes the pairwise clock synchronization as a special case.

Remark 2: The result in (15) indicates that if the belief at θ_j has an infinite variance, the message from θ_j to its neighbor sensor θ_i has no impact on θ_i's belief update. Notice that every sensor except the reference node has an infinite variance in their beliefs initially, in order to save energy, a sensor can keep silence and only receive messages at the beginning stage of synchronization, and start to broadcast when its belief changes.

Remark 3: The proposed algorithm is robust to both packet losses and node failures. For the former issue, packets correspond to time-stamps obtained in two-way message exchanges between two sensors. From (21) and (22), it can be seen that both $C_{j \to i}^{(l)}$ and $\mu_{j \to i}^{(l)}$, i.e., the variance and mean of the message $m_{j \to i}^{(l)}(\theta_i)$, depend on the number of rounds of message exchange N and the values of time-stamps. If time-stamps in the k^{th} round are lost, N in (21) and (22) would become $N - 1$,

and the only consequence is that $C_{j\longrightarrow i}^{(l)}$ increases due to less timing information. Moreover, this increased variance is further reflected in the stage of belief update in (24) and (27). For the latter issue, since our algorithm is fully distributed, communications occur only between neighboring sensors and no global structure is required. When any sensor, for example S_j (which is a neighbor of S_i), failed after the l^{th} iteration of belief updates, S_i will set $C_j^{(l+1)} = \infty$ in (27) and its updated belief $b_i^{(l+1)}(\theta_i)$ will automatically remove the contribution from the failed node S_j.

III. Convergence Analysis

It is well known that algorithms based on belief propagation converge if the underlying topology has no loops, for example in tree structure. However, for general topology, the issue of convergence is poorly understood and difficult to prove. As observed numerically in [27], when there exist loops, these algorithms can diverge. A sufficient condition for convergence was proposed in [28] for Gaussian graphical models. Unfortunately, it requires the knowledge of the joint posterior distribution of all variables, and is difficult to verify for large sensor networks. Nevertheless, we can establish convergence of our proposed algorithm as follows.

Theorem 1: The beliefs at variable nodes $\{\theta_i\}_{i=1}^M$ in Algorithm 1 always converge for strongly connected networks[1] in the sense that there exists a unique fixed point $\{P_i^*, \mu_i^*\}$ such that

$$\lim_{l\longrightarrow\infty} P_i^{(l)} = P_i^* \tag{28}$$

and

$$\lim_{l\longrightarrow\infty} \mu_i^{(l)} = \mu_i^*. \tag{29}$$

Proof: Notice from (24) that $P_i^{(l)}$ is updated independently of $\mu_i^{(l)}$, we first analyze the convergence property of $P_i^{(l)}$. Next, notice from (27) that the update of $\mu_i^{(l)}$ involves $P_i^{(l)}$, we analyze the convergence property of $\mu_i^{(l)}$ under the assumption that $P_i^{(l)}$ has converged.

1) Convergence of $P_i^{(l)}$: For notation simplicity, we denote $\alpha \triangleq \frac{\sigma^2}{4N}$ and $K_i^{(l)} \triangleq \left[P_i^{(l)}\right]^{-1}$. The update equation for $P_i^{(l)}$ in (24) is then rewritten as

$$K_i^{(l+1)} = \sum_{j\in\mathcal{B}_i} \frac{1}{\alpha + \frac{1}{K_j^{(l)}}}. \tag{30}$$

Since $P_i^{(l)}$ is the variance, we always have $0 \le P_i^{(l)} \le +\infty$ and hence $K_i^{(l)} \ge 0$ for any $i \in \{1,\ldots,M\}$ and $l = 0,1,2,\ldots$. Denote the size of $\mathcal{B}_i$ as $|\mathcal{B}_i|$, which is the number of S_i's neighboring sensors, and bring $K_j^{(l)} \ge 0$ into (30), we have

$$0 \le K_i^{(l)} \le \frac{|\mathcal{B}_i|}{\alpha} \le \frac{M}{\alpha}, \quad \forall\, l = 0,1,2,\ldots. \tag{31}$$

[1]strongly connected network: a network is called strongly connected if a path, with either single hop or multiple hops, exists between any two nodes.

Therefore, the sequence $\{K_i^{(0)}, K_i^{(1)}, K_i^{(2)}, \ldots\}$ is bounded for any $i \in \{1,\ldots,M\}$. Furthermore, we show in the following that this sequence is also monotonic increasing. First, we have

$$\begin{aligned} K_i^{(l+1)} - K_i^{(l)} &= \sum_{j\in\mathcal{B}_i} \frac{1}{\alpha + \frac{1}{K_j^{(l)}}} - \sum_{j\in\mathcal{B}_i} \frac{1}{\alpha + \frac{1}{K_j^{(l-1)}}} \\ &= \sum_{j\in\mathcal{B}_i} \frac{K_j^{(l)} - K_j^{(l-1)}}{(1+\alpha K_j^{(l)})(1+\alpha K_j^{(l-1)})} \\ &\ge \frac{1}{(1+M)^2} \sum_{j\in\mathcal{B}_i} \left[K_j^{(l)} - K_j^{(l-1)}\right] \end{aligned} \tag{32}$$

where we used the upper-bound in (31). Denote the M-by-M adjacency matrix of a graph as $\mathbf{A}$ with its $(i,j)^{\text{th}}$ element given by

$$\mathbf{A}(i,j) = \begin{cases} 1, & S_i \text{ and } S_j \text{ are neighbors}, i \ne j \\ 0, & \text{otherwise.} \end{cases}$$

And denoting $\mathbf{K}^{(l)} \triangleq [K_1^{(l)}, K_2^{(l)}, \ldots, K_M^{(l)}]$, we can rewrite the inequality in (32) as

$$\begin{aligned} \mathbf{K}^{(l+1)} - \mathbf{K}^{(l)} &\ge \frac{1}{(1+M)^2} \times \mathbf{A}\left[\mathbf{K}^{(l)} - \mathbf{K}^{(l-1)}\right] \\ &\ge \frac{1}{(1+M)^{2l}} \times \mathbf{A}^l \left[\mathbf{K}^{(1)} - \mathbf{K}^{(0)}\right]. \end{aligned} \tag{33}$$

Notice that $\mathbf{K}^{(0)} = [0,\ldots,0]$ due to $P_i^{(0)} = +\infty$ for $i = 1,\ldots,M$. Together with the fact that $K_i^{(l)} \ge 0$ for all l, we have

$$\mathbf{K}^{(1)} - \mathbf{K}^{(0)} \ge \mathbf{0}. \tag{34}$$

Moreover, the $(i,j)^{\text{th}}$ element of $\mathbf{A}^l$ counts the number of paths of length l between S_i and S_j and hence is always nonnegative. Therefore, we have

$$\mathbf{A}^l \left[\mathbf{K}^{(1)} - \mathbf{K}^{(0)}\right] \ge \mathbf{0}. \tag{35}$$

Putting (35) into (33) gives

$$\mathbf{K}^{(l+1)} - \mathbf{K}^{(l)} \ge \mathbf{0}, \tag{36}$$

and hence $K_i^{(l+1)} - K_i^{(l)} \ge 0$ for any $i \in \{1,\ldots,M\}$. Therefore, the sequence $\{K_i^{(0)}, K_i^{(1)}, K_i^{(2)}, \ldots\}$ is monotonic increasing.

From the monotone convergence theorem [29], every bounded monotone sequence is convergent. Therefore, we can draw the conclusion that the sequence $\{K_i^{(0)}, K_i^{(1)}, K_i^{(2)}, \ldots\}$ is convergent, and denoting its limit as K_i^*, we have

$$\lim_{l\longrightarrow\infty} K_i^{(l)} = K_i^* \tag{37}$$

which equals to (28).

2) Convergence of $\mu_i^{(l)}$: For the update equation of $\mu_i^{(l)}$, putting the definition of $\mathbf{T}_{i,j}$ into (22), we have

$$\begin{aligned}\nu_{j\longrightarrow i}^{(l)} &= \mu_j^{(l)} - \frac{1}{2N}\mathbf{1}^T\mathbf{T}_{i,j} \\ &= \mu_j^{(l)} - \theta_j + \theta_i - \underbrace{\frac{1}{2N}\sum_{n=1}^{N} Z_n}_{\triangleq Z_{j,i}} \end{aligned} \tag{38}$$

where $Z_{j,i}$ represents the average of Gaussian random delays in the procedure of message exchange between S_i and S_j, and it is independent of the iteration number l. Putting (38) into (27), the update equation can be rewritten as

$$\begin{aligned}\underbrace{\mu_i^{(l)} - \theta_i}_{\triangleq \tilde{\mu}_i^{(l)}} &= \frac{1}{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}} \\ &\quad\cdot \sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}[\underbrace{\mu_j^{(l-1)} - \theta_j}_{\triangleq \tilde{\mu}_j^{(l-1)}} - Z_{j,i}] \end{aligned} \tag{39}$$

with $i \in \{1,\ldots,M\}$ and $j \in \{0,1,\ldots,M\}$. Notice that for $j = 0$, the variable node θ_0 corresponds to the reference node S_0. Since the belief at S_0 does not change, we have $\theta_0 = 0$, $\mu_0^{(l)} = 0$ and $P_0^{(l)} = 0$ for all l, and hence $\tilde{\mu}_0^{(l-1)} = 0$ and $C_{0\longrightarrow i}^{(l-1)} = \alpha$ from (21). Therefore, if the reference node is a direct neighbor of S_i, i.e., $0 \in \mathcal{B}_i$, the update (39) is equivalent to

$$\tilde{\mu}_i^{(l)} = \frac{\sum_{j\in\mathcal{B}_i, j\neq 0}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}\left[\tilde{\mu}_j^{(l-1)} - Z_{j,i}\right]}{\alpha + \sum_{j\in\mathcal{B}_i, j\neq 0}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}}. \tag{40}$$

Otherwise, if S_i and S_0 are not directly connected, i.e., $0 \notin \mathcal{B}_i$, the update (39) is

$$\tilde{\mu}_i^{(l)} = \frac{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}\left[\tilde{\mu}_j^{(l-1)} - Z_{j,i}\right]}{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^{(l-1)}\right]^{-1}} \tag{41}$$

with $i, j \in \{1,\ldots,M\}$. Furthermore, with l large enough, it can be assumed that $P_j^{(l)}$ has converged to P_j^* for $j = 1,\ldots,M$, and hence $C_{j\longrightarrow i}^{(l-1)}$ converges to $C_{j\longrightarrow i}^* \triangleq \alpha + P_j^*$. Putting $\{C_{j\longrightarrow i}^*\}_{j=1}^M$ into (40) and (41), and combining these two cases, we can rewrite (39) as

$$\tilde{\mu}_i^{(l)} = \begin{cases} \dfrac{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}\left[\tilde{\mu}_j^{(l-1)} - Z_{j,i}\right]}{\alpha + \sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}}, & 0 \in \mathcal{B}_i, \\ \dfrac{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}\left[\tilde{\mu}_j^{(l-1)} - Z_{j,i}\right]}{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}}, & 0 \notin \mathcal{B}_i \end{cases} \tag{42}$$

where $i, j \in \{1,\ldots,M\}$.

To further simplify (42), we define an M-by-M matrix $\mathbf{Q}$ whose $(i,j)^{\text{th}}$ element is

$$\mathbf{Q}(i,j) = \begin{cases} \dfrac{\left[C_{j\longrightarrow i}^*\right]^{-1}}{\alpha + \sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}}, & 0 \in \mathcal{B}_i, j \in \mathcal{B}_i, i \neq j, \\ \dfrac{\left[C_{j\longrightarrow i}^*\right]^{-1}}{\sum_{j\in\mathcal{B}_i}\left[C_{j\longrightarrow i}^*\right]^{-1}}, & 0 \notin \mathcal{B}_i, j \in \mathcal{B}_i, i \neq j \\ 0, & \text{otherwise.} \end{cases} \tag{43}$$

Moreover, denoting $\tilde{\boldsymbol{\mu}}^{(l)} \triangleq [\tilde{\mu}_1^{(l)}, \tilde{\mu}_2^{(l)}, \ldots, \tilde{\mu}_M^{(l)}]$ and

$$\mathbf{Z} \triangleq \begin{bmatrix} 0 & Z_{2,1} & \cdots & Z_{N,1} \\ Z_{1,2} & 0 & \cdots & Z_{N,2} \\ \vdots & & & \vdots \\ Z_{1,N} & \cdots & Z_{N-1,N} & 0 \end{bmatrix}$$

we can rewrite (42) as

$$\tilde{\boldsymbol{\mu}}^{(l)} = \mathbf{Q}\tilde{\boldsymbol{\mu}}^{(l-1)} - \operatorname{diag}\{\mathbf{QZ}\} \tag{44}$$

where $\operatorname{diag}\{\mathbf{A}\}$ represents an N-by-1 vector formed by diagonal elements of the matrix $\mathbf{A}$. If $\tilde{\boldsymbol{\mu}}^{(l)}$ converges, it is easy to see that the sequence $\{\mu_i^{(0)}, \mu_i^{(1)}, \mu_i^{(2)}, \ldots\}$ is convergent for any $i \in \{1,\ldots,M\}$. And we show in the following that $\tilde{\boldsymbol{\mu}}^{(l)}$ converges pointwise as l increases.

First, for finite integers $l > 0$ and $n > 0$, by using (44), we have

$$\begin{aligned}\tilde{\boldsymbol{\mu}}^{(n+l)} - \tilde{\boldsymbol{\mu}}^{(l)} &= \left[\tilde{\boldsymbol{\mu}}^{(n+l)} - \tilde{\boldsymbol{\mu}}^{(n+l-1)}\right] + \left[\tilde{\boldsymbol{\mu}}^{(n+l-1)} - \tilde{\boldsymbol{\mu}}^{(n+l-2)}\right] \\ &\quad + \cdots + \left[\tilde{\boldsymbol{\mu}}^{(l+1)} - \tilde{\boldsymbol{\mu}}^{(l)}\right] \\ &= \mathbf{Q}^{n+l-1}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right] + \mathbf{Q}^{n+l-2}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right] \\ &\quad + \cdots + \mathbf{Q}^{l}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right] \qquad (45) \\ &= \mathbf{Q}^{l}\sum_{k=0}^{n-1}\mathbf{Q}^{k}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right] \\ &< \mathbf{Q}^{l}\sum_{k=0}^{+\infty}\mathbf{Q}^{k}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right] \qquad (46) \end{aligned}$$

where (45) uses the fact that $\mathbf{Q}$ and $\mathbf{Z}$ do not change during the iteration. Since all elements in $\mathbf{Q}$ are nonnegative, together with the fact that its row sums $\sum_{j=1}^M \mathbf{Q}(i,j) \leq 1$ for all $i \in \{1,\ldots,M\}$ and $\sum_{j=1}^M \mathbf{Q}(i,j) < 1$ for the i^{th} row with $0 \in \mathcal{B}_i$, the matrix $\mathbf{Q}$ is substochastic. Moreover, notice that $\mathbf{Q}$ corresponds to a graph matrix of a strongly connected network, $\mathbf{Q}$ is irreducible [31]. Based on these two facts, denoting the eigenvalues of $\mathbf{Q}$ as $\{\lambda_i\}_{i=1}^M$, it follows from Perron-Frobenius theorem that $-1 < \lambda_i < 1$ for $i = 1,\ldots,M$[31], [32].

Next, by factorizing $\mathbf{Q}$ with eigenvalue decomposition as $\mathbf{Q} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with M eigenvalues along the diagonal and $\mathbf{V}$ is a matrix formed by eigenvectors, we can rewrite (46) as

$$\tilde{\boldsymbol{\mu}}^{(n+l)} - \tilde{\boldsymbol{\mu}}^{(l)} < \mathbf{V}\left(\boldsymbol{\Lambda}^l\sum_{k=0}^{+\infty}\boldsymbol{\Lambda}^k\right)\mathbf{V}^{-1}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right]. \tag{47}$$

Since $-1 < \lambda_i < 1$ for $i = 1,\ldots,M$, the sum of power series $\{\lambda_i^k\}_{k=0}^{+\infty}$ must converge, and $\sum_{k=0}^{+\infty}\boldsymbol{\Lambda}^k$ is a diagonal matrix

with its $(i,i)^{\text{th}}$ element given by $\frac{1}{1-\lambda_i}$. Defining this matrix as $\mathbf{\Sigma}$ and putting it into (47), we have

$$\tilde{\boldsymbol{\mu}}^{(n+l)} - \tilde{\boldsymbol{\mu}}^{(l)} < \mathbf{V}\boldsymbol{\Lambda}^l\mathbf{\Sigma}\mathbf{V}^{-1}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right]. \tag{48}$$

Notice that $\tilde{\boldsymbol{\mu}}^{(1)}$, $\tilde{\boldsymbol{\mu}}^{(0)}$, $\mathbf{V}$, and $\mathbf{\Sigma}$ are fixed, and moreover, $\boldsymbol{\Lambda}^l$ is a diagonal matrix with its $(i,i)^{\text{th}}$ element given by λ_i^l, we can rewrite (48) as

$$\tilde{\mu}_i^{(n+l)} - \tilde{\mu}_i^{(l)} < \eta_i\lambda_i^l, \quad \forall\, i = 1,\ldots,M \tag{49}$$

where η_i is a constant defined by the product of the i^{th} row of $\mathbf{V}$ and the vector $\mathbf{\Sigma}\mathbf{V}^{-1}\left[\tilde{\boldsymbol{\mu}}^{(1)} - \tilde{\boldsymbol{\mu}}^{(0)}\right]$. Since $-1 < \lambda_i < 1$ for $i = 1,\ldots,M$, for any real number $\epsilon > 0$, we can always find certain $l > 0$ such that $\eta_i\lambda_i^l < \epsilon$, and hence $\tilde{\mu}_i^{(n+l)} - \tilde{\mu}_i^{(l)} < \epsilon$. Therefore, according to Cauchy's criterion, we can finally draw the conclusion that the sequence $\{\tilde{\mu}_i^{(0)}, \tilde{\mu}_i^{(1)}, \tilde{\mu}_i^{(2)}, \ldots\}$ converges for any $i \in \{1,\ldots,M\}$. And hence the sequence $\{\mu_i^{(0)}, \mu_i^{(1)}, \mu_i^{(2)}, \ldots\}$ is convergent for any $i \in \{1,\ldots,M\}$. ■

IV. Simulation Results and Discussions

To validate the performances of our proposed algorithm, simulation results are presented and compared to consensus algorithms. As introduced in Section I, existing algorithms in [13]–[16] are based on the average consensus principle. Since message delays are not considered in [13]–[15], their performances deteriorate significantly when there exist fixed delays or random delays. Therefore, we compare our algorithm with the consensus algorithm in [16] which explicitly considers random delays. The error of the consensus algorithm is measured by difference between the estimated clock offset and the average value of all clock offsets. That is

$$\text{error}(\theta_i)_{\text{consensus}} = \left|\hat{\theta}_i - \frac{1}{M}\sum_{i=0}^{M-1}\theta_i\right| \tag{50}$$

for $i \in \{0,1,\ldots,M-1\}$. On the other hand, our proposed algorithm adjusts clocks to their true values, and hence avoids the issue of time discontinuity in the consensus algorithm. And the error is measured by difference between the estimated clock offset and its true value. That is

$$\text{error}(\theta_i)_{\text{proposed}} = \left|\hat{\theta}_i - \theta_i\right| \tag{51}$$

for $i \in \{1,\ldots,M-1\}$. The parameters used in simulations are as follows. Clock offsets and fixed delays are uniformly drawn from ranges $[-30,30]$ and $[0,10]$, respectively. The variance of the random delay is 1. The number of message exchange rounds is 4. Each point in the figures is an average of 10 000 independent simulation runs. Mean squared error (MSE) at each point is presented to measure the estimation accuracy.

Simulations are carried out for a network of 25 nodes as shown in Fig. 5(a). All nodes are distributed in a standard grid pattern and communicate through a tree-structured topology. First, the convergence rate at two nodes is presented in Fig. 5(b). Node A is directly connected to the reference node, while Node B lies far away. Since Node B is 4 hops away from the reference node, its belief stays unchanged at the beginning. This is unavoidable as it takes time for messages to propagate to distant sensors.

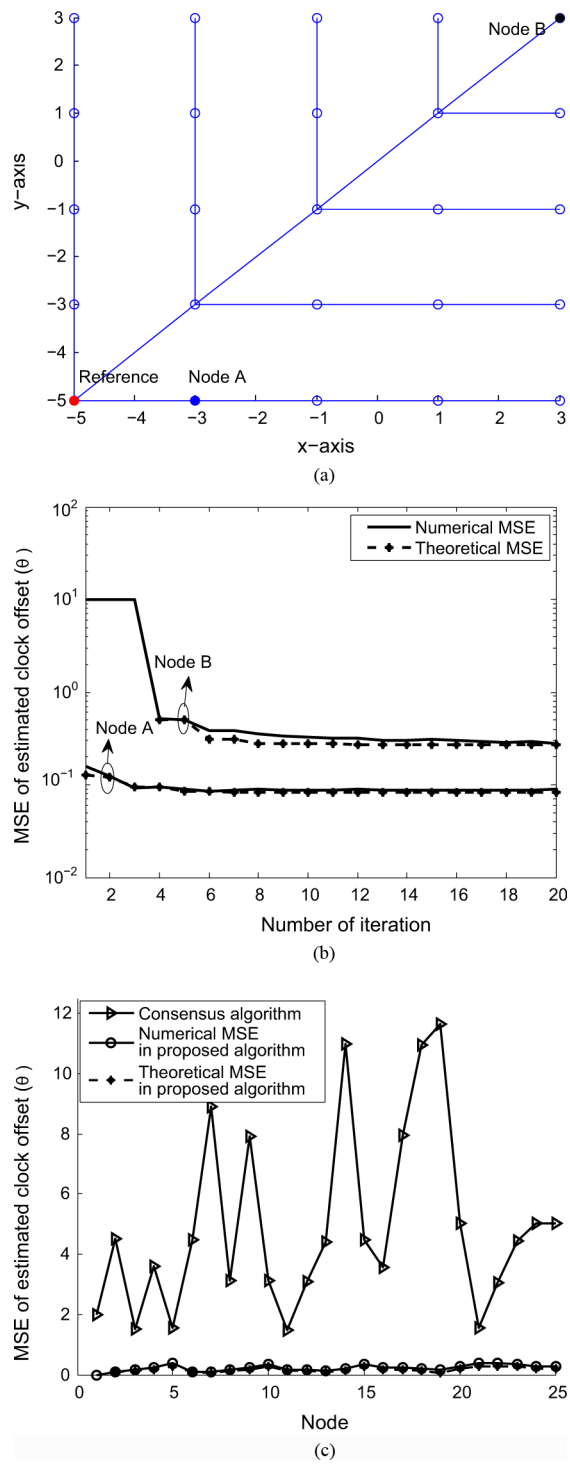
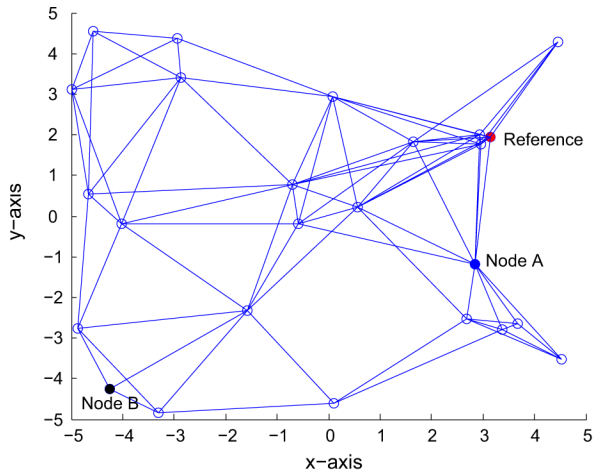
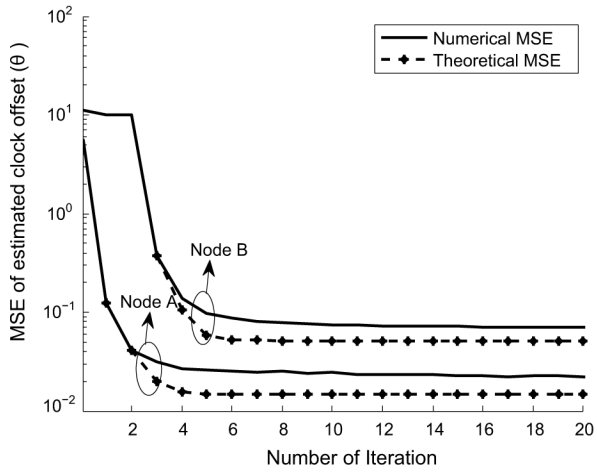


Fig. 5. Convergence rate and MSE for network with tree topology. (a) Network topology. (b) Convergence rate of the proposed algorithm at Node A and Node B. (c) MSE of estimated clock offset at individual sensors.

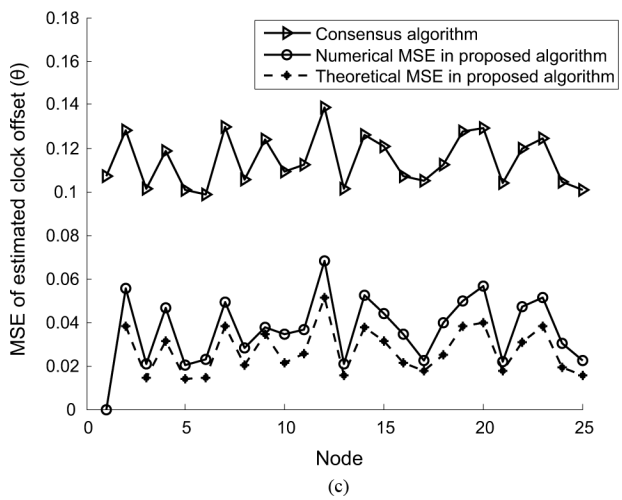
However, once a sensor starts to update its belief, it converges quickly in 4 iterations. Furthermore, due to the fact that beliefs are represented in the forms of Gaussian, their variances correspond to the theoretical MSE for estimations in each iteration, and it shows in Fig. 5(b) that the numerical and theoretical MSE overlap after convergence, which confirms that beliefs converge to marginal distributions exactly for tree topologies. Next, the converged MSE at individual sensors are shown in Fig. 5(c),



(a)



(b)



(c)

Fig. 6. Convergence rate and MSE for network with loopy topology. (a) Network topology. (b) Convergence rate of the proposed algorithm at Node A and Node B. (c) MSE of estimated clock offset at individual sensors.

where 20 iterations are carried to ensure convergence for both algorithms. It can be seen that the theoretical MSE of the proposed algorithm overlaps with its numerical MSE at individual sensors. Moreover, the proposed algorithm outperforms the consensus algorithm and reaches an accuracy at the order of 10^{-1}.

Another set of simulations is conducted on a loopy network as shown in Fig. 6(a). All 25 nodes are randomly distributed in the squared area and form a communication topology with multiple loops. Convergence performances at two nodes are presented in Fig. 6(b). Similar conclusion about their convergence rate can be drawn as in Fig. 5(b). On the other hand, the numerical MSE of the proposed algorithm is lower bounded by its theoretical MSE, and such a gap exists because beliefs converge to marginal distributions approximately for loopy topologies [24], [25]. When compared to the consensus algorithm, the proposed algorithm still show improved performance, and achieves an accuracy at the order of 10^{-2}, as shown in Fig. 6(c).

V. Conclusion

In this paper, we propose a fully distributed algorithm based on belief propagation for global clock synchronization in wireless sensor networks. The proposed algorithm requires communications only between neighboring sensors and is implemented distributedly in the network. Therefore, it has low overhead and can achieve robust and scalable synchronization. Moreover, it is shown analytically that the proposed algorithm converges for strongly connected networks. Compared to existing algorithms based on the average consensus principle, our proposed algorithm synchronizes clocks with a consistent reference value instead of adjusting clocks to an average value. This helps avoiding the issue of time discontinuity. Simulation results show that our proposed algorithm achieves better accuracy than consensus algorithms. Furthermore, the belief updated at each sensor provides an accurate prediction on the algorithm's performance in terms of MSE.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] N. Bulusu and S. Jha, *Wireless Sensor Networks: A Systems Perspective*. Reading, MA: Artech House, 2005.

[3] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 3, pp. 281–323, 2005.

[4] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," Univ. Calif. Center for Embedded Network Sensing, Los Angeles, CA, 2003.

[5] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Trans. Comput.*, vol. 36, no. 8, pp. 933–940, Aug. 1987.

[6] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, 1st ed. New York: Wiley, 2005.

[7] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. 2nd Int. Conf. Embedded Netw. Sens. Syst.*, 2004, pp. 39–49.

[8] J. V. Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *Proc. Int. Workshop on Wireless Sens. Netw. Appl. (WSNA)*, San Diego, CA, Sep. 2003.

[9] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu, "Tiny-sync: Tight time synchronization for wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 3, no. 8, Jun. 2007.

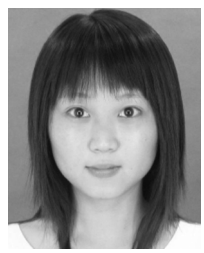
[10] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. 5th Symp. Operat. Syst. Design Implement.*, Boston, MA, Dec. 2002, pp. 147–163.

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

[12] H. Dai and R. Han, "TSync: A lightweight bidirectional time synchronization service for wireless sensor networks," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 1, pp. 125–139, 2004.

[13] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–225, Feb. 2006.

[14] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *Proc. 45th IEEE Conf. Decision and Contr.*, San Diego, CA, Dec. 2006, pp. 4915–4920.
[15] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proc. IEEE Conf. Decision and Contr.*, New Orleans, LA, Dec. 2007, pp. 2289–2294.
[16] G. Xiong and S. Kishore, "Analysis of distributed consensus time synchronization with Gaussian delay over wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, p. 9, 2009.
[17] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1085–1099, May 2005.
[18] A.-S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2725–2748, 2006.
[19] O. Simeone and U. Spagnolini, "Distributed synchronization in wireless networks," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 81–97, Sep. 2008.
[20] K.-L. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 766–777, Apr. 2007.
[21] M. Leng and Y.-C. Wu, "On clock synchronization algorithms for wireless sensor networks with unknown delay," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 182–190, Jan. 2010.
[22] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe, "Some improved and generalized estimation schemes for clock synchronization of listening nodes in wireless sensor networks," *IEEE Trans. Commun.*, vol. 58, no. 1, pp. 63–67, Jan. 2010.
[23] F. R. Kschischang, B. J. Frey, and H.-S. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.
[25] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding Belief Propagation and its Generalization," in *Exploring Artificial Intelligence in the New Millennium*. New York: Elsevier, 2003, ch. 8, pp. 239–236, ISBN 1558608117.
[26] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.
[27] P. Rusmevichientong and B. V. Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 745–765, 2001.
[28] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum interpretation and analysis of Gaussian belief propagation," *Adv. Neural Inf. Process. Syst.*, vol. 18, 2006.
[29] J. Bibby, "Axiomatisations of the average and a further generalization of monotonic sequences," *Glasgow Math. J.*, vol. 15, pp. 63–65, 1974.
[30] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem and some of its applications," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 62–75, Mar. 2005.
[31] C. D. Meyer, "Matrix analysis and applied linear algebra book and solutions manual," *SIAM: Soc. Indust. Appl. Math.*, Feb. 2001.
[32] E. Seneta, *Nonnegative Matrices and Markov Chains*, 2nd ed. New York: Springer, 1981.
[33] P. Sommer and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks," in *Proc. Int. Conf. Inf. Process. Sens. Netw. (IPSN)*, San Francisco, CA, Apr. 2009, pp. 37–48.
[34] R. Carli and S. Zampieri, "Networked clock synchronization based on second order linear consensus algorithms," in *Proc. 49th IEEE Conf. Decision and Contr.*, Atlanta, GA, Dec. 2010, pp. 7259–7264.
[35] W.-P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Bayesian detection in bounded height tree networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4042–4051, Oct. 2009.
[36] W.-P. Tay, J. N. Tsitsiklis, and M. Z. Win, "On the impact of node failures and unreliable communications in dense sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2535–2546, Jun. 2008.



Mei Leng received the B.Eng. degree from the University of Electronic Science and Technology of China (UESTC) in 2005 and the Ph.D. degree from The University of Hong Kong in 2011.

She is currently a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University. Her current research interests include statistical signal processing, optimization, machine learning, as well as Bayesian analysis, with applications to wireless sensor networks and wireless communication systems.

Yik-Chung Wu received the B.Eng. (EEE) degree in 1998 and the M.Phil. degree in 2001 from The University of Hong Kong (HKU). He received the Ph.D. degree in 2005 from Texas A&M University, College Station.

From August 2005 to August 2006, he was with the Thomson Corporate Research, Princeton, NJ, as a Member of Technical Staff. Since September 2006, he has been with the University of Hong Kong as an Assistant Professor. His research interests are in the general area of signal processing and communication systems, and in particular, receiver algorithm design, synchronization techniques, channel estimation and equalization.

Dr. Wu is currently serving as an Associate Editor for the IEEE COMMUNICATIONS LETTERS.