The HKU Scholars Hub    The University of Hong Kong    香港大學學術庫

| Title | Distributed Clock Skew and Offset Estimation in Wireless Sensor Networks: Asynchronous Algorithm and Convergence Analysis |
|---|---|
| Author(s) | Du, J; Wu, YC |
| Citation | IEEE Transactions on Wireless Communications, 2013, v. 12 n. 11, p. 5908-5917 |
| Issued Date | 2013 |
| URL | http://hdl.handle.net/10722/199095 |
| Rights | IEEE Transactions on Wireless Communications. Copyright © IEEE. |

# Distributed Clock Skew and Offset Estimation in Wireless Sensor Networks: Asynchronous Algorithm and Convergence Analysis

Jian Du and Yik-Chung Wu

*Abstract*—In this paper, we propose a fully distributed algorithm for joint clock skew and offset estimation in wireless sensor networks based on belief propagation. In the proposed algorithm, each node can estimate its clock skew and offset in a completely distributed and asynchronous way: some nodes may update their estimates more frequently than others using outdated message from neighboring nodes. In addition, the proposed algorithm is robust to random packet loss. Such algorithm does not require any centralized information processing or coordination, and is scalable with network size. The proposed algorithm represents a unified framework that encompasses both classes of synchronous and asynchronous algorithms for network-wide clock synchronization. It is shown analytically that the proposed asynchronous algorithm converges to the optimal estimates with estimation mean-square-error at each node approaching the centralized Cramér-Rao bound under any network topology. Simulation results further show that the convergence speed is faster than that corresponding to a synchronous algorithm.

*Index Terms*—Clock synchronization, wireless sensor network, factor graph, asynchronous algorithm.

## I. INTRODUCTION

**W**IRELESS sensor networks (WSNs) have been widely used in environmental and emergency monitoring [1], [2], event detection [3] and object tracking [4]. To perform distributed information processing in WSNs, a common clock across the network is usually required to guarantee the nodes act in a collaborative and synchronized fashion. Unfortunately, clock oscillator in each sensor node has its own imperfection and both clock skew (frequency difference) and clock offset (phase difference) are present. Therefore, time synchronization [5] appears as one of the most important research challenges in the design of WSNs.

Existing time synchronization algorithms can be categorized into two main classes. One is pairwise synchronization [6]–[17] where protocols are primarily designed to synchronize two nodes. The other is network-wide synchronization where protocols are designed to synchronize a large number of nodes in the network [18]–[30]. Network-wide clock synchronization is much more challenging due to limited radio

range. Nodes in a sensor network cannot directly communicate with every other node, but they have to do it via multi-hop. Traditionally, network-wide clock synchronization in WSNs relies on spanning tree or clustered-based structure. Under such structures, synchronization is achieved through layer-by-layer pairwise synchronization. Such protocols, like time-synchronization protocol for sensor network (TPSN) [18] and pairwise broadcast synchronization (PBS) [19], suffer large overhead in building and maintaining the tree or cluster structure, and are vulnerable to sudden node failures.

Without global structure or special nodes, by exchanging pulses emitted by oscillators, sensors are synchronized to transmit and receive at the same time in [20]–[22]. However, these algorithms cannot provide a precise clock reading at the sensor node. On the other hand, fully distributed synchronization based on averaged consensus algorithms have been proposed in [23]–[28]. Unfortunately, as shown in [26], [29], consensus protocol is not optimal and the performance will deteriorate when message delay exists. Besides, as average-consensus based algorithm seeks to reach global average in the whole network, it has slow convergence [27] (in order of hundreds of iterations before convergence). More recently, [29] pioneered the fully distributed network-wide clock offset estimation algorithm based on belief propagation (BP), and found that its performance is superior to consensus algorithms. However, ignoring the effect of clock skew would significantly increase the re-synchronization frequency. Moreover, [29] considers a parallel implementation with message exchange carried out in a synchronous fashion. Notwithstanding, in many practical scenarios, the inter-sensor message exchange is asynchronous since random data packet losses may occur, and different nodes may update at different frequencies. At present, it is not clear the impact of these disturbance factors on the performance of synchronization algorithms.

This work advances the state-of-the-art distributed synchronization in the following ways: 1) The distributed algorithm is fairly general and can cope with both clock skews as well as offsets over the whole network in parallel. 2) It represents a unified framework that encompasses both classes of synchronous [29], [30] and asynchronous algorithms. 3) The convergence of the proposed method under asynchronous environments is formally proved. The convergence result is derived for vector variable case, in which the Perron-Frobenious theorem used in [29] is not applicable. 4) With the adoption of a different message passing rule from [29], the mean-

square error (MSE) performance of the derived algorithm is shown to approach the centralized Cramér-Rao bound (CRB) asymptotically. Simulations show that the convergence speed of asynchronous algorithm is faster than its synchronous counterpart.

The rest of this paper is organized as follows. The system model is presented in Section II. A fully distributed asynchronous clock skew and offset estimation algorithm based on BP is derived in Section III. The convergence of the proposed asynchronous algorithm is analyzed in Section IV. Simulation results are given in Section V and, finally, conclusions are drawn in Section VI.

*Notations*: Boldface uppercase and lowercase letters are used for matrices and vectors, respectively. Superscript $T$ denotes transpose. The symbol $\boldsymbol{I}_N$ represents the $N \times N$ identity matrix. Notation $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{R})$ stands for the probability density function (pdf) of a Gaussian random vector $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{R}$. The symbol $\propto$ represents the linear scalar relationship between two real valued functions and $|\mathcal{V}|$ denotes the cardinality of set $\mathcal{V}$. For two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\boldsymbol{X} \succ \boldsymbol{Y}$ means that $\boldsymbol{X} - \boldsymbol{Y}$ is a positive definite matrix, and $\boldsymbol{X} \succeq \boldsymbol{Y}$ means that $\boldsymbol{X} - \boldsymbol{Y}$ is a positive semi-definite matrix.

## II. System Model

Consider a general multi-hop sensor network with $M$ sensor nodes distributed in a field as shown in Fig. 1. Let $\mathcal{V} = \{1, \ldots, M\}$ denotes the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge is denoted by $\{i, j\}$ if node $i$ and node $j$ can communicate directly. In the example shown in Fig. 1, the vertices are depicted by circles and the edges by lines connecting these circles. The set of neighbors of node $i$ is denoted by $\mathcal{I}(i) \subseteq \mathcal{V}$ with the definition that $\mathcal{I}(i) \triangleq \{j \in \mathcal{V} | \{i, j\} \in \mathcal{E}\}$. It is assumed that the radio coverage area of a node is circular with a specific radius so that each pair of nodes can exchange message only when their distance is less than both of their communication radiuses. Furthermore, it is assumed that any two distinct nodes can communicate with each other through a finite number of hops. Such a network will be referred to as a strongly connected network.

With the imperfection of oscillators and possible environmental changes, each node has a local clock with possibly different clock skew and offset. The relationship between real time $t$ and the local clock reading is modeled as

$$c_i(t) = \alpha_i t + \theta_i, \qquad i = 1, \cdots, M, \qquad (1)$$

where $\alpha_i$ and $\theta_i$ are the clock skew and offset of node $i$, respectively.

To estimate and compensate such clock skews and offsets, a two-way time-stamp message exchange mechanism was proposed for pairwise clock synchronization [19]. Specifically, as shown in Fig. 2, between one-hop neighboring nodes $i$ and $j$, at the $n^{th}$ round of time-stamp exchange, node $i$ sends a synchronization message to node $j$ at $t_n^1$ with its local clock reading $c_i(t_n^1)$ embedded in the message. Node $j$ records its time $c_j(t_n^2)$ at the reception of that message and replies to node $i$ at $c_j(t_n^3)$. The replied message contains both
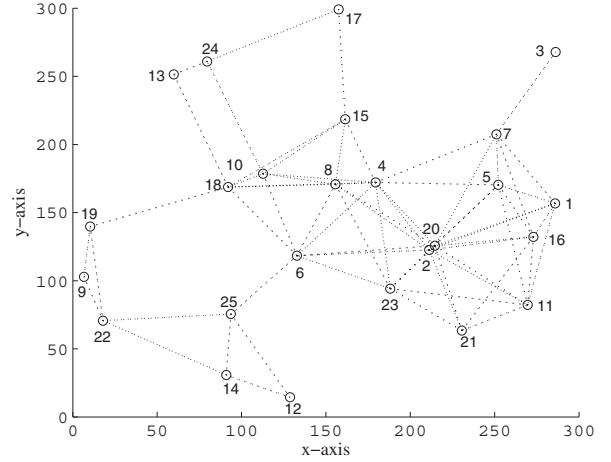
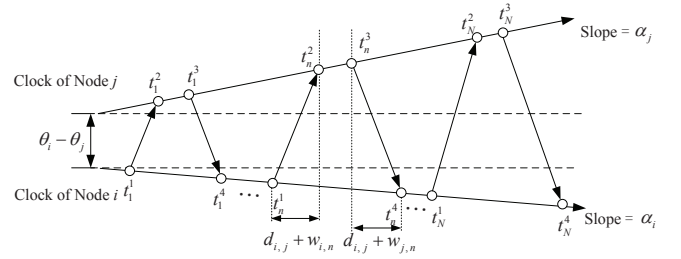

Fig. 1.　WSN topology with 25 nodes randomly distributed.



Fig. 2.　Two way message exchange between node $i$ and $j$ in the WSN.

time stamps $c_j(t_n^2)$ and $c_j(t_n^3)$. Then, node $i$ records the reception time from node $j$'s reply as $c_i(t_n^4)$. $N$ rounds of such message exchange are performed between each pair of nodes to establish a relationship between the nodes $i$'s and $j$'s clocks. In particular, for the $n^{th}$ round time-stamp exchange, we can write

$$\frac{1}{\alpha_j}[c_j(t_n^2) - \theta_j] = \frac{1}{\alpha_i}[c_i(t_n^1) - \theta_i] + d_{i,j} + w_{j,n}, \quad (2)$$

and

$$\frac{1}{\alpha_j}[c_j(t_n^3) - \theta_j] = \frac{1}{\alpha_i}[c_i(t_n^4) - \theta_i] - d_{j,i} - w_{i,n}, \quad (3)$$

where $w_{j,n}$ and $w_{i,n}$ denote independent and identically distributed (i.i.d.) Gaussian random delay during the $n^{th}$ round of time-stamp exchange, with zero mean and variances $\sigma_j^2$, $\sigma_i^2$, respectively; $d_{i,j}$ and $d_{j,i}$ represent the fixed message delay during which node $i/j$ sends message to node $j/i$, respectively. Under the assumption that the network topology does not change during the clock synchronization process, we have $d_{i,j} = d_{j,i}$. Adding (2) and (3) and stacking all resultant equations for $N$ rounds of time-stamp exchange, we obtain

$$\boldsymbol{A}_{j,i}\boldsymbol{\beta}_j + \boldsymbol{A}_{i,j}\boldsymbol{\beta}_i = \boldsymbol{z}_{j,i}, \qquad (4)$$

where $\boldsymbol{A}_{j,i}$ and $\boldsymbol{A}_{i,j}$ are $N$-by-2 matrices with the $n^{th}$ row being $[c_j(t_n^2) + c_j(t_n^3), -2]$ and $-[c_i(t_n^1) + c_i(t_n^4), -2]$, respectively; $\boldsymbol{\beta}_j \triangleq [\frac{1}{\alpha_j}, \frac{\theta_j}{\alpha_j}]^T$ and $\boldsymbol{\beta}_i \triangleq [\frac{1}{\alpha_i}, \frac{\theta_i}{\alpha_i}]^T$; and $\boldsymbol{z}_{j,i}$ is an $N$ dimensional vector with the $n^{th}$ element being $w_{j,n} - w_{i,n}$. Since $w_{j,n}$ and $w_{i,n}$ are both i.i.d. Gaussian, it is easy to obtain

$z_{j,i} \sim \mathcal{N}(z_{j,i}|\mathbf{0}, \sigma_{i,j}^2 \mathbf{I}_N)$, where $\sigma_{i,j}^2 = \sigma_i^2 + \sigma_j^2$. The goal is to establish global synchronization (i.e., estimate $\alpha_i$ and $\theta_i$ in each node) based on the local observations $\mathbf{A}_{j,i}$ and $\mathbf{A}_{i,j}$.

## III. ASYNCHRONOUS DISTRIBUTED ESTIMATION

In this section, the asynchronous distributed clock parameter estimation algorithm is derived based on BP. In the following, message exchange means BP message passing since two-way time-stamp exchange has been completed.

### A. BP Framework

For the reason that the established clock relationships during two-way time-stamp exchanges involve interaction between neighboring nodes, the optimal clock estimate at each node requires the marginalization of joint posterior distribution of all $\boldsymbol{\beta}_i$, which is

$$g_i(\boldsymbol{\beta}_i) \propto \int \cdots \int \prod_{i=1}^{M} p(\boldsymbol{\beta}_i) \prod_{\{i,j\} \in \mathcal{E}} p(\mathbf{A}_{i,j}, \mathbf{A}_{j,i}|\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) \tag{5}$$
$$d\boldsymbol{\beta}_1 \ldots d\boldsymbol{\beta}_{i-1} d\boldsymbol{\beta}_{i+1} d\boldsymbol{\beta}_M,$$

where $p(\boldsymbol{\beta}_i)$ is the prior distribution of $\boldsymbol{\beta}_i$; $p(\mathbf{A}_{i,j}, \mathbf{A}_{j,i}|\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = \mathcal{N}(\mathbf{A}_{j,i}\boldsymbol{\beta}_j|\mathbf{A}_{i,j}\boldsymbol{\beta}_i, \sigma_{i,j}^2 \mathbf{I}_N)$ is the likelihood function obtained from (4). Node 1 is assumed to be the reference node with $p(\boldsymbol{\beta}_1) = \delta(\boldsymbol{\beta}_1 - [1,0]^T)$, and its parameters need not to be estimated. The computation of $g_i(\boldsymbol{\beta}_i)$ in (5) needs to gather all information in a central processing unit. Besides, for the arbitrary network topology, the corresponding $|\mathcal{V}|$ and $|\mathcal{E}|$ can be very large leading to the computationally demanding integration (5).

Although the joint posterior distribution of $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M$ (integrand in (5)) is complicated due to the local interactions of sensor nodes, it is a product of local likelihood functions, each of which depends on a subset of the variables. Such a nice property can be conveniently revealed in a factor graph [31], over which the computation of $g_i(\boldsymbol{\beta}_i)$ for all $i$ can be efficiently accomplished in a distributed way. One example of factor graph is shown in Fig. 3. In this factor graph, local synchronization parameters $\boldsymbol{\beta}_i$, $i = 1, \cdots, M$, are represented by variables nodes (circles). If two sensor nodes $i$ and $j$ are within the communication range of each other, the corresponding variables $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$ are linked by a factor node (local function) $f_{i,j} = f_{i,j} \triangleq p(\mathbf{A}_{i,j}, \mathbf{A}_{j,i}|\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)$. On the other hand, the factor node $f_i \triangleq p(\boldsymbol{\beta}_i)$ denotes the prior information.

The message passing algorithm operated on the factor graph involves two kinds of messages: One is the message from factor node $f_{j,i}$ to a variable node $\boldsymbol{\beta}_i$, defined as [31]

$$m_{f_{j,i} \to i}^{(l)}(\boldsymbol{\beta}_i) = \int m_{j \to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j) f_{j,i} d\boldsymbol{\beta}_j, \tag{6}$$

where $l$ denotes the time of message exchange and $m_{j \to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j)$ is the other kind of message from the variable node to the factor node, which is simply the product of the incoming messages on the other links, i.e.,

$$m_{j \to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j) = \prod_{f \in \mathcal{B}(\boldsymbol{\beta}_j) \backslash f_{j,i}} m_{f \to j}^{(l-1)}(\boldsymbol{\beta}_j), \tag{7}$$
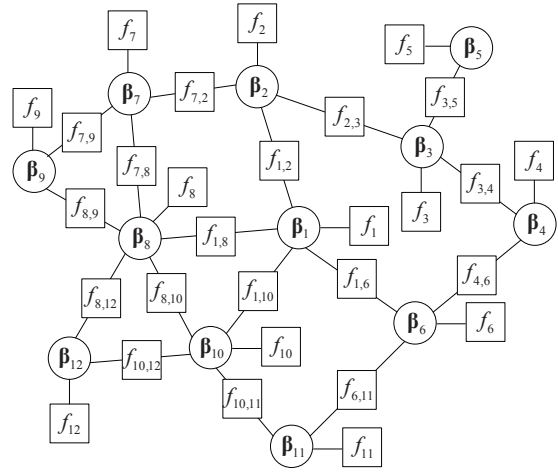


Fig. 3.   The factor graph for clock synchronization in a WSNs.

where $\mathcal{B}(\boldsymbol{\beta}_j)$ denotes the set of neighboring factors of $\boldsymbol{\beta}_j$ on the factor graph. In particular, under such message computation rule, the message from factor node $f_i$ to $\boldsymbol{\beta}_i$ is always equals to the prior distribution $p(\boldsymbol{\beta}_i)$ [31].

During the first round of message passing, it is reasonable to set initial messages from factor node to variable node $m_{f_i \to i}^{(0)}(\boldsymbol{\beta}_i)$ and $m_{f_{j,i} \to i}^{(0)}(\boldsymbol{\beta}_i)$ as $p(\boldsymbol{\beta}_i)$ and non-informative message $\mathcal{N}(\boldsymbol{\beta}_i|\mathbf{0}, +\infty \mathbf{I}_2)$, respectively. Assuming $p(\boldsymbol{\beta}_i) = m_{f_{j,i} \to i}^{(1)}(\boldsymbol{\beta}_i)$ is in Gaussian form (if there is no prior information, we can set the mean to be zero and set the variance to be a large value, i.e., non-informative prior). Then, $m_{j \to f_{j,i}}^{(1)}(\boldsymbol{\beta}_j)$ being the product of Gaussian functions in (7) is also a Gaussian function [37]. Furthermore, based on the fact that the likelihood function $f_{j,i}$ is also Gaussian, according to (6), $m_{f_{j,i} \to i}^{(1)}(\boldsymbol{\beta}_i)$ is a Gaussian function. Thus during each round of message exchange, all the messages are Gaussian functions and only the mean vectors and covariance matrices need to be exchanged between neighboring factor nodes and variable nodes.

In general, for the $l^{th}$ ($l = 2, 3, \cdots$) round of message exchange, factor node $f_{j,i}$ receives message $m_{j \to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j)$ in the form of $\mathcal{N}(\boldsymbol{\beta}_j|\mathbf{v}_{j \to f_{j,i}}^{(l)}, \mathbf{C}_{j \to f_{j,i}}^{(l)})$ from their neighboring variable nodes and then computes a message using (6):

$$m_{f_{j,i} \to i}^{(l)}(\boldsymbol{\beta}_i) = \int m_{j \to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j) f_{j,i} d\boldsymbol{\beta}_j$$
$$= \int \mathcal{N}(\boldsymbol{\beta}_j|\mathbf{v}_{j \to f_{j,i}}^{(l)}, \mathbf{C}_{j \to f_{j,i}}^{(l)}) \tag{8}$$
$$\times \mathcal{N}(\mathbf{A}_{i,j}\boldsymbol{\beta}_i|\mathbf{A}_{j,i}\boldsymbol{\beta}_j, \sigma_{i,j}^2 \mathbf{I}_N) d\boldsymbol{\beta}_j$$

As the convolution of a pair of Gaussian function is also Gaussian function [37], after some algebraic manipulations, we obtain $m_{f_{j,i} \to i}^{(l)}(\boldsymbol{\beta}_i) \propto \mathcal{N}(\boldsymbol{\beta}_i|\mathbf{v}_{f_{j,i} \to i}^{(l)}, \mathbf{C}_{f_{j,i} \to i}^{(l)})$, where the covariance matrix and mean vector are given by

$$\left[\mathbf{C}_{f_{j,i} \to i}^{(l)}\right]^{-1} = \mathbf{A}_{i,j}^T \left[\sigma_{i,j}^2 \mathbf{I}_N + \mathbf{A}_{j,i}\mathbf{C}_{j \to f_{j,i}}^{(l)} \mathbf{A}_{j,i}^T\right]^{-1} \mathbf{A}_{i,j}, \tag{9}$$

and

$$\boldsymbol{v}_{f_{j,i}\to i}^{(l)} = \boldsymbol{C}_{f_{j,i}\to i}^{(l)} \boldsymbol{A}_{i,j}^T \boldsymbol{A}_{j,i} \left\{ \boldsymbol{A}_{j,i}^T \boldsymbol{A}_{j,i} + \sigma_{i,j}^2 \left[ \boldsymbol{C}_{j\to f_{j,i}}^{(l)} \right]^{-1} \right\}^{-1}$$
$$\times \left[ \boldsymbol{C}_{j\to f_{j,i}}^{(l)} \right]^{-1} \boldsymbol{v}_{j\to f_{j,i}}^{(l)}. \tag{10}$$

On the other hand, using (7), the message passed from the variable node to the factor node is given by the product of Gaussian distributions, which is

$$m_{j\to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j) = \prod_{f\in\mathcal{B}(\boldsymbol{\beta}_j)\backslash f_{j,i}} m_{f\to j}^{(l-1)}(\boldsymbol{\beta}_j)$$
$$\propto \mathcal{N}(\boldsymbol{\beta}_j | \boldsymbol{v}_{j\to f_{j,i}}^{(l)}, \boldsymbol{C}_{j\to f_{j,i}}^{(l)}), \tag{11}$$

where

$$\left[ \boldsymbol{C}_{j\to f_{j,i}}^{(l)} \right]^{-1} = \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_j)\backslash f_{j,i}} \left[ \boldsymbol{C}_{f\to j}^{(l-1)} \right]^{-1} \tag{12}$$

and

$$\boldsymbol{v}_{j\to f_{j,i}}^{(l)} = \boldsymbol{C}_{j\to f_{j,i}}^{(l)} \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_j)\backslash f_{j,i}} \left[ \boldsymbol{C}_{f\to j}^{(l-1)} \right]^{-1} \boldsymbol{v}_{f\to j}^{(l-1)}. \tag{13}$$

Furthermore, during each round of message passing, each node can compute the belief for $\boldsymbol{\beta}_i$ as the product of all the incoming messages from neighboring factor nodes, which is given by

$$b^{(l)}(\boldsymbol{\beta}_i) = \prod_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} m_{f\to i}^{(l-1)}(\boldsymbol{\beta}_i). \tag{14}$$

According to (9), (10) and (14), we can easily obtain (15) at the bottom of this page. Finally, the estimate of $\boldsymbol{\beta}_i$ in the $l^{th}$ iteration is (16).

### B. Asynchronous Message Update

In practical WSNs, there is neither factor nodes nor variable nodes. These two kinds of messages $m_{j\to f_{j,i}}^{(l)}(\boldsymbol{\beta}_j)$ and $m_{f_{j,i}\to i}^{(l)}(\boldsymbol{\beta}_i)$ are computed locally at node $j$, and only $m_{f_{j,i}\to i}^{(l)}(\boldsymbol{\beta}_i)$ is sent from node $j$ to node $i$ during each round of

message exchange of BP. Let $m_{j\to i}^{(l)}(\boldsymbol{\beta}_i) = \mathcal{N}(\boldsymbol{\beta}_i | \boldsymbol{\gamma}_{j\to i}^{(l)}, \boldsymbol{\Gamma}_{j\to i}^{(l)})$ represent the physical message from node $j$ to node $i$. Putting (12) and (13) into (9) and (10), we have (17) and (18), where $\boldsymbol{\Gamma}_j$ and $\boldsymbol{\gamma}_j$ are the covariance matrix and mean vector of prior distribution of $\boldsymbol{\beta}_j$, respectively, and they will never change during the updating process.

As shown in (17) and (18), from the perspective of node $j$, the outgoing message covariance $\boldsymbol{\Gamma}_{j\to i}^{(l)}$ and mean vector $\boldsymbol{\gamma}_{j\to i}^{(l)}$ computed by node $j$ at time $l$ depends on the incoming message covariance $\boldsymbol{\Gamma}_{k\to j}^{(l-1)}$ and $\boldsymbol{\gamma}_{k\to j}^{(l-1)}$ from node $j$'s neighbour (i.e., $k\in\mathcal{I}(j)\backslash i$) at time $l-1$. However, in many situations, the inter-sensor message exchange is possibly asynchronous due to random data packet dropouts, and different nodes may update their messages at different frequencies. If every node is allowed to update its belief only after receiving updated messages from all its neighbors, the convergence speed of the distributed algorithm would be slow. Thus, some nodes should be allowed to update their beliefs more frequently than others, as long as they receive some of the updates from their neighboring nodes within a predetermined time period. It means that when node $j$ computes $\boldsymbol{\Gamma}_{j\to i}^{(l)}$, it may only have $\boldsymbol{\Gamma}_{k\to j}^{(s)}$ computed by node $k\in\mathcal{I}(j)\backslash i$ with $s\le l-1$. In order to capture these asynchronous properties of message exchanges, we introduce the totally asynchronous model [32] as follows.

Let the message covariance matrices and mean vectors available to node $j$ at time $l$ are $\boldsymbol{\Gamma}_{k\to j}^{(\tau_j^k(l-1))}$ and $\boldsymbol{\gamma}_{k\to j}^{(\tau_j^k(l-1))}$, where $0\le\tau_j^k(l-1)\le l-1$. Without loss of generality, we assume that node $j$ computes its outgoing messages to its neighboring nodes according to a discrete time set $\mathcal{L}_j\subseteq\{0,1,2,\ldots\}$. According to (17) and (18), the asynchronous message covariance and mean evolution are defined as (19) and (20) at the bottom of the next page. We assume $\lim_{l\to\infty}\tau_j^k(l) = \infty$ for all $\{k,j\}\in\mathcal{E}$, which guarantees that old information is eventually purged out of the network, and that each node eventually exchanges messages with its neighboring nodes.

The asynchronous iterative algorithm is summarized as

$$b^{(l)}(\boldsymbol{\beta}_i) \sim \mathcal{N}\Big(\boldsymbol{\beta}_i | \Big[ \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} \big[ \boldsymbol{C}_{f\to i}^{(l-1)} \big]^{-1} \Big]^{-1} \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} \big[ \boldsymbol{C}_{f\to i}^{(l-1)} \big]^{-1} \boldsymbol{v}_{f\to i}^{(l-1)}, \Big[ \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} \big[ \boldsymbol{C}_{f\to i}^{(l-1)} \big]^{-1} \Big]^{-1} \Big). \tag{15}$$

$$\hat{\boldsymbol{\beta}}_i^{(l)} = \int \boldsymbol{\beta}_i b^{(l)}(\boldsymbol{\beta}_i) d\boldsymbol{\beta}_i = \Big[ \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} \big[ \boldsymbol{C}_{f\to i}^{(l-1)} \big]^{-1} \Big]^{-1} \sum_{f\in\mathcal{B}(\boldsymbol{\beta}_i)} \big[ \boldsymbol{C}_{f\to i}^{(l-1)} \big]^{-1} \boldsymbol{v}_{f\to i}^{(l-1)}. \tag{16}$$

$$\big[ \boldsymbol{\Gamma}_{j\to i}^{(l)} \big]^{-1} = \boldsymbol{A}_{i,j}^T \Big[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \Big[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k\in\mathcal{I}(j)\backslash i} \big[ \boldsymbol{\Gamma}_{k\to j}^{(l-1)} \big]^{-1} \Big]^{-1} \boldsymbol{A}_{j,i}^T \Big]^{-1} \boldsymbol{A}_{i,j}. \tag{17}$$

$$\boldsymbol{\gamma}_{j\to i}^{(l)} = \boldsymbol{\Gamma}_{j\to i}^{(l)} \boldsymbol{A}_{i,j}^T \Big[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \Big[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k\in\mathcal{I}(j)\backslash i} \big[ \boldsymbol{\Gamma}_{k\to j}^{(l-1)} \big]^{-1} \Big]^{-1} \boldsymbol{A}_{j,i}^T \Big]^{-1}$$
$$\times \boldsymbol{A}_{j,i} \Big[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k\in\mathcal{I}(j)\backslash i} \big[ \boldsymbol{\Gamma}_{k\to j}^{(l-1)} \big]^{-1} \Big]^{-1} \Big[ \boldsymbol{\Gamma}_j^{-1} \boldsymbol{\gamma}_j + \sum_{k\in\mathcal{I}(j)\backslash i} \big[ \boldsymbol{\Gamma}_{k\to j}^{(l-1)} \big]^{-1} \boldsymbol{\gamma}_{k\to j}^{(l-1)} \Big], \tag{18}$$

follows. The algorithm is started by setting the messages from node $j$ to node $i$ as $m_{j \to i}^{(0)}(\boldsymbol{\beta}_i) = \mathcal{N}(\boldsymbol{\beta}_i; \mathbf{0}, +\infty \boldsymbol{I}_2)$ [1]. Each node $i$ computes its outgoing message according to (19) and (20) at independent time $l \in \mathcal{L}_i$ with its available $[\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l-1))}]^{-1}$ and $\boldsymbol{\gamma}_{j \to i}^{(\tau_i^j(l-1))}$. The corresponding belief of node $i$ at time $l$ is computed as

$$b^{(l)}(\boldsymbol{\beta}_i) \sim \mathcal{N}(\boldsymbol{\beta}_i | \boldsymbol{\mu}_i^{(l)}, \boldsymbol{P}_i^{(l)}), \qquad (21)$$

where the belief covariance matrix is

$$\boldsymbol{P}_i^{(l)} = \left[ \boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} \left[ \boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l-1))} \right]^{-1} \right]^{-1}, \qquad (22)$$

and mean vector is

$$\boldsymbol{\mu}_i^{(l)} = \boldsymbol{P}_i^{(l)} \left[ \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\gamma}_i + \sum_{j \in \mathcal{I}(i)} \left[ \boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l-1))} \right]^{-1} \boldsymbol{\gamma}_{j \to i}^{(\tau_i^j(l-1))} \right]. \qquad (23)$$

The iterative computation terminates when (21) converges or the maximum number of iterations is reached. Then each sensor computes its clock skew and offset according to

$$\hat{\alpha}_i = 1/\boldsymbol{\mu}_i^{(l)}(1), \quad \hat{\theta}_i = \boldsymbol{\mu}_i^{(l)}(2)/\boldsymbol{\mu}_i^{(l)}(1), \qquad (24)$$

where $\boldsymbol{\mu}_i^{(l)}(k)$ denotes the $k^{th}$ element of $\boldsymbol{\mu}_i^{(l)}$.

## IV. ASYNCHRONOUS BP CONVERGENCE ANALYSIS

It is important to note that the BP message updates (8) and (11) are specially designed for the computation of marginal functions (e.g., $g_i(\boldsymbol{\beta}_i)$ in (5)) on cycle-free FG and it is known that the beliefs will converge to the exact marginal functions. On the other hand, the BP algorithm may be applied to FG with cycles, but since messages will be passed multiple times on a given edge, no convergence can be guaranteed [34]. Although some of the most exciting applications of BP algorithm like the decoding of turbo codes and low-density parity-check codes [31] do not exhibit divergence in the simulations even under loopy FG, there are still many applications where BP do diverge. General sufficient condition for convergence of loopy FGs is available in [35] but it requires the knowledge of the joint posterior distribution of

---

[1]Since the message updating using (19) and (20) only involves inverse of covariance matrix, in practice, we can set the inverse of the initial covariance matrix as 0.

all unknown variables as shown in the integrand of (5), and is difficult to verify for large-scale dynamic networks. Reference [29] proved the convergence of BP in the context of distributed clock offset synchronization, by exploiting the Perron-Frobenius theorem in the context of matrices with nonnegative elements. However, in the vector variable case (both clock skew and offset), the BP message covariance matrices contain negative elements, and the analysis in [29] is not applicable. Besides, the effect of asynchronous message-update was not addressed in [29]. In the following, we will prove the convergence of asynchronous vector BP messages in distributed clock synchronization.

Defining the operator $\mathbb{F}_{j \to i}(\cdot)$ corresponding to the update of the message covariance in (19), the following properties are first established.

**Lemma 1.** The updating operator $\mathbb{F}_{j \to i}(\cdot)$ satisfies the following properties:

Property i): $\mathbb{F}_{j \to i}(\mathbf{0}) = \mathbf{0}$.

Property ii): $\mathbb{F}_{j \to i}(\boldsymbol{X}) \succ \mathbf{0}$, if $\boldsymbol{X} \succ \mathbf{0}$.

Property iii): $\mathbb{F}_{j \to i}(\boldsymbol{X}) \succeq \mathbb{F}_{j \to i}(\boldsymbol{Y})$, if $\boldsymbol{X} \succeq \boldsymbol{Y} \succ \mathbf{0}$.

*Proof*: Property i) is apparent according to (19). The proof of property ii) is given as follows. Let $\boldsymbol{X} \succ 0$, it is obvious that $\boldsymbol{X}^{-1} \succ \mathbf{0}$, which means $\boldsymbol{y}^T \boldsymbol{X}^{-1} \boldsymbol{y} \geq 0$ for any $\boldsymbol{y}$. Putting $\boldsymbol{y} = \boldsymbol{A}_{j,i} \boldsymbol{x}$, we have $\boldsymbol{x}^T \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \boldsymbol{x} \geq \mathbf{0}$. As sum of positive definite and positive semi-definite matrices is positive definite, we have $\left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \succ \mathbf{0}$. Since $\boldsymbol{A}_{i,j}$ is of full column rank, we obtain $\boldsymbol{A}_{i,j}^T \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \boldsymbol{A}_{i,j} \succ \mathbf{0}$. Thus, property ii) is proved. For the proof of property iii), let $\boldsymbol{X} \succeq \boldsymbol{Y} \succ \mathbf{0}$, then we have $\boldsymbol{Y}^{-1} - \boldsymbol{X}^{-1} \succ \mathbf{0}$ [39], which means $\boldsymbol{y}^T (\boldsymbol{Y}^{-1} - \boldsymbol{X}^{-1}) \boldsymbol{y} \geq 0$ for any $\boldsymbol{y}$. Let $\boldsymbol{y} = \boldsymbol{A}_{j,i}^T \boldsymbol{x}$, we have $\boldsymbol{x}^T \boldsymbol{A}_{j,i} \boldsymbol{Y}^{-1} \boldsymbol{A}_{j,i}^T \boldsymbol{x} \geq \boldsymbol{x}^T \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \boldsymbol{x}$. Hence, we have $\left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \succeq \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{Y}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1}$. Due to the fact that $\boldsymbol{A}_{i,j}$ is of full column rank, we have $\boldsymbol{A}_{i,j}^T \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{X}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \boldsymbol{A}_{i,j} \succeq \boldsymbol{A}_{i,j}^T \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \boldsymbol{Y}^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \boldsymbol{A}_{i,j}$, which is equivalent to $\mathbb{F}_{j \to i}(\boldsymbol{X}) \succeq \mathbb{F}_{j \to i}(\boldsymbol{Y})$. ∎

To consider the updates of all message covariance matrices, we introduce the following definitions. Let $\boldsymbol{\Xi}^{(\tau(l-1))} \triangleq \left[ [\boldsymbol{\Gamma}_{1 \to k}^{(\tau_k^1(l-1))}]^{-1}; \ldots; [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l-1))}]^{-1}; \ldots; [\boldsymbol{\Gamma}_{r \to M}^{(\tau_M^r(l-1))}]^{-1}; \boldsymbol{\Gamma}_1^{-1}; \ldots; \boldsymbol{\Gamma}_M^{-1} \right]$ be the collection of all available message covariance

$$\left[ \boldsymbol{\Gamma}_{j \to i}^{(l)} \right]^{-1} = \begin{cases} \underbrace{\boldsymbol{A}_{i,j}^T \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \left[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \backslash i} \left[ \boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))} \right]^{-1} \right]^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \boldsymbol{A}_{i,j}}_{\triangleq \mathbb{F}_{j \to i} \left( \boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \backslash i} \left[ \boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))} \right]^{-1} \right)}, & l \in \mathcal{L}_j, \\[2em] \left[ \boldsymbol{\Gamma}_{j \to i}^{(l-1)} \right]^{-1}, & \text{otherwise.} \end{cases} \qquad (19)$$

$$\boldsymbol{\gamma}_{j \to i}^{(l)} = \begin{cases} \boldsymbol{\Gamma}_{j \to i}^{(l)} \boldsymbol{A}_{i,j}^T \left[ \sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \left[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \backslash i} \left[ \boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))} \right]^{-1} \right]^{-1} \boldsymbol{A}_{j,i}^T \right]^{-1} \boldsymbol{A}_{j,i} \\ \times \left[ \boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \backslash i} \left[ \boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))} \right]^{-1} \right]^{-1} \left[ \boldsymbol{\Gamma}_j^{-1} \boldsymbol{\gamma}_j + \sum_{k \in \mathcal{I}(j) \backslash i} \left[ \boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))} \right]^{-1} \boldsymbol{\gamma}_{k \to j}^{(\tau_j^k(l-1))} \right], & l \in \mathcal{L}_j, \\[1em] \boldsymbol{\gamma}_{j \to i}^{(l-1)}, & \text{otherwise.} \end{cases} \qquad (20)$$

(including prior covariance) matrices in the network at time $l$, and $\boldsymbol{\Xi}^{(l)} \triangleq \left[[\boldsymbol{\Gamma}_{1 \to k}^{(l)}]^{-1}; \ldots; [\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1}; \ldots; [\boldsymbol{\Gamma}_{r \to M}^{(l)}]^{-1}\right]$ be the collection of all outgoing message covariances in the network at time $l$. Define $\boldsymbol{\Xi}^{(l)} \succeq^b \mathbf{0}$ if its component $[\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1} \succeq \mathbf{0}$; and $\boldsymbol{\Xi}^{(l)} \succeq^b \boldsymbol{\Xi}^{(l-1)}$ if their corresponding components satisfy $[\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1} \succeq [\boldsymbol{\Gamma}_{j \to i}^{(l-1)}]^{-1}$. The same definitions apply to $\boldsymbol{\Xi}^{(\tau(l))}$. Furthermore, we define the function $\mathbb{F} \triangleq (\mathbb{F}_{1 \to k}, \ldots, \mathbb{F}_{j \to i}, \ldots, \mathbb{F}_{r \to M})$ which satisfies $\boldsymbol{\Xi}^{(l+1)} = \mathbb{F}(\boldsymbol{\Xi}^{(\tau(l))})$. Then we have the following lemma.

**Lemma 2.** $\boldsymbol{\Xi}^{(l)}$ and $\boldsymbol{\Xi}^{(\tau(l-1))}$ satisfy the following properties:

Property iv): If $\boldsymbol{\Xi}^{(l)} \succeq^b \boldsymbol{\Xi}^{(l-1)}$, then $\boldsymbol{\Xi}^{(\tau(l))} \succeq^b \boldsymbol{\Xi}^{(\tau(l-1))}$.
Property v): If $\boldsymbol{\Xi}^{(\tau(l))} \succeq^b \boldsymbol{\Xi}^{(\tau(l-1))}$, then $\mathbb{F}(\boldsymbol{\Xi}^{(\tau(l))}) \succeq^b \mathbb{F}(\boldsymbol{\Xi}^{(\tau(l-1))})$ or equivalently $\boldsymbol{\Xi}^{(l+1)} \succeq^b \boldsymbol{\Xi}^{(l)}$.

*Proof*: The proofs of properties iv) and v) rest on the basic definitions that $[\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1}$ represents the message covariance matrix sends from node $j$ to node $i$ at time $l$, and $[\boldsymbol{\Gamma}_{j \to i}^{(\tau(l))}]^{-1}$ represents message covariance matrix received by node $i$ at time $l$. If $[\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1} \succeq [\boldsymbol{\Gamma}_{j \to i}^{(l-1)}]^{-1}$, it is obvious that the received covariance will satisfy $[\boldsymbol{\Gamma}_{j \to i}^{(\tau(l))}]^{-1} \succeq [\boldsymbol{\Gamma}_{j \to i}^{(\tau(l-1))}]^{-1}$. Since $\boldsymbol{\Xi}^{(l)}$ and $\boldsymbol{\Xi}^{(\tau(l))}$ contain $[\boldsymbol{\Gamma}_{j \to i}^{(l)}]^{-1}$ and $[\boldsymbol{\Gamma}_{j \to i}^{(\tau(l))}]^{-1}$ as components respectively, property iv) is obvious. On the other hand, property v) is apparent since each of the corresponding components in $\boldsymbol{\Xi}^{(\tau(l))}$ and $\boldsymbol{\Xi}^{(\tau(l-1))}$ satisfies property i) or iii) in Lemma 1. ∎

Now we present the convergence property of the covariance matrix in the local beliefs.

**Theorem 1.** For the totally asynchronous clock synchronization algorithm, the covariance matrix $\boldsymbol{P}_i^{(l)}$ of belief $b_i^{(l)}(\boldsymbol{\beta}_i)$ at each node converges to a positive definite matrix regardless of network topology.

*Proof*: Initially, all messages are non-informative, that is, $\boldsymbol{\Gamma}_{j \to i}^{\tau(-1)} = \boldsymbol{\Gamma}_{j \to i}^{(0)} = \infty \boldsymbol{I}_2$. From (19), properties i) and ii), we obtain that $\left[\boldsymbol{\Gamma}_{j \to i}^{(l)}\right]^{-1} \succ \mathbf{0}$ only if $\boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))}]^{-1} \succ \mathbf{0}$. Therefore, the first batch of nodes having outgoing covariance $\left[\boldsymbol{\Gamma}_{j \to i}^{(l)}\right]^{-1} \succ \mathbf{0}$ must have $\boldsymbol{\Gamma}_j^{-1} \succ \mathbf{0}$, i.e., informative prior. Let the first message updating event in the network occurs at time $s$. We have $\boldsymbol{\Xi}^{(s)} \succeq^b \boldsymbol{\Xi}^{(s-1)}$. Applying property iv), we further obtain $\boldsymbol{\Xi}^{\tau(s)} \succeq^b \boldsymbol{\Xi}^{\tau(s-1)}$.

Suppose $\boldsymbol{\Xi}^{(\tau(l))} \succeq^b \boldsymbol{\Xi}^{(\tau(l-1))}$ for $l \geq s$, according to property v), $\boldsymbol{\Xi}^{(l+1)} \succeq^b \boldsymbol{\Xi}^{(l)}$. Thus $\boldsymbol{\Xi}^{(\tau(l+1))} \succeq^b \boldsymbol{\Xi}^{(\tau(l))}$ for $l \geq s$ due to property iv). Hence, by induction the updating

relationship of $\boldsymbol{\Xi}^{(\tau(l))}$ is

$$\ldots \succeq^b \boldsymbol{\Xi}^{(\tau(l))} \ldots \succeq^b \boldsymbol{\Xi}^{(\tau(s))} \succeq^b \mathbf{0}. \tag{25}$$

Focusing on node $i$, we obtain

$$\ldots \succeq \boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l))}]^{-1} \ldots \succeq \boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(s))}]^{-1}. \tag{26}$$

Since a strongly connected network is considered, there must be one of $[\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l'-1))}]^{-1} \succ \mathbf{0}$ for some $l' \geq s$, and therefore (26) is lower bounded by the all-zero matrix. Furthermore, since $\infty \boldsymbol{I}_2 \succeq \boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))}]^{-1}$, according to property iii), $\mathbb{F}_{j \to i}(\infty \boldsymbol{I}_2) \succeq \mathbb{F}_{j \to i}(\boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l-1))}]^{-1})$. Using the definition of $\mathbb{F}_{j \to i}(\cdot)$ in (19), this is equivalent to $\frac{1}{\sigma_{i,j}^2} \boldsymbol{A}_{i,j}^T \boldsymbol{A}_{i,j} \succeq [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l))}]^{-1}$. Therefore, we can add an upper bound to (26) and obtain (27). Then, applying matrix inverse to (27) and using the definition of $\boldsymbol{P}_i^{(l)}$ in (22) results in

$$\boldsymbol{P}_i^{(l')} \succeq \boldsymbol{P}_i^{(l'+1)} \succeq \ldots \succeq \left[\boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} \frac{1}{\sigma_{i,j}^2} \boldsymbol{A}_{i,j}^T \boldsymbol{A}_{i,j}\right]^{-1} \succ \mathbf{0}, \tag{28}$$

where the inequality relationship is due to the fact that if $\boldsymbol{X}, \boldsymbol{Y} \succ 0$ and $\boldsymbol{X} \succeq \boldsymbol{Y}$, then $\boldsymbol{Y}^{-1} \succeq \boldsymbol{X}^{-1}$ [39]. Consequently, such non-increasing positive definite matrix sequence $\boldsymbol{P}_i^{(l)}$ in (28) converges to a positive definite matrix [40]. ∎

The importance of Theorem 1 is that the covariance matrices of belief always converge regardless of network topology as long as informative prior exists. Next, we show the convergence of belief mean vectors.

**Theorem 2.** For the totally asynchronous clock synchronization algorithm, the mean vector $\boldsymbol{\mu}_i^{(l)}$ of the belief $b^{(l)}(\boldsymbol{\beta}_i)$ converges to a constant vector regardless of the network topology.

*Proof*: From (25) in the proof of Theorem 1, we can readily see that $\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l))}$ satisfies: $\ldots \succeq [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l))}]^{-1} \succeq \ldots \succeq [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(s))}]^{-1} \succeq \mathbf{0}$. If there is a path from any node with informative prior to node $k$, according to property ii), there must be a time instant $l'$ after which $\ldots \succeq [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l'+1))}]^{-1} \succeq \ldots \succeq [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l'))}]^{-1} \succ \mathbf{0}$. Hence $\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l'))}$ is convergent [40]. On the other hand, if there is no path from any node with informative prior to node $k$, we have $\ldots = [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l))}]^{-1} = \ldots = [\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(0))}]^{-1} = \mathbf{0}$. Either case implies $\boldsymbol{\Gamma}_{k \to j}^{(\tau_j^k(l))}$ converges

$$\boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} \frac{1}{\sigma_{i,j}^2} \boldsymbol{A}_{i,j}^T \boldsymbol{A}_{i,j} \succeq \ldots \succeq \boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l'+1))}]^{-1} \succeq \boldsymbol{\Gamma}_i^{-1} + \sum_{j \in \mathcal{I}(i)} [\boldsymbol{\Gamma}_{j \to i}^{(\tau_i^j(l'))}]^{-1} \succ \mathbf{0}. \tag{27}$$

$$\boldsymbol{\gamma}_{j \to i}^{(l)} = \begin{cases} \boldsymbol{\Gamma}_{j \to i}^{(*)} \boldsymbol{A}_{i,j}^T \left[\sigma_{i,j}^2 \boldsymbol{I}_N + \boldsymbol{A}_{j,i} \left[\boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(*)}]^{-1}\right]^{-1} \boldsymbol{A}_{j,i}^T\right]^{-1} \boldsymbol{A}_{j,i} \\ \times \left[\boldsymbol{\Gamma}_j^{-1} + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(*)}]^{-1}\right]^{-1} \left[\boldsymbol{\Gamma}_j^{-1} \boldsymbol{\gamma}_j + \sum_{k \in \mathcal{I}(j) \setminus i} [\boldsymbol{\Gamma}_{k \to j}^{(*)}]^{-1} \boldsymbol{\gamma}_{k \to j}^{(\tau_j^k(l-1))}\right], & l \in \mathcal{L}_j, \\ \boldsymbol{\gamma}_{j \to i}^{(l-1)}, & \text{otherwise.} \end{cases} \tag{29}$$

to a matrix $\boldsymbol{\Gamma}_{k\to j}^{(*)}$. From (19), if $\boldsymbol{\Gamma}_{k\to j}^{(\tau_j^k(l))}$ converges, we have $\boldsymbol{\Gamma}_{j\to i}^{(l)}$ also converges to a fixed matrix $\boldsymbol{\Gamma}_{j\to i}^{(*)}$. Then, (20) can be rewritten as (29). Without loss of generality, define $\boldsymbol{\gamma}^{(l)}$ as a vector containing all $\boldsymbol{\gamma}_j$ and outgoing message mean $\boldsymbol{\gamma}_{j\to i}^{(l)}$ with ascending index first on $j$ and then on $i$ ($\boldsymbol{\gamma}_j$ can be interpreted as $\boldsymbol{\gamma}_{j\to j}$ for the ordering), and $\boldsymbol{\gamma}^{(l-1)}$ is the vector constituted by available message means with the same ordering. It should be noticed that the order of $\boldsymbol{\gamma}_{j\to i}^{(l)}$ arranged in $\boldsymbol{\gamma}^{(l)}$ can be arbitrary as long as it does not change after the order is fixed. Then, (29) can be expressed as

$$\boldsymbol{\gamma}^{(l)} = \boldsymbol{Q}^{(l)}\boldsymbol{\gamma}^{(l-1)}, \tag{30}$$

where the specific structure of $\boldsymbol{Q}^{(l)}$ depends on the messages sent and received at time $l$. Notice that $\boldsymbol{Q}^{(l)}$ is time-varying due to asynchronous updating. The convergence condition for the asynchronous system (30) turns out to be related to the system matrix of the corresponding synchronous system [32, p. 434], [33, p. 14]. Consider $\mathcal{L}_j = \{0, 1, 2, \ldots\}$ for all $j = 1, 2, \ldots, M$, the asynchronous system (30) becomes a synchronous one:

$$\boldsymbol{\gamma}^{(l)} = \boldsymbol{Q}\boldsymbol{\gamma}^{(l-1)}, \tag{31}$$

where $\boldsymbol{Q}$ is now independent of iteration number $l$. The necessary and sufficient convergence condition for the asynchronous iteration (30) is $\rho(|\boldsymbol{Q}|) < 1$ [32, p. 434], where $|\boldsymbol{Q}|$ denotes the matrix whose elements are the absolute values of those in $\boldsymbol{Q}$. Next, we prove that $\rho(|\boldsymbol{Q}|) < 1$.

First, construct the new linear iteration as

$$\boldsymbol{x}^{(r)} = \tilde{\boldsymbol{Q}}\boldsymbol{x}^{(r-1)}, \tag{32}$$

where $\tilde{\boldsymbol{Q}} = |\boldsymbol{Q}|$, $\boldsymbol{x}^{(r)}$ is a vector with the same structure as $\boldsymbol{\gamma}^{(r)}$ and $\boldsymbol{x}^{(0)} = \boldsymbol{\gamma}^{(0)}$. Since there is always a positive value $\eta$, satisfying $\eta > \sum_{i\neq j} |[\tilde{\boldsymbol{Q}}]_{i,j}|$ for all $i$, we have $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$ is strictly diagonally dominant and then $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$ is nonsingular [41]. Hence, the arbitrary initial value $\boldsymbol{x}^{(0)}$ can be expressed in terms of the eigenvectors of $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$ as $\boldsymbol{x}^{(0)} = \sum_{d=1}^{D} c_d\boldsymbol{q}_d$, where $D$ is the dimension of matrix $\tilde{\boldsymbol{Q}}$ and $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_D$ are the eigenvectors of $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$. Since the eigenvectors of $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$ are the same as those of $\tilde{\boldsymbol{Q}}$, and the eigenvalues of $\eta\boldsymbol{I} + \tilde{\boldsymbol{Q}}$ are $\eta + \lambda_d$ ($1 \leqslant d \leqslant D$), where $\lambda_d$ is the eigenvalue of $\tilde{\boldsymbol{Q}}$, we have

$$\boldsymbol{x}^{(r)} = \tilde{\boldsymbol{Q}}^r\boldsymbol{x}^{(0)} = \sum_{d=1}^{D} c_d\lambda_d^r\boldsymbol{q}_d. \tag{33}$$

Without loss of generality, suppose $\lambda_d$ are arranged in descending order as

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_D|. \tag{34}$$

Let the eigenvalue with the largest magnitude has a multiplicity of $d_0$. Then $\lambda_d/\lambda_1 < 1$ for $d > d_0$ and $(\lambda_d/\lambda_1)^r = 0$ if $r$ is large enough. We then obtain

$$\lim_{r\to\infty} \boldsymbol{x}^{(r)} = \lambda_1^r \sum_{d=1}^{d_0} c_d\boldsymbol{q}_d. \tag{35}$$

On the other hand, putting $j = 1$ into (19), and noting $\boldsymbol{\Gamma}_1^{-1} = \infty\boldsymbol{I}_2$, we obtain $[\boldsymbol{\Gamma}_{1\to i}^{(l)}]^{-1} = \frac{1}{\sigma_{i,1}^2}\boldsymbol{A}_{i,1}^T\boldsymbol{A}_{i,1}$, for $l \in \mathcal{L}_i$. But since this outgoing covariance from the reference

node is independent of time $l$, we can combine the two cases in (19). Substituting this result into (20), we have $\boldsymbol{\gamma}_{1\to i}^{(l)} = \frac{1}{\sigma_{i,1}^2}[\boldsymbol{A}_{i,1}^T\boldsymbol{A}_{i,1}]^{-1}\boldsymbol{A}_{i,1}^T\boldsymbol{A}_{1,i}\boldsymbol{\beta}_1$, which shows that $\boldsymbol{\gamma}_{1\to i}^{(l)}$ is also independent of time $l$. Consequently, according to (31), $\boldsymbol{\gamma}_{1\to i}^{(l)} = [\boldsymbol{Q}]_{1:2,1:D}\boldsymbol{\gamma}^{(l-1)}$ and $[\boldsymbol{Q}]_{1:2,1:D} = [\boldsymbol{I}_2, \boldsymbol{0}]$. Hence, $[[\boldsymbol{Q}]_{1:2,1:D}]\boldsymbol{x}^{(0)} = \boldsymbol{x}_{1\to i}^{(0)} = \boldsymbol{x}_{1\to i}^{(1)}$. In general, we also have $\boldsymbol{x}_{1\to i}^{(r)} = \boldsymbol{x}_{1\to i}^{(0)}$ for all $r$. Therefore, we can put $\boldsymbol{x}^{(r)}(m_i) = \boldsymbol{\gamma}_{1\to i}^{(l)} \triangleq \xi_c$ being a constant into (35) to obtain $\lambda_1^r = \frac{\xi_c}{\sum_{d=1}^{d_0} c_d\boldsymbol{q}_d(m_i)}$ for $r$ large enough. Substituting it back into (35) yields

$$\lim_{r\to\infty} \boldsymbol{x}^{(r)} = \frac{\xi_c \sum_{d=1}^{d_0} c_d\boldsymbol{q}_d}{\sum_{d=1}^{d_0} c_d\boldsymbol{q}_d(m_i)}. \tag{36}$$

It is obvious that $\boldsymbol{x}^{(r)}$ does not change when $r$ is large enough, and therefore, $\boldsymbol{x}^{(r)}$ in (32) converges. Hence, the spectrum radius $\rho(\tilde{\boldsymbol{Q}}) = \rho(|\boldsymbol{Q}|) < 1$ [42], and according to [32, p. 434], the asynchronous version of the iteration given by (30) converges. Finally, with $\boldsymbol{\mu}_i^{(l)}$ defined in (23), since $\boldsymbol{P}_i^{(l)}$, $\boldsymbol{\Gamma}_{j\to i}^{(l)}$ and $\boldsymbol{\gamma}_{j\to i}^{(l)}$ converge, we can draw the conclusion that the vector sequence $\{\boldsymbol{\mu}_i^{(1)}, \boldsymbol{\mu}_i^{(2)}, \ldots\}$ converges. ∎

Theorems 1 and 2 reveal that the BP messages converge. Next, we address how good is the clock parameters estimate (24) based on the converged message mean $\boldsymbol{\mu}_i^* = \lim_{l\to+\infty} \boldsymbol{\mu}_i^{(l)}$. Since the prior $p(\boldsymbol{\beta}_i)$ and likelihood function $p(\boldsymbol{A}_{i,j}, \boldsymbol{A}_{j,i}|\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)$ are both Gaussian distribution and it is known that if Gaussian BP (synchronous or asynchronous) converges, the means of the beliefs computed by BP equal the means of the marginal posterior distribution [35], [36], i.e., $\boldsymbol{\mu}_i^* = \hat{\boldsymbol{\beta}}_i^{\text{MMSE}} \triangleq \int \cdots \int \boldsymbol{\beta}_i p(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_M|\{\boldsymbol{A}_{i,j}\}_{\{i,j\}\in\mathcal{E}})d\boldsymbol{\beta}_2\cdots d\boldsymbol{\beta}_M$. Stacking $\hat{\boldsymbol{\beta}}_i^{\text{MMSE}}$ into a block vector $\hat{\boldsymbol{\beta}}^{\text{MMSE}} = [(\hat{\boldsymbol{\beta}}_2^{\text{MMSE}})^T, \ldots, (\hat{\boldsymbol{\beta}}_M^{\text{MMSE}})^T]^T$ gives

$$\hat{\boldsymbol{\beta}}^{\text{MMSE}} = \int \cdots \int [\boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_M^T]^T$$
$$\times p(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_M|\{\boldsymbol{A}_{i,j}\}_{\{i,j\}\in\mathcal{E}})d\boldsymbol{\beta}_2 \ldots d\boldsymbol{\beta}_M. \tag{37}$$

It is obvious that $\boldsymbol{\mu}^* = [(\boldsymbol{\mu}_2^*)^T, \ldots, (\boldsymbol{\mu}_M^*)^T]^T$ equals the centralized joint MMSE estimator $\hat{\boldsymbol{\beta}}^{\text{MMSE}}$. In case of non-informative prior, $\hat{\boldsymbol{\beta}}^{\text{MMSE}}$ is the mean of the joint likelihood function. Since the mean and maximum of a Gaussian distribution are the same, $\boldsymbol{\mu}^*$ equals the centralized joint maximum likelihood (ML) estimator under non-informative prior.

**Theorem 3.** Under non-informative prior of $\boldsymbol{\beta}_i$, the MSE of the estimator $[\frac{1}{\boldsymbol{\mu}_2^*(1)}, \frac{\boldsymbol{\mu}_2^*(2)}{\boldsymbol{\mu}_2^*(1)}, \ldots, \frac{1}{\boldsymbol{\mu}_M^*(1)}, \frac{\boldsymbol{\mu}_M^*(2)}{\boldsymbol{\mu}_M^*(1)}]^T$ obtained from the converged BP message mean vectors $\boldsymbol{\mu}_i^*$ asymptotically approaches the centralized CRB of $\boldsymbol{\zeta} = [\theta_2, \alpha_2, \ldots, \theta_M, \alpha_M]^T$, where the CRB is given by (42) in the Appendix.

*Proof*: As discussed after (37), under non-informative prior, $\boldsymbol{\mu}^*$ equals the centralized joint ML estimator of $[\boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_M^T]^T$. Due to $\boldsymbol{\beta}_i = [\frac{1}{\alpha_i}, \frac{\theta_i}{\alpha_i}]^T$ and from the invariance property of ML estimator [38], $[\frac{1}{\boldsymbol{\mu}_2^*(1)}, \frac{\boldsymbol{\mu}_2^*(2)}{\boldsymbol{\mu}_2^*(1)}, \ldots, \frac{1}{\boldsymbol{\mu}_M^*(1)}, \frac{\boldsymbol{\mu}_M^*(2)}{\boldsymbol{\mu}_M^*(1)}]^T$ is the ML estimator of $\boldsymbol{\zeta} = [\theta_2, \alpha_2, \ldots, \theta_M, \alpha_M]^T$, with the corresponding MSE

Fig. 4.	Convergence performance of estimated clock skew at two nodes.
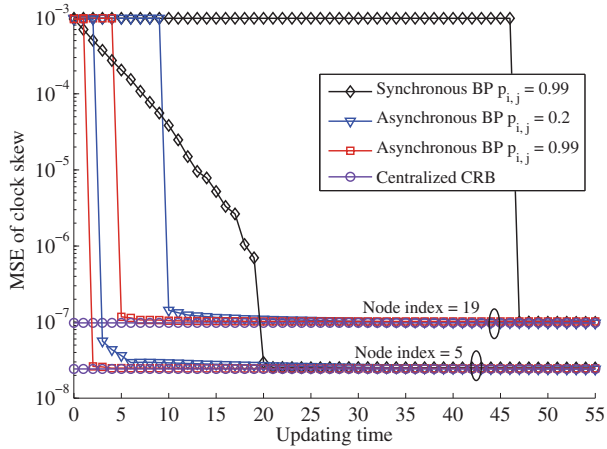


Fig. 5.	Convergence performance of estimated clock offset at two nodes.

asymptotically approaches the centralized CRB of $\zeta$ derived in (42) in the Appendix. ∎

Synchronous message updating, i.e., $\mathcal{L}_1 = \ldots = \mathcal{L}_M$ and $\tau_j^k(l-1) = l-1$, is obviously a special case of (19) and (20). Hence, Theorem 1, Theorem 2 and Theorem 3 also apply to the synchronous BP.

## V. SIMULATION RESULTS

This section presents numerical results to assess the performance of the proposed algorithm. Simulation results of estimation mean-square-error (MSE) are presented for random networks with 25 nodes randomly located in an area of size $[0, 300] \times [0, 300]$. Each node can only communicate with the sensor nodes that are within its radio range, which is assumed to be 90. In each simulation, clock skews $\alpha_i$ and clock offsets $\theta_i$ are uniformly distributed in the range $[-0.945, 1.055]$ and $[-5.5, 5.5]$, respectively. The fixed delay $d_{i,j}$ is uniformly distributed in $[8, 12]$ and variance of random delay $\sigma_i^2 = 0.05$ is assumed to be identical for all nodes. 5000 Monte-carlo simulation trials were performed to obtain the average performance of each point in all the figures presented in this section. Without loss of generality, Node 1 is selected as the reference node with $\boldsymbol{\beta}_1 = [1,0]^T$, and $p(\boldsymbol{\beta}_1) = \delta(\boldsymbol{\beta}_1 - [1,0]^T)$. For the other nodes, non-informative prior is assumed $p(\boldsymbol{\beta}_i) = \mathcal{N}(\boldsymbol{\beta}_i; \mathbf{0}, +\infty \boldsymbol{I}_2)$. The probability of node $i$ successfully pass a message to its direct neighboring node $j$ is $p_{i,j}$ for $\{i, j\} \in \mathcal{E}$. With $p_{i,j} \neq 1$, we can emulate an asynchronous network. To serve as a reference of the distributed estimation performance, the CRB for centralized estimation is derived in the Appendix.

Fig. 4 shows the MSE of the clock skew estimations in nodes 19 and 5 as a function of updating time $\{0, 1, 2, \ldots\}$ for the topology of WSN shown in Fig. 1. The number of time-stamp exchange rounds is $N = 20$ at the beginning. Synchronous schedule, asynchronous schedule and centralized CRB are plotted for comparison. The synchronous algorithm can only be updated when each node has successfully received updated messages from all its neighboring nodes. It can be seen from the figure that for both synchronous and asynchronous algorithms, MSEs touch the corresponding CRBs,
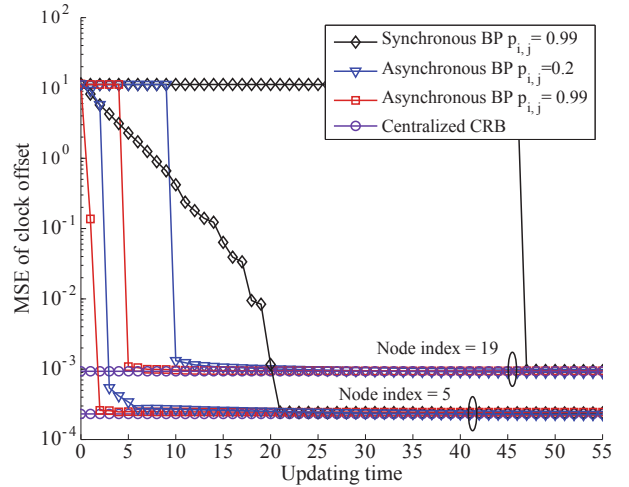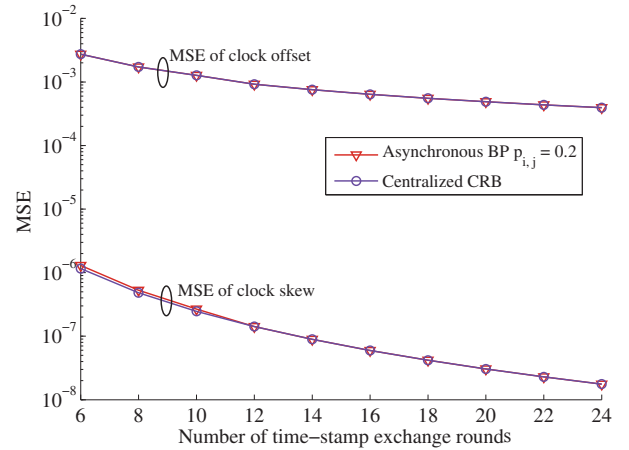


Fig. 6.	MSE of clock skew and offset averaged over the whole network under random network topologies.

which are supported by Theorem 3. However, due to the random packet losses, their convergence speeds differ. Even for high probability of successful transmission ($p_{i,j} = 0.99$), the network with synchronous schedule has to wait for all nodes to receive newly updated information from all neighbours, thus it presents slow convergence. For the same $p_{i,j}$, asynchronous scheduling shows extremely fast convergence, since each node updates independently. Furthermore, even with very low probability of successful transmission ($p_{i,j} = 0.2$), asynchronous scheduling can also converge within 10 iterations. However, with such a small $p_{i,j}$, synchronous scheduling would waste most of its time in waiting for updated messages,

and shows extremely slow convergence. The convergence properties of nodes 5 and 19 are also compared in Fig. 4. As node 5 being a neighbour of the reference node, while node 19 being much far away, node 5 converges faster than node 19. Besides, we observe that the further away from the reference node, the larger is the corresponding CRB, i.e., $\text{CRB}(\alpha_{19}) > \text{CRB}(\alpha_5)$. Fig. 5 shows the corresponding results for the clock offset estimation. It can be seen from the figure that same conclusions as in Fig. 4 can be drawn. Finally,

Fig. 6 shows the MSE for clock skews and offsets averaged over all nodes versus the number of time-stamp exchange rounds $N$. $p_{i,j} = 0.2$ is assumed for the network. The MSE is computed after the asynchronous BP algorithm runs for 30 updating iterations. 5000 random network topologies were generated for averaging. As shown in the figure, the network MSE achieves the best performance as it reaches the CRB. This figure also shows that the proposed algorithm can achieve the best performance even under a small number of time-exchange rounds.

## VI. CONCLUSIONS

In this paper, an asynchronous fully distributed clock skew and offset estimation algorithm for WSNs was proposed. The algorithm is based on asynchronous BP and is easy to be implemented by exchanging limited information between neighboring sensor nodes. The proposed algorithm can handle random packet losses and allows some nodes to compute faster and execute more iterations than others. It was shown analytically that the totally asynchronous algorithm converges regardless of the network topology, and the MSE of the clock parameter estimates reaches the centralized CRB asymptotically. Simulations further showed that the asynchronous algorithm converges faster than its synchronous counterpart.

## APPENDIX

We derive the centralized CRB under the assumption that all information over the network can be gathered in a center. First, rewrite (2) and (3) as

$$
\begin{bmatrix} c_j(t_n^2) & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{\alpha_j} \\ \frac{\theta_j}{\alpha_j} \end{bmatrix}}_{\boldsymbol{\beta}_j} = \begin{bmatrix} c_i(t_n^1) & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{\alpha_i} \\ \frac{\theta_i}{\alpha_i} \end{bmatrix}}_{\boldsymbol{\beta}_i} + d_{i,j} + w_{j,n},
$$

(38)

and

$$
\begin{bmatrix} c_j(t_n^3) & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{\alpha_j} \\ \frac{\theta_j}{\alpha_j} \end{bmatrix}}_{\boldsymbol{\beta}_j} = \begin{bmatrix} c_i(t_n^4) & -1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{\alpha_i} \\ \frac{\theta_i}{\alpha_i} \end{bmatrix}}_{\boldsymbol{\beta}_i} - d_{j,i} - w_{i,n}.
$$

(39)

Stacking (38) and (39) in matrix form with the assumption $d_{i,j} = d_{j,i}$, we have (40), where $\mathbf{1}_N$ is an all one $N$ dimensional vector and $\boldsymbol{n}_{j,i} \sim \mathcal{N}(\boldsymbol{n}_{j,i}|\mathbf{0}, \text{diag}[\sigma_j^2, \sigma_i^2] \otimes \boldsymbol{I}_N)$ where the symbol $\otimes$ denotes the Kronecker product.

Define $\boldsymbol{y} \in \mathbb{R}^{2N|\mathcal{E}|\times 1}$ with $-\boldsymbol{T}_{1,i}\boldsymbol{\beta}_1$ arranged in ascending order with respect to index $i$, with $i \in \mathcal{I}(1)$ and the remaining

elements being zeros, and define $\boldsymbol{\xi} \triangleq [\boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_M^T, \boldsymbol{d}^T]^T$ with vector $\boldsymbol{d}$ containing elements $d_{i,j}$ with ascending order first with respect to $j$ and then with respect to $i$. Then stacking (40) for all $i$ and $j$, we obtain

$$
\boldsymbol{y} = \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{n}, \tag{41}
$$

where $\boldsymbol{n}$ contains $\boldsymbol{n}_{i,j}$ with ascending order first with respect to $j$ and then with respect to $i$. Notice that $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{n}|\mathbf{0}, \boldsymbol{\Delta})$ with $\boldsymbol{\Delta}$ is a block diagonal matrix containing $\boldsymbol{\Delta}_{i,j} = \text{diag}[\sigma_j^2, \sigma_i^2] \otimes \boldsymbol{I}_N$ as diagonal block. Since (41) is a standard linear model, the CRB for $\boldsymbol{\xi}$ is given by $\text{CRB}(\boldsymbol{\xi}) = [\boldsymbol{H}^T \boldsymbol{\Delta}^{-1} \boldsymbol{H}]^{-1}$ [38].

The ultimate goal is to estimate the clock offsets and skews $\boldsymbol{\zeta} \triangleq [\theta_2, \alpha_2, \ldots, \theta_M, \alpha_M]^T$. Since $\boldsymbol{\xi}$ is a related to $\boldsymbol{\kappa} \triangleq [\boldsymbol{\zeta}^T, \boldsymbol{d}^T]^T$ through a transformation, thus we can express the CRB matrix of $\boldsymbol{\zeta}$ as [38]

$$
\text{CRB}(\boldsymbol{\zeta}) = \left(\frac{\partial \boldsymbol{\kappa}}{\partial \boldsymbol{\xi}}\right)\text{CRB}(\boldsymbol{\xi})\left(\frac{\partial \boldsymbol{\kappa}}{\partial \boldsymbol{\xi}}\right)^T. \tag{42}
$$

It can be easily inferred that $\partial \boldsymbol{\kappa}/\partial \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{\frac{1}{2}|\mathcal{E}|} \end{bmatrix}$ with $\boldsymbol{\Sigma}$ being a $2(M-1)$-by-$2(M-1)$ block diagonal matrix with the $m^{th}$ diagonal block being $\begin{bmatrix} -\alpha_{m+1}\theta_{m+1} & \alpha_{m+1} \\ -\alpha_{m+1}^2 & 0 \end{bmatrix}$.

## REFERENCES

[1] S. Ehsan, K. Bradford, M. Brugger, B. Hamdaoui, Y. Kovchegov, D. Johnson, and M. Louhaichi, "Design and analysis of delay tolerant sensor networks for monitoring and tracking free-roaming animals," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1220–1227, Mar. 2011.

[2] E. Msechu and G. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 400–414, Jan. 2012.

[3] D. Arifler, "Information theoretic approach to detecting systematic node destructions in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4730–4738, 2008.

[4] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[5] B. Sadler and A. Swami, "Synchronization in sensor networks: an overview," in *Proc. 2006 IEEE Military Commun. Conf.*, pp. 1–6.

[6] J.-S. Kim, J. Lee, E. Serpedin, and K. Qaraqe, "Robust clock synchronization in wireless sensor networks through noise density estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3035–3047, July 2011.

[7] I. Skog and P. Handel, "Synchronization by two-way message exchanges: Cramer-Rao bounds, approximate maximum likelihood, and offshore submarine positioning," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2351–2362, Apr. 2010.

[8] M. Leng and Y.-C. Wu, "Low complexity maximum likelihood estimators for clock synchronization of wireless sensor nodes under exponential delays," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4860–4870, Oct. 2011.

$$
\underbrace{\begin{bmatrix} c_j(t_1^2) & -1 \\ \vdots & \vdots \\ c_j(t_N^2) & -1 \\ c_j(t_1^3) & -1 \\ \vdots & \vdots \\ c_j(t_N^3) & -1 \end{bmatrix}}_{\triangleq \boldsymbol{T}_{j,i}} \boldsymbol{\beta}_j - \underbrace{\begin{bmatrix} c_i(t_1^1) & -1 \\ \vdots & \vdots \\ c_i(t_N^1) & -1 \\ c_i(t_1^4) & -1 \\ \vdots & \vdots \\ c_i(t_N^4) & -1 \end{bmatrix}}_{\triangleq \boldsymbol{T}_{j,i}} \boldsymbol{\beta}_i - d_{i,j}\begin{bmatrix} \mathbf{1}_N \\ -\mathbf{1}_N \end{bmatrix} = \underbrace{\begin{bmatrix} w_{j,1} \\ \vdots \\ w_{j,N} \\ -w_{i,1} \\ \vdots \\ -w_{i,N} \end{bmatrix}}_{\triangleq \boldsymbol{n}_{j,i}}. \tag{40}
$$

[9] A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "A factor graph approach to clock offset estimation in wireless sensor networks," *IEEE Trans. Inf. Theory,* vol. 58, no. 7, pp. 4244–4260, July 2012.

[10] A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "Time-varying clock offset estimation in two-way timing message exchange in wireless sensor networks using factor graphs," in *2012 IEEE International Conf. Acoustics, Speech, Signal Process.*

[11] D. R. Jeske, "On maximum-likelihood estimation of clock offset," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 53–54, Jan. 2005.

[12] K.-L. Noh, E. Serpedin, and B. 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.

[13] K.-Y. Cheng, K.-S. Lui, Y.-C. Wu, and V. Tam, "A distributed multihop time synchronization protocol for wireless sensor networks using pairwise broadcast synchronization," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1764–1772, Apr. 2009

[14] B. R. Hamilton, X. Ma, Q. Xhao, and J. Xu, "ACES: adaptive clock estimation and synchronization using Kalman filtering," in *Proc. 2008 ACM Int. Conf. Mobile Comput. Netw.*, pp. 152–162.

[15] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *Proc. 2005 ACM Sensys Conf.*, pp. 130–141.

[16] Q. Gao, K. J. Blow, and D. J. Holding, "Simple algorithm for improving time synchronization in wireless sensor networks," *Electron. Lett.*, vol. 40, pp. 889, 2004.

[17] D. Tulone, "Resource-efficient time estimation for wireless sensor networks," in *Proc. 2004 DIALM-POMC Workshop Found. Mobile Comput.*, p. 42–50.

[18] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. 2003 Int'l Conf. Embedded Netwo. Sensor Systems*, pp. 138–149.

[19] 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, Sept. 2008.

[20] 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.

[21] 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.

[22] O. Simeone and U. Spagnolini, "Distributed synchronization in wireless networks," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 81–97. Sept. 2008.

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

[24] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," in in *Proc. 2006 IEEE Conf. Decision Contr.*, pp. 4915–4920.

[25] B. Choi, H. Liang, X. Shen, and W. Zhuang, "DCS: distributed asynchronous clock synchronization in delay tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 491–504, 2012.

[26] 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, 9 pages, 2009.

[27] J. He, P. Cheng, L. Shi, and J. Chen, "Clock synchronization for random mobile sensor networks," in *Proc. 2012 IEEE Conf. Decision Contr.*, pp. 2712–2717.

[28] F. Kirsch and M. Vossiek, "Distributed Kalman filter for precise and robust clock synchronization in wireless networks," in *Proc. 2009 Int. Conf. Radio Wireless Symp.*, pp. 455–458.

[29] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5404–5414, Nov. 2011.

[30] J. Du and Y.-C. Wu, "Fully distributed clock skew and offset estimation in wireless sensor networks," in *Proc. 2013 IEEE Acoustics Speech Signal Process. Conf.*

[31] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 498–519, Feb. 2001.

[32] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, 1997.

[33] D. P. Bertsekas and J. N. Tsitsiklis, "Some aspects of parallel and distributed iterative algorithms—a survey," *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

[34] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation,* vol. 13, no. 10, pp. 2173–2200, 2001.

[35] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Machine Learning Research,* vol. 7, Oct. 2006.

[36] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4753–4766, Nov. 2006.

[37] A. Papoulis and S. U. Pillai, *Random Variables and Stochastic Processes,* 4th ed. McGraw-Hill, 2002.

[38] S. M. Kay, *Fundamentals of Statistical Signal Processing Estimation Theory.* Prentice-Hall, 1993.

[39] S. Puntanen, G. Styan, and J. Isotalo, *Matrix Tricks for Linear Statistical Models Our Personal Top Twenty.* Springer, 2011.

[40] A. M. Sarhan, N. M. EI-Shazly, and E. M. Shehata, "On the existence of extremal positive defnite solutions of the nonlinear matrix equation $\boldsymbol{X}^r + \sum_{i=1}^m \boldsymbol{A}_i^* \boldsymbol{X}^{\delta_i} \boldsymbol{A}_i = \boldsymbol{I}$," *Mathematical and Computer Modelling*, vol. 51, no. 9–10, 2010.

[41] R. A. Horn and C. R. Johnson, *Matrix Analysis* Cambridge University Press, 1985.

[42] J. W. Demmel *Applied Numerical Linear Algebra.* SIAM, 1997.

**Jian Du** received the B.Eng. and M.Eng. degrees in communication and information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006 and 2009, respectively. He is currently working towards his Ph.D. degree with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong. His research interests are in distributed signal processing, statistical learning and inferences for wireless communication systems and smart grid.

**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 Croucher Foundation scholarship in 2002 to study Ph.D. degree at Texas A&M University, College Station, and graduated in 2005. 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 HKU, currently as an Associate Professor. He was a visiting scholar at Princeton University, in summer 2011. His research interests are in general area of signal processing and communication systems, and in particular distributed signal processing and communications; optimization theories for communication systems; estimation and detection theories in transceiver designs; and smart grid. Dr. Wu served as an Editor for IEEE COMMUNICATIONS LETTERS, and is currently an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS and the *Journal of Communications and Networks*.