Time Synchronization for Random Mobile Sensor Networks

Jianping He, Peng Cheng, *Member, IEEE*, Jiming Chen, *Senior Member, IEEE*, Ling Shi, and Rongxing Lu, *Member, IEEE*

***Abstract*—Mobile sensor nodes have a wide spectrum of applications, such as social networks and habitat monitoring. Time synchronization is a critical issue for most applications using mobile sensor networks. However, due to the limited communication range, mobile sensor nodes can only exchange their information when they are sufficiently close for contact. Moreover, random movements of the nodes render the performance analysis of any time synchronization protocols challenging. In this paper, we introduce the relation graph for modeling the random contact of mobile sensor nodes and evaluate the probability that the network can be synchronized within an arbitrary time. We adapt the previous maximum time synchronization (MTS) protocol, which can drive the clocks of all sensor nodes to a common value by utilizing their own neighboring information. We provide an analytical lower bound of the probability that the time synchronization can be finished within any given time. The obtained theories and algorithms are applied for several fundamental problems, and it is proven that a better connectivity is beneficial for convergence. For linearizable graph, we provide an efficient way to calculate the exact probabilities. Extensive numerical examples demonstrate the effectiveness of our results.**

***Index Terms*—Clock synchronization, finite-time convergence, maximum consensus, mobile sensor networks.**

I. Introduction

RANDOM mobile sensor networks (RMSNs) are characterized by dynamic and random connections of links among nodes due to their random mobility. Mobile sensor networks are attracting increased research interest, e.g., delay-tolerant networks (DTNs) and social networks [2]–[5], estimation and control [6], [7], synchronization [8], traffic monitoring [9], etc. Time synchronization is an important requirement in RMSNs for providing accurate time information of data collections and for energy conservation. There are many applications where accurate time synchronization of sensor nodes is crucial, such as mobile target tracking, event detection, speed estimating, etc., [9]–[11]. Moreover, accurate time synchronization is crucial for energy-efficient sleep scheduling mechanisms as it provides the possibility of setting nodes into the sleeping mode [12]–[14]. A detailed survey for the significance of time synchronization in sensor networks can be found in [10] and [15].

The topology of the mobile sensor network changes randomly and depends on the probability distribution of contacts among the nodes. Thus, in general, the topology of the network being surely connected or jointly (union) connected, cannot be guarenteed, which renders the design and theoretical analysis of any time synchronization protocols challenging.

Many synchronization protocols for sensor networks have been provided by researchers recently, including traditional tree-based and distributed consensus-based protocols [8], [12], [16]–[18], [20], [21], [25], [26]. Traditional protocols [16]–[18] require a root node and are tree based, which means that they are fragile to link or there are node failures. Thus, these traditional protocols are not suitable for clock synchronization in RMSNs. Existing distributed protocols [8], [12], [25], [26] and the associated theoretical results are obtained by assuming that the topology of the network is connected or jointly (union) connected, which, however, no longer holds in RMSNs. These protocols also have slow convergence speed in general. For instance, the protocol in [25] is based on gossip averaging algorithms [29], and the protocols GTSP in [26] and ATS in [8] are based on average consensus algorithm [30]. These protocols have slow convergence speed as pointed out in [19].

Carli *et al.* [27], [28] treat the different clock speeds as unknown constant disturbances and the different clock offsets as different initial conditions for the system dynamics. Based on this technique, they further develop a proportional–integral (PI) controller to the time synchronization algorithm. Compared with ATS, although the algorithms proposed in [27] and [28] have slightly slower convergence speed, they outperform ATS in terms of robustness against process noise and measurement noise and time-varying clock drifts. Recently, by using the PI control principle, Chen *et al.* [22] propose a feedback-based synchronization (FBS) scheme to compensate the clock drift caused by both internal perturbation and external disturbance, which achieves highly accurate time synchronization. Leng

Manuscript received September 4, 2013; revised February 16, 2014; accepted February 17, 2014. Date of publication February 20, 2014; date of current version October 14, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 612223005, by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20120101110139, and by the National Program for Special Support of Top-Notch Young Professionals. The work of L. Shi was supported by the Hong Kong Research Grants Council through the General Research Fund under Grant 618612. This paper was presented in part at the 51st Annual IEEE Conference on Decision and Control, Maui, HI, USA, December 10–12, 2012. The review of this paper was coordinated by Prof. Y. Fang. *(Corresponding author: P. Cheng.)*

J. He, P. Cheng, and J. Chen are with the State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: jphe@iipc.zju.edu.cn; pcheng@iipc.zju.edu.cn; jmchen@ieee.org).

L. Shi is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: eesling@ust.hk).

R. Lu is with the Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: rxlu@ntu.edu.sg).

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/TVT.2014.2307076

and Wu [23] address how to jointly estimate clock offset and clock skew with unknown delay in time synchronization by utilizing belief propagation for achieving better synchronization accuracy than the average-consensus-based algorithms [24]. However, the algorithm introduces more computational complexity and information storage. A detailed survey for adopting a statistical signal processing approach for handling delay in time synchronization is provided in [32].

Most of the aforementioned algorithms are developed for deterministic wireless sensor networks (WSNs). There are some works that investigate time synchronization for RMSNs. For example, Liao and Barooah [31] model the dynamic network as a Markov chain, and they propose an algorithm for estimating the offsets and skews under the assumption that the union of all graphs is connected. Choi *et al.* [13] present a distributed clock synchronization (DCS) in DTNs, which is a class of RMSNs. However, DCS is still an average-consensus-based algorithm, which has slow convergence speed. Therefore, it is of great interest to develop a distributed time synchronization protocol that has faster converging speed for RMSNs.

The main contributions of this paper are summarized as follows.

1) We define the relation graph to describe the contact relationship between nodes in RMSNs. Unlike the classic communication graph, two nodes in the relation graph are neighbor nodes if and only if they have positive contact probability parameters. Based on the relation graph, we relax the basic convergence condition and provide theoretical analysis of the protocol for RMSNs.
2) The maximum time synchronization (MTS) protocol in [19] is revised to solve the time synchronization in random sensor networks. We prove that the skew and offset can be compensated so as to converge simultaneously. Meanwhile, we provide theoretical results for estimating the lower bound of the probability of finite-time convergence, and develop an algorithm for computing the lower bound for linearizable graphs.
3) We apply the developed theories and algorithms for solving several fundamental problems. Specifically, we prove that an RMSN, which has better connectivity in the relation graph, has higher probability of convergence within a given time. Furthermore, based on this theoretical result, we present a method that provides a better lower bound of the probability of finite-time convergence. In addition, we derive the condition when the probability of finite-time convergence can be increased by adding new nodes into the network.
4) We conduct extensive simulations that prove the effectiveness of our algorithm and theories. Moreover, our protocol converges much faster than the latest average-consensus-based protocol [13], which is also designed for time synchronization of RMSNs.

The remainder of this paper is organized as follows. In Section II, the network and clock models for RMSNs are introduced. In Section III, the maximum-value-based protocol for time synchronization is proposed, and the proof for its convergence is provided, as well as the analysis of the probability of the finite-time convergence. In Section IV, several fundamental problems are solved based on the developed theories and algorithms. Simulation results are presented in Section V and Section VI concludes this paper.

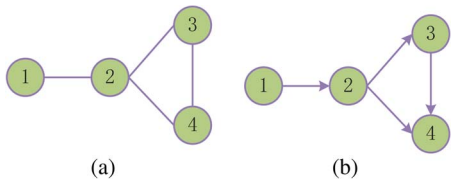


Fig. 1. Relation graph and its subgraph. (a) Relation graph $\mathcal{G}$. (b) Subgraph $\mathcal{G}_{14}$.

II. Problem Formulation

A. Random Mobile Network Model

Consider n nodes indexed by $i = 1, 2, \ldots, n$, with random mobility. Assume that the nodes have the same transmission range $\mathcal{R}$ and that every two nodes can communicate with each other when they are within their transmission range, which is referred to as a communication contact. Since the nodes move randomly, the communication contact between two nodes depends on the probability distribution of the nodes' mobility. In our network model, the occurrence of the contacts between any two nodes is assumed to follow the Poisson distribution, i.e., the contact rate of nodes i and j is a Poisson distribution with the parameter $\lambda_{ij} \geq 0$ for $i, j = 1, 2, \ldots, n$. We assume that each λ_{ij} is a constant for $i, j = 1, 2, \ldots, n$. Similar assumptions have been used in other existing works to analyze the data dissemination [2], [3] in DTNs and the multicast capacity [4]. Under this assumption, the random communication used in [34] can be also viewed as a special contact case caused by the random mobility of nodes. If node i cannot make contact with node j, we set $\lambda_{ij} = 0$, and if node i always makes contact with node j, we set $\lambda_{ij} = \infty$. Note that a node always has contact with itself; therefore, $\lambda_{ii} = \infty$ for all i. Let $\mathbf{B} = \{\mathbf{B}_{ij}\}$ be a matrix with its element $\mathbf{B}_{ij} = 1$ for $\lambda_{ij} > 0$ or $\lambda_{ij} = \infty$, and $\mathbf{B}_{ij} = 0$ for $\lambda_{ij} = 0$.

The RMSNs are modeled as a relation graph as follows, which describes the contact relationship between the nodes (see also Fig. 1).

Definition 2.1: A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is defined as a relation graph of a random mobile WSN if and only if its adjacency matrix is equal to $\mathbf{B}$.

Definition 2.2: $\mathcal{G}_{sd}$ is defined as a subgraph of $\mathcal{G}$, which is composed of all paths from s to d in $\mathcal{G}$.

Let $\mathcal{V} = \{i : i = 1, 2, \ldots, n\}$ be the set of the mobile sensor nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ be the set of edges. It is obvious that $\{i, j\} \in \mathcal{E}$ iff $\lambda_{ij} > 0$ for $i, j \in \mathcal{V}$. Let $\mathcal{N}_i = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$ be the set of neighbors of node i. Then, one obtains that $j \in \mathcal{N}_i$ iff $\lambda_{ij} > 0$. Since the sensor nodes are assumed to have the same transmission range $\mathcal{R}$, relation graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is an undirected connected graph.

For the RMSNs, we have assumed that the contacts between any two nodes follow the Poisson distribution with fixed contact parameters λ_{ij} for $i, j = 1, 2, \ldots, n$. Thus, the contact

relationship between nodes is fully depicted by the contact probability between nodes, i.e., contact parameter matrix $\lambda = \{\lambda_{ij}\}$, with nonnegative elements, can characterize the contact relationship between nodes completely. Note that a nonnegative matrix can be modeled as a graph based on classical graph theory; thus, we utilize the given relation graph to model the random mobile networks in this paper. As a consequence, the contacts between nodes are modeled as a deterministic relation graph. Based on this deterministic relation graph, we shall provide theoretical analysis of the algorithm in the remainder of this paper.

Remark 2.3: In [17]–[19], where deterministic networks are considered, it is crucial to assume that the network is connected or jointly connected for network-wide clock synchronization. This assumption no longer holds in RMSNs, as a connected relation graph only guarantees that the network is connected with probability 1 when time $t \to \infty$, i.e., this condition is relaxed so that the relation graph for RMSNs is connected.

Example 2.4: Consider an RMSN with its contact parameter matrix $\lambda = \{\lambda_{ij}\}$ as

$$\lambda = \begin{bmatrix} \infty & 2 & 0 & 0 \\ 2 & \infty & 1 & 3 \\ 0 & 1 & \infty & 1 \\ 0 & 3 & 1 & \infty \end{bmatrix}.$$

Then, it obtains the adjacency matrix $\mathbf{B}$ of the relation graph as

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

Thus, according to the two given definitions, we obtain relation graph $\mathcal{G}$ and one of its subgraph $\mathcal{G}_{14}$ as follows.

Note that the communication collision will decrease the successful communication probability between two nodes and thus can be regarded as that the contact probability of these two node is decreased. It is assumed that the communication delay can be ignored in this paper, as in [8] and [26]. Note that, if this is not the case, there are several methods that can be utilized to handle the packet delivery delay (see, e.g., [10], [12], [23], and [24] for transmission delay compensation).

B. Clock Model

Consider a commonly used linear hardware clock model as follows:

$$\tau_i(t) = a_i t + b_i \tag{1}$$

where a_i is the hardware clock skew, which determines the clock speed, and b_i is the hardware clock offset. Obviously, if the clock oscillator of each node i is perfect, then $a_i = 1$ and $b_i = 0$. Since the real time t is unavailable to each node, both a_i and b_i cannot be computed [8]. However, by comparing the local clock readings of nodes i and j, we can obtain a relative clock between them, which is given by $\tau_i(t) = (a_i/a_j)\tau_j(t) + (b_i - (a_i/a_j)b_j) = a_{ji}\tau_j(t) + b_{ji}$.

Since the value of the hardware clock cannot be adjusted manually [26], a logical clock is developed to represent the synchronized time, which is given by

$$L_i(t) = \hat{a}_i \tau_i(t) + \hat{b}_i = \hat{a}_i a_i t + \hat{a}_i b_i + \hat{b}_i$$

where $\hat{a}_i$ and $\hat{b}_i$ are two estimated parameters. Our goal is to find parameters $(\hat{a}_i, \hat{b}_i)$ and $(\hat{a}_j, \hat{b}_j)$ for $i,\ j \in \mathcal{V}$ such that $L_i(t) = L_j(t), \forall\ i, j \in \mathcal{V}$. Therefore, the objective of the time synchronization algorithm is to find $(\hat{a}_i(k), \hat{b}_i(k))$ for each sensor node $i, i \in \mathcal{V}$, such that

$$\lim_{k\to\infty} \hat{a}_i(k) a_i = a_v \tag{2}$$

$$\lim_{k\to\infty} \hat{a}_i(k) b_i + \hat{b}_i(k) = b_v \tag{3}$$

where k denotes the iteration of the algorithm.

III. Algorithm and Convergence Analysis

Here, we will propose a time synchronization algorithm and further provide some theoretical analysis on its convergence and finite-time convergence based on the relation graph model.

A. Revised Maximum Time Synchronization

Let $a_{ij} = a_j/a_i$ be the relative skew of node i with respect to node j. The estimation of the relative skew is crucial to the accuracy of synchronization. Define $s_{ij}(k)$ as a single estimation of relative skew, which is estimated by

$$s_{ij}(k) = \frac{\tau_j(t_k) - \tau_j(t_{k-1})}{\tau_i(t_k) - \tau_i(t_{k-1})}, \qquad i, j \in \mathcal{V} \tag{4}$$

where $(\tau_i(t_k), \tau_j(t_k))$ and $(\tau_i(t_{k-1}), \tau_j(t_{k-1}))$ are two pairs of time sampling, and k denote that $(k+1)$th node i receives a message from node j. In this paper, the following equation is used to estimate each relative skew a_{ij}:

$$a_{ij}(k) = \frac{s_{ij}(k) + (k-1)a_{ij}(k-1)}{k}, \qquad k \in \mathcal{N}^+. \tag{5}$$

By some manipulation, (5) can be rewritten as

$$a_{ij}(k) = \frac{1}{k}\Sigma_{l=1}^{k} s_{ij}(l), \qquad k \in \mathcal{N}^+$$

which means that $a_{ij}(k)$ is the average of k times estimation of each relative skew a_{ij} for $i,\ j \in \mathcal{V}$. Thus, increasing the communication times between the nodes will decrease the estimate error of each relative skew.

As shown in Fig. 2, node j sends its hardware clock time-stamp to a neighboring node i when they are contacting, whereas node i records its current hardware clock reading when it receives the packet from node j, and then node i computes $s_{ij}(k)$ and $a_{ij}(k)$ based on (4) and (5), respectively. Note the fact that, if two neighboring nodes send information to each other at the same time, then they cannot receive any messages from each other due to the interference. Thus, when two nodes contact with each other, one of them will send its information

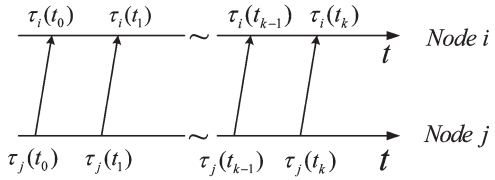


Fig. 2. Relative skew estimation.

first, and the other will only send its information after it receives the packet.

The MTS algorithm [19] is revised as follows to handle time synchronization for RMSNs.

Algorithm 1 *Revised MTS (RMTS)*

1: Set initial conditions $\hat{a}_i = 1$ and $\hat{b}_i = 0$ to each node i for $i \in \mathcal{V}$.
2: If node i contacts with node j at time t, then j first sends a packet to node i, including its current hardware clock reading $\tau_j(t)$ and its current parameters $\{\hat{a}_j, \hat{b}_j\}$.
3: If node i receives the packet from node j for the first time, then it records its current clock reading as $\tau_i(t_0)$ and the packet message of node j as $[\tau_j(t_0), \hat{a}_j, \hat{b}_j]$. Then, it sends a packet, including $[\tau_i(t_0), \hat{a}_i, \hat{b}_i]$, back to node j and stores $[\tau_i(t_0), \tau_j(t_0)]$. Otherwise, go to step 4.
4: If node i receives the packet from node j and has a historical record $[\tau_i(t_{k-1}), \tau_j(t_{k-1})]$, then it computes the $s_{ij}(k)$ and $a_{ij}(k)$ by (4) and (5), respectively.
5: Node i computes q_{ij} by $q_{ij} = a_{ij}(k)\hat{a}_j/\hat{a}_i$. If $q_{ij} > 1$, then

$$\hat{a}_i \leftarrow a_{ij}(k)\hat{a}_j,$$
$$\hat{b}_i \leftarrow \hat{a}_j\tau_j(t) + \hat{b}_j - a_{ij}(k)\hat{a}_j\tau_i(t). \tag{6}$$

If $q_{ij} = 1$, then

$$\hat{b}_i \leftarrow \max_{x=i,j}\left\{\hat{a}_x\tau_x(t) + \hat{b}_x\right\} - \hat{a}_i\tau_i(t). \tag{7}$$

6: Node i sends a packet, including its current hardware clock reading $\tau_i(t)$ and current parameters $\{\hat{a}_i, \hat{b}_i\}$, to node j.
7: Node i stores information $[\tau_i(t_k), \tau_j(t_k), a_{ij}(k)]$, deletes historical record $[\tau_i(t_{k-1}), \tau_j(t_{k-1}), a_{ij}(k-1)]$, and then goes to step 2.

For RMTS, unlike MTS in [19], each node does not send its information periodically to the neighboring nodes for saving energy, and two nodes will communicate with each other and update their clocks only when they make contact with each other, i.e., when the distance between two nodes is less than communication range $\mathcal{R}$. To ensure that two nodes can communicate when they are neighbors with each other, a typical neighbor discovery protocol can be used as in [33]. Since $a_{ij}(k)$ in (5) is the average of k times estimation and used in RMTS, the estimate error of a_{ij} can be decreased by increasing communication times between nodes i and j. In step 5 of RMTS, q_{ij} satisfies

$$q_{ij} = \frac{a_{ij}\hat{a}_j}{\hat{a}_i} = \frac{\hat{a}_j a_j}{\hat{a}_i a_i}$$

which is the ratio of two nodes' logical clock skew. Hence, node i can know from q_{ij} whether its neighbor node j has the larger logical clock skew.

Since the clock skew, which denotes the clock speed, is compensated in the given algorithm, it can decrease the frequency of synchronization. In particular, if the clock skew compensation can get the same logical clock skews for all nodes, then after the completion of synchronization, there is no need to synchronize again. Assume that nodes i and j can communicate with each other successfully when they are contacting. Since the estimation error of each a_{ij} is decreasing with the communication times between nodes i and j, it is ignored in this paper.

Remark 3.1: For RMTS, the only requirement is that each node i sends its hardware clock reading $\tau_i(t)$ and parameters $\hat{a}_i(t)$ and $\hat{b}_i(t)$, which is also essential for the average-consensus-based algorithms, e.g., DCS [13] and ATS [31]. Meanwhile, for RMTS, when node i receives the message from node j, it only computes the logical clocks of both node j and itself, and then updates its clock based on the maximum logical clock. However, for average-consensus-based algorithms, node i has to compute the average of the logical clocks of node j and itself, and additionally updating according to the average logical clock. Hence, RMTS has lower computation complexity than that of average-consensus-based algorithms. Furthermore, RMTS can converge in finite time with a positive probability, which monotonically increases with time t (the details will be given later), whereas the average-consensus-based algorithms converge asymptotically. Hence, every node needs less communication time for RMTS to achieve time synchronization than that for average-consensus-based algorithms. In summary, RMTS has lower computation complexity and faster convergence speed and is thus more energy-efficient than average-consensus-based algorithms.

B. Convergence of RMTS

Before proving the convergence of our algorithm, we introduce some preliminarily results. Due to the calibration issue of the clock oscillator, the clock oscillators of different nodes are slightly different, which means that the clock skews are slightly different. Define $a_{\max} = \max_{i\in\mathcal{V}} a_i$. Let node v be the node whose clock skew is equal to $a_{\max}$, i.e., $a_v = a_{\max}$, and its hardware clock $\tau_v(t)$ is given by

$$\tau_v(t) = a_v t + b_v. \tag{8}$$

Let $\mathcal{V}_v(t)$ be the set of nodes whose logical clock skew and offset are equal to a_v and b_v at time t, respectively. Define $|\mathcal{V}_v(t)|$ as the number of nodes belonging to the set $\mathcal{V}_v(t)$ at time t. Since the initial condition satisfies $\hat{a}_i = 1$ and $\hat{b}_i = 0$ for $i \in \mathcal{V}$ in RMTS, it follows from the definition of $\mathcal{V}_v(t)$ that $|\mathcal{V}_v(0)| = 1$.

Theorem 3.2: Suppose that relation graph $\mathcal{G}$ is connected. By using RMTS, the skew and offset of each node $i, i \in \mathcal{V}$, converge at the same time and satisfy

$$\begin{cases} \Pr\left\{\lim_{t\to\infty} \hat{a}_i(t)a_i = a_v\right\} = 1 \\ \Pr\left\{\lim_{t\to\infty} \hat{a}_i(t)b_i + \hat{b}_i(t) = b_v\right\} = 1. \end{cases} \tag{9}$$

Proof: (Sketch of proof) Note that $|\mathcal{V}_v(t)| \leq n$ holds for any time t. If $|\mathcal{V}_v(t)| < n$, we can prove that $|\mathcal{V}_v(t)|$ will strictly increase with positive probability. Thus

$$\Pr\left\{\lim_{t\to\infty} |\mathcal{V}_v(t)| = n\right\} = 1$$

which yields (9). The detailed proof is given in Appendix A. ■

C. Finite-Time Convergence of RMTS

Finite-time convergence of time synchronization algorithms is an important and interesting problem in sensor networks as an algorithm with finite-time convergence is more practical and energy efficient. Here, we will consider the lower bound of the probability for the finite-time convergence of MTS.

Note that each node i becomes one node of $\mathcal{V}_v$ when it obtains messages a_v and b_v during its contacts with a node in $\mathcal{V}_v$. Hence, given time t, the probability of node i being a node of $\mathcal{V}_v(t)$ is equivalent to the probability of node v having delivered its a_v and b_v successfully to node i. For i, $j \in \mathcal{V}$, let T_{ij} be a random variable, which denotes the time needed for node i to successfully transmit a message to node j. Let $f_{ij}(t)$ be the probability density function of T_{ij}, and let $F_{ij}(t) = \int_0^t f_{ij}(\tau)d\tau$ be the probability distribution function of T_{ij}. Clearly, $F_{ii}(t) = 1$ for any time t.

Theorem 3.3: Suppose the relation graph $\mathcal{G}$ of an RMSN is connected. Then

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}\} \geq \Pi_{i\in\mathcal{V}} F_{vi}(t). \tag{10}$$

Proof: From the proof of Theorem 3.2, it can be observed that node i in $\mathcal{V} - \mathcal{V}_v$ will become a node in $\mathcal{V}_v$ after it contacts with one node j in $\mathcal{V}_v$. Hence, for each node i, when it obtains the messages of a_v and b_v from the source node v at time t, it has $\hat{a}_i(t)a_i = a_v$ and $\hat{a}_i(t)b_i + \hat{b}_i(t) = b_v$, which means that $L_i(t) = \tau_v(t)$. It follows from the definition of $F_{ij}(t)$ that

$$\Pr\{L_i(t) = \tau_v(t)\} = F_{vi}(t), \qquad i \in \mathcal{V}$$

and note the fact that

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}\} \geq \Pi_{i\in\mathcal{V}} \Pr\{L_i(t) = \tau_v(t)\}$$

always holds true, which yields (10). ■

Theorem 3.3 provides a lower bound of the probability of finite-time convergence. The lower bound in (10) can be tight for some special relation graphs, e.g., a linear graph with two nodes or a star graph with node v as the central node. For example, consider star graph $\mathcal{G}$ with central node v; time synchronization in Algorithm 1 is reached at time t if and only if every node i in the network has contact with node v before time t, and the events of that node v contacting different leaves $i, i \neq v$, are independent with each other. Hence, we follow from Theorem 3.3 that

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}\} = \Pi_{i\in\mathcal{V}} F_{vi}(t) \tag{11}$$

which is a tight bound. Specifically, note that

$$f_{vi}(t) = \lambda_{vi} e^{-\lambda_{vi}t}, \qquad i \in \mathcal{N}_v$$

and thus

$$F_{vi}(t) = \int_0^t \lambda_{vi} e^{-\lambda_{vi}\tau} d\tau = 1 - e^{-\lambda_{vi}t}, \qquad i \in \mathcal{N}_v. \tag{12}$$

By substituting (12) into (11), we have

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}\} = \prod_{i\neq v, i\in\mathcal{V}} (1 - e^{-\lambda_{vi}t})$$

which is the probability of finite-time convergence for the star relation graph.

Meanwhile, when there is only one path connecting nodes v and i, and if node i has obtained the messages from node v, then all nodes included in this path must have received the messages. Based on this, therefore, Theorem 3.3 can be further simplified.

Corollary 3.4: Suppose that relation graph $\mathcal{G}$ is a tree. Then

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}\} \geq \prod_{i\in\mathcal{V}_1} F_{vi}(t) \tag{13}$$

where $\mathcal{V}_1 = \{i | d_i = 1, i \in \mathcal{V}\}$ is the set of leaves.

The given theoretical results give the lower bound of the probability of finite-time convergence for RMTS, where the lower bound depends on probability distribution functions $F_{vi}(t)$ for $i \in \mathcal{V}$. Then, we show how to calculate each $F_{vi}(t)$.

By referring to the theoretical results in [2], we have the following Lemma.

Lemma 3.5: Assume a message is delivered from i_0 to i_m according to path $i_0 \to i_1 \to \cdots \to i_m$. Then

$$f_{i_0 i_m}(t) = f_{i_0 i_1}(t) \otimes f_{i_1 i_2}(t) \otimes \cdots \otimes f_{i_{m-1} i_m}(t) \tag{14}$$

where $\otimes$ is the convolution operator.

From Lemma 3.5, if there is only one path from node i_0 to node i_m in graph $\mathcal{G}$, then combining the definition of convolution with [2, Th. 2] yields

$$\begin{aligned} f_{i_0 i_m}(t) &= \int_0^t \int_{t_1}^t \cdots \int_{t_{m-2}}^t \left(\prod_{l=1}^m \lambda_l e^{-\lambda_l(t_l - t_{l-1})}\right) \\ &\qquad \times dt_{m-1} dt_{m-2} \cdots dt_1 \\ &= \sum_{l=1}^m C_l^m \lambda_l e^{-\lambda_l t} \end{aligned} \tag{15}$$

where $\lambda_l = \lambda_{i_{l-1} i_l}$ and $C_l^m = \prod_{s=1, s\neq l}^m (\lambda_s/\lambda_s - \lambda_l)$ (when $\lambda_s = \lambda_l$, it is obtained from taking limit). From (15), we have

$$F_{i_0 i_m}(t) = \int_0^t \left(\sum_{l=1}^m C_l^m \lambda_l e^{-\lambda_l \tau}\right) d\tau$$

from which one has

$$\lim_{t\to\infty} F_{i_0 i_m}(t) = \int_0^{\infty} \sum_{l=1}^{m} \left(C_l^m \lambda_l e^{-\lambda_l \tau}\right) d\tau$$
$$= \sum_{l=1}^{m} \left(C_l^m \int_0^{\infty} \lambda_l e^{-\lambda_l \tau} d\tau \right)$$
$$= \sum_{l=1}^{m} C_l^m = 1. \quad (16)$$

Remark 3.6: There is at least one path from node v to each node i in a connected graph $\mathcal{G}$, and it follows from (16) that the probability distribution function over a path will be equal to 1 when $t \to \infty$; thus, we have $\lim_{t\to\infty} F_{vi}(t) = 1$ for $i \in \mathcal{V}$. Hence, from Theorem 3.3, we have

$$\Pr\left\{\lim_{t\to\infty} L_i(t) = \tau_v(t), i \in \mathcal{V}\right\} = 1$$

which implies that Theorem 3.3 is equivalent to Theorem 3.2 when $t \to \infty$.

Suppose that there are p different paths from node i to node j in graph $\mathcal{G}$, where p paths do not have the same nodes except nodes i and j. Let $f_{ij}^{\ell}(t), \ell \in \{1, 2, \ldots, p\}$ be the probability density function of that node i successfully delivers a message to node j according to ℓth path between them. Then, define the following recursion:

$$g_{ij}^{\ell}(t) = g_{ij}^{\ell-1}(t) \left(1 - \int_0^t f_{ij}^{\ell}(\tau) d\tau\right) + f_{ij}^{\ell}(t)$$
$$\times \left(1 - \int_0^t g_{ij}^{\ell-1}(\tau) d\tau\right) + g_{ij}^{\ell-1} f_{ij}^{\ell}(t) \quad (17)$$

for $\ell = 1, 2, \ldots, p$, with boundary condition of $g_{ij}^0(t) = 0$.

Lemma 3.7: Suppose that there are p different paths between nodes i and j. Then, $f_{ij}(t) = g_{ij}^p(t)$ for $i,j \in \mathcal{V}$.

To obtain each $F_{vi}(t)$ in (10), we need to first compute $f_{vi}(t)$. Note that, if two nodes j and l in $\mathcal{G}_{vi}$ are connected by p $(p > 1)$ different paths, then $f_{jl}(t)$ can be calculated from Lemma 3.7 where the probability density function is given. Note that the nodes connecting between node j and node l are used to deliver messages from j to l. Thus, each $f_{vi}(t)$ can be also calculated from $f_{jl}(t)$, and we do not have to consider the deliver probability of that nodes connecting between node j and node l. Therefore, we propose the following algorithm to simplify the graph $\mathcal{G}_{vi}$.

Algorithm 2: *A Simplified Graph Method*

1: Let $\mathcal{G}_{sd}(0)$ be a subgraph $\mathcal{G}_{vi}$ of $\mathcal{G}$, where $s = v$ and $d = i$.
2: For graph $\mathcal{G}_{sd}(l-1), l \in N^+$, if there are two or more different paths between i to j for $i,j \in \mathcal{G}_{sd}(l-1)$, then remove the nodes on these paths, except nodes i and j, and built an edge $\{i, j\}$ between nodes i and j.
3: Denote the new graph as $\mathcal{G}_{sd}(l)$.

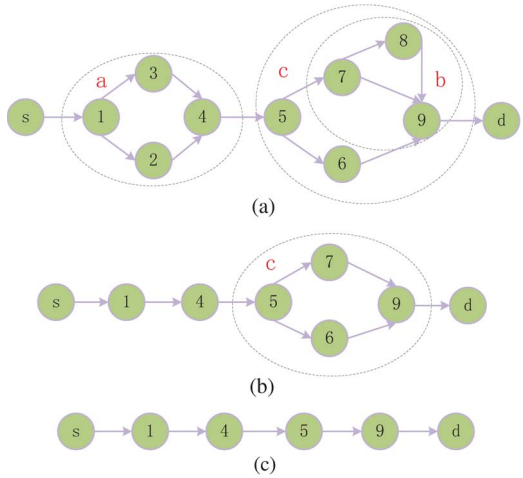


Fig. 3. Example for the simplified graph method. (a) Graph $\mathcal{G}_{sd}(0)$. (b) Graph $\mathcal{G}_{sd}(1)$. (c) Graph $\mathcal{G}_{sd}(2)$.

Definition 3.8: Graph $\mathcal{G}$ is defined as a linearizable graph if all its subgraphs $\mathcal{G}_{ij}$ for $i,j \in \mathcal{V}$ can be simplified as a linear graph by Algorithm 2.

Example 3.9: Given subgraph $\mathcal{G}_{sd}(0)$ as shown in Fig. 3(a), one sees that between nodes 1 and 4 and nodes 7 and 9, there are two different paths, respectively. From step 2, remove nodes 2, 3, and 8, and built edge {1, 4} and {7, 9}. Then, we obtain $\mathcal{G}_{sd}(1)$ as shown in Fig. 3(b). Repeating this process, we obtain $\mathcal{G}_{sd}(2)$ as shown in Fig. 3(c), which is already a linear graph.

In fact, many graphs in practice are linearizable graphs, e.g., a star graph, a linear graph, a tree, and a ring graph. For a linearizable graph, we propose the following algorithm to compute each $F_{vi}(t)$.

Algorithm 3: *Compute* F_{vi} of $\mathcal{G}$

1: For each node $i, i \neq v, i \in \mathcal{V}$, establish the subgraph $\mathcal{G}_{vi}$ of graph $\mathcal{G}$, and let $\mathcal{G}_{sd}(0) = \mathcal{G}_{vi}$.
2: If there are p $(p > 1)$ different paths between x to y for $x, y \in \mathcal{G}_{sd}(l-1)$, compute each $f_{xy}^m(t)$ by using (14) for $m = 1, 2, \ldots, p$ and then compute $f_{xy}(t)$ by using Lemma 3.7 based on (17).
3: Simplify the graph $\mathcal{G}_{sd}(l-1)$ to get graph $\mathcal{G}_{sd}(l)$. If $\mathcal{G}_{sd}(l)$ is not a linear graph, then go to step 2.
4: Compute $f_{sd}(t)$ by using (14); then, the $F_{sd}(t)$ is given by $F_{sd}(t) = \int_0^t f_{sd}(\tau) d\tau$.

In Algorithm 3, each $f_{ij}(t)$ for $\{i, j\}$ in $\mathcal{G}_{sd}(l)$ can be computed by using (14) and Lemma 3.7 based on $\mathcal{G}_{sd}(l-1)$. Note that each $f_{ij}(t)$ for $\{i, j\}$ in $\mathcal{G}_{sd}(0)$ is given, which satisfies $f_{ij}(t) = \lambda_{ij} e^{-\lambda_{ij} t}$. Therefore, by Algorithm 3, we can get F_{vi} for $i \in \mathcal{V}$.

Example 3.10: Assume that $\mathcal{G}_{sd}(0)$ is given as shown in Fig. 3(a). Since $f_{ij}(t)$ is known for each $\{i, j\}$ in $\mathcal{G}_{sd}(0)$, $f_{14}(t)$ and $f_{79}(t)$ can be obtained by using (14) and Lemma 3.7, which means that $f_{ij}(t)$ have been given for each $\{i, j\}$ in $\mathcal{G}_{sd}(1)$. Similarly, by using (14) and Lemma 3.7, we get f_{59}. Hence,

each $f_{ij}(t)$ can be obtained for each $\{i,j\}$ in $\mathcal{G}_{sd}(2)$, and note that $\mathcal{G}_{sd}(2)$ in Fig. 3(c) is a linear graph; thus, one obtains

$$f_{sd}(t) = f_{s1}(t) \otimes f_{14}(t) \otimes f_{45}(t) \otimes f_{59}(t) \otimes f_{9d}(t).$$

Then, $F_{sd}(t)$ is given by $F_{sd}(t) = \int_0^t f_{sd}(\tau)d\tau$.

IV. Two Fundamental Problems

Here, the previously obtained theories and algorithms are applied to answer two fundamental problems: 1) whether better connectivity of relation graph enhances the convergence of the synchronization algorithm; and 2) whether the probability of finite-time convergence can be increased by adding new nodes into the network.

Let $\mathcal{G}$ be a relation graph of a random sensor network. Let $\mathcal{G}_a$ and $\mathcal{G}_b$ be two subgraphs of graph $\mathcal{G}$, and assume that there are same nodes in $\mathcal{G}_a$ and $\mathcal{G}_b$, i.e., $\mathcal{G}_a \subseteq \mathcal{G}$, $\mathcal{G}_b \subseteq \mathcal{G}$, and $\mathcal{V}_a = \mathcal{V}_b$, where $\mathcal{V}_a$ and $\mathcal{V}_b$ are the node set of $\mathcal{G}_a$ and $\mathcal{G}_b$, respectively.

Theorem 4.1: Assume that $\mathcal{G}_a$ and $\mathcal{G}_b$ are connected. Then

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_a\} \leq \Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_b\} \tag{18}$$

if and only if $\mathcal{G}_a \subseteq \mathcal{G}_b$, and the equal sign in (18) holds if and only if $\mathcal{G}_a = \mathcal{G}_b$.

Proof: The proof is given in Appendix B. ■

Remark 4.2: The given theorem proves the fact that the better connectivity of the network is beneficial for the convergence of the algorithm. It should be noticed that the definition of relation graph is critical for the rigorous proof of Theorem 4.1.

From the definition of a relation graph, we know that each edge between any two nodes in the relation graph denotes that there is a positive contact probability between these two nodes. Furthermore, $\mathcal{G}_a \subset \mathcal{G}_b$ means that there are at least two nodes that can make contact with each other in network $\mathcal{G}_b$ but not in $\mathcal{G}_a$. Thus, from Theorem 4.1, we have that increasing the mobility of a node such that it can contact more nodes in the network will increase the convergence speed of the synchronization algorithm. Moreover, by using Theorem 4.1, it gives a tighter lower bound of the probability of finite-time convergence for some relation graphs, e.g., a ring graph. The following example is employed to illustrate the results.

Example 4.3: Consider a ring relation graph $\mathcal{G}_a$ with $2m+1$ nodes, and that the contact parameter between any two neighbor nodes is λ. There exist two nodes i and j, with the shortest length of all paths from node v to both of them is equal to m. Remove edges $\{i,j\}$ and $\{j,i\}$ from $\mathcal{G}_a$ to get a new graph $\mathcal{G}_b$. It is clear to see that $\mathcal{G}_b \subset \mathcal{G}_a$. From Theorem 4.1, we have

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_a\} \geq \Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_b\}.$$

Meanwhile, from Corollary 3.4, we have

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_b\} \geq F_{vi}(t)F_{vj}(t)$$

as $\mathcal{G}_b$ is a tree. In addition, note in $\mathcal{G}_b$ that $F_{vi}(t)$ and $F_{vj}(t)$ satisfy $F_{vi}(t) = F_{vj}(t)$ and

$$\begin{aligned} F_{vi}(t) &= \int_0^t e^{-\lambda\tau}\frac{\lambda^{m+1}\tau^m}{m!}d\tau \\ &= 1 - e^{-\lambda t} - \sum_{l=1}^{m}\frac{\lambda^l t^l e^{-\lambda t}}{l!}. \end{aligned} \tag{19}$$

Hence, we have

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_a\} \geq \left(1 - \sum_{l=0}^{m}\frac{\lambda^l t^l e^{-\lambda t}}{l!}\right)^2 \tag{20}$$

where $0! = 1$. Since the ring graph is a linearizable graph, we can calculate each $F_{vi}(t)$ by Algorithm 3, and then the lower bound can be obtained from Theorem 3.3. However, when m is large, the lower bound obtained from Theorem 3.3 is relative conservative as it decreases exponentially with m, which is much faster than that derived from (20). Obtaining the lower bound from Theorem 3.3 will also have high computational complexity as it needs to calculate each $F_{vi}(t), \forall i, i \in \mathcal{G}$.

We now consider whether the probability of finite-time convergence can be increased by adding new nodes into the network. Assume that there are m new nodes $i_1, i_2, \ldots, i_m$ joining network $\mathcal{G}_a$ to form a new network $\mathcal{G}'_a$. Comparing $\mathcal{G}_a$ and $\mathcal{G}'_a$, a cycle is called a new cycle of $\mathcal{G}'_a$ here only when there are nodes belonging to node set $\{i_1, i_2, \ldots, i_m\}$ in the cycle. Then, if there is a path from nodes i and j for $i, j \in \mathcal{G}_a$ and all the other nodes in this path belong to node set $\{i_1, i_2, \ldots, i_m\}$ in $\mathcal{G}'_a$, then we build a new edge between node i and node j into $\mathcal{G}_a$, and obtain a new graph, which is denoted $\mathcal{G}_b$. By using Theorem 4.1, we can obtain the following result.

Theorem 4.4: Suppose that graph $\mathcal{G}_a$ is connected, and the m new nodes are $i_1, i_2, \ldots, i_m$. If $a_{i_l} < a_v$ holds for $l = 1, 2, \ldots, m$, then

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_a\} < \Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_b\} \tag{21}$$

if and only if that there exist at least one new cycle in $\mathcal{G}'_a$.

Proof: The condition that $a_{i_l} < a_v$ holds for $l = 1, 2, \ldots, m$ guarantees that node v respectively in $\mathcal{G}_a$ and $\mathcal{G}'_a$ are the same node. Clearly, $\mathcal{G}_a \subseteq \mathcal{G}_b$. Furthermore, there is at least one path from nodes i and j for $i, j \in \mathcal{G}_a$ with all the other nodes between nodes i and j belonging to node set $\{i_1, i_2, \ldots, i_m\}$ in $\mathcal{G}'_a$ if and only if there is at least one new cycle in $\mathcal{G}'_a$. Thus, there is $\mathcal{G}_a \subset \mathcal{G}_b$ if and only if there exists at least one new cycle in $\mathcal{G}'_a$. The theorem is then immediately derived from Theorem 4.1. ■

The given theorem gives a necessary and sufficient condition to determine whether adding new nodes can enhance the convergence. In fact, from the theorem, we can see that the positive effect can only be guaranteed when the new joint nodes construct new cycles for the network, which can increase the probability of finite-time convergence. Thus, if the new joint nodes only make contact with one node in the network, i.e., no new cycles can be generated, these new nodes do not help speed up the limit time convergence.

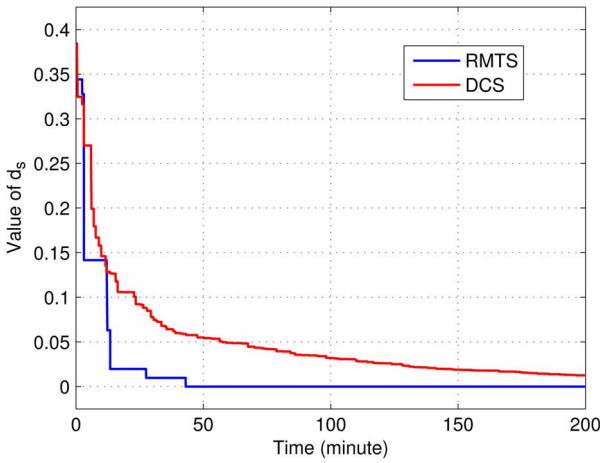


Fig. 4. Comparison d_s between RMTS and DCS with $\lambda = 0.5$.

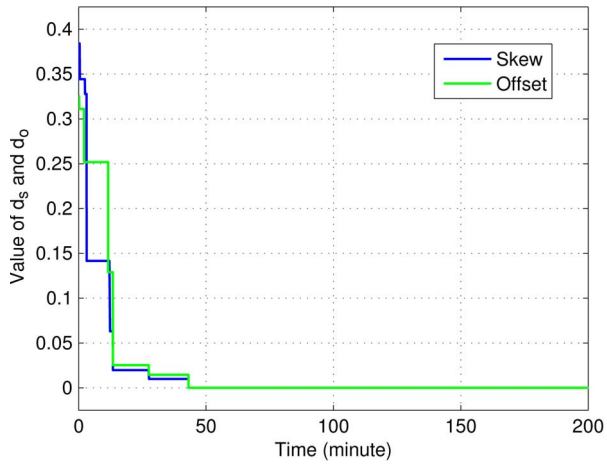


Fig. 5. Comparison between skew and offset using RMTS with $\lambda = 0.5$.

V. Simulation

For the simulation examples, each a_i is randomly selected from set [0.8, 1.2], and offset b_i is randomly selected from set [0, 0.4] following uniform distribution. For the simplicity of showing performance, we define two functions as follows:

$$d_s(t) = \max_{i,j\in\mathcal{V}} \{\hat{a}_i(t)a_i - \hat{a}_j(t)a_j\}$$
$$d_o(t) = \max_{i,j\in\mathcal{V}} \left\{\hat{a}_i(t)b_i + \hat{b}_i(t) - \left(\hat{a}_j(t)b_j + \hat{b}_j(t)\right)\right\} \quad (22)$$

where $d_s(t)$ and $d_o(t)$ denote the maximum difference (error) of the logical skew and offset between any two nodes, respectively. Clearly, the global time synchronization is reached if and only if $d_s(t) = 0$ and $d_o(t) = 0$. Then, we compare our synchronization RMTS with DCS in [13], where DCS is a latest average-consensus-based time synchronization algorithm in DTNs, which is a class of RMSNs. All the results are obtained by Matlab 7.0,

Consider a random mobile WSN with a linear relation graph denoted by $\mathcal{G}_l$ and 30 nodes, and each λ_{ij} of two neighbor nodes is equal to λ, i.e., any pair of neighboring nodes has the same contact probability. For example, 30 nodes are uniformly distributed in a belt-type region, and each node moves randomly in a fixed geographic area to monitor the events. This practical scenario corresponds to the given linear relation graph. Fig. 4 shows the dynamic of d_s under both RMTS and DCS with $\lambda = 0.5$, from which it can be observed that it takes 43 min for RMTS to reach skew synchronization, whereas it takes more than 200 min for DCS to obtain comparable accuracy, i.e., RMTS has much faster convergence speed of skew compensation than that of DCS. Choi *et al.* [13] pointed out that DCS cannot finish the skew and offset compensation simultaneously in most cases. However, in Fig. 5, the skew and offset synchronization can be achieved simultaneously under RMTS, which again supports Theorem 3.2. Therefore, RMTS can synchronize the nodes' logical clocks much faster than DCS.

Next, we compare the convergence time of RMTS and DCS for varying contact probability of neighboring nodes. The convergence bound of RMTS is set as $d_s < 10^{-8}$, and DCS is set as $d_s < 10^{-2}$, where $d_s < 10^{-8}$ and $d_s < 10^{-2}$ are two finite bounds (set a larger bound to DCS as it has much slower convergence speed). By conducting 5000 times of simulations, the average convergence time for RMTS and DCS under different λ are shown in Fig. 6. It is observed that RMTS always converges much faster than DCS in spite of the fact that the bound $d_s < 10^{-8}$ for RMTS is much smaller than $d_s < 10^{-2}$ for DCS. Hence, under different contact probabilities, RMTS has faster convergence speed than DCS. We also compare the energy efficiency of RMTS and DCS. Define the average communication times for each node, i.e., the average number of communication times for each node during the convergence time (where the convergence bounds are also $d_s < 10^{-8}$ and $d_s < 10^{-2}$ for RMTS and DCS, respectively), to be the energy consumption of the algorithms. Larger average communication times mean that it needs more energy for the algorithms to converge. Fig. 7 shows the average communication times for each node of both RMTS and DCS under different λ. It is observed that, for RMTS, the average communication times for each node is about 25, whereas for DCS, it is about 200. Hence, RMTS is much more energy efficient than DCS. Note that two nodes will communicate with each other once they contact, and then they will update their clocks based on the algorithm. Hence, for both RMTS and DCS, different contact probabilities will lead to the different convergence time (see Fig. 6) without affecting the average communication times (see Fig. 7).

From Lemma 3.5, $f_{vi}(t)$ can be computed by (14) for the linear graph. Let node 1 be node v with $a_1 = 1.20$, and let $\lambda = 1$. From (15), we have

$$f_{1,30}(t) = \lambda^m e^{-\lambda t}\frac{t^{m-1}}{(m-1)!} = \frac{t^{29}}{29!}e^{-t} \quad (23)$$

which provides a theoretical value of $f_{1,30}(t)$ for each time t. Note that, if node 30 has obtained the message from node 1, then all nodes in the network have obtained the message from node 1, which means that the synchronization is reached. Thus, for each λ, a theoretical value of the probability of finite-time convergence of RMTS is established from (23), and an associated simulation value is obtained from statistical analysis of 5000 times of simulations. The comparison between simulation

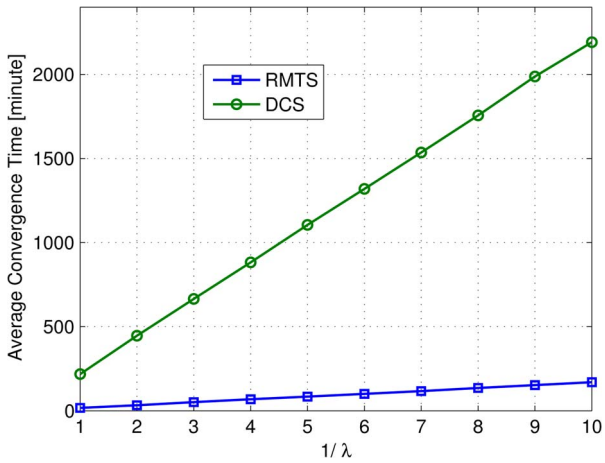


Fig. 6. Comparison of the convergence time of RMTS and DCS under different λ.

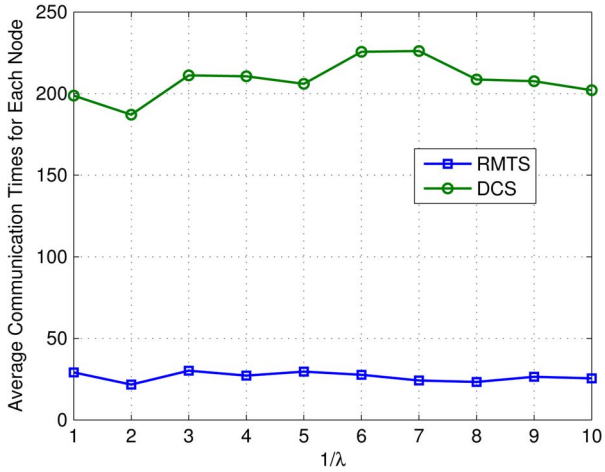


Fig. 7. Comparison of the communication times of RMTS and DCS under different λ.

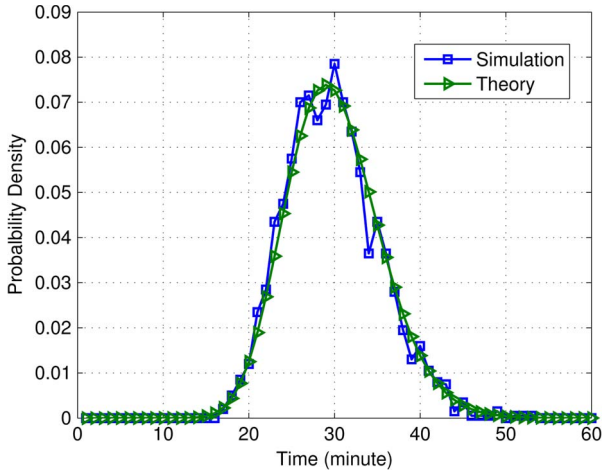


Fig. 8. Probability in simulation and theory with $\lambda = 1$.

and theoretical results is shown in Fig. 8. It is observed that the two values are close, which is the same as the results of Theorem 3.3. Meanwhile, we study the robustness for RMTS with packet loss for this linear relation graph, and the associated simulation results are shown in Fig. 9, which is also obtained from statistical analysis of 5000 times of simulations for each

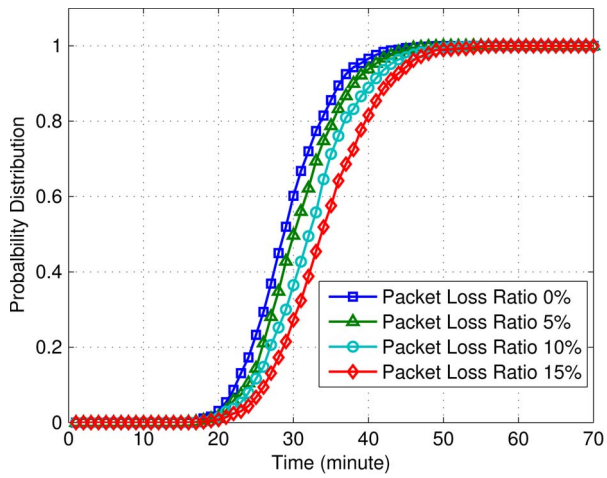


Fig. 9. Comparison the finite-time convergence of RMTS under different packet loss ratios.

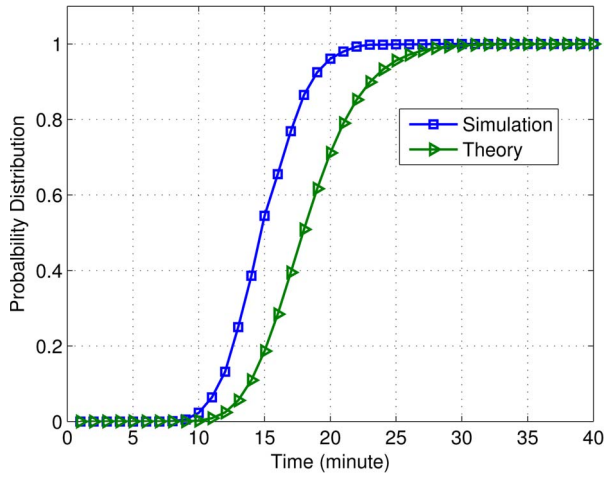


Fig. 10. Probability in simulation and lower bound estimation in theory.

given packet loss ratio. It is obvious that RMTS is robust to packet loss; although for any given a time, the larger packet loss ratio may decrease the probability of finite-time convergence for RMTS.

We now provide simulations to illustrate Theorems 4.1 and 4.4, respectively, for the two fundamental problems given in Section IV. For the given linear graph, we add a new node 31, and assume that node 31 can contact with nodes 1 and 30 with their contact parameters being equal to 1; hence, the new graph $\mathcal{G}_r$ will be a ring graph with $\lambda_{ij} = \lambda = 1$ for $i, j \in \mathcal{V}_r$. This new ring graph can be viewed as a practical scenario where all nodes are uniformly distributed in a ring-type region and where every node moves randomly in a fixed geographic area to monitor the events. From Example 4.3, it follows from (20) that

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_r\} \geq \left(1 - \sum_{l=0}^{15} \frac{t^l e^{-t}}{l!}\right)^2 \qquad (24)$$

which establishes a lower bound of the probability of finite-time convergence of RMTS for $\mathcal{G}_r$. The empirical probability of finite-time convergence of RMTS for $\mathcal{G}_r$ in simulation is obtained from statistical analysis of 5000 times of simulations. The comparison between the empirical results from simulation

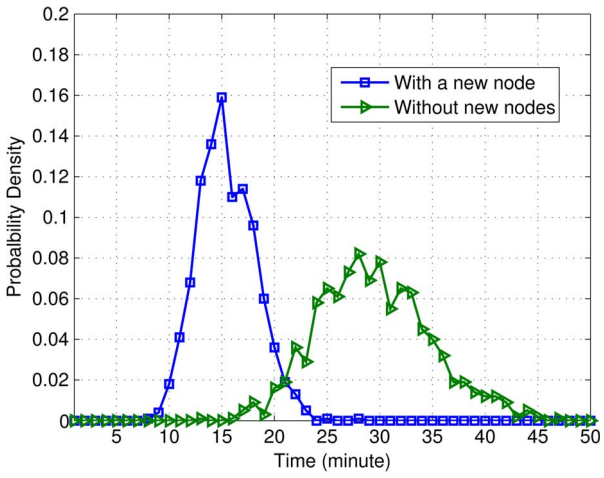


Fig. 11. Compare the convergence time of RMTS for $\mathcal{G}_l$ and $\mathcal{G}_r$.

and the estimated lower bound in theory is shown in Fig. 10. It is observed that the theoretical lower bound is always lower than but close to the empirical results, which illustrates that our proposed method is effective. Comparing $\mathcal{G}_r$ with $\mathcal{G}_l$, there is a new node 31, which constructs a new cycle in $\mathcal{G}_r$. Thus, from Theorem 4.4, the convergence speed of RMTS for $\mathcal{G}_r$ should be faster than that for $\mathcal{G}_l$. Similarly, we give the probability of finite-time convergence of RMTS in simulation, which is obtained from statistical analysis of 5000 times of simulations for both $\mathcal{G}_l$ and $\mathcal{G}_r$. As shown in Fig. 11, it is clear that, under $\mathcal{G}_r$, the average convergence time is about 15 min, whereas under $\mathcal{G}_l$ without the new nodes, the average convergence time is about 30 min. Thus, the average convergence time under $\mathcal{G}_r$ is much faster than that under $\mathcal{G}_l$, which again supports the results of Theorem 4.4.

VI. Conclusion

We have investigated time synchronization for RMSNs in this paper. The MTS protocol in [19] is revised to handle the clock synchronization for the random mobile networks. The protocol has faster convergence speed and achieves both the skew and offset compensations simultaneously. By using a relation graph to model the random mobile networks, we proved the convergence of our algorithm and provided a lower bound of the probability of finite-time convergence. Moreover, for linearizable graphs, an algorithm is developed to compute the probability of finite-time convergence. Extensive simulations demonstrated the effectiveness of our results. Future directions include motion design of sensor nodes and experimental validation of the results.

Appendix A
Proof of Theorem 3.2

Proof: Since $\hat{a}_i(0) = 1$ and $\hat{b}_i(0) = 1$, we have $\hat{a}_i(0)a_i = a_i$ and $\hat{a}_i(0)b_i + \hat{b}_i(0) = b_i$ for $i \in \mathcal{V}$, i.e., the logical clock of every node is the same as its hardware clock at initial time. Meanwhile, without loss of generality, assume node v has the largest logical clock skew. If node i contacts with j at time t, only when $q_{ij} > 1$, node i will update its logical clock such that $\hat{a}_i a_i = \hat{a}_j a_j$ and $\hat{a}_i b_i + \hat{b}_i = \hat{a}_j b_j + \hat{b}_j$. Thus, during the iteration of RMTS, the logical clock skew of each node in the network is less than or equal to $a_{\max}$. Note that, for node $i, i \in \mathcal{V}_v$, we have $\hat{a}_i a_i = a_v = a_{\max}$ and $\hat{a}_i b_i + \hat{b}_i = b_v$. We can infer that $\hat{a}_i a_i \geq \hat{a}_j a_j$ holds for $\forall\, j \in \mathcal{V}$, i.e., $q_{ij} \leq 1$ holds at any time t, which means that node i ($i \in \mathcal{V}_v(t)$) will no longer update its logical clock. Therefore, each node i in $\mathcal{V}_v(t)$ will maintain its logical clock skew and offset during the iteration, which means that $|\mathcal{V}_v(t)|$ is nondecreasing.

Let $\mathcal{V} - \mathcal{V}_v(t)$ be the set of the nodes that are not in $\mathcal{V}_v(t)$, i.e., if node j is not in the set $\mathcal{V}_v(t)$, then node j is in $\mathcal{V} - \mathcal{V}_v(t)$. Since the logical clock skew of each node in the network is less than or equal to $a_{\max}$ during the iteration of RMTS, we have $\hat{a}_j(t)a_j < a_v$ for $\forall\, j \in \mathcal{V} - \mathcal{V}_v(t)$. Thus, if node $i, i \in \mathcal{V}_v(t)$ contacts with node $j, j \in \mathcal{V} - \mathcal{V}_v(t)$ at time t, then it follows from step 5 of RMTS that node j will update its logical clock such that $\hat{a}_j a_j = \hat{a}_i a_i = a_v$ and $\hat{a}_j b_j + \hat{b}_j = \hat{a}_i b_i + \hat{b}_i = \hat{b}_v$. Then, one obtains that $|\mathcal{V}_v(t^+)| = |\mathcal{V}_v(t)| + 1$ and $|\mathcal{V} - \mathcal{V}_v(t^+)| = |\mathcal{V} - \mathcal{V}_v(t)| - 1$, where t^+ is the finish time of the contact between them. Hence, $|\mathcal{V}_v(t)|$ will strictly increase when node i ($i \in V_v(t)$) contacts with node j ($j \in \mathcal{V} - \mathcal{V}_v(t)$).

If $|\mathcal{V}_v(t)| = n$ at time t, then one obtains that all nodes of the network are in $\mathcal{V}_v(t)$; thus, $\hat{a}_i a_i = a_v$ and $\hat{a}_i b_i + \hat{b}_i = b_v$ for $i \in \mathcal{V}$, which implies that (9) holds true. Otherwise, we have $|\mathcal{V}_v(t)| < n$. Since relation graph $\mathcal{G}$ is connected, there is at least one edge $\{i, j\}$ in $\mathcal{G}$ for $i \in \mathcal{V}_v(t)$ and $j \in \mathcal{V} - \mathcal{V}_v(t)$. Thus, we have $\lambda_{ij} > 0$. Clearly, the probability that node i has contacted with node j at time t' for $t' > t$ satisfies $1 - e^{-\lambda_{ij}(t'-t)}$. Then, we have the probability of $|\mathcal{V}_v(t')| \geq |\mathcal{V}_v(t)| + 1$ being equal to $1 - e^{-\lambda_{ij}(t'-t)}$. Thus, $|\mathcal{V}_v(t)|$ will strictly increase with probability 1 when $t \to \infty$. Note that $|\mathcal{V}_v(t)| \leq n$. Hence, we have

$$\Pr\left\{\lim_{t\to\infty} |\mathcal{V}_v(t)| = n\right\} = 1$$

which yields (9). Since after each node i contacting with one node in $\mathcal{V}_v$, it finishes both of its skew and offset compensation. Therefore, the skew and offset compensation for each node $i, i \in \mathcal{V}$ can be finished simultaneously. ■

Appendix B
Proof of Theorem 4.1

Proof: Note that $\mathcal{G}_a \subseteq \mathcal{G}$, $\mathcal{G}_b \subseteq \mathcal{G}$, and $\mathcal{V}_a = \mathcal{V}_b$, and in particular, there are some node v with $a_v = a_{\max}$.

Let the paths from node v to each i, which exist in $\mathcal{G}_b$ but not in $\mathcal{G}_a$, be the new paths of $\mathcal{G}_b$, and the other paths from node v to each node i, which exist in both $\mathcal{G}_b$ and $\mathcal{G}_a$, be the original paths of $\mathcal{G}_b$. Let $e_i(t)$ be the event that node v successfully transmits a message to node i before the time t. Let $e_i^0(t)$ and $e_i^1(t)$, respectively, be the events that node v successfully transmits a message to node i before time t, respectively, through the original and new paths of node i for $i \in \mathcal{V}$. Clearly, $e_i(t) = e_i^0(t) \cup e_i^1(t)$ for $\forall\, i, i \in \mathcal{V}_b$. Then

$$\Pr\{L_i(t) = \tau_v(t), i \in \mathcal{V}_b\} = \Pr\{e_1(t) \cap e_2(t) \cap \cdots \cap e_n(t)\}. \quad (25)$$

Note that $e_i^1(t)=\emptyset$ in $\mathcal{G}_a, \forall\, i, i\in\mathcal{V}_a$, which yields

$$\Pr\left\{L_i(t)=\tau_v(t), i\in\mathcal{V}_a\right\}=\Pr\left\{e_1^0(t)\cap e_2^0(t)\cap\cdots\cap e_n^0(t)\right\}. \tag{26}$$

Hence, it follows that

$$\begin{aligned}
&\Pr\left\{e_1(t)\cap e_2(t)\cap\cdots\cap e_n(t)\right\}\\
&\quad=\Pr\left\{\left[e_1^0(t)\cup e_1^1(t)\right]\cap\left[e_2^0(t)\cup e_2^1(t)\right]\right.\\
&\qquad\left.\cap\cdots\cap\left[e_n^0(t)\cup e_n^1(t)\right]\right\}\\
&\quad\geq\Pr\left\{\left(e_1^0(t)+e_1^1(t)-e_1^0(t)\cap e_1^1(t)\right)\cap\left[e_2^0(t)\cup e_2^1(t)\right]\right.\\
&\qquad\left.\cap\cdots\cap\left[e_n^0(t)\cup e_n^1(t)\right]\right\}\\
&\quad\geq\Pr\left\{e_1^0(t)\cap\left[e_2^0(t)\cup e_2^1(t)\right]\cap\cdots\cap\left[e_n^0(t)\cup e_n^1(t)\right]\right\}\\
&\qquad+\Pr\left(e_1^1(t)-e_1^0(t)\cap e_1^1(t)\right)\\
&\qquad\times\Pr\left(e_2^0(t)\right)\times\cdots\times\Pr\left(e_n^0(t)\right)\\
&\quad\geq\cdots\\
&\quad\geq\Pr\left\{e_1^0(t)\cap e_2^0(t)\cap\cdots\cap e_n^0(t)\right\}+P
\end{aligned}\tag{27}$$

where

$$P=\sum_{i=1}^{n}\left\{\Pr\left(e_i^1(t)-e_i^0(t)\cap e_i^1(t)\right)\prod_{j=1,j\neq i}^{n}\Pr\left(e_j^0(t)\right)\right\}.$$

From the definitions of $e_i^1(t)$ and $e_i^0(t)$ that $e_i^1(t)\not\subseteq e_i^0(t)$ holds when $e_i^1(t)\neq\emptyset$, which means that $e_i^1(t)-e_i^0(t)\cap e_i^1(t)=\emptyset$ iff $e_i^1(t)=\emptyset$. Thus, $\mathcal{G}_a\subset\mathcal{G}_b$ if and only if that there exists node i with $\Pr(e_i^1(t)-e_i^0(t)\cap e_i^1(t))>0$, which means that

$$\begin{aligned}
&\Pr\left\{e_1(t)\cap e_2(t)\cap\cdots\cap e_n(t)\right\}\\
&\qquad>\Pr\left\{e_1^0(t)\cap e_2^0(t)\cap\cdots\cap e_n^0(t)\right\}
\end{aligned}$$

and we have $\mathcal{G}_a=\mathcal{G}_b$ iff $\Pr(e_i^1(t)-e_i^0(t)\cap e_i^1(t))=0$ holds for every node i as $e_i^1(t)=\emptyset$, which means that

$$\begin{aligned}
&\Pr\left\{e_1(t)\cap e_2(t)\cap\cdots\cap e_n(t)\right\}\\
&\qquad=\Pr\left\{e_1^0(t)\cap e_2^0(t)\cap\cdots\cap e_n^0(t)\right\}.
\end{aligned}$$

Therefore, this theorem is obtained immediately by combining both (25) and (26) with the results in (27). ■

REFERENCES

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

[2] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *Proc. ACM MobiHoc*, 2009, pp. 299–308.

[3] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *Proc. IEEE INFOCOM*, 2009, pp. 1422–1430.

[4] U. Lee, S.-Y. Oh, K.-W. Lee, and M. Gerla, "Scalable multicast routing in delay tolerant networks," in *Proc. IEEE ICNP*, 2008, pp. 218–227.

[5] H. Zhou, J. Chen, J. Fan, Y. Du, and K. Sajal, "ConSub: Incentive-based content subscribing in selfish opportunistic mobile networks," *IEEE J. Sel. Area Commun.*, vol. 31, no. 9, pp. 669–679, Sep. 2013.

[6] X. Zhang, "Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking," *IEEE Autom. Control.*, vol. 56, no. 10, pp. 2429–2444, Oct. 2011.

[7] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Autom. Control.*, vol. 57, no. 9, pp. 2609–2614, Oct. 2012.

[8] L. Schenato and F. Fiorentin, "Average timesynch: A consensus-based protocol for time synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, Sep. 2011.

[9] X. Li, W. Shu, M. Li, P. Luo, H. Huang, and M-Y. Wu, "Performance evaluation of vehicle-based mobile sensor networks for traffic monitoring," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1647–1653, May 2009.

[10] B. Sundararaman, U. Buy, and A. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 281–323, May 2005.

[11] R. Parker and S. Valaee, "Vehicular node localization using received-signal-strength indicator," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3371–3380, Nov. 2007.

[12] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *Proc. IEEE CDC*, 2006, pp. 4915–4920.

[13] 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, Mar. 2012.

[14] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *Proc. IEEE INFOCOM*, 2004, pp. 564–574.

[15] N. M. Freris, H. Kowshik, and P. R. Kumar, "Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation," *Proc. IEEE*, vol. 98, no. 11, pp. 1828–1846, Nov. 2010.

[16] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. ACM OSDI*, 2002, pp. 147–163.

[17] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. ACM SenSys*, 2003, pp. 138–149.

[18] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. ACM SenSys*, 2004, pp. 39–49.

[19] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Autom. Control.*, vol. 59, no. 3, pp. 660–675, Mar. 2014.

[20] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental limits on synchronizing clocks over networks," *IEEE Autom. Control.*, vol. 56, no. 6, pp. 1352–1364, Jun. 2011.

[21] C. Lenzen, T. Locher, and R. Wattenhofer, "Clock synchronization: Open problems in theory and practice," in *Proc. ACM SofSem*, 2010, pp. 61–70.

[22] J. Chen, Q. Yu, Y. Zhang, H-H. Chen, and Y. Sun, "Feedback based clock synchronization in wireless sensor networks: A control theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2963–2973, Jul. 2010.

[23] M. Leng and Y.-C. Wu, "On clock synchronization algorithms for wireless sensor networks under unknown delay," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 182–190, Jan. 2010.

[24] 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.

[25] N. Marechal, J. Pierrot, and J. Gorce, "Fine synchronization for wireless sensor networks using gossip averaging algorithms," in *Proc. IEEE ICC*, 2008, pp. 4963–4967.

[26] S. Philipp and W. Roger, "Gradient clock synchronization in wireless sensor networks," in *Proc. IPSN*, 2009, pp. 37–48.

[27] R. Carli, E. Elia, and S. Zampieri, "A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks," in *Proc. IEEE CDC-ECC*, 2011, pp. 7512–7517.

[28] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri, "A randomized linear algorithm for clock synchronization in multi-agent systems," in *Proc. IEEE CDC*, 2012, pp. 20–25.

[29] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

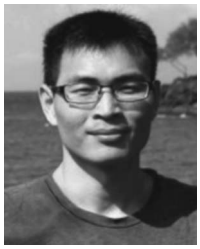
[30] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[31] C. Liao and P. Barooah, "Clock skew and offset estimation from relative measurements in mobile networks with Markovian switching topology," *Automatica*, vol. 49, no. 10, pp. 3015–3022, 2013.

[32] 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.

[33] D. Zhang, T. He, Y. Liu, and Y. Gu, "Poster: neighbor discovery with distributed quorum system," in *Proc. ACM SenSys*, 2011, pp. 369–370.

[34] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 6103–6107, Nov. 2012.



Jianping He received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China.

He is currently a member of the Networked Sensing and Control Group and a Postdoctoral Research Fellow with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China. His research interests include time synchronization, consensus, distributed optimization, and security problems in wireless sensor networks.

Peng Cheng (M'10) received the B.E. degree in automation and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively.

He is currently with the State Key Laboratory of Industrial Control Technology and an Associate Professor with the Department of Control Science and Engineering, Zhejiang University. His research interests include networked sensing and control, cyber–physical systems, and robust control.

Dr. Cheng served as the Publicity Cochair for the 2013 10th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. He currently serves as an Associate Editor for the *International Journal of Distributed Sensor Networks* and the *International Journal of Communication Systems*.

Jiming Chen (M'08–SM'11) received the B.Sc. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2000 and 2005, respectively.

In 2006, he was a Visiting Researcher with the Institut National de Recherche en Informatique et en Automatique (INRIA), Paris, France, in 2007; with the National University of Singapore, Singapore; and from 2008 to 2010, with the University of Waterloo, Waterloo, ON, Canada. He is currently a Full Professor with the Department of Control Science and Engineering and a Coordinator with the Networked Sensing and Control Group, State Key Laboratory of Industrial Control Technology, Zhejiang University. His research interests include estimation and control over sensor networks, sensor and actuator networks, and coverage and optimization in sensor networks.

Dr. Chen served as the Technical Program Committee (TPC) Cochair for the Ad hoc and Sensor Network Symposium of the 2011 IEEE Conference on Global Communications; as the General Cochair for the 2009 and 2010 Association for Computing Machinery IWCMC International Wireless Communications and Mobile Computing Conference; as the Medium Access Control Track Cochair for the 2010 International Wireless Internet Conference; as the Publicity Cochair for the 2011 IEEE International Conference on Mobile Ad Hoc and Sensor Systems (IEEE MASS), the 2011 IEEE International Conference on Distributed Computing in Sensor Systems, and the 2012 International Conference on Distributed Computing Systems (IEEE ICDCS); and as a TPC member for the 2010 and 2012 IEEE ICDCS, the 2010 IEEE MASS, the 2011 IEEE International Conference on Sensor and Ad Hoc Communications and Networks, and the 2011 and 2012 IEEE Conference on Computer Communications. He has served as a Guest Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Computer Communication* (Elsevier), *Wireless Communication and Mobile Computer* (Wiley), and the *Journal of Network and Computer Applications* (Elsevier). He currently serves as Associate Editor for several international journals, including the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE NETWORKS.

Ling Shi received the B.S. degree in electrical and electronic engineering from Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2002 and the Ph.D. degree in control and dynamical systems from California Institute of Technology, Pasadena, CA, USA, in 2008.

He is currently an Assistant Professor with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. His research interests include networked control systems, wireless sensor networks, and distributed control.

Rongxing Lu (S'09–M'11) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2006 and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012.

From May 2012 to April 2013, he was a Postdoctoral Fellow with the University of Waterloo. Since May 2013, he has been an Assistant Professor with the Division of Communication Engineering, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His research interests include wireless network security, big data security and privacy, network coding security, and applied cryptography.