


A consensus-based distributed method of clock synchronization for sensor networks

International Journal of Distributed
Sensor Networks
2017, Vol. 13(3)
© The Author(s) 2017
DOI: 10.1177/1550147717697320
journals.sagepub.com/home/ijdsn


Yong Qiao¹, Wenlun Yang¹ and Minyue Fu^{1,2}

Abstract

In this study, an innovative distributed method for achieving external clock synchronization is presented. Based on discrete-time clock models, it can realize synchronization of clocks while enabling their time to change at the same pace simultaneously. Stronger robustness against noisy measurements and clock rate drifts is gained by combining both controller and estimator design methods into the protocol. Additionally, a specifically designed communication scheme is proposed to make our protocols independent on global physical time. To render our protocols more practical, the control variable for clock synchronization is ensured bounded and a stopping criterion for implementation of the protocols is established. Finally, performance of the method is illustrated by certain numerical simulations.

Keywords

Distributed method, clock synchronization, consensus-based, wireless sensor networks

Date received: 10 September 2016; accepted: 7 February 2017

Academic Editor: Olivier Sentieys

Introduction

Technological advancement of low-cost and low-power programmable sensors has been witnessed during these years. These sensors are capable of sensing, recording, and processing data, as well as communicating via wireless channels. They can be distributed in large areas to monitor physical phenomena, making up wireless sensor networks (WSNs). In WSNs, sensor nodes can be embedded in the environment to be static, or they are enabled to be mobile with physical artifacts carrying them.¹

WSNs are closely coupled to application backgrounds.² Climate monitoring, target positioning, and event detection are time-critical applications in which local clocks of all sensors must have the same time value. However, oscillators in low-cost sensors result in different local clock time with distinct drifting speed. Thus, clock synchronization is a significant and interesting topic attracting focuses of many researches. Although successful clock synchronization protocols

for wired networks have been developed for decades, they are inappropriate for WSNs for facts that WSNs have a wider deployment, constraint power supply, and a higher demand on robustness.

Literature review

For clock synchronization in WSNs, it includes the following method patterns:³ master-to-slave versus peer-to-peer synchronization. Master-to-slave protocols, among which we will list some to review, assign certain

¹School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

²The State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, P.R. China

Corresponding author:

Minyue Fu, School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia.
Email: minyue.fu@newcastle.edu.au



nodes as masters. Reference Broadcast Sync seeks to reduce non-deterministic time delay using partial aggregations and conserves energy via the post facto scheme.⁴ However, it has to divide the network into multiple clusters and choose master nodes for each cluster, which increases workloads of communication and computation. Time-Diffusion Sync Protocol⁵ utilizes a diffusion of messages, in which all nodes participate to establish network-wide balanced time. A radial tree structure is formed with an election and re-election cycle where each node decides to become a master or a diffused-leader on its own. With the structure, an iterative weighted-averaging process can be carried out. Timing-sync Protocol for Sensor Networks (TPSN)⁶ builds up a spanning tree along whose branches pairwise synchronization is executed by sender–receiver handshake⁷ exchanges, before the phase of synchronization. Master-node failure is a common problem with which these methods are confronted. In another method called Flooding Time Sync Protocol, receiving nodes record incoming time-stamps' arrival instants and normalize them, after which they update their own time-stamps and then broadcast them to their neighbors.⁸ This method achieves robustness to root failure to some extent, but not completely.

Compared with master-to-slave protocols, peer-to-peer protocols are distributed without center nodes. Then, problems caused by master-node failure do not exist. Average Time Sync (ATS)⁹ utilizes a cascade of two consensus algorithms to achieve the consensus of clock skews and that of offsets separately by tuning compensation parameters. In Distributed Time Sync,¹⁰ neighboring nodes exchange time-stamp packets on a one-to-one basis. Any participating node can become a reference node by simply not adjusting its own clock during the synchronization procedure, which converts synchronization into solving an optimization problem. To obtain faster convergence and better energy efficiency, Clustered Consensus Time Synchronization (CCTS)¹¹ combines the Distributed Consensus Time Synchronization (DCTS) algorithm with the clustering technique. However, it still compensates skew and offset parameters separately and requires additional energy for clustering.

However, method patterns can be roughly classified into internal versus external synchronization. The physical-clock rate which depends on the oscillator frequency usually cannot be tuned directly, while external time of local clocks can be synchronized more easily.¹² Thus, external clocks, also called virtual clocks, can be modeled to make synchronization much easier. However, only certain literature, such as CCTS¹¹ and FLOPSYNC-2,¹³ presents relatively specific relationships between actual and virtual clocks.

Statement of contribution

Compared with previous research work, our article contributes to the following innovations as far as we know:

1. Unlike most previous literature,¹⁴ clock skews of sensor nodes in our method can be time-varying, which makes the model more apposite in implementation.
2. Synchronization of clock rates and that of initial local time are achieved separately by most works^{9,15} with several coupled algorithms. However, on one hand, this separation can result in slower convergence, which means heavy burdens of communication and computation. On the other hand, performance of coupled algorithms would interact with each other, which may have an undesirable accumulated effect on synchronization precision. Like certain literature,¹⁶ our method combines and solves these two tasks jointly to achieve external synchronization with light-weight linear protocols.
3. Inspired by both controller and estimator design methods, our protocol combines their advantages as in the literature^{9,13} to follow drifts of clock time with noise filtering. Besides, control inputs used to adjust time of local clocks are proved to be bounded, which was rarely considered by previous researches.
4. Certain application details usually omitted in theoretical researches are considered and explained in this article. First, the virtual-time model is clearly distinguished from the physical-clock model to make this external synchronization more applicable. Then, a specifically designed communication scheme is proposed to explicitly show how sensor nodes execute packets' receiving and transmitting with absence of global time information. Meanwhile, inspired by literature,⁹ the communication scheme is modified to expand its application with asynchronous communication.
5. A stopping criterion for our synchronization protocols is established in a distributed way. Then, the synchronization process can be ended in finite time, which is significant in practical implementation.

Notation and graph theory

Notation

For a number p , $|p|$ denotes its modulus; $\mathbf{1}$ represents a column vector with N dimensions, whose entries are all equal to 1; $\mathbf{0}$ denotes the matrix with a proper order and with all entries equal to 0; \mathbf{I} represents an N -

dimension identity matrix; and the superscripts T and H denote the transpose of real matrices and conjugate transpose of complex matrices, respectively.

Graph theory

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of a non-empty node set $\mathcal{V} = \{1, 2, \dots, N\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where an edge of \mathcal{G} is a pair of unordered nodes. The in-neighbor set of node i is denoted as \mathcal{N}_i , that is, $\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}$. In WSNs, if j is an in-neighbor of i , this has the physical meaning that sensor j is located within the listening distance of sensor i . An undirected graph is called connected if any pair of nodes can be connected by a path of edges in \mathcal{E} . The diameter \mathfrak{D} of a graph is the longest path among the shortest ones between any pair of nodes.

An extended neighbor graph, denoted as $\mathcal{D}(t)$, is a directed graph with at most one arc between each pair of ordered nodes and with one self-loop of each vertex on event time t . If extended neighbor graphs on all event instants within a time interval make up \mathcal{G} , then vertex j belongs to \mathcal{N}_i in \mathcal{G} as long as this establishes at least once in $\mathcal{D}(t)$ in that interval.

For a graph \mathcal{G} , each edge (j, i) is associated with a weight $\xi_{ij} \neq 0$. Then, the ij th entry of a Laplacian matrix L is defined as

$$L(i, j) = \begin{cases} -\xi_{ij} & i \neq j, j \in \mathcal{N}_i \\ 0 & i \neq j, j \notin \mathcal{N}_i \\ \sum_{j \in \mathcal{N}_i} \xi_{ij} & i = j \end{cases}$$

Lemma 1. Zero is an eigenvalue of the Laplacian matrix L with $\mathbf{1}$ as a corresponding right eigenvector. All non-zero eigenvalues have positive real parts. For an undirected graph, 0 is a simple eigenvalue if and only if the graph is connected.¹⁷

Clock models and proposed synchronization protocols

Problem formulation

First, one classical simplified clock-time model¹⁸ can be presented as

$$\tau_i(t) = \alpha_i t + o_i \quad (1)$$

where $\tau_i(t)$ represents the local clock time of node i on the instant when the global physical time is t . $\alpha_i > 0$ stands for the local clock rate of node i . o_i denotes the initial value of $\tau_i(t)$, that is, $o_i = \tau_i(0)$. According to the model, each local clock time has its own relation with t before being synchronized. Therefore, clock synchronization methods aim to tune all local time to eventually reach consensus values corresponding with the same

global instants. The following are certain key aspects in this research area nowadays, which are not solved totally:

1. When α_i is time varying, denoted as $\alpha_i(t)$, instead of constant in practice, new protocols are needed for clock synchronization.
2. Since $\alpha_i(t), i = 1, 2, \dots, N$, is different from each other and actually global time is unavailable to all nodes, a communication scheme has to be specifically designed for the execution of clock synchronization. Besides, the application of designed protocols should be able to be extended to cases with asynchronous communication.
3. Synchronization of local time and that of local time variations should be achieved simultaneously for the sake of synchronization accuracy and power conservation.

In this article, we explain how we try to solve the above concerns.

Models and protocols

Continuous clock synchronization poses a high demand for power supply, which is not practical for WSNs. So, we establish a discrete-time sampling model of each local clock as

$$\tau_i(k+1) = \tau_i(k) + \Delta_i(k) \quad (2)$$

where $k = 0, 1, \dots$ denotes the k th sampling from the start of the synchronization. Accordingly, t_k represents the global instant on which the sampling occurs. $\tau_i(k+1)$ denotes node i 's local time sampled after one sampling period T_k from t_k . $\Delta_i(k) > 0$ corresponding to T_k represents clock i 's local time variation between t_k and t_{k+1} . Since all local time drifts at distinct speed, $\Delta_i(k), i = 1, 2, \dots, N$, has a many-to-one mapping relationship with T_k , that is, usually $\Delta_i(k) \neq \Delta_j(k)$, if $i \neq j$.

Remark 1. Here, we assume that all nodes have the same sampling period from the global view. And a specific communication scheme can be designed as follows to make this assumption realizable in practice. First, the signal for clock synchronization wakens one of the nodes, supposed as node 1, from the low-power state known as the sleeping mode. And this node broadcasts packets containing information for synchronization to its neighbors on periodic transmission instants in its local timescale. With the MAC-layer stamping technique, node 1's neighbors can be assumed to receive its packet immediately when it broadcasts the packet. Then, as soon as receiving the packet, neighbors of node 1 are triggered to have their local time sampled and broadcast their packets to their own neighbors.

The information of k ensures that each node only broadcasts its packets once around the same instant. To make this scheme easily understood, its schematic diagram is shown in Figure 1. In this way, all nodes have the same sampling instants t_k which lead to the same sampling, as well as transmitting period T_k . And T_k is a monotonic function of Δ_1 , where Δ_1 denotes the local transmission period of node 1. There is an implicit assumption that sampling for the in-packet and broadcasting of the out-packet of the same node are triggered almost simultaneously, namely, time delay between them can be ignored compared to the period's timescale. It is kind of a case with synchronous communication. However, when that time delay cannot be ignored for some low-cost sensors or it accumulates tremendously in a very sparse and large net, our method has to be implemented with asynchronous communication, which will be specifically stated in the implementation part. Additionally, since $\alpha_1(t)$ is drifting while global physical time changes at the same pace, Δ_1 may lead to value changes of T_k . Thus, $\Delta_i(k), i \geq 2$ can be treated as $f(\alpha_i(k), T_k)$ —a function of $\alpha_i(k)$ and T_k . Due to above statements, it is rational to have the assumption as follows.

Assumption 1. Values of $\Delta_i(k), i = 1, 2, \dots, N$, are uniformly bounded when $k = 0, 1, \dots$

However, there are reasons that prevent us from utilizing this physical sampling model directly. First, it is not practical to adjust oscillators or counters of these sensors which may be deployed in inaccessible or atrocious environments. Additionally, this discrete-time form has the implicit disadvantage of causing time discontinuity, which may lead agents to making unexpected faults such as missing significant events or recording the same thing multiple times. These critical issues are usually omitted by methods for internal synchronization.

As a solution to above problems, a model of virtual local clock whose time can be directly tuned by a control input is established. Virtual-clock synchronization only needs achieving for cooperative events requiring for common time. And physical clocks, which are referred to for individual time events, still work due to their own mechanical schemes. On t_{k+1} , the virtual-clock model is presented as

$$\bar{\tau}_i(k+1) = \bar{\tau}_i(k) + \Delta_i(k) + u_i(k) \quad (3)$$

where $\bar{\tau}_i(k)$ stands for the virtual local time of node i and $\bar{\tau}_i(0) = \tau_i(0)$. The communication scheme presented in Remark 1 still works. And the specific process of packets' receiving and transmitting of node j is shown in Figure 2.

Thus, synchronization can be achieved by designing protocols for $u_i(k)$, which help all virtual-time update and converge to a common value. According to the

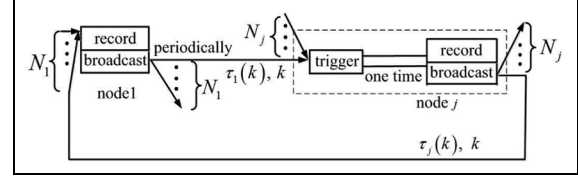


Figure 1. Schematic diagram of communication between node 1 and its neighbors.

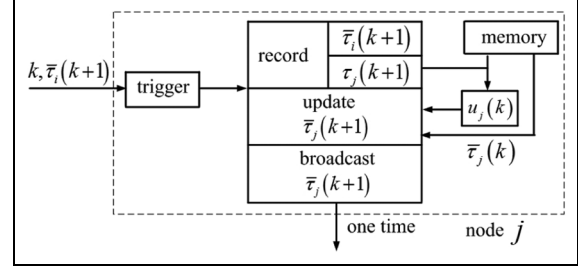


Figure 2. Schematic diagram of packets' recording and broadcasting.

required function, the control input can be divided into two components as follows

$$u_i(k) = u_{s_i}(k) + u_{\Delta_i}(k) \quad (4)$$

where $u_{\Delta_i}(k)$ is to ensure drifts of $\tilde{\Delta}_i(k) = \Delta_i(k) + u_{\Delta_i}(k)$ to be within a sufficiently narrow bounded interval eventually, which will be discussed later. The protocol for $u_{s_i}(k)$ is proposed as

$$\begin{cases} d_i(k+1) = \varepsilon \sum_{j \in N_i} l_{ij}(\bar{\tau}_j(k) - \bar{\tau}_i(k)) \\ u_{s_i}(k+1) = \sum_{j \in N_i} l_{ij}(\bar{\tau}_i(k) - \bar{\tau}_j(k)) + u_{s_i}(k) \\ \quad + d_i(k+1) - \mu d_i(k) \end{cases}$$

where $d_i(k)$ is an auxiliary variable here and $d_i(0) = 0$. l_{ij} is edge (j, i) 's weight. ε and μ are parameters to be identified in the following section.

Remark 2. Since $\bar{\tau}_j(k)$ and $\bar{\tau}_i(k)$ correspond to the same global instant, the difference between them can be easily gained by packet exchanges. That is, though all nodes have no access to the exact value of t_k , $\bar{\tau}_j(k) - \bar{\tau}_i(k)$ in the above expressions can be obtained in practice.

The vector form of above expressions can be expressed as

$$\begin{cases} \mathbf{d}(k+1) = -\varepsilon \mathbf{L} \bar{\boldsymbol{\tau}}(k) \\ \mathbf{u}_s(k+1) = \mathbf{L} \bar{\boldsymbol{\tau}}(k) + \mathbf{u}_s(k) + \mathbf{d}(k+1) - \mu \mathbf{d}(k) \end{cases} \quad (5)$$

where \mathbf{L} consisting of l_{ij} is a Laplacian matrix. Since sensors in the same network have almost the same listening distance and they usually have enough power to

establish message channels within these distances, wireless channels can be considered as being symmetric. Then it is reasonable to have the following assumption.

Assumption 2. The underlying graph of the network's message-channel topology is connected undirected.

Due to this assumption, $l_{ij} = l_{ji}$ and zero is one simple eigenvalue of \mathbf{L} which is symmetric.

Remark 3. When $k \geq 1$, combining the two expressions in equation (5) yields

$$\mathbf{u}_s(k+1) = \mathbf{u}_s(k) + (1-\varepsilon)\mathbf{L}\bar{\boldsymbol{\tau}}(k) + \mu\varepsilon\mathbf{L}\bar{\boldsymbol{\tau}}(k-1) \quad (6)$$

from which it can be seen that the effect of noise mainly from $\Delta(k)$ as part of $\bar{\boldsymbol{\tau}}(k)$ on $\mathbf{u}_s(k+1)$ can be reduced to certain extent by the combination form and choice of parameters. Said differently, $\mathbf{u}_s(k+1)$ can be taken as the update of an estimator with a filtering effect of noisy inputs (short-term jitter of clock crystal, measurement and quantization errors, etc.). This is an advantage over purely controller-based protocols.

However, we utilize the transfer function method by taking $\mathbf{L}\bar{\boldsymbol{\tau}}(k)$ and $\mathbf{u}_s(k)$ as the input and output of a system, respectively. This system is presented as Figure 3, where

$$Q(z) = -\frac{1}{z-\mu}, G(z) = \frac{(\varepsilon-1)z-\mu\varepsilon}{(z-1)\varepsilon}, H(z) = z-1$$

In this way, equation (5) can be converted into a feedback control system which is linear and time-invariant without uncertainty and model errors. Then, its closed-loop transfer function can be figured out as

$$G_{cls}(z) = \frac{(1-\varepsilon)z + \mu\varepsilon}{z(z-1)}$$

which shows the stability of this control system. Thus, as long as messages are exchanged regularly enough, the input of this system is bounded, which ensures $\mathbf{u}_s(k)$ not to be divergent. Then, values of $\mathbf{u}_s(k)$ can be gained in practice.

After the above analysis, we take the design of $u_{\Delta_i}(k)$ into consideration. Above all, we have to define one function as

$$P_i(k) = \begin{cases} \frac{|\tilde{\Delta}_i(k) - \tilde{\Delta}_i(k-1)|^2 - \gamma^2}{|\tilde{\Delta}_i(k) - \tilde{\Delta}_i(k-1)|^2} & , \left| \tilde{\Delta}_i(k) - \tilde{\Delta}_i(k-1) \right| \geq \gamma \\ 0 & , \text{Otherwise} \end{cases}$$

where $k \geq 1$ and γ is a preset sufficiently small positive parameter. Then, we can define another function as

$$\eta_i(k) = -\frac{P_i(k+1) - P_i(k)}{\bar{\boldsymbol{\tau}}_i(k+1) - \bar{\boldsymbol{\tau}}_i(k)}$$

where $k \geq 1$ and $\eta_i(0) = 0$. Then, $u_{\Delta_i}(k)$ is designed as

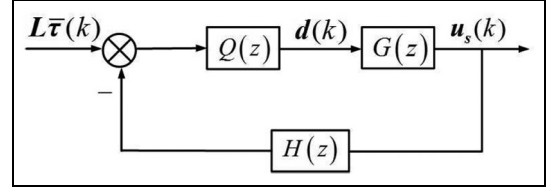


Figure 3. Scheme of the feedback control system.

$$u_{\Delta_i}(k) = (|u_{s_i}(k)| + |\Delta_i(k)| + \beta)\text{sgn}[\eta_i(k)]$$

where β is a sufficiently small positive parameter. With Remark 3 and Assumption 1, it is easy to figure out that $u_{\Delta_i}(k)$ is bounded, which implies that the overall input $u_i(k)$ and $\tilde{\Delta}_i(k)$ are also bounded.

Then, one Lyapunov function can be established in the following form

$$V(k) = \sum_{i=1}^N P_i(k)$$

Using the definition of $P_i(k)$, it can be figured out that $V(k) \geq 0$ establishes for $\forall i$ and all possible values of $|\tilde{\Delta}_i(k) - \tilde{\Delta}_i(k-1)| \geq 0$ when $k \geq 1$. Furthermore, the difference equation of the Lyapunov function can be obtained as

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = \sum_{i=1}^N [P_i(k+1) - P_i(k)] \\ &= -\sum_{i=1}^N \{\eta_i(k)[\bar{\boldsymbol{\tau}}_i(k+1) - \bar{\boldsymbol{\tau}}_i(k)]\} \\ &= -\sum_{i=1}^N \{\eta_i(k)[\Delta_i(k) + u_{s_i}(k) + u_{\Delta_i}(k)]\} \\ &= -\sum_{i=1}^N \{\eta_i(k)[u_{s_i}(k) + \Delta_i(k) \\ &\quad + (|u_{s_i}(k)| + |\Delta_i(k)| + \beta)\text{sgn}[\eta_i(k)]]\} \end{aligned}$$

where $k \geq 1$. Since $|u_{s_i}(k)| + |\Delta_i(k)| + \beta$ is positive and its value must be greater than $u_{s_i}(k) + \Delta_i(k)$, we can obtain that

$$\text{sgn}[u_{s_i}(k) + \Delta_i(k) + (|u_{s_i}(k)| + |\Delta_i(k)| + \beta)\text{sgn}[\eta_i(k)]]$$

equates with $\text{sgn}[\eta_i(k)]$, which implies that $\Delta V(k) \leq 0$. Said differently, if there exists one node whose value of $\tilde{\Delta}_i(k)$ has a drift larger than γ , $\Delta V < 0$ would be satisfied, otherwise $\Delta V = 0$. Thus, drifts between neighboring values of $\tilde{\Delta}_i(k)$, for $\forall i \in [1, N]$, have the trend to move inside the interval $[0, \gamma)$ due to above analysis. Then, there should exist one finite sampling instant, namely t_{k_m} , when neighboring values of $\tilde{\Delta}_i(k)$, for $\forall i \in [1, N]$, have the difference less than γ . This further

implies that when $k \geq k_m$, $\tilde{\Delta}_i(k)$ can be treated as a constant value as long as γ is sufficiently small.

Stability and consensus achieving analysis

Stability here means the input-to-state stability (ISS) of the whole system, which can ensure virtual time of all nodes to have bounded values respectively, while consensus represents that all bounded stable values of the virtual time converge to the same one asymptotically.

Theorem 1. With Assumption 1 and Assumption 2, the whole system can achieve stability if and only if the parameters in equation (5) satisfy that $1 < \varepsilon < 1 + \frac{1}{\rho(L)}$ and $0 < \mu < 1 - \frac{1}{\varepsilon}$, where $\rho(L)$ is the spectral radius of L .

Proof: Since L is symmetric, then U , an N -dimension unitary matrix, should exist and satisfy

$$U^H L U = \Lambda = \text{diag}\{\lambda_1, \dots, \lambda_N\} \quad (7)$$

where λ_i represents L 's one eigenvalue. Using Lemma 1, symbols of these eigenvalues can be arranged as $0 = \lambda_1 < \lambda_2 \dots \leq \lambda_N$. Combining equation (3) with above protocols obtains the state-space expression as

$$\begin{pmatrix} \bar{\tau}(k+1) \\ \mathbf{u}_s(k+1) \\ \mathbf{d}(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} \\ (1-\varepsilon)L & \mathbf{I} & -\mu\mathbf{I} \\ -\varepsilon L & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \bar{\tau}(k) \\ \mathbf{u}_s(k) \\ \mathbf{d}(k) \end{pmatrix} + \begin{pmatrix} \tilde{\Delta}(k)^T & \mathbf{0} & \mathbf{0} \end{pmatrix}^T \quad (8)$$

where $\mathbf{0}$ denotes an N -dimension square matrix with all entries equal to 0. Then, equation (8) can be decoupled by utilizing the unitary transformation

$$\begin{cases} \hat{\mathbf{d}}(k) = U^H \mathbf{d}(k), & \hat{\mathbf{u}}_s(k) = U^H \mathbf{u}_s(k) \\ \hat{\tau}(k) = U^H \bar{\tau}(k), & \hat{\Delta}(k) = U^H \tilde{\Delta}(k) \end{cases} \quad (9)$$

Let $\bar{U} = \text{diag}\{U, U, U\}$. Multiplying it with equation (8) in both sides yields

$$\begin{aligned} & \bar{U}^H (\bar{\tau}(k+1)^T \quad \mathbf{u}_s(k+1)^T \quad \mathbf{d}(k+1)^T)^T \\ &= \bar{U}^H \begin{pmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} \\ (1-\varepsilon)L & \mathbf{I} & -\mu\mathbf{I} \\ -\varepsilon L & \mathbf{0} & \mathbf{0} \end{pmatrix} \bar{U}^H \begin{pmatrix} \bar{\tau}(k) \\ \mathbf{u}_s(k) \\ \mathbf{d}(k) \end{pmatrix} \\ &+ \bar{U}^H (\tilde{\Delta}(k)^T \quad \mathbf{0} \quad \mathbf{0})^T \end{aligned}$$

which can be transformed to the decoupled vector form as

$$\begin{pmatrix} \hat{\tau}(k+1) \\ \hat{\mathbf{u}}_s(k+1) \\ \hat{\mathbf{d}}(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} \\ (1-\varepsilon)\Lambda & \mathbf{I} & -\mu\mathbf{I} \\ -\varepsilon\Lambda & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\tau}(k) \\ \hat{\mathbf{u}}_s(k) \\ \hat{\mathbf{d}}(k) \end{pmatrix} + \begin{pmatrix} \hat{\Delta}(k)^T & \mathbf{0} & \mathbf{0} \end{pmatrix}^T$$

The component-wise form of it is

$$\begin{pmatrix} \hat{\tau}_i(k+1) \\ \hat{u}_{s_i}(k+1) \\ \hat{d}_i(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ (1-\varepsilon)\lambda_i & 1 & -\mu \\ -\varepsilon\lambda_i & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{\tau}_i(k) \\ \hat{u}_{s_i}(k) \\ \hat{d}_i(k) \end{pmatrix} + \begin{pmatrix} \hat{\Delta}_i(k) & 0 & 0 \end{pmatrix}^T \quad (10)$$

With values of $\tilde{\Delta}_i(k)$ being bounded, the theorem of ISS¹⁹ helps conclude that stability can be achieved if the system matrix achieves Schur stability. The eigenvalue expression of the system matrix is

$$\begin{aligned} f(\delta) &= \begin{vmatrix} \delta - 1 & -1 & 0 \\ (\varepsilon - 1)\lambda_i & \delta - 1 & \mu \\ \varepsilon\lambda_i & 0 & \delta \end{vmatrix} \\ &= \delta^3 - 2\delta^2 + [1 + (\varepsilon - 1)\lambda_i]\delta - \mu\varepsilon\lambda_i \end{aligned}$$

Then, Schur stability can be achieved by utilizing the Jury stability criterion directly. According to the criterion, a test chart needs tabulating as Table 1 first, where

$$\begin{aligned} b_0 &= \begin{vmatrix} -\mu\varepsilon\lambda_i & 1 \\ 1 & -\mu\varepsilon\lambda_i \end{vmatrix}, & b_1 &= \begin{vmatrix} -\mu\varepsilon\lambda_i & -2 \\ 1 & 1 + (\varepsilon - 1)\lambda_i \end{vmatrix}, \\ b_2 &= \begin{vmatrix} -\mu\varepsilon\lambda_i & 1 + (\varepsilon - 1)\lambda_i \\ 1 & -2 \end{vmatrix} \end{aligned}$$

Then, following constraints

$$\begin{cases} f(1) > 0, f(-1) < 0 \\ |-\mu\varepsilon\lambda_i| < 1, |b_0| > |b_2| \end{cases}$$

have to be satisfied according to the criterion. Therefore, the value range of ε is figured out as $1 < \varepsilon < 1 + \frac{1}{\lambda_i}$. After choosing the value of ε , one value satisfying $0 < \mu < 1 - \frac{1}{\varepsilon}$ could be chosen for μ . With $\rho(L)$ being assumed to be available,²⁰ the value ranges presented in this theorem can be gained by letting $\rho(L)$ take the place of λ_i .

When parameters are chosen within the ranges mentioned above, all eigenvalues of the system matrix are included in a unit circle, that is, the system matrix is Schur-stable. Then, values of all state vectors in the state-space expression are bounded, depending on the

Table 1. Chart of Jury stability criterion.

Linage	δ^0	δ^1	δ^2	δ^3
1	$-\mu\varepsilon\lambda_i$	$1 + (\varepsilon - 1)\lambda_i$	-2	1
2	1	-2	$1 + (\varepsilon - 1)\lambda_i$	$-\mu\varepsilon\lambda_i$
3	b_0	b_1	b_2	

bounded values of inputs due to ISS.¹⁹ Therefore, the whole system achieves stability. Till now, the proof is completed.

Remark 4. The simplicity of designing only two parameters may be at the expense of stability performance. If more emphasis has to be attached on this index, equation (5) can be modified as

$$\begin{cases} \mathbf{d}(k+1) = -\mathbf{Y}\mathbf{L}\tau(k) \\ \mathbf{u}_s(k+1) = \mathbf{L}\tau(k) + \mathbf{u}_s(k) + \mathbf{d}(k+1) - \mu\mathbf{d}(k) \end{cases}$$

where $\mathbf{Y} = \text{diag}\{\varepsilon_1, \dots, \varepsilon_N\}$ and $\mu = \text{diag}\{\mu_1, \dots, \mu_N\}$. In this way, more parameters have to be designed to make $1 < \varepsilon_i < 1 + \frac{1}{\lambda_i}$ and $0 < \mu_i < 1 - \frac{1}{\varepsilon_i}$ satisfied. However, better stability performance may be achieved.

Theorem 2. With Assumption 1 and Assumption 2, time of all virtual local clocks converges to the same time-varying stable value

$$\tau_c(k) = \frac{1}{N} \sum_{i=1}^N \left(\sum_{j=0}^{k-1} \tilde{\Delta}_i(j) + \tau_i(0) \right) \quad (11)$$

asymptotically under the presented protocols.

Proof: Above all, we let

$$e_i(k) = \bar{\tau}_i(k) - \frac{1}{N} \sum_{j=1}^N \bar{\tau}_j(k)$$

Then, for all nodes, its vector form is

$$\mathbf{e}(k) = \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \bar{\boldsymbol{\tau}}(k) \quad (12)$$

which implies that the average consensus of $\bar{\boldsymbol{\tau}}(k)$ can be achieved if $\mathbf{e}(k)$ converges to $\mathbf{0}$. Here, $\mathbf{0}$ denotes an N -dimension column vector whose entries are all 0. Since $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H$ can be obtained using equation (7), then with $\mathbf{L}\mathbf{1} = \mathbf{0}$, we get $\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H\mathbf{1} = \mathbf{0}$ which infers that $\boldsymbol{\Lambda}\mathbf{U}^H\mathbf{1} = \mathbf{0}$. Since $\boldsymbol{\Lambda}$ consists of λ_i in the order mentioned in equation (7), it can be further inferred that $\mathbf{U}^H\mathbf{1} = (x_1, 0, \dots, 0)^T$, where x_1 represents the first entry of this column vector. Then, we can get the following process of inference

$$\begin{aligned} \mathbf{U}^H \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{U} &= \mathbf{I} - \frac{1}{N} \mathbf{U}^H \mathbf{1}\mathbf{1}^T \mathbf{U} \\ &= \mathbf{I} - \frac{1}{N} (x_1, 0, \dots, 0)^T (x_1, 0, \dots, 0) \\ &= \text{diag} \left\{ 1 - \frac{1}{N} x_1^2, \mathbf{I}_{N-1} \right\} \end{aligned}$$

Due to the fact that $\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T$ is also a symmetric Laplacian matrix with a special form, 0 must be its

eigenvalue. In other words, $1 - \frac{1}{N} x_1^2$ must equate with 0. Then, the above inference is further inferred as

$$\mathbf{U}^H \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{U} = \text{diag}\{0, \mathbf{I}_{N-1}\}$$

Thus, by letting $\hat{\mathbf{e}}(k) = \mathbf{U}^H \mathbf{e}(k)$, equation (12) can be transformed as

$$\mathbf{U}^H \mathbf{e}(k) = \mathbf{U}^H \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \mathbf{U} \mathbf{U}^H \bar{\boldsymbol{\tau}}(k)$$

which infers

$$\hat{\mathbf{e}}(k) = \text{diag}\{0, \mathbf{I}_{N-1}\} \hat{\boldsymbol{\tau}}(k) \quad (13)$$

This implies that $\hat{e}_1(k) \equiv 0$, while $\hat{e}_i(k) = \hat{\tau}_i(k)$ when $i \geq 2$.

Then, the fact obtained in the previous subsection that $\hat{\Delta}(\infty)$ can be treated as being constant helps infer that state vectors in the state-space expression are constant when k tends to infinity. Therefore, formulas can be implied from equation (10) as

$$\begin{cases} \hat{u}_{s_i}(\infty) = (1 - \varepsilon) \lambda_i \hat{\tau}_i(\infty) + \hat{u}_{s_i}(\infty) - \mu \hat{d}_i(\infty) \\ \hat{d}_i(\infty) = -\varepsilon \lambda_i \hat{\tau}_i(\infty) \end{cases}$$

Combining these two expressions yields

$$(\mu\varepsilon + 1 - \varepsilon) \lambda_i \hat{\tau}_i(\infty) = 0$$

Then with $0 = \lambda_1 < \lambda_2 \dots \leq \lambda_N$ and $\mu\varepsilon + 1 - \varepsilon < 0$, $\hat{\tau}_i(\infty) \equiv 0$, $i \geq 2$, can be concluded. Thus, $\hat{\mathbf{e}}(\infty) \equiv \mathbf{0}$ can be yielded with the relation already inferred from equation (13). Since \mathbf{U}^H does not have 0 as its eigenvalue, it can be concluded that $\mathbf{e}(\infty) \equiv \mathbf{0}$ due to $\hat{\mathbf{e}}(k) = \mathbf{U}^H \mathbf{e}(k)$. Then till now, virtual local time of all nodes has been proved to be able to achieve asymptotic consensus under above protocols.

To make the virtual common-time value more explicit, the case $i = 1$ needs a specific discussion. By substituting $i = 1, \lambda_1 = 0$ into equation (10), we have

$$\begin{cases} \hat{\tau}_1(k+1) = \hat{\tau}_1(k) + \hat{u}_{s_1}(k) + \hat{\Delta}_1(k) \\ \hat{u}_{s_1}(k+1) = \hat{u}_{s_1}(k) - \mu \hat{d}_1(k) \\ \hat{d}_1(k+1) = \hat{d}_1(k) = 0 \end{cases}$$

By recursion of above expressions, $\hat{\tau}_1(k)$ can be expressed as

$$\hat{\tau}_1(k) = \sum_{j=0}^{k-1} \hat{\Delta}_1(j) + \hat{\tau}_1(0) + k \hat{u}_{s_1}(0)$$

As a constant factor of k , $\hat{u}_{s_1}(0)$ in the last term can be removed by letting $\mathbf{u}_s(0) = \mathbf{0}$ when that protocol gets initiated. In this way, a more compact form without uncertainty can be gained as

$$\hat{\tau}_1(k) = \sum_{j=0}^{k-1} \hat{\Delta}_1(j) + \hat{\tau}_1(0)$$

Due to the following process of inference

$$\begin{aligned} \lim_{k \rightarrow \infty} \bar{\tau}(k) &= \lim_{k \rightarrow \infty} \mathbf{U} \hat{\tau}(k) \\ &= \lim_{k \rightarrow \infty} \mathbf{U} \left(\hat{\tau}_1(k) \quad \dots \quad \hat{\tau}_N(k) \right)^T \\ &= \mathbf{U} \left(\sum_{j=0}^{k-1} \hat{\Delta}_1(j) + \hat{\tau}_1(0) \quad 0 \quad \dots \quad 0 \right)^T \\ &= \mathbf{U} \left\{ \text{diag}\{1, \mathbf{0}_{N-1}\} \mathbf{U}^H \left[\sum_{j=0}^{k-1} \tilde{\Delta}(j) + \bar{\tau}(0) \right] \right\} \\ &= \mathbf{U} \left\{ (\mathbf{I} - \text{diag}\{0, \mathbf{I}_{N-1}\}) \mathbf{U}^H \left[\sum_{j=0}^{k-1} \tilde{\Delta}(j) + \bar{\tau}(0) \right] \right\} \\ &= \left[\mathbf{I} - \left(\mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T \right) \right] \left[\sum_{j=0}^{k-1} \tilde{\Delta}(j) + \bar{\tau}(0) \right] \\ &= \frac{1}{N} \mathbf{1}\mathbf{1}^T \left[\sum_{j=0}^{k-1} \tilde{\Delta}(j) + \bar{\tau}(0) \right] \end{aligned}$$

the value of virtual common time can be denoted as

$$\tau_c(k) = \frac{1}{N} \sum_{i=1}^N \left(\sum_{j=0}^{k-1} \tilde{\Delta}_i(j) + \tau_i(0) \right) \quad (14)$$

Remark 5. It can be seen that virtual common time changes at a variation with the consensus value $\frac{1}{N} \sum_{i=1}^N \sum_{j=0}^{k-1} \tilde{\Delta}_i(j)$ which is achieved by all nodes asymptotically. Simultaneously, initial time errors also converge to the common value $\frac{1}{N} \sum_{i=1}^N \tau_i(0)$ which is the initial time of the virtual common clock. Synchronization of time variations implies that virtual time of all nodes can change at the same pace in a relatively long period of time. It decreases the frequency of executing re-synchronization, which reduces workloads and energy consumption of transmission and computation.

Besides, when clock skews are constant, consensus of local time variations also means consensus of local clock skews. In this case, it is easy to figure out that $\Delta_i = \alpha_i T$ due to all concerned definitions. So Δ_i is a constant value for node i . Therefore, there is no need for the additional control u_{Δ_i} to make $\tilde{\Delta}_i(k)$ constant. Instead, let $\tilde{\Delta}_i = \Delta_i$ directly.

Then, equation (14) can be transformed to

$$\tau_c(k) = \frac{1}{N} \sum_{i=1}^N (\alpha_i T k + \tau_i(0)) \quad (15)$$

Let α_c , $\tau_c(0)$, and T stand for the skew, initial time, and transmission period of the virtual common clock, respectively, then we get $\alpha_c = \frac{1}{N}(\alpha_1 + \dots + \alpha_N)$ and $\tau_c(0) = \frac{1}{N}(\tau_1(0) + \dots + \tau_N(0))$.

Implementation

Communication scheme

In this subsection, certain detailed attributes of the communication scheme presented in Remark 1 are supplemented. That communication means can be summarized as a reactive flooding scheme with the MAC-layer time-stamping. The MAC-layer technique does not only have the function mentioned in Remark 1, but also has the function of preventing transmission collisions. Specifically, message transmission between two nodes does not interfere with transmission between other nodes. Reactive, similar with the term post facto in RBS,⁴ represents that sensor nodes are triggered to find communication links instead of getting routing information known and stored proactively. This attribute leads to energy savings for the fact that no energy is consumed for attempting to maintain routines at all time. The flooding scheme has been explained in Remark 1. By flooding, packets are transmitted only over multiple short distances between pairs of neighbors instead of a single long path, which reduces energy consumption.

Application with asynchronous communication

Modified communication scheme. In WSNs, if message channels are full duplex, the communication graph can be connected undirected due to Assumption 2. However, in most cases of WSNs, sensor nodes cannot transmit packets immediately when they receive packets. That is to say, these channels are half-duplex, which implies that underlying communication graphs on sampling instants are unidirectional. Here, it is worth reminding that the message-channel graph of the whole net is still connected undirected. Then, the communication scheme in Remark 1 has to be modified so as to be applicable in this case. Node i updates its own virtual time-stamp immediately when it records its local physical time, which is triggered by receiving the incoming packet from node j . This event's global instant is denoted as \bar{t}_{ik}^j . Then node i enters the state of waiting for broadcasting its out-packets. When one channel is ready for transmission on the global instant t_{ik} which is after relatively short time delay from \bar{t}_{ik}^j , node i broadcasts its current virtual time to its neighbors. And with information of k contained in packets, out-packet broadcasting of each node can be triggered only once in the k th transmission. Thus, it can be seen that every time when a node receives one in-packet, it updates its virtual time without necessity in waiting for all neighbors' messages. Therefore, robustness to packet losses and node failure can be gained. This communication scheme with the MAC-layer stamping technique also contributes to the generation of extended

neighbor graphs which generate non-symmetric stochastic matrices.

Now let $\cup_i \{t_{ik}\}_{k=0}^{+\infty}$ denote the set whose elements are all broadcasting instants of all nodes. By rearranging its elements due to their values, it can also be denoted as an ordered set $\mathfrak{T} = \{\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_\nu, \dots\}$, where \mathcal{T}_ν increases monotonically as ν increases. Relative to the broadcasting instant of node 1 in the k th transmission, other nodes' broadcasting instants have time delay $t_i^k, i = 2, \dots, N$. And corresponding to the maximum delay value t_{\max}^k , we define $\bar{i} = \arg_{i \in N} \max t_i^k$. Then, one component in \mathfrak{T} whose value corresponds to $t_{\bar{i}k}$ is denoted as $\mathcal{T}_{\bar{i}k}$. Since T_k determined by node 1 generally has a much larger timescale than the time delay does, $\bar{\nu}_{k+1} - \bar{\nu}_k = N$ can be satisfied under the above broadcasting scheme. Namely, it can be ensured that the broadcasting instant of each node appears once in the time interval $(\mathcal{T}_{\bar{i}k}, \mathcal{T}_{\bar{i}k+1}]$ —that is, for $\forall i$, there always exists k and l so that $\mathcal{T}_l = t_{i(k+1)}$, where $\bar{\nu}_k < l \leq \bar{\nu}_{k+1}, k = 0, 1, \dots$. Then, due to the above statements and the first theorem in literature,⁹ the union of communication graphs on all event instants within $(\mathcal{T}_{\bar{i}k}, \mathcal{T}_{\bar{i}k+1}]$ can be treated as one connected undirected graph. Thus, the protocols can be applied during the time interval.

Comparison to existing methods. Compared with existing ideas of implementation,⁹ our method of addressing the asynchronous problem is more practical. Actually, Schenato and Fiorentin⁹ implicitly assume that one of the following two conditions has to establish when its implementation method is applied. One is that all nodes start the synchronization on the same global instant. The other is that all physical-clock skews are constant. However, both conditions are not practical due to the problem formulation. Specifically, a counter-example is shown in Figure 4, where there are four nodes making up the net. From the literature,⁹ $T_i^k = t_{i(k+1)} - t_{ik}$ denotes the transmission period of node i and T_{\max}^k represents the maximum value among all nodes' periods in

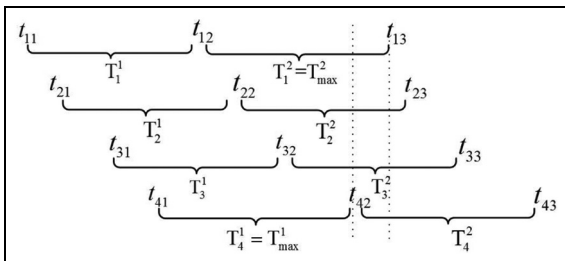


Figure 4. A counter-example of the existing implementation method.

the k th transmission. Then, transmitting instants of each node should have appeared at least once in the time interval $[t_{42}, t_{13}]$. However, this is obviously not true according to Figure 4.

As for the method proposed in this article, from subsection “Modified communication scheme,” $\mathfrak{T} = \{t_{11}, t_{21}, t_{31}, t_{41}, t_{12}, t_{22}, t_{32}, t_{42}, t_{13}, t_{23}, t_{33}, t_{43}\}$ can be obtained. Thus, we can get $\mathcal{T}_0 = t_{11}, \mathcal{T}_{\bar{\nu}_1} = t_{41}, \mathcal{T}_{\bar{\nu}_2} = t_{42}, \mathcal{T}_{\bar{\nu}_3} = t_{43}$ and $\bar{\nu}_{k+1} - \bar{\nu}_k = 4, k = 1, 2, 3$. And it can be seen that the broadcasting instant of each node appears once in the time interval $(\mathcal{T}_{\bar{\nu}_k}, \mathcal{T}_{\bar{\nu}_k+1}]$, $k = 1, 2, 3$. Therefore, confronted with the same case, our protocols can be applied in these intervals with the modified communication scheme. That’s to say, in contrast, our method is more applicable in practice.

Stopping criterion for synchronization protocols

Though it has been proven that clock synchronization can be achieved asymptotically, Pottie and Kaiser²¹ show that a considerable amount of energy has to be conserved in practice by finishing the synchronization process in finite time. Yadav and Salapaka²² has proved that average-consensus protocols whose underlying graphs are undirected or strongly connected balanced can be ended in finite time by designing minimum and maximum consensus protocols.

Since synchronization of clock time and that of time variations are achieved simultaneously, our stopping criterion is to utilize virtual-time variations. Thus, the protocols are designed as

$$\begin{aligned} y_i(k+1) &= \max_{j \in \mathcal{N}_i} y_j(k) \\ z_i(k+1) &= \max_{j \in \mathcal{N}_i} z_j(k) \end{aligned} \quad (16)$$

where $y(0) = \bar{\Delta}(0)$, $z(0) = \bar{\Delta}(0)$, and $\bar{\Delta}_i(k) = \bar{\tau}_j(k+1) - \bar{\tau}_j(k)$. From the literature,²² after every \mathcal{D} . time steps, minimum and maximum values among virtual-time variations increase and decrease strictly, respectively. And all nodes can have access to the value of \mathcal{D} by a distributed method presented in the literature.²³ It is worth remarking that we take the sampling period as one time step in this criterion. Then, we define $t_{h\mathcal{D}} = t_0 + \sum_{j=0}^{h\mathcal{D}-1} T_j$, $h = 1, 2, \dots$ as the global time instants when the maximum/minimum protocols are reset, that is, $y_i(h\mathcal{D})$ and $z_i(h\mathcal{D})$ of the above two protocols are set to be equal with the virtual variation $\bar{\Delta}_i(h\mathcal{D} - 1)$ respectively. Maximum and minimum consensus values are defined as \bar{x}_h and \underline{x}_h , respectively. Then, the proposed synchronization protocols can stop when difference $\varepsilon_h = \bar{x}_h - \underline{x}_h < \rho$, where ρ is a preset parameter which is small enough to ensure \bar{x}_h and \underline{x}_h close enough to the desired consensus value.

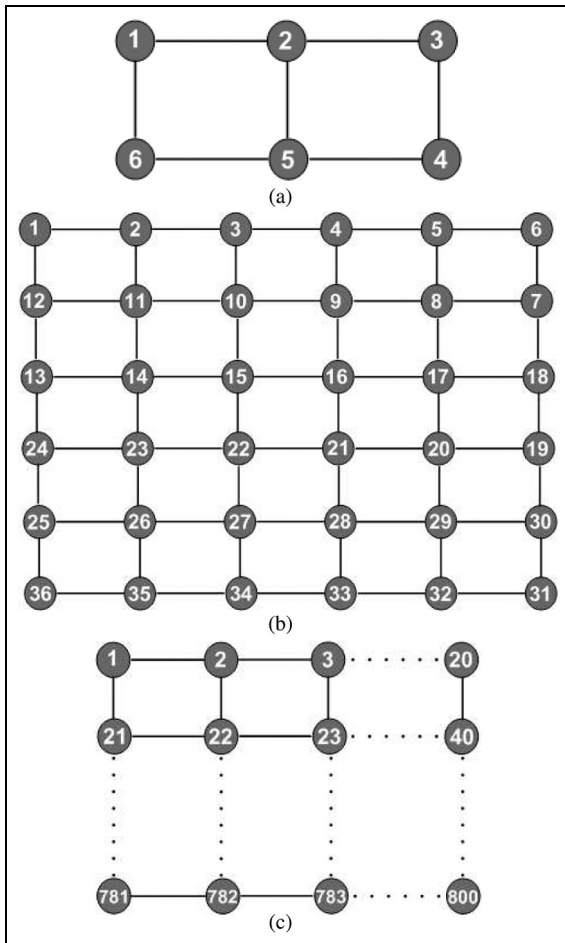


Figure 5. Topology graphs of nets with different scales: (a) topology graph of a 6-node grid, (b) topology graph of a 36-node grid, and (c) topology graph of an 800-node grid.

After all nodes stop exchanging messages and running protocols for synchronization, they can return to low-power states to save energy. Then a whole strategy for clock synchronization is accomplished.

Numerical simulations

In this section, performance of proposed protocols is illustrated by presenting certain simulation results.

We consider three examples of WSN grids consisting of 6 nodes, 36 nodes, and 800 nodes, respectively, for two main reasons. On one hand, the scalability of our proposed method can be illustrated when the scale of the network differs. On the other hand, details that may not be seen clearly in the figure of a large net can be displayed explicitly in the figure of a small net.

Above all, Figure 5 shows the underlying graphs of nets with different scales, on which our simulation is carried. It can be seen from the figure that all graphs are connected undirected. Figure 6 shows how local

time of all nodes in Figure 5 changes corresponding to the physical time before synchronization. In the figure, curves with different colors represent local clocks of different nodes.

Simulation of synchronization process

Parameters in the protocols and those needed in the simulation are shown in Table 2.

Then, running the protocols under the graph shown in Figure 5(a) can get the simulation results shown in Figure 7. Changes of virtual time of all local clocks corresponding to global physical time t_k are shown in Figure 7(a). It can be seen that initial time errors are compensated and synchronization errors reduce asymptotically.

In this figure, black asterisk markers represent the scatter diagram of the virtual common clock in equation (14). Thus, it can be seen that the synchronization curves can follow the form of equation (14) finally.

Figure 7(b) shows the consensus of virtual-time variations of all local clocks. In this figure, black asterisk markers represent the scatter diagram of the virtual common-time variations as part of equation (14). Then, the two figures show that virtual-time and time variations can almost simultaneously converge to consensus values shown in equation (14). Then according to the definition of \mathcal{D} , the diameter of the graph shown in Figure 5(a) is 4. Thus, running stopping protocols under this graph can get Figure 8 which shows the changes of virtual-time variations with the stopping criterion.

In this figure, the blue circle is the stopping margin picked out by the criterion when ρ is assigned the value in Table 2.

Simulation of changing topologies

With Assumption 2 established, this subsection shows the simulation results when there are changes in the topology.

One more node is added to the net. This simulates certain physical cases. For example, there may be a node off-line or off-power at first. Then, on some physical instant, it is put on line or on power. And when the node is added into the net, other nodes may have been in the synchronization process for a spell. Then, the above situation can be equivalent to the change from Figures 5(a) to 9(a). And the simulation results of this process are shown in Figure 10.

In Figure 10(a), the cyan asterisk denotes the local clock time of node 7 when it is added into the net, and the cyan curve represents the changes of its virtual clock time after it joins with other nodes in running

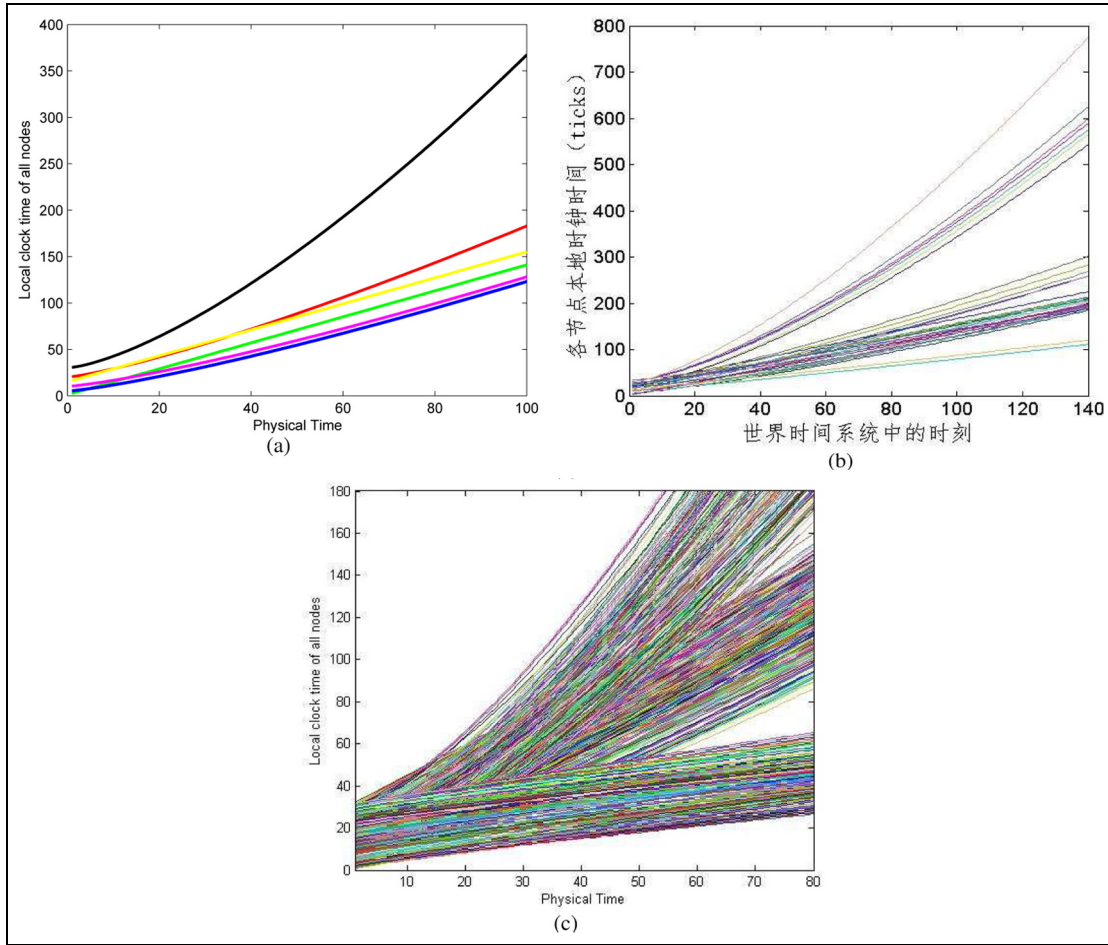


Figure 6. Local clock time of all nodes before synchronization: (a) 6-node grid, (b) 36-node grid, and (c) 800-node grid.

Table 2. Simulation parameters.

Parameters in protocols			
ε	μ	$\rho(L)$	ρ
1.30	0.23	1.60	0.5
Network-level parameters			
Description	Settings		
Communication standard	IEEE 802.15.4		
Transmission rate	250 kbps		
MAC protocol	iQueue-MAC ²⁴		
Packet size (L_p)	960 bits		
Simulation time	3 h		

protocols. Similarly, in Figure 10(b), the cyan curve shows the changes of virtual-time variations of node 7. From these figures, we can see that when the new node is added, the previous trend of synchronization is disturbed; however, after node 7 runs the protocols, all nodes can achieve the synchronization eventually.

One node is lost in the net. This can simulate the cases that there are packet losses or node failure in the net. These cases can be equivalent to the change from Figure 5(a) to Figure 9(b). Figure 9(b) can represent that node 3 and node 5 cannot acquire the packets from node 4 or node failure happens to node 4. Then,

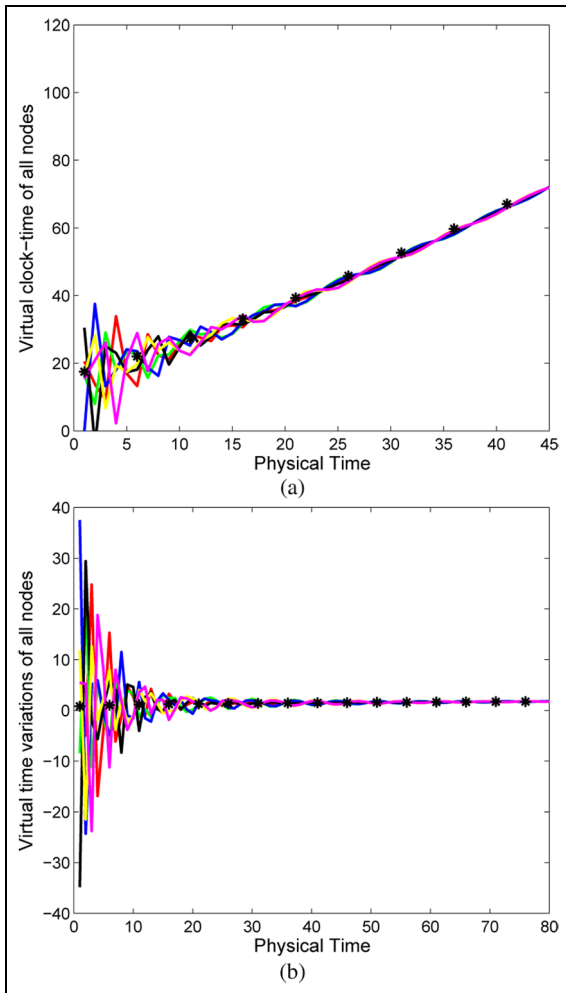


Figure 7. Clock synchronization of the grid with six nodes: (a) changes of virtual time of all local clocks and (b) changes of virtual-time variations of all local clocks.

simulation results of this process are shown in Figure 11. In Figure 11(a), the black asterisk denotes the virtual local time of node 4 when it is lost. And it can be seen that after the loss of node 4, the virtual-time errors between other nodes become large for a while, which is specifically shown in Figure 12. Figure 12 is obtained by taking the virtual clock of node 1 as the reference. With the coordinate (1,1) in this figure corresponding to node 1 in Figure 9(b), the X-Y plane can denote the WSN grid in Figure 9(b). Figure 12(a) shows the virtual-time errors on the instant before the loss of node 4, and Figure 12(b) shows the virtual-time errors on the instant after the loss. Then, it can be seen that errors become large between these two instants. However, after remained nodes continue to run protocols, the synchronization can be achieved finally. Similarly, Figure 11(b) shows that the loss of node 4 can influence the errors between virtual-time variations in the process; however, virtual-time variations of all

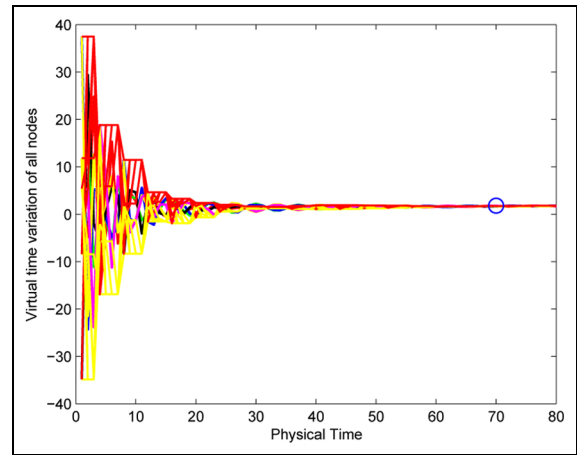


Figure 8. Changes of virtual-time variations with the stopping criterion.

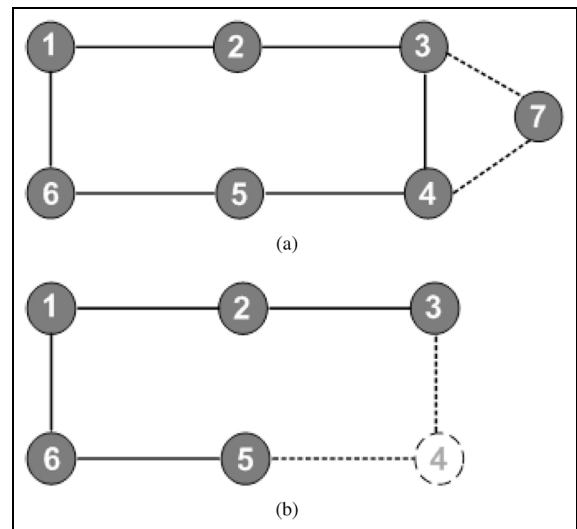


Figure 9. Topology graphs after changes: (a) number of nodes increases and (b) number of nodes decreases.

remained nodes can still achieve the synchronization eventually. It is worth mentioning that the black lines in Figure 11 show how virtual-time and time variations of node 4 would change without connection with the net.

Simulation of the nets with larger scales

In this section, our protocols are run under the graphs shown in Figure 5(b) and (b), respectively. Then, simulation results can be obtained as shown in Figures 13 and 14 respectively. From these two figures, it can be seen that synchronization of virtual clock time and that of virtual-time variations can still be achieved in the larger nets. So, our protocols can be applied in nets with different scales. Due to the definition of \mathcal{D} , the

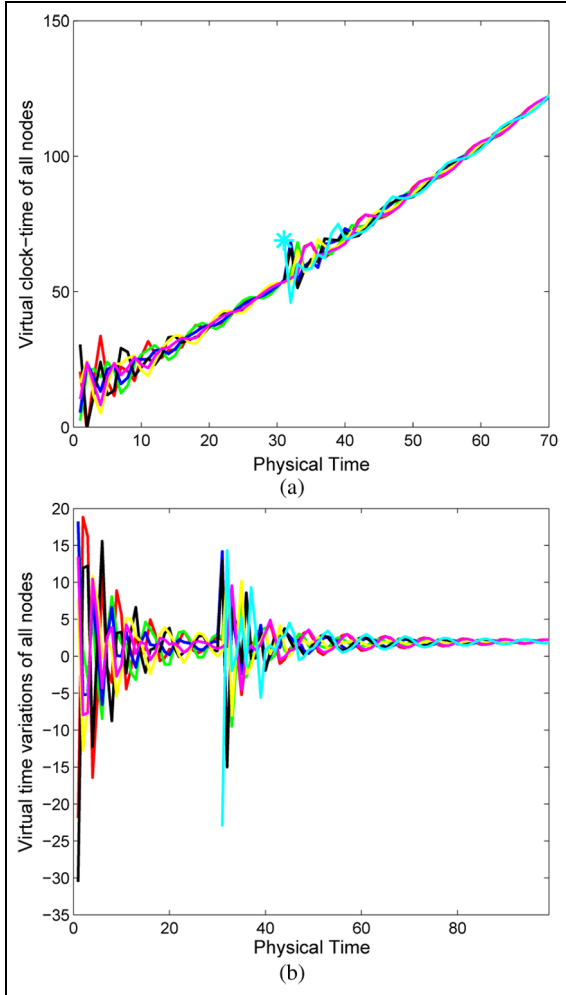


Figure 10. Synchronization of the net with one more node: (a) synchronization of virtual local time and (b) synchronization of virtual-time variations.

diameter of the graph shown in Figure 5(b) is 11. And here, we let $\rho = 1.5$ as the preset synchronization accuracy. Then, in Figure 13(b), the blue circle is still the margin picked out by the stopping criterion. And it is worth mentioning that the time of achieving the same preset synchronization accuracy in the larger net is longer than that in the smaller net by comparing Figure 13(b) with Figure 8.

Comparison of synchronization errors

In this section, protocols in ATS⁹ method and those in our method are run under the graph shown in Figure 5(b). Then, Figure 15 is obtained by taking the virtual clock of node 1 as the reference to compare the synchronization errors of these two methods. With coordinates (1,1,0), (2,1,0) and (1,2,0) in this figure corresponding to node 1, 2 and 12 in Figure 5(b) respectively, the X-Y plane can denote the WSN grid in Figure 5(b). From

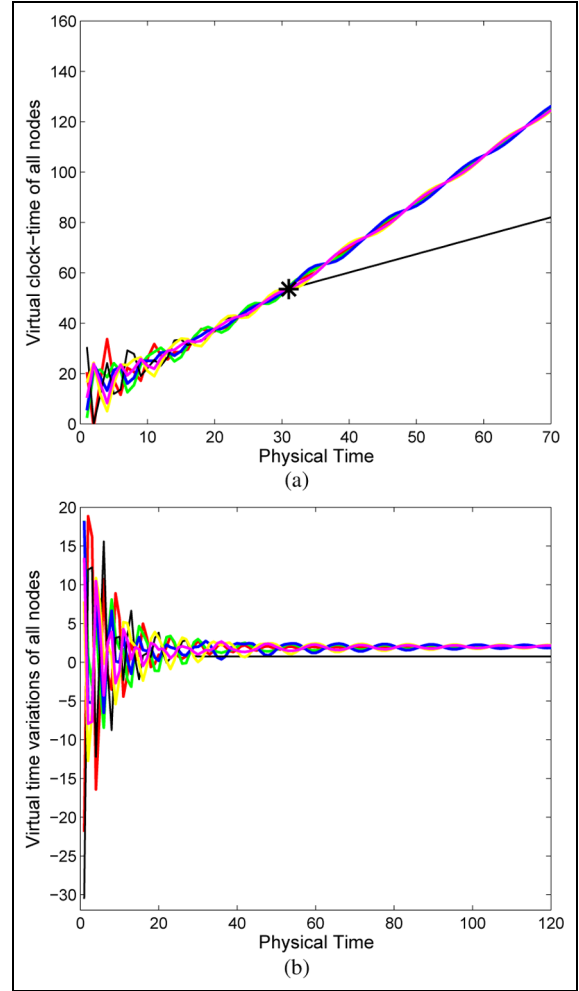


Figure 11. Synchronization of the net with one node lost: (a) synchronization of virtual local time and (b) synchronization of virtual-time variations.

Figure 15, it can be seen that synchronization errors between virtual clocks from both methods are almost proportional to the physical distance between nodes, which is an advantage over TPSN.⁶ And comparing Figure 15(a) with Figure 15(b) shows that our method has smaller synchronization errors than the ATS⁹ method does under the graph shown in Figure 5(b).

Energy consumption analysis

In this section, a recognized energy consumption model²⁵ for WSNs is utilized to analyze the energy consumption of our protocol. This model contains two main expressions as follows

$$E_{tx}(L_p, \bar{d}) = \begin{cases} L_p E_{elec} + L_p \epsilon_{fs} \bar{d}^2, & \bar{d} < \bar{d}_0 \\ L_p E_{elec} + L_p \epsilon_{mp} \bar{d}^4, & \bar{d} \geq \bar{d}_0 \end{cases} \quad (17)$$

$$E_{rx}(L_p) = L_p E_{elec}$$

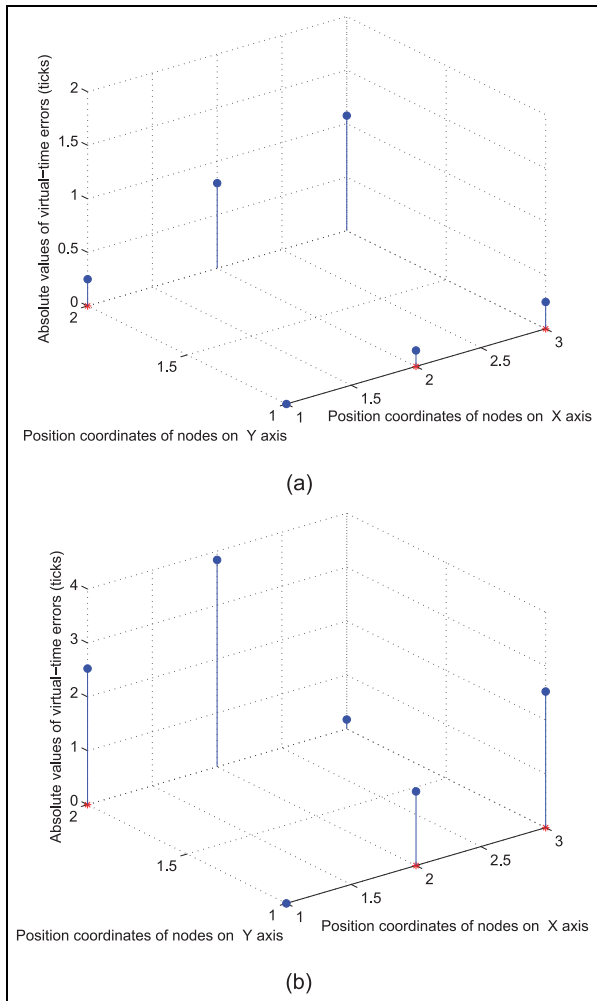


Figure 12. Comparison of errors before and after the loss: (a) errors before the loss and (b) errors after the loss.

where fs is short for free-space model and mp is short for multi-path model. The physical meanings and typical values of all symbols in the above equations are shown in Table 3.

Then, with the focus on relatively short-distance transmissions, the transmission distance between nodes in the topology shown in Figure 5(a) is set as $\bar{d} = 30$ m. And the initial energy of each node is set as $E_0 = 2$ J. Then, the simulation results of energy consumption can be obtained in Figure 16. Figure 16(a) shows the residual energy of each node after the k th packet transmission of the grid, where $k = 1, 2, \dots, 70$. Due to the stopping margin shown in Figure 8, the residual energy on $k = 70$ in Figure 16(a) is the ultimate energy of each node when synchronization is completed. Then, it can be seen that packet exchanges for our protocol can be achieved with reasonable energy consumption. Besides, the capacity of batteries on sensors is much larger than E_0 , which can support the application of our protocols.

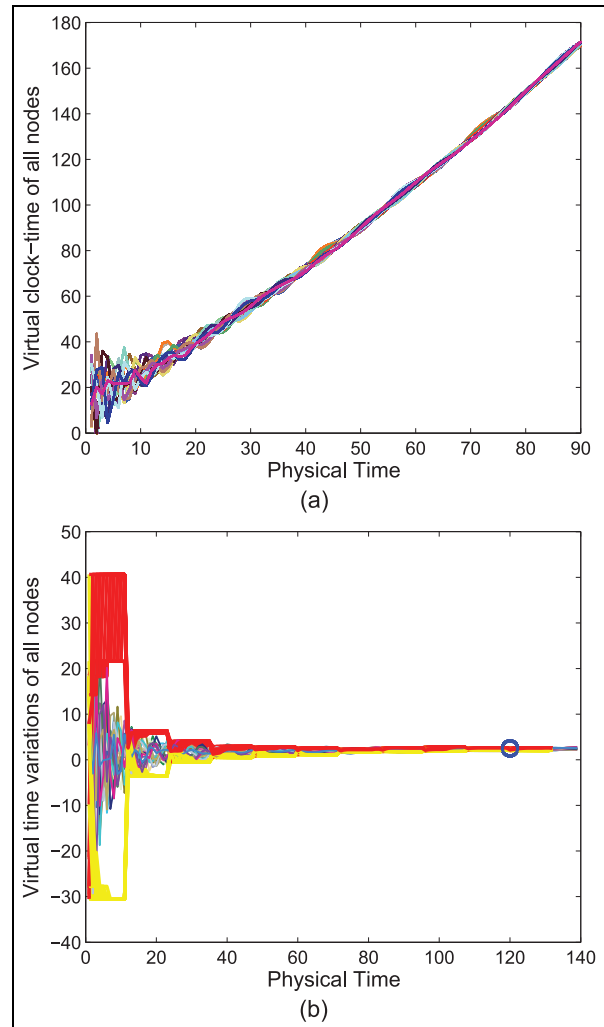


Figure 13. Clock synchronization of the grid with 36 nodes: (a) changes of virtual time of all local clocks and (b) changes of virtual-time variations of all local clocks.

Figure 16(b) shows the energy consumption of each node per 3 min with different packet transmission periods. Then, it can be seen that when the transmission period is small, the energy consumption can still be reasonable. And a tradeoff between the convergence rate and energy consumption can be made by choosing the values of transmission period.

Conclusion and future work

A new distributed protocol is proposed to achieve synchronization of clock time and that of time variations simultaneously. Its control inputs are robust against noise interference with achievable bounded values. Also, this article provides specific explanations on the virtual clock modeling and communication schemes independent on global physical time. The

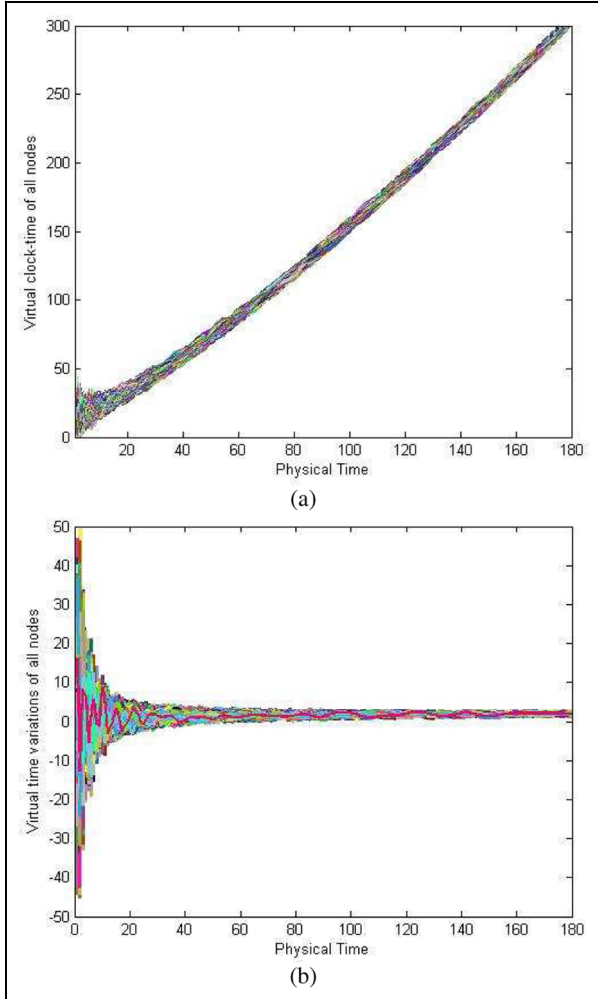


Figure 14. Clock synchronization of the grid with 800 nodes: (a) changes of virtual time of all local clocks and (b) changes of virtual-time variations of all local clocks.

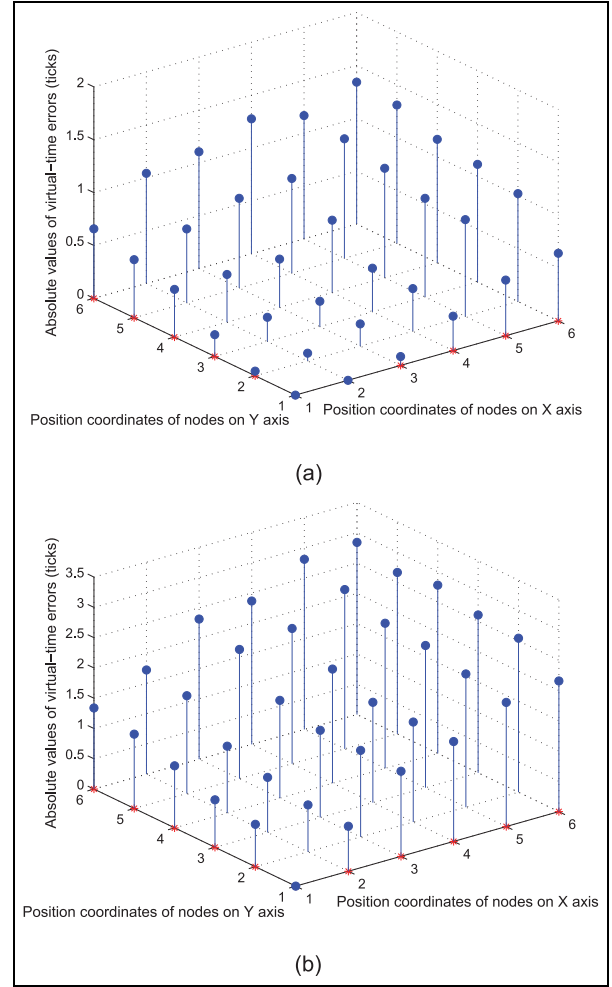


Figure 15. Comparison of synchronization errors: (a) synchronization errors of our method and (b) synchronization errors of the existing method.

Table 3. Symbols and their typical values in the energy consumption model.

Symbols	Meaning	Values
$E_{tx}(L_p, \bar{d})$	Energy for transmitting L_p bits over \bar{d} meters	
$E_{rx}(L_p)$	Energy for receiving L_p -bit data	
E_{elec}	Energy for receiving or transmitting 1-bit data	50 nJ/bit
ϵ_{fs}	Power of transmission amplifier	100 pJ/bit/m ²
ϵ_{mp}	Power of transmission amplifier	0.0013 pJ/bit/m ⁴
\bar{d}_0	Distance threshold	80 m

modified communication scheme enhances robustness against node failure and packet losses and then has a wider application. Finding a stopping margin for the protocols in finite time can conserve more energy. Future work may include researches on optimal

choices of parameters to gain better stability and consensus performance. And we plan to conduct more experiments to compare the performance of our method with that of others as certain literature²⁶ did in the future.

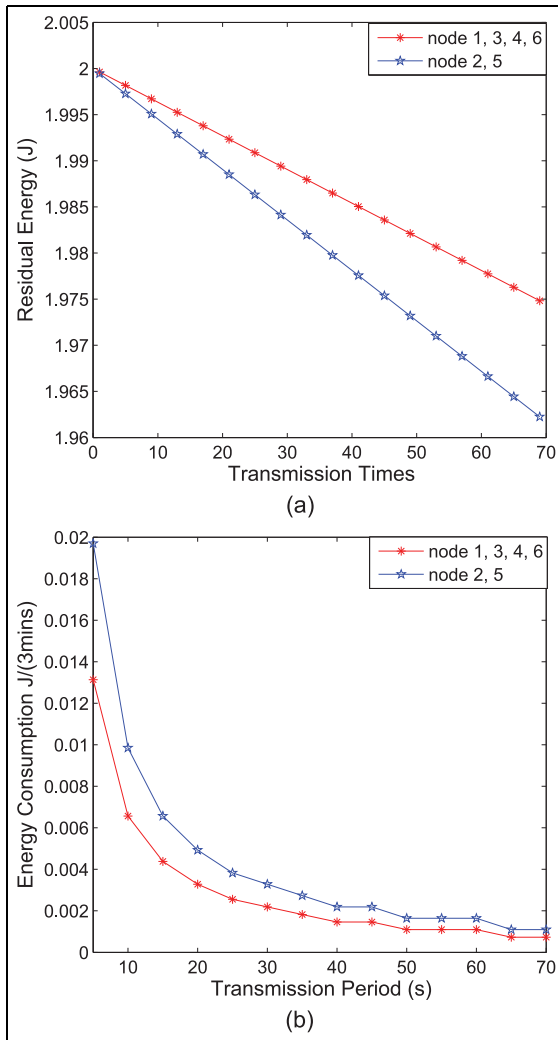


Figure 16. Energy consumption of the grid with six nodes: (a) changes of residual energy due to transmission times and (b) energy consumption with different packet transmission periods.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The work was supported by the National Natural Science Foundation of China (grant no. 61273113) and the Zhejiang Provincial Natural Science Foundation of China (grant no. LR13F030002).

References

- Sundaraman B, Buy U and Kshemkalyani AD. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Netw* 2005; 3: 281–323.
- Kaur B and Kaur A. A survey of time synchronization protocols for wireless sensor networks. *Int J Comput Sci Mob Comput* 2013; 2: 100–106.
- Faizulkhakov YR. Time synchronization methods for wireless sensor networks: survey. *Progr Comput Softw* 2007; 79: 214–226.
- Elson J, Girod L and Estrin D. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Oper Syst Rev* 2002; 36: 147–163.
- Su W and Akyildiz I. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE ACM Trans Netw* 2005; 13: 384–397.
- Ganeriwal S, Kumar R and Srivastava M. Timing-sync protocol for sensor networks. In: *Proceedings of the international conference embedded networked sensor systems*, Los Angeles, CA, 5–7 November 2003, pp.138–149. New York: ACM.
- Mills D. Internet time synchronization: the network time protocol. *IEEE Trans Commun* 1991; 39: 1482–1493.
- Maroti M, Simon G, Kusy B, et al. The flooding time synchronization protocol. In: *Proceedings of the international conference embedded networked sensor systems*, Baltimore, MD, 3–5 November 2004, pp.39–49. New York: ACM.
- Schenato L and Fiorentin F. Average TimeSync: a consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* 2011; 47: 1878–1886.
- Yoon S, Veerarittiphan C and Sichitiu ML. Tiny-sync: tight time synchronization for wireless sensor networks. *ACM Trans Sens Netw* 2007; 3: 8.
- Wu J, Zhang L, Bai Y, et al. Cluster-based consensus time synchronization for wireless sensor networks. *IEEE Sens J* 2015; 15: 1404–1413.
- Swain AR and Hansdah RC. A weighted average-based external clock synchronisation protocol for wireless sensor networks. *Int J Sens Netw* 2012; 12: 89–105.
- Terraneo F, Rinaldi L, Maggio M, et al. FLOPSYNC-2: efficient monotonic clock synchronisation. In: *Proceedings of the IEEE real-time systems symposium*, Rome, 2–5 December 2014, pp.11–20. New York: IEEE.
- Maggs MK, O’Keefe SG and Thiel DV. Consensus clock synchronization for wireless sensor networks. *IEEE Sens J* 2012; 12: 2269–2277.
- Simeone O, Spagnolini U, Bar-Ness Y, et al. Distributed synchronization in wireless networks. *IEEE Signal Process Mag* 2008; 25: 81–97.
- He JP, Cheng P, Shi L, et al. Time synchronization in WSNs: a maximum-value-based consensus approach. *IEEE Trans Autom Control* 2014; 59: 660–674.
- Agavev R and Chebotarev P. On the spectra of nonsymmetric Laplacian matrices. *Linear Algebra Appl* 2005; 399: 157–178.
- Mock M, Frings R and Nett E. Continuous clock synchronization in wireless real-time applications. In: *Proceedings 19th IEEE symposium on reliable distributed systems SRDS-2000*, Nurnberg, 16–18 October 2000, pp.125–132. New York: IEEE.
- Khalil HK. Lyapunov stability. In: Khalil HK (ed.) *Non-linear systems*. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002, pp.174–180.

20. Zhang XD. Two sharp upper bounds for the Laplacian eigenvalues. *Linear Algebra Appl* 2004; 376: 207–213.
21. Pottie G and Kaiser W. Wireless integrated network sensors. *Commun ACM* 2000; 43: 51–58.
22. Yadav V and Salapaka MV. Distributed protocol for determining when averaging consensus is reached. In: *Proceedings of the annual allerton conference*, Monticello, IL, 26–28 September 2007, pp.715–720.
23. Haldar S. An ‘All Pairs Shortest Paths’ distributed algorithm using $2n^2$ messages. *J Algorithm* 1997; 24: 20–36.
24. Zhuo SG, Wang Z, Song YQ, et al. A traffic adaptive multi-channel MAC protocol with dynamic slot allocation for WSNs. *IEEE Trans Mob Comput* 2016; 15: 1600–1613.
25. Heinzelman WB, Chandrakasan AP and Balakrishnan H. An applicaiton-specific protocol architecture for wireless microsensor networks. *IEEE Trans Wirel Commun* 2002; 1: 660–670.
26. He JP, Li H, Chen JM, et al. Study of consensus-based time synchronization in wireless sensor networks. *ISA Trans* 2014; 53: 347–357.