Cluster-Based Consensus Time Synchronization for Wireless Sensor Networks

Jie Wu, Liyi Zhang, Yu Bai, and Yunshan Sun

***Abstract*— This paper proposes a new time synchronization algorithm for wireless sensor networks, named clustered consensus time synchronization (CCTS). This algorithm is developed on the base of the distributed consensus time synchronization (DCTS) algorithm. However, to obtain faster convergence in the clock synchronization of node and better energy efficiency, the clustering technique is incorporated into the algorithm. The CCTS includes two parts: 1) intracluster time synchronization and 2) intercluster time synchronization. In the intracluster time synchronization, the improved DCTS is applied. The cluster head is responsible for exchanging messages within the cluster. The average value of skew compensation parameters of intracluster virtual clock and the average value of intracluster virtual clocks are used to update the skew compensation parameter and offset compensation parameter, respectively. In the intercluster time synchronization, cluster heads exchange messages via gateway nodes. To update the clock compensation parameters of the network virtual clocks, clock compensation parameters of intracluster virtual clocks of every cluster head are assigned with corresponding weights based on the size of each cluster. The simulation results show that the proposed algorithm reduces the communication traffic compared with the DCTS algorithm, and improves the convergence rate due to the combination of clustering topologies.**

***Index Terms*— Clock offset, clock skew, clustering, time synchronization, wireless sensor networks.**

I. INTRODUCTION

WIRELESS sensor networks (WSN) have drawn much attention from the academia to industry in recent years due to their broad range of applications [1], [2], such as object localization or tracking, environmental monitoring, scientific exploration in dangerous environments and so forth [3], [4]. In particular, many applications require global time synchronization, that is, all the nodes of the network achieve and maintain a common notion of time [5].

For most sensor network applications, the data sensed are meaningless and useless without time and space information of an event. The accurate time synchronization is required in many circumstances in network applications. For example, media access control using TDMA needs accurate time information, so that transmissions do not interfere; the sleep scheduling needs nodes to sleep and wake up at the same time. In addition, some special applications such as mobile target localization and event detection also need synchronization.

A variety of centralized and distributed synchronization protocols and algorithms have been proposed based on different applications since 2002, when J. Elson [6], for the first time, proposed sensor network time synchronization at HotNets-I International Conference.

Centralized synchronization protocols, such as *Reference-Broadcast Synchronization* (RBS) [7], *Timing-sync Protocol for Sensor Networks* (TPSN) [8], and *Flooding Time Synchronization Protocol* (FTSP) [9], need a reference node or root node to provide a reference time for the whole network. This kind of protocol usually has a fast convergence rate and little synchronization error. But due to the hierarchical structure, a centralized synchronization protocol has to deal with a WSN's dynamic topology caused by new nodes joining and failure. Therefore, centralized synchronization protocols are often designed with complexity logic along with the high computational complexity, poor robustness, and large energy consumption. Other disadvantages of centralized synchronization protocols include their cumulative errors and unbalanced network synchronization accuracy [10].

Distributed synchronization protocols, such as *Time-Diffusion Synchronization Protocol* (TDP) [11], *Average TimeSync* (ATS) [12], and *Consensus Clock Synchronization* (CCS) [13], use local information to achieve whole network synchronization. The convergence rate of this type of protocols, which rely on the network topology, is unfavorably low compared to centralized synchronization protocols. But because the nodes are not hierarchical, they can easily adapt to a WSN's dynamic topology. In this paper, we choose distributed synchronization protocols to achieve time synchronization of WSN.

Distributed Consensus Time Synchronization (DCTS) algorithms is one kind of distributed time synchronization algorithm, including ATS, CCS, *Global Clock Synchronization* (GCS) [14], *Distributed Time Synchronization Protocol* (DTSP) [15], *Gradient Time Synchronization Protocol* (GTSP) [16], *Energy-Efficient Gradient Time Synchronization Protocol* (EGTSP) [17], and *Second Order Distributed Consensus Time Synchronization* (SO-DCTS) [18]. Since it does not require reference nodes and nodes in the

Manuscript received September 10, 2014; accepted October 12, 2014. Date of publication October 16, 2014; date of current version December 11, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61340034, and the Research Program of Application Foundation and Advanced Technology of Tianjin, China under Grant 13JCYBJC15600. The associate editor coordinating the review of this paper and approving it for publication was Dr. Nitaigour P. Mahalik.

J. Wu and Y. Bai are with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: wuqufuhua@gmail.com; tjubaiyu@gmail.com).

L. Zhang and Y. Sun are with the College of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China (e-mail: zhangliyi@tjcu.edu.cn; sunyunshan@tjcu.edu.cn).

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/JSEN.2014.2363471

network maintain a consensus virtual time, such algorithms have good scalability and robustness. However, some disadvantages of DCTS related to the fundamental problem of clock synchronization have been pointed out [14], [16]. The DCTS has a high communication traffic and its convergence rate is slower than that of centralized synchronization protocol, because the constraints of network topology and the properties of synchronization method are based on diffusion and iteration.

Clustering algorithms have been widely used in WSNs [19], [20]. Based on the clustering topology, the network is divided into several clusters. Each cluster contains a cluster-head and a number of cluster member nodes, the cluster-head is elected according to certain mechanisms and it is responsible for the information collection, the data integration, and the inter-cluster communication. Therefore, clustering topology reduces the communication traffic between the nodes and thus reduces the energy consumption and prolongs the network lifetime.

In this paper, taking a hybrid approach, we incorporate clustering techniques into distributed consensus algorithms, and propose the *Clustered Consensus Time Synchronization* (CCTS) algorithm. In this algorithm, the network is divided into overlapping clusters and the time synchronization process is divided into intra-cluster time synchronization and inter-cluster time synchronization.

The rest of the paper is organized as follows. In section II, we formulate the problem and describe the graph theory and consensus algorithm. Section III introduces a model for the clock. In section IV, we propose the Clustered Consensus Time Synchronization algorithm. The performance of the proposed algorithm is analyzed and verified with simulation results in section V. In section VI, we draw conclusions and discuss the direction of future work.

II. Graphs and Consensus Algorithm

The topology of wireless sensor networks can be represented by the graph theory. Let $G = (V, E, A)$ be an undirected network, where $V = \{1, 2, \ldots, n\}$ represents the set of nodes, n is the number of nodes. $E \subseteq V \times V$ represents set of edges between the nodes, i.e., $(i, j) \in E$ if node j can send message to node i. $A = [a_{ij}]$ is a weighted adjacency matrix with nonempty adjacency elements a_{ij}. $a_{ij} > 0$ if and only if there is a directed edge in G, i.e., $(i, j) \in E$. The set of neighbors of node i is denoted by $D_i = \{j \in V : (i, j) \in E\}$, where $d_i = |D_i|$ is the degree of node i. Let $x_i \in \mathrm{R}^{\mathrm{n}}$ be the state of node i, then the linear model of consensus algorithm [21] is given by

$$x_i(k+1) = x_i(k) + \sum_{j \in D_i} a_{ij}\ [x_j(k) - x_i(k)] \tag{1}$$

Where k is the iteration rounds. In practice, the nodes cannot know the states of all the other nodes. However, the consensus algorithm is an iterative process and through exchanging the messages with neighbors the nodes can reach a consensus state [21], that is, $lim_{t\to\infty} x_i(t) = \frac{1}{n}\sum_{j=1}^{n} x_j(t_0)$, where $x(t_0)$ is the initial state of node. Therefore, if the update equations of clock in proposed algorithm satisfy the model of the consensus algorithm as shown in (1), the clocks of nodes will reach a consensus clock.

In order to make the proposed algorithm satisfy the convergence conditions of the consensus algorithm described in [21], we assume wireless sensor networks to have the following characteristics: communications between nodes are bidirectional, i.e., if $(i, j) \in E$, then $(j, i) \in E$; the graph is connected, that is, there is a path connecting any two arbitrary nodes of the graph. However, since the communication quality depends on many environmental conditions and energy reserves, without corresponding solutions the symmetric communication links are rare or at least not reliable. In order to deal with the problem of asymmetric links, we assume that all the nodes can use power control to vary the amount of transmit power, that is, the nodes can adjust the transmission range to reach the other nodes in the cluster if needed. These assumptions are reasonable due to technological advances in radio hardware and this power control is also used in some power aware protocols [22]–[24].

III. Clock Model

The timing system of WSNs is composed of an oscillator and a counter, the crystal oscillator generates a pulse signal and the counter controls clock frequency. Based on the oscillator's angular frequency, the counter increases its value to represent the local clock $C(t)$ of a node. The local clock at absolute time t is defined as the same form proposed in [13], which is given by

$$C(t) = \epsilon \int_{t_0}^{t} \omega(\tau)d\tau + c(t_0) \tag{2}$$

Where ω is the angular frequency of the oscillator, ϵ is a proportional coefficient that translates the timer counter ticks into the clocks unit of time. The initial clock $c(t_0)$ is the clock reading at time t_0. For the convenience of description, these clocks appear as a simple time varying function, then the first order model of the local clock of node i proposed in [13] is given by

$$C_i(t) = \alpha_i t + \beta_i \tag{3}$$

Where α_i and β_i are respectively the skew and offset of the node i relative to the absolute time. The skew α_i determines how much time the clock will gain or lose over a given period, it depends on the hardware manufacturing process as well as on ambient effects such as temperature, pressure, and the circuit voltage. For a perfect clock the skew is 1. The offset β_i represents clock difference between the initial value and the absolute time and it is primarily caused by inaccuracy of the first setting time, the ideal offset is 0.

The α_i of each node is different, so the nodes' clocks will be inconsistent. Although the nodes' clocks were calibrated at a certain time, since the skews of nodes are not static, they will deviate from each other over time [25]. However, in order to maintain the continuity of the local clock, we do not directly modify the local clock, but establish the virtual clocks according to the local clock via the clock compensation parameters which include skew and offset compensation parameter respectively. So the time synchronization algorithm should be performed periodically in order to keep clock compensation parameters up to date. The virtual clock of node i is given by

$$\hat{C}_i(t) = f(\hat{s}_i, \hat{o}_i) \tag{4}$$

Where $\hat{s}_i$ and $\hat{o}_i$ are the skew and offset compensation parameters of the virtual clock of node i respectively. In this paper, we establish the intra-cluster virtual clock and the network virtual clock in intra-cluster time synchronization and inter-cluster time synchronization respectively.

Consensus time synchronization algorithm is not designed to synchronize the clocks of all the nodes to a reference node or to the absolute time, but through the periodic iteration to update clock compensation parameters of the virtual clocks of nodes and let virtual clocks reach a consensus clock.

$$C^c(t) = \alpha^c t + \beta^c \tag{5}$$

Where α^c and β^c are the skew and the offset of the consensus clock relative to the absolute time.

Therefore, this paper uses the consensus algorithm to achieve the virtual clocks of all the nodes synchronized by updating the clock compensation parameters, that is

$$\lim_{t\to\infty} \hat{C}_i(t) = C^c(t) \tag{6}$$

DCTS algorithm uses the consensus algorithm to synchronize the virtual clocks of nodes, but cannot avoid the drawback of slower convergence rate and larger communication traffic. In this paper, these problems are overcome by applying the improved DCTS in intra-cluster time synchronization. In order to reduce the communication traffic, the cluster-head serves as the routing node which is responsible for intra-cluster communication. The average value of skew compensation parameters of intra-cluster virtual clocks and the average value of intra-cluster virtual clocks are used to update the clock compensation parameters. In the inter-cluster time synchronization, cluster-heads exchange messages via the gateway nodes. In order to accelerate the convergence rate, the clock compensation parameters of intra-cluster virtual clocks of each individual cluster are assigned with corresponding weights. These weights are based on the size of each cluster and are used to update the clock compensation parameters of the network virtual clock.

IV. Clustered Consensus Time Synchronization Algorithm

Sensor nodes are randomly distributed within the target area and form a cluster-based network by Low-energy Adaptive Clustering Hierarchy (LEACH) [26], a protocol architecture for microsensor networks. LEACH includes a distributed cluster formation technique that enables self-organization of large numbers of nodes, and algorithms for adapting clusters and rotating cluster-head election to evenly distribute the energy load among all the nodes. After the cluster-heads are elected, the clusters overlap each other through the power control. Nodes in the overlapping region are called overlap-nodes.

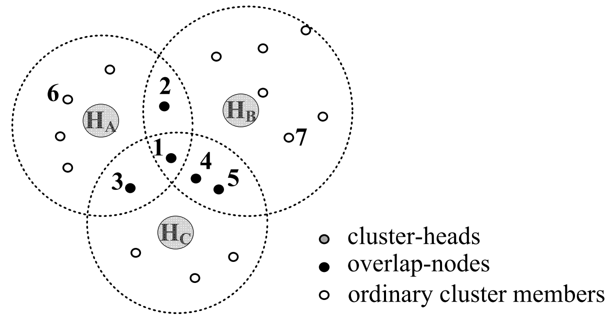

Fig. 1. Schematic illustration of the local topology of a network.

The gateway nodes which are responsible for the message exchange between clusters are elected from the overlap-nodes.

Each node is assigned with an ID number for identification. After the formation of the network, cluster-heads broadcast a message containing their own ID numbers to their cluster member nodes to notice them the identity of their cluster-heads. Cluster member nodes receive the messages and record the ID numbers. Nodes recording only one ID number mark themselves as the ordinary cluster member nodes, whereas nodes at the overlapping regions record more than one ID numbers and exhibit themselves as the overlap-nodes. After receiving the message from the cluster-head, cluster member nodes respond with a message containing their own ID numbers and the ID numbers of their cluster-heads which they can communicate with.

After receiving the response messages, cluster-heads check the ID numbers of other cluster-heads with whom communicate through overlap-nodes and mark them as the neighboring cluster-heads, and then the gateway nodes are elected from these overlap-nodes. When different overlap-nodes can communicate with the same neighboring cluster-head, the overlap-node whose message arrives first is elected as the gateway node only.

The local topology of the network is schematically illustrated in Fig.1. Clusters A, B, and C mutually overlap one to another, and nodes 1 to 5 represent overlap-nodes shown as closed circles. Cluster-heads (H) and ordinary cluster member nodes are represented by shade circles and small open circles respectively. In Fig.1, cluster B and C are the neighboring clusters, and nodes 1, 4, and 5 are their overlap-nodes. Assuming overlap-nodes 1, 4, and 5 have received the message from cluster-head B, then after they receive the message from cluster-head C, they will respond to cluster-head C a message containing the ID number of the cluster-head B. If the message of node 4 is first to arrive, then node 4 is elected as the gateway node between cluster B and C, and cluster-head C will ignore the messages from node 1 and 5. In order to respond to message loss, an ordinary cluster member node will send a message to cluster-head if it does not receive the synchronization messages in one period [27]. However, if there is no response message from cluster-head, then the ordinary cluster member node will announce itself be a new cluster-head and send a message to its neighbor nodes in the next synchronous period.

The first stage of the algorithm is the intra-cluster time synchronization. Cluster-heads calculate the average values of the skew compensation parameters of intra-cluster virtual clocks of nodes within their clusters and the average values of intra-cluster virtual clocks of nodes, and then they update the clock compensation parameters of intra-cluster virtual clocks and simultaneously broadcast them to the neighboring nodes. Cluster member nodes receive the messages and update the local intra-cluster virtual clock compensation parameters to achieve the synchronization of intra-cluster virtual clocks.

The second stage of the algorithm is the inter-cluster time synchronization. The cluster-heads exchange their intra-cluster virtual clocks and their clock compensation parameters through gateway nodes. The received messages are given corresponding weights according to the size of each cluster. Then cluster-heads update skew and offset compensation parameters of network virtual clocks in order to achieve the synchronization of network virtual clocks.

A. Intra-Cluster Time Synchronization

According to (4) and the local clock model which is given by (3), each node within the cluster establishes the intra-cluster virtual clock, which is given by

$$\hat{C}_i^{v}(t) = \hat{s}_i^{v} C_i(t) + \hat{o}_i^{v} \tag{7}$$

Where $\hat{s}_i^{v}$ and $\hat{o}_i^{v}$ are the skew and offset compensation parameters respectively of intra-cluster virtual clock of node i. Substituting (3) into (7), we have

$$\hat{C}_i^{v}(t) = \hat{s}_i^{v}\alpha_i t + \hat{s}_i^{v}\beta_i + \hat{o}_i^{v} = \alpha_i^{v} t + \beta_i^{v} \tag{8}$$

Where $\alpha^{v} = \hat{s}^{v}\alpha$ and $\beta^{v} = \hat{s}^{v}\beta + \hat{o}^{v}$ are respectively the skew and the offset of intra-cluster virtual clock of node relative to the absolute time.

Cluster-heads H_m start the synchronization process of intra-cluster, where $m \in (1, M)$ is the label of cluster-head and M is the number of clusters. The cluster-head sends synchronization message which contains its own ID number, the local clock, the intra-cluster virtual clock, and the skew compensation parameter to the cluster member nodes and records the sending time stamping of local clock $C_m(t)$. Upon receiving this synchronization message, cluster member nodes record the receiving time stamping of local clock $C_i(t)$, intra-cluster virtual clock, and skew compensation parameters and reply this information to the cluster-head. Based on $\hat{s}_i^{v}$ and $\hat{C}_i^{v}(t)$, the cluster-head m calculates $\bar{\hat{s}}_m^{ave}$, the average value of intra-cluster virtual clock skew compensation parameters of cluster member nodes and $\bar{\hat{C}}_m^{ave}$, the average value of intra-cluster virtual clocks.

$$\bar{\hat{s}}_m^{ave}(t) = \frac{1}{N_m}\sum_{i=1}^{N_m} \hat{s}_i^{v}(t) \tag{9}$$

$$\bar{\hat{C}}_m^{ave}(t) = \frac{1}{N_m}\sum_{i=1}^{N_m} \hat{C}_i^{v}(t) \tag{10}$$

Where N_m is the number of cluster member nodes in cluster m, $i = 1, 2, \ldots, N_m$ is the label of the cluster member nodes. To distinguish the label of cluster-head from that of cluster member nodes, we designate the label of cluster-head as $i = N_m + 1$. It should be noted that there is an implicit assumption, that is, the process of messages exchange is instantaneous and the time delay can be ignored. Therefore, this paper exploits the MAC layer message exchange mechanism whose time-stamping in the physical layer will satisfy the above assumption, and has been used in most of the time synchronization protocols [7], [12], [28].

1) Skew Compensation in Intra-Cluster Time Synchronization: As the absolute time t is unknown, it is impossible to calculate α_i and β_i of the local clock given by (3). However, it is still possible to obtain α_{ij} the relative skew of two clocks [12].

$$\alpha_{ij}(k+1) = \frac{\alpha_j(k+1)}{\alpha_i(k+1)} = \frac{C_j(k+1) - C_j(k)}{C_i(k+1) - C_i(k)} \tag{11}$$

Where $\alpha_{ij}(k) = \frac{1}{\alpha_{ji}(k)}$.

Based on $\bar{\hat{s}}_m^{ave}$, cluster-heads update the intra-cluster virtual clock skew compensation parameters with

$$\hat{s}_m^{v}(k+1) = (1-\rho_s)\hat{s}_m^{v}(k) + \rho_s \bar{\hat{s}}_m^{ave}(k)\alpha_{mi}(k) \tag{12}$$

The cluster-heads broadcast intra-cluster virtual clock skew compensation parameters to cluster member nodes, then cluster member node i updates its own intra-cluster virtual clock skew compensation parameter with

$$\hat{s}_i^{v}(k+1) = (1-\rho_s')\hat{s}_i^{v}(k) + \rho_s'\hat{s}_m^{v}(k)\alpha_{im}(k) \tag{13}$$

Where $\rho_s \in (0, 1)$ and $\rho_s' \in (0, 1)$ are the weight parameters. Equations (12) and (13) are the update equations of skew compensation parameters of the intra-cluster virtual clocks.

In order to prove skews compensated satisfy the consensus algorithm, we show the verification processes as follows:

Substituting (9) into (12), we have

$$\hat{s}_m^{v}(k+1) = (1-\rho_s)\hat{s}_m^{v}(k) + \frac{\rho_s}{N_m}\sum_{i=1}^{N_m}\hat{s}_i^{v}(k)\alpha_{mi}(k) \tag{14}$$

By multiplying α_m at both sides of (14), we have

$$\begin{aligned}
\hat{s}_m^{v}(k+1)\alpha_m &= (1-\rho_s)\hat{s}_m^{v}(k)\alpha_m + \frac{\rho_s}{N_m}\sum_{i=1}^{N_m}\hat{s}_i^{v}(k)\alpha_{mi}(k)\alpha_m \\
\alpha_m^{v}(k+1) &= (1-\rho_s)\alpha_m^{v}(k) + \frac{\rho_s}{N_m}\sum_{i=1}^{N_m}\alpha_i^{v}(k) \\
&= \alpha_m^{v}(k) + \frac{\rho_s}{N_m}\sum_{i=1}^{N_m}[\alpha_i^{v}(k) - \alpha_m^{v}(k)]
\end{aligned} \tag{15}$$

By multiplying α_i at both sides of (13), we have

$$\begin{aligned}
\hat{s}_i^{v}(k+1)\alpha_i &= (1-\rho_s')\hat{s}_i^{v}(k)\alpha_i + \rho_s'\hat{s}_m^{v}(k)\alpha_{im}(k)\alpha_i \\
\alpha_i^{v}(k+1) &= (1-\rho_s')\alpha_i^{v}(k) + \rho_s'\alpha_m^{v}(k) \\
&= \alpha_i^{v}(k) + \rho_s'[\alpha_m^{v}(k) - \alpha_i^{v}(k)]
\end{aligned} \tag{16}$$

By (15) and (16), the skews of intra-cluster virtual clocks satisfy the consensus algorithm as shown in (1). So after iterative updates, the skews of intra-cluster virtual clocks of nodes will converge to consensus.

2) Offset Compensation in Intra-Cluster Time Synchronization: After the skew compensation is applied, we can suppose that the intra-cluster virtual clocks of nodes have the same skew. By (8), the intra-cluster virtual clock of node i can be rewritten as

$$\hat{C}_i^v(t) = \alpha^v t + \frac{\alpha^v}{\alpha_i}\beta_i + \hat{o}_i^v = \alpha^v t + \beta_i^v \tag{17}$$

Where $\beta_i^v = \frac{\alpha^v}{\alpha_i}\beta_i + \hat{o}_i^v$ is the offset of intra-cluster virtual clock of node i.

Based on $\hat{C}_m^{ave}$, cluster-head m updates the intra-cluster virtual clock offset compensation parameter with

$$\hat{o}_m^v(k+1) = \hat{o}_m^v(k) + \rho_o[\bar{\hat{C}}_m^{ave}(k) - \hat{C}_m^v(k)] \tag{18}$$

Cluster-head m broadcasts intra-cluster virtual clock to cluster member nodes, cluster member node i updates its own intra-cluster virtual clock offset compensation parameter with

$$\hat{o}_i^v(k+1) = \hat{o}_i^v(k) + \rho_o'[\hat{C}_m^v(k) - \hat{C}_i^v(k)] \tag{19}$$

Where $\rho_o \in (0,1)$ and $\rho_o' \in (0,1)$ are the weight parameters. Equations (18) and (19) are the update equations of offset compensation parameters of the intra-cluster virtual clocks.

In order to prove offsets compensated satisfy the consensus algorithm, we show the verification processes as follows:

Substituting (10) into (18), we have

$$\hat{o}_m^v(k+1) = \hat{o}_m^v(k) + \frac{\rho_o}{N_m}\sum_{i=1}^{N_m}[\hat{C}_i^v(k) - \hat{C}_m^v(k)] \tag{20}$$

By (17), we have $\hat{C}_i^v(k) - \hat{C}_m^v(k) = \beta_i^v(k) - \beta_m^v(k)$, and since $\beta_m^v = \frac{\alpha^v}{\alpha_m}\beta_m + \hat{o}_m^v$, then

$$\begin{aligned}\frac{\alpha^v}{\alpha_m}\beta_m + \hat{o}_m^v(k+1) &= \frac{\alpha^v}{\alpha_m}\beta_m + \hat{o}_m^v(k) \\ &\quad + \frac{\rho_o}{N_m}\sum_{i=1}^{N_m}[\beta_i^v(k) - \beta_m^v(k)] \\ \beta_m^v(k+1) &= (1-\rho_o)\beta_m^v(k) + \frac{\rho_o}{N_m}\sum_{i=1}^{N_m}\beta_i^v(k) \\ &= \beta_m^v(k) + \frac{\rho_o}{N_m}\sum_{i=1}^{N_m}[\beta_i^v(k) - \beta_m^v(k)] \end{aligned} \tag{21}$$

By (19), we have

$$\begin{aligned}\frac{\alpha^v}{\alpha_i}\beta_i + \hat{o}_i^v(k+1) &= \frac{\alpha^v}{\alpha_i}\beta_i + \hat{o}_i^v(k) + \rho_o'[\beta_m^v(k) - \beta_i^v(k)] \\ \beta_i^v(k+1) &= (1-\rho_o')\beta_i^v(k) + \rho_o'\beta_m^v(k) \\ &= \beta_i^v(k) + \rho_o'[\beta_m^v(k) - \beta_i^v(k)] \end{aligned} \tag{22}$$

By (21) and (22), the offsets of intra-cluster virtual clocks of nodes satisfy the consensus algorithm as shown in (1). Therefore, after iterative updates, the offsets of intra-cluster virtual clocks of nodes will converge to consensus.

The matrix form of iterative process shown as (15), (16), (21), and (22) is $\mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k)$, where $\mathbf{x}(k)$ is the state matrix of the k-th iteration. $\mathbf{W} = (w_{ij})$ is the update weight matrix. Since (15) and (21) have the same form, then let $\rho_s = \rho_o = \rho$. In (15) and (21), $1-\rho$ and $\frac{\rho}{N_m}$ are respectively the update weights of cluster-head m and that of cluster member node i in the clock parameter update equations of cluster-head. Given that the update weights of each node are equal, we have

$$1-\rho = \frac{\rho}{N_m} \Rightarrow \rho = \frac{N_m}{N_m+1} \Rightarrow w_{ij}^v = \frac{1}{N_m+1} \tag{23}$$

Similarly, in (16) and (22), letting $\rho_s' = \rho_o' = \rho'$. $1-\rho'$ and ρ' are respectively the update weights of cluster member node i and that of cluster-head m in the clock parameter update equations of cluster member node. Given that the update weights of each node are equal, i.e., $1-\rho' = \rho' \Rightarrow \rho' = 1/2 \Rightarrow w_{ij}^v = 1/2$. Therefore, the update weight matrix of intra-cluster virtual clock is

$$w_{ij}^v = \begin{cases} 0, & (i,j) \notin E \\ \frac{1}{2}, & (i,j) \in E, \ and \quad i \le N_m \\ \frac{1}{N_m+1}, & (i,j) \in E, \ and \quad i = N_m+1. \end{cases} \tag{24}$$

B. Inter-Cluster Time Synchronization

Define the induced graph of G by $g = (V_g, E_g)$, where $V_g = \{1, 2, \ldots, M\}$ is the set of clusters and the set of edges E_g represents the communication links between clusters. $(m,l) \in E_g$ represents that cluster m and l overlap each other. Let $\tilde{D}_m = \{l \in V_g : (m,l) \in E_g\}$ denote the set of neighboring clusters of cluster m, and $\tilde{d}_m = |\tilde{D}_m|$ is the degree of cluster m.

As shown in Fig. 1, cluster A and B overlap each other. According to the graph theory representation, $\tilde{G}_{AB} = (\tilde{V}, \tilde{E})$ denotes the subgraph composed by nodes of cluster A and B in the cluster-based topology network. G_{AB} denotes the distributed network composed by the same nodes, where $\tilde{V}$ denotes the set of nodes in cluster A and B, and $\tilde{E}$ denotes the communication links between the nodes in $\tilde{G}_{AB}$. As illustrated in Fig. 1, since nodes in one cluster and those in its adjacent clusters all become neighbors, the new graph $\tilde{G}_{AB}$ is much better-connected than G_{AB} [21]. It is obvious that G is a subgraph of $\tilde{G}$. According to [21], $\tilde{G}$ is a connected graph, and the states of nodes will converge to consensus in the consensus algorithm.

Inter-cluster time synchronization is achieved by gateway nodes. According to (4) and the intra-cluster virtual clock given by (7), cluster-heads establish the network virtual clocks, which is given by

$$\hat{C}_m^w(t) = \hat{s}_m^w \hat{C}_m^v(t) + \hat{o}_m^w \tag{25}$$

Where $\hat{s}_m^w$ and $\hat{o}_m^w$ are the skew and offset compensation parameters of network virtual clock of cluster-head m respectively. Substituting (7) into (25), then

$$\begin{aligned}\hat{C}_m^w(t) &= \hat{s}_m^w \hat{s}_m^v \alpha_m t + \hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w \\ &= \alpha_m^w t + \beta_m^w \end{aligned} \tag{26}$$

Where $\alpha^w = \hat{s}^w\hat{s}^v\alpha$ and $\beta^w = \hat{s}^w(\hat{s}^v\beta + \hat{o}^v) + \hat{o}^w$ are respectively the skew and the offset of network virtual clock of node relative to the absolute time.

1) Skew Compensation in Inter-Cluster Time Synchronization: Cluster m and l are overlapping clusters, and their relative skew of intra-cluster virtual clock is

$$\alpha_{ml}^{v} = \frac{\alpha_l^v}{\alpha_m^v} = \frac{\hat{C}_l^v(k+1) - \hat{C}_l^v(k)}{\hat{C}_m^v(k+1) - \hat{C}_m^v(k)} \tag{27}$$

Assuming that cluster m overlaps with r clusters, as the skew compensation parameters of network virtual clocks of every cluster are assigned with the corresponding weights based on the size of each cluster, then cluster m updates network virtual clock skew compensation parameter with

$$\begin{aligned}\hat{s}_m^w(k+1) &= \frac{N_m\hat{s}_m^w(k) + N_1\hat{s}_1^w(k)\alpha_{m1}^v + \cdots + N_r\hat{s}_r^w(k)\alpha_{mr}^v}{N_m + N_1 + \cdots + N_r} \\ &= \frac{N_m\hat{s}_m^w(k)}{N_m + \sum_{l=1}^r N_l} + \frac{\sum_{l=1}^r N_l\hat{s}_l^w(k)\alpha_{ml}^v}{N_m + \sum_{l=1}^r N_l} \end{aligned} \tag{28}$$

Equation (28) is the update equation of skew compensation parameters of the network virtual clocks. To verify whether skews compensated satisfy the consensus algorithm, we multiply α_m^v at both sides of (28), we have

$$\begin{aligned}\hat{s}_m^w(k+1)\alpha_m^v &= \frac{N_m\hat{s}_m^w(k)\alpha_m^v}{N_m + \sum_{l=1}^r N_l} + \frac{\sum_{l=1}^r N_l\hat{s}_l^w(k)\alpha_{ml}^v\alpha_m^v}{N_m + \sum_{l=1}^r N_l} \\ \alpha_m^w(k+1) &= \frac{N_m\alpha_m^w(k)}{N_m + \sum_{l=1}^r N_l} + \frac{\sum_{l=1}^r N_l\alpha_l^w(k)}{N_m + \sum_{l=1}^r N_l} \\ &= \alpha_m^w(k) + \frac{\sum_{l=1}^r N_l[\alpha_l^w(k) - \alpha_m^w(k)]}{N_m + \sum_{l=1}^r N_l} \end{aligned} \tag{29}$$

By (29), the skews of network virtual clocks satisfy consensus algorithm as shown in (1). So after iterative updates, the skews of network virtual clocks of cluster-heads will converge to consensus.

2) Offset Compensation in Inter-Cluster Time Synchronization: After the skew compensation is applied, we can suppose that the network virtual clocks of adjacent cluster-heads have the same skew, and thus the network virtual clock of cluster-head m as shown in (26) can be rewritten as

$$\begin{aligned}\hat{C}_m^w(t) &= \alpha^w t + \hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w \\ &= \alpha^w t + \beta_m^w \end{aligned} \tag{30}$$

Where $\beta_m^w = \hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w$ is the offset of network virtual clock of cluster-head m.

Based on the network virtual clocks of neighboring clusters, cluster-head m updates network virtual clock offset compensation parameter with

$$\hat{o}_m^w(k+1) = \hat{o}_m^w(k) + \frac{\sum_{l=1}^r N_l[\hat{C}_l^w(k) - \hat{C}_m^w(k)]}{N_m + \sum_{l=1}^r N_l} \tag{31}$$

Equation (31) is the update equation of offset compensation parameters of the network virtual clocks. By (30), we have $\hat{C}_m^w(k) - \hat{C}_l^w(k) = \beta_m^w(k) - \beta_l^w(k)$, and since $\beta_m^w = \hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w$, we verify whether offsets compensated satisfy the consensus algorithm, then

$$\begin{aligned}&\hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w(k+1) \\ &\quad = \hat{s}_m^w(\hat{s}_m^v\beta_m + \hat{o}_m^v) + \hat{o}_m^w(k) + \frac{\sum_{l=1}^r N_l[\hat{C}_l^w(k) - \hat{C}_m^w(k)]}{N_m + \sum_{l=1}^r N_l} \\ &\beta_m^w(k+1) = \beta_m^w(k) + \frac{\sum_{l=1}^r N_l[\beta_l^w(k) - \beta_m^w(k)]}{N_m + \sum_{l=1}^r N_l} \end{aligned} \tag{32}$$

By (32), the offsets of the network virtual clocks satisfy the consensus algorithm as shown in (1). So after iterative updates, the offsets of network virtual clocks of cluster-heads will converge to consensus.

By (29) and (32), the update weights of cluster-head m and that of neighboring cluster-heads in the clock parameter update equations of cluster-head m is

$$w_{ml}^c = \frac{N_l}{N_m + \sum_{t=1}^r N_t}. \tag{33}$$

V. Performance Analysis and Simulation

We perform analyses and simulations on convergence rate, communication traffic, the skew, the offset, and the time synchronization error in MATLAB. The size of network is 200 nodes distributed randomly. The network is divided into 25 clusters, the numbers of cluster member nodes of each cluster are distributed randomly. The crystal oscillators run at 32768 Hz, therefore 1 tick = 1/32768 Hz = 30.5μs, where one tick is one oscillation period. The skew rates are assigned randomly from a normal distribution with mean = 1 and standard deviation = 20 PPM, nodes are given an initial error distribution between 0 and 400 ticks, the fixed delay are assigned randomly from a uniform distribution with mean = 0 and standard deviation = 20μs. The CCTS is compared with the classic DCTS.

A. Convergence Rate

Reference [30] maps the iteration process of consensus algorithm into the Markov chain domain, and deduces the relationship between the convergence rate and the network parameters. With reference to [30], we analyze the convergence rate of CCTS. Firstly, we map the iteration process of CCTS into the state transition process of Markov chain. Mapping relationship is

$$\begin{aligned} &G = (V, E) \rightarrow M_G(S, \mathbf{P}) \\ &V = S, \quad S = \{s_1, s_2, \ldots, s_n\} \\ &\mathbf{W} = \mathbf{P}, \quad w_{ij} = p_{ij} \end{aligned} \tag{34}$$

Where s_i is the state of the Markov chain. S is the set of s_i, and it corresponds to the set of nodes. The Markov one step transition probability matrix $\mathbf{P}$ is the weight matrix $\mathbf{W}$. By [30], the second largest eigenvalue of an Markov chain M_G determines convergence rate, that is

$$k = \frac{1}{\log(1/\lambda_2)} \tag{35}$$

Where λ_2 satisfies Cheeger's inequality [31] $1 - 2\Phi \le \lambda_2 \le 1 - \Phi^2/2$, where Φ is the conduction coefficient of the

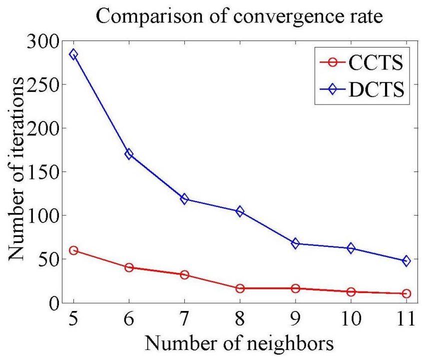

Fig. 2. Comparison of CCTS and DCTS on convergence rate.

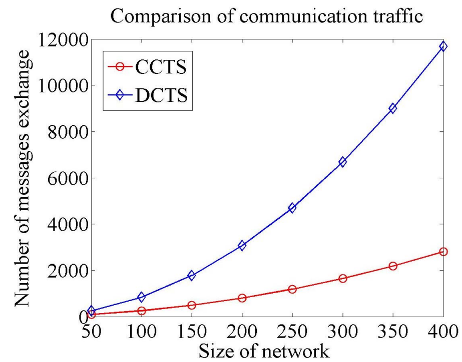

Fig. 3. Comparison of CCTS and DCTS on communication traffic in a single-step iteration.

Markov chain. By [32], there is $\Phi \approx \frac{d'}{2n'}$ in cycle network, then Cheeger's inequality states that

$$1 - d'/n' \leq \lambda_2 \leq 1 - d'^2/8n'^2 \tag{36}$$

Where d' is the degree of node, and n' denotes the size of the network. We substitute (36) into (35), then the number of iterations to convergence is obtained.

$$\frac{n'}{d'} \leq k \leq \frac{8n'^2}{d'^2} \tag{37}$$

In intra-cluster time synchronization, nodes can obtain the information of all the nodes within the cluster, so the network is fully connected. Then, there are $d' = N_m$ and $n' = N_m + 1$. By (37), the number of iterations required to ensure nodes' clocks converge to consensus is

$$\frac{N_m + 1}{N_m} \leq k \leq \; 8(\frac{N_m + 1}{N_m})^2 \tag{38}$$

By (38), the number of iterations of intra-cluster virtual clocks required to converge to consensus is only related to the number of nodes in the cluster.

In inter-cluster time synchronization, network is divided into M clusters, so there are $d' = \tilde{d}$ and $n' = M$ in (37). Then the number of iterations is

$$\frac{M}{\tilde{d}} \leq k \leq \frac{8M^2}{\tilde{d}^2} \tag{39}$$

By (39), the number of iterations of network virtual clocks required to converge to consensus is related to the ratio of the cluster number of the network and the neighbor numbers of each cluster.

In Fig. 2, the number of iterations of CCTS and that of DCTS are compared when the mean square error (MSE) of clock is below 10^{-5}. It shows that the convergence rate of networks can be much faster in CCTS than DCTS when the neighbor numbers of nodes increase.

B. Communication Traffic

In DCTS, a node with $d_i = |D_i|$ neighbors needs $1 + |D_i|$ messages in a single-step iteration, and thus the network with n nodes needs $\sum_{i=1}^{n}(1 + |D_i|)$ messages. In CCTS, the network with M clusters and n nodes needs $1 + N_m$ messages within the cluster and n messages for the whole network in the intra-cluster time synchronization. In the inter-cluster time synchronization, the number of messages exchanged between the clusters is $1 + 3|\tilde{D}_m|$ and in the whole network is $\sum_{m=1}^{M}(1 + 3|\tilde{D}_m|)$. So, in CCTS, the number of messages exchanged in a single-step iteration is $n + \sum_{m=1}^{M}(1 + 3|\tilde{D}_m|)$. The comparison of the communication traffic between CCTS and DCTS in a single-step iteration is shown in Fig. 3. It shows the number of messages exchanged in the networks can be much higher in DCTS than CCTS when the size of network gets larger.

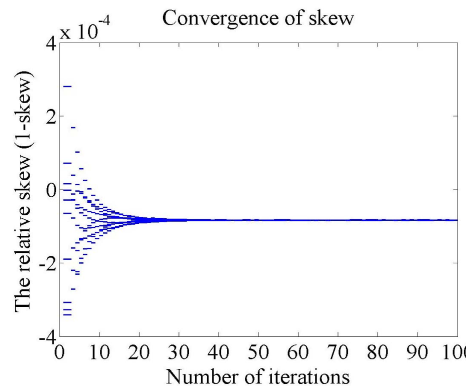

Fig. 4. The convergence of skews of intra-cluster virtual clocks for a cluster of ten nodes in intra-cluster time synchronization.

C. Synchronization Error

1) Intra-Cluster Time Synchronization: The size of cluster is 10 nodes that distribute randomly in the cluster. Fig. 4 shows the convergence of skews of intra-cluster virtual clocks during the 100 rounds of skew compensations. After 31 rounds compensations, the biggest difference of skews of nodes is 30.2μs which is less than 1 tick, that is, the skews of intra-cluster virtual clocks of 10 nodes in the cluster achieve the consensus.

Histograms of the offset errors of intra-cluster virtual clocks for nodes before (a) and after (b) the offset compensations are shown in Fig. 5. Before the offset compensation, the maximum difference of offsets can reach up to 360 ticks, that is, the maximum and minimum value of offsets are 379 ticks and 19 ticks respectively. But after 20 rounds offset compensations,

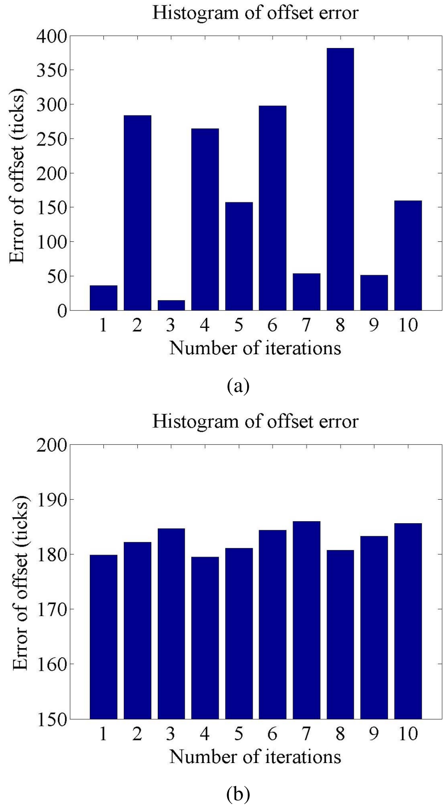

Fig. 5. Histograms of the offset errors of intra-cluster virtual clocks of nodes within the cluster (a) before and (b) after 20 rounds of offset compensations.

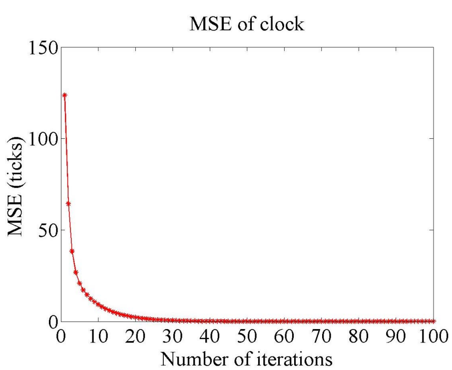

Fig. 6. MSE of intra-cluster virtual clock for intra-cluster time synchronization.

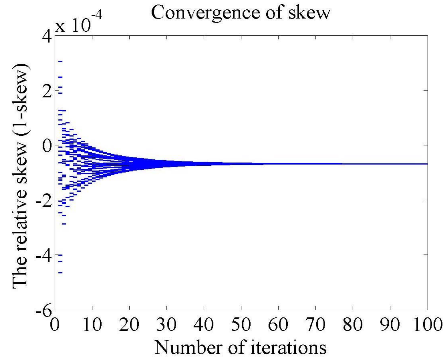

Fig. 7. The convergence of skews of network virtual clocks of 25 clusters in inter-cluster time synchronization.

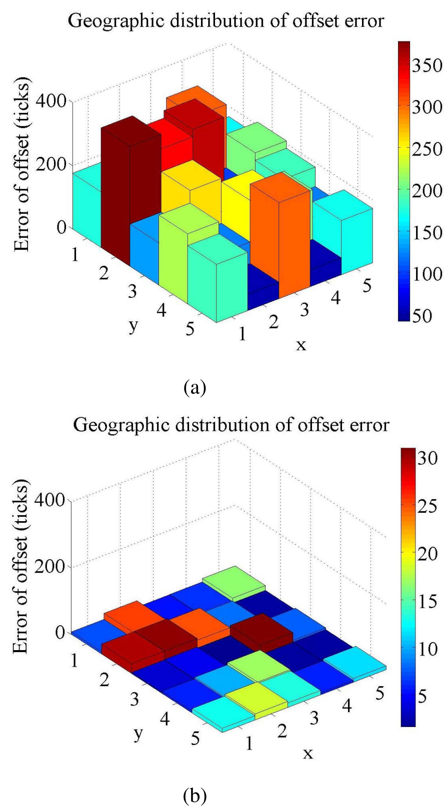

Fig. 8. The geographic distributions of the offset errors of network virtual clocks of cluster-heads (a) before and (b) after 30 rounds of offset compensations.

as shown in Fig. 5 (b), the offset errors between any nodes are no larger than 10 ticks.

When both offset compensations and skew compensations are applied, the clock MSE of intra-cluster time synchronization are obtained as shown in Fig. 6. The MSE of intra-cluster virtual clocks decreases when the iterations increase.

2) Inter-Cluster Time Synchronization: The network is divided into 25 clusters distributed randomly in a 5×5 grid. Fig. 7 shows the convergence of skews of network virtual clocks during the 100 rounds of skew compensations. After 43 rounds compensations, the biggest difference of skews of nodes is $29.4\mu s$ which is less than 1 tick, indicating that the skews of network virtual clocks of cluster-heads in the network achieve the consensus.

Geographic distributions of the offset errors of network virtual clocks for clusters before (a) and after (b) the offset compensations are shown in Fig. 8. Before the offset compensation, the maximum difference of offsets can reach up to 343 ticks. But after 30 rounds offset compensations, as shown in Fig. 8 (b), the offset errors of network virtual clocks between any pair of cluster-heads are no larger than 30 ticks.

When both offset compensations and skew compensations are applied, the clock MSE of inter-cluster time synchronization is obtained as shown in Fig. 9. The curve leaps up on the second data point and declines from the third data point. The reason for this phenomenon is that the offset compensation of the inter-cluster time synchronization actually starts from the second iteration. After intra-cluster

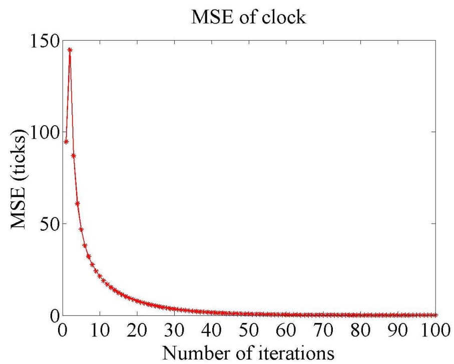

Fig. 9. MSE of network virtual clock for inter-cluster time synchronization.

time synchronization, the intra-cluster virtual clocks of nodes in the cluster achieve consensus. However, there are deviations between intra-cluster virtual clocks of every cluster. In the stage of inter-cluster time synchronization, the offset compensation actually starts from the second iteration and the skew compensation in the first iteration does not reduce the clock deviations but aggravates them, causing that the curve of MSE starts to decline from the third iteration on.

VI. Conclusion and Future Work

A new algorithm for time synchronization of WSNs was proposed in this paper. This algorithm is based on the consensus algorithms but incorporated with the clustering technique. The basic idea of CCTS is to use local information to achieve a global agreement. The proposed algorithm reduces the communication traffic compared to the DCTS and improves the convergence rate due to the incorporation of clustering topology. Finally, a set of simulations are carried out to show the successful performance of the proposed protocol.

However, several shortcomings of CCTS ought to be improved in the future study. On the one hand, the synchronization period of CCTS is not adaptive in respect of the delay, so that the clock compensation cannot be made in unpredictable environments. On the other hand, the CCTS only is established on the base of initial consensus, so that it is not sufficient to deal with the dynamic topology problem that requires resynchronization. Since the energy of nodes will reduce and communication range of clusters will become smaller over time, the number of neighbor clusters will reduce, leading to the reduced convergence rate over time (Fig. 2). Owing to the characteristic of clustering topology and the feature of consensus algorithms, CCTS is realized in the network where a large number of messages and status information are exchanged between nodes. Since CCTS is modeled in the context of no message loss, its performance will probably be compromised due to the message loss and the cluster membership change when deals with the unpredictable packet losses and high probability of node failures in the real WSNs.

Therefore, our future work will focus on solving the problem of asymmetric link and clock delay estimation based on the prediction in dynamic and unpredictable environments. Moreover, the problem of message loss and adaptive resynchronization cycle based on the delay and synchronization errors will also be addressed.

Acknowledgment

The authors would like to thank editors and anonymous reviewers for their outstanding detailed comments. They would also like to thank H. Wu for her assistance in the manuscript preparation.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, 2007.

[2] S. Misra and A. Vaish, “Reputation-based role assignment for role-based access control in wireless sensor networks,” *Comput. Commun.*, vol. 34, no. 3, pp. 281–294, 2011.

[3] K. Plarre and P. R. Kumar, “Object tracking by scattered directional sensors,” in *Proc. 44th IEEE Conf. Decision Control*, Seville, Spain, Dec. 2005, pp. 3123–3128.

[4] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: A survey,” *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul./Aug. 2004.

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

[6] J. Elson and K. Römer, “Wireless sensor networks: A new regime for time synchronization,” in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2003, vol. 33, no. 1, pp. 149–154.

[7] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proc. ACM SIGOPS Oper. Syst. Rev.*, 2002, vol. 36, no. SI, pp. 147–163.

[8] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proc. 1st Int. Embedded Netw. Sensor Syst.*, Los Angeles, CA, USA, 2003, pp. 138–149.

[9] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, Baltimore, MD, USA, 2004, pp. 39–49.

[10] Z. Dengchang, A. Zhulin, and X. Yongjun, “Time synchronization in wireless sensor networks using max and average consensus protocol,” *Int. J. Distrib. Sensor Netw.*, vol. 2013, pp. 1–10, Feb. 2013, Art. ID 192128.

[11] W. Su and I. F. Akyildiz, “Time-diffusion synchronization protocol for wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 384–397, Apr. 2005.

[12] L. Schenato and G. Gamba, “A distributed consensus protocol for clock synchronization in wireless sensor network,” in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, USA, Dec. 2007, pp. 2289–2294.

[13] M. K. Maggs, S. G. O’Keefe, and D. V. Thiel, “Consensus clock synchronization for wireless sensor networks,” *IEEE Sensors J.*, vol. 12, no. 6, pp. 2269–2277, Jun. 2012.

[14] Q. Li and D. Rus, “Global clock synchronization in sensor networks,” *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, Feb. 2006.

[15] R. Solis, V. S. Borkar, and P. R. Kumar, “A new distributed time synchronization protocol for multihop wireless networks,” in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, USA, Dec. 2006, pp. 2734–2739.

[16] P. Sommer and R. Wattenhofer, “Gradient clock synchronization in wireless sensor networks,” in *Proc. 8th ACM/IEEE Conf. Inf. Process. Sensor Netw.*, San Francisco, CA, USA, Apr. 2009, pp. 37–48.

[17] K. Apicharttrisorn, S. Choochaisri, and C. Intanagonwiwat, “Energy-efficient gradient time synchronization for wireless sensor networks,” in *Proc. 2nd Int. Conf. Comput. Intell., Commun. Syst., Netw.*, Liverpool, U.K., Jul. 2010, pp. 124–129.

[18] G. Xiong and S. Kishore, “Second order distributed consensus time synchronization algorithm for wireless sensor networks,” in *Proc. IEEE Conf. Global Telecommun.*, New Orleans, LA, USA, Nov./Dec. 2008, pp. 1–5.

[19] Y. Liao, H. Qi, and W. Li, “Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks,” *IEEE Sensors J.*, vol. 13, no. 5, pp. 1498–1506, May 2013.

[20] C.-T. Cheng, C. K. Tse, and F. C. M. Lau, “A clustering algorithm for wireless sensor networks based on social insect colonies,” *IEEE Sensors J.*, vol. 11, no. 3, pp. 711–721, Mar. 2011.

[21] 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.

[22] G. Miao, N. Himayat, G. Y. Li, and S. Talwar, "Distributed interference-aware energy-efficient power optimization," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1323–1333, Apr. 2011.

[23] S.-H. Lee, E. Choi, and D.-H. Cho, "Timer-based broadcasting for power-aware routing in power-controlled wireless ad hoc networks," *IEEE Commun. Lett.*, vol. 9, no. 3, pp. 222–224, Mar. 2005.

[24] A. Zappone, Z. Chong, E. A. Jorswieck, and S. Buzzi, "Energy-aware competitive power control in relay-assisted interference wireless networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1860–1871, Apr. 2013.

[25] M. Baunach, "Handling time and reactivity for synchronization and clock drift calculation in wireless sensor/actuator networks," in *Proc. 3rd Int. Conf. Sensor Netw. (SENSORNETS)*, Lisbon, Portugal, 2014, pp. 63–72.

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[27] M. S. Stankovic, S. Stankovic, and K. H. Johansson, "Distributed time synchronization in lossy wireless sensor networks," in *Proc. 3rd IFAC Workshop Distrib. Estimation Control Netw. Syst.*, 2012, vol. 3, no. 1, pp. 25–30.

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

[29] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[30] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, USA, Dec. 2006, pp. 4915–4920.

[31] R. Kannan, "Markov chains and polynomial time algorithms," in *Proc. 35th IEEE Annu. Symp. Found. Comput. Sci.*, Santa Fe, NM, USA, Nov. 1994, pp. 656–671.

[32] L. Li, L. Yong-Pan, Y. Hua-Zhong, and W. Hui, "Convergence analysis and accelerating design for distributed consensus time synchronization protocol in wireless sensor networks," *J. Electron. Inf. Technol.*, vol. 32, no. 9, pp. 2045–2051, 2010.

Jie Wu is currently pursuing the Ph.D. degree at the School of Electronic Information Engineering, Tianjin University, Tianjin, China.

His current research interests include wireless sensor networks time synchronization and sensor node localization.

Liyi Zhang received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2003.

He is currently a Professor and the Director of the Graduate School and the Office of Discipline Construction, Tianjin University of Commerce, Tianjin, China. He is also a Tutor of Post-Graduate Candidates of Information and Communication Engineering with the School of Electronic Information Engineering, Tianjin University, Tianjin.

He is a Committee Member of the Circuits and Systems subcommittee of the Chinese Institute of Electronics, and the Director of the Universities Working Committee of the Tianjin Institute of Communications.

His current research interests include signal detection and processing, neural network, blind equalization and optimization, higher-order spectrum theory and application, and wireless sensor networks.

Yu Bai received the Ph.D. degree in telecommunication and electronic engineering from Tianjin University, Tianjin, China, in 2009. In 2009, he joined Tianjin University as an Instructor, where he has been a Post-Doctoral Research Fellow since 2010.

His research interests include radio frequency identification, signal processing, and filter theory and design.

Yunshan Sun received the Ph.D. degree from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2012.

His current research interests include blind image restoration, neural network, and blind equalization and optimization.