ISA Transactions 53 (2014) 347–357

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research Article

Study of consensus-based time synchronization in wireless sensor networks

Jianping He, Hao Li, Jiming Chen, Peng Cheng *

State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Zheda Road 38#, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 3 May 2013
Received in revised form
22 October 2013
Accepted 2 November 2013
Available online 25 November 2013
This paper was recommended for publication by Dr. Q.-G. Wang

Keywords:
Wireless sensor networks
Consensus-based time synchronization
Maximum and minimum consensus
Experimental study

ABSTRACT

Recently, various consensus-based protocols have been developed for time synchronization in wireless sensor networks. However, due to the uncertainties lying in both the hardware fabrication and network communication processes, it is not clear how most of the protocols will perform in real implementations. In order to reduce such gap, this paper investigates whether and how the typical consensus-based time synchronization protocols can tolerate the uncertainties in practical sensor networks through extensive testbed experiments. For two typical protocols, i.e., Average Time Synchronization (ATS) and Maximum Time Synchronization (MTS), we first analyze how the time synchronization accuracy will be affected by various uncertainties in the system. Then, we implement both protocols on our sensor network testbed consisted of Micaz nodes, and investigate the time synchronization performance and robustness under various network settings. Noticing that the synchronized clocks under MTS may be slightly faster than the desirable clock, by adopting both maximum consensus and minimum consensus, we propose a modified protocol, MMTS, which is able to drive the synchronized clocks closer to the desirable clock while maintaining the convergence rate and synchronization accuracy of MTS.

1. Introduction

Time synchronization in wireless sensor networks (WSNs) is crucial in many applications, e.g., mobile target tracking [1], even detection [2], coverage [3], speed estimating [4], management [5], monitoring [6], etc. An accurate synchronized time can help to extend the applications, and to save energy by providing a coordinated wake-up and sleep schedule [7]. More detailed surveys on the importance of time synchronization in WSNs are given in [8,9].

Many time synchronization protocols have been proposed in the past few years, e.g., RBS [10], TPSN [11], FTSP [12], etc. However, most of these protocols are root-based or tree-based time synchronization protocols, which are sensitive to the dynamic network topology. Thus, in order to enhance the robustness and scalability of the protocols, consensus concept, e.g., average consensus [13,14], has been introduced to solve the time synchronization problem in WSNs recently, which is called consensus-based time synchronization [15–18]. Compared with the traditional root-based or tree-based time synchronization protocols, consensus-based time synchronization protocols are fully distributed without requiring any certain reference node. Meanwhile, the consensus-based time synchronization protocols are able to simultaneously compensate both the clock offset, i.e., instantaneous clock difference, and the clock skew, i.e., clock speed, which can prolong the re-synchronization period and thus reducing communication and energy costs. The existing consensus-based time synchronization protocols can be divided into two categories, i.e., average consensus-based, e.g., [16,19,20], and maximum consensus-based, e.g., [17].

There have been some theoretical and simulation investigations over different consensus-based time synchronization protocols. These investigations are mostly based on some assumptions, such as reliable communication, no communication delay, and no packet loss [16,17]. However, in real wireless sensor network, these assumptions do not hold again, e.g., the communication delay is inevitable, which may render the theoretical results false. Hence, the experimental verification of time synchronization protocols in real hardware testbed is essential and meaningful, as the experimental results can provide strong support to apply the time synchronization protocols to applications [10,11,16]. Considering numerous challenges in experimental study due to various uncertainties and interferences, it is of great interest to evaluate and compare the consensus-based time synchronization protocols by experimental study. In this paper, we establish a sensor network testbed and evaluate the performance of two typical consensus-based time synchronization protocols, Average TimeSynch (ATS) [16] and Maximum TimeSynch (MTS) [17],

* Corresponding author.
E-mail address: pcheng@iipc.zju.edu.cn (P. Cheng).

0019-0578/$ - see front matter
http://dx.doi.org/10.1016/j.isatra.2013.11.001

through extensive experiments. The main contributions of this paper are summarized as follows:

1. To the best of our knowledge, it is the first experimental work on evaluating and comparing the performance of two latest consensus-based time synchronization protocols, i.e., ATS and MTS.
2. We analytically establish the relationship between the time synchronization accuracy and the bound of uncertainty for both ATS and MTS. We implement ATS and MTS on our sensor network testbed with around 20 Micaz nodes. Under different experimental system parameters, the time synchronization performance and robustness for ATS and MTS are compared through extensive experiments. The experimental results show the advantages of MTS over ATS in terms of convergence rate and synchronization accuracy.
3. Noticing that the synchronized clocks under MTS may be slightly faster than the desired clock, we propose a revised protocol, MMTS by adopting both maximum consensus and minimum consensus. The experimental results show that MMTS is able to drive the network-wide clocks closer to the desirable clock while maintaining the advantages of MTS.

The remainder of the paper is organized as follows. Section 2 discusses the related work. In Section 3, the preliminaries for both ATS and MTS are introduced. Section 4 analyzes the relationship between the time synchronization accuracy and the bound of uncertainty for both ATS and MTS. Section 5 presents the experimental testbed. The experimental results on ATS and MTS, and the revised protocol, MMTS, are given in Section 6. Finally, Section 7 draws conclusions.

2. Related work

Many experimental works have been provided to validate different traditional time synchronization protocols [10,11,21,22]. For example, some experiments under the traditional root-based or tree-based time synchronization protocols such as RBS [10] and TPSN [11] have been done in a sensor network testbed. In these experiments, one reference node is essential according to the protocols. In [21], the authors model the error uncertainty of skew detection with Micaz motes, and propose an On-Demand Synchronization (ODS) protocol. ODS develops an uncertainty-driven mechanism to adjust clock calibration interval of each node adaptively and individually instead of periodic synchronization to reduce energy consumption. The experiments under ODS are carried out on the Micaz mote platform. Ferrari et al. in [22] present a new flooding architecture Glossy for WSNs, which achieves fast flooding in the whole network and implicit time synchronization. Based on Berkeley Mica2 motes testbed, Solis et al. in [19] implement a distributed algorithm to achieve accurate time synchronization in large multihop wireless networks by exploiting the large number of global constraints. Based on CC2430 chips, the experiments on Feedback-Based Synchronization (FBS) are conducted in [23]. In [24], Round-Robin Timing Exchange (RRTE) protocol is proposed for small-scale WSNs and implemented on a Zigbee development kit, U-NET01. However, only a few experimental studies have been carried out for consensus-based time synchronization protocols [16]. Especially, there have been no experimental studies on the comparisons between the average consensus-based protocols, e.g., ATS, and the maximum consensus-based protocols, e.g., MTS. Hence, we conduct extensive experiments to evaluate and compare consensus-based protocols ATS and MTS in this paper.

The uncertainties, e.g., communication delay and measurement noise, are still fundamental limits in time synchronization [25], which makes the perfect time synchronization quite difficult. Hence, recently, there are some works provided to solve these problems [26–29]. Based on the two-way message-exchange mechanism, Leng and Wu propose three estimators for the joint estimation of the clock offset and skew under unknown delay [26] and a low-complexity Maximum likelihood estimator under exponential delays [27]. In [28], the Maximum Likelihood Estimates (MLE) of the clock offset are obtained under symmetric exponential link delays, and the authors present improved estimators based on MLE and generalize its scope by considering asymmetric exponential delays in [29]. This paper studies the performance of ATS and MTS under uncertainties through theoretical analysis and experiments.

3. Preliminaries

3.1. Clock model

Referring to [7,16,17,30], assume that each node i in the network is equipped a hardware clock with its clock reading $\tau_i(t)$ as

$$\tau_i(t) = \alpha_i t + \beta_i, \tag{1}$$

where α_i is the hardware clock skew and β_i is the hardware clock offset. Note that the real time t is unavailable to the sensor nodes, and both α_i and β_i cannot be computed [16]. However, by comparing the hardware clock readings of two nodes i and j, a relative clock is obtained as

$$\tau_i(t) = \frac{\alpha_i}{\alpha_j}\tau_j(t) + \left(\beta_i - \frac{\alpha_i}{\alpha_j}\beta_j\right) = \alpha_{ji}\tau_j(t) + \beta_{ji}.$$

A simple experiment is provided to verify the above linear model of the hardware clock. Two Micaz sensor nodes, denoted with i and j, are utilized. In the experiment, node i periodically broadcasts its current hardware clock reading $\tau_i(t)$ to node j, where the broadcast period is set to 30 s, and node j records its current hardware clock reading $\tau_j(t)$ when it receives the message from node i and stores $(\tau_i(t), \tau_j(t))$. The experiment lasts for 30 min. Each pair of $(\tau_i(t), \tau_j(t))$ is plotted in Fig. 1. It is observed that each $\tau_i(t)$ and $\tau_j(t)$ satisfies

$$\tau_j(t) = 0.99996758\tau_i(t) + 9.1217489. \tag{2}$$

Hence, it follows from the above equation that the relative hardware clock can be modeled by a linear function.

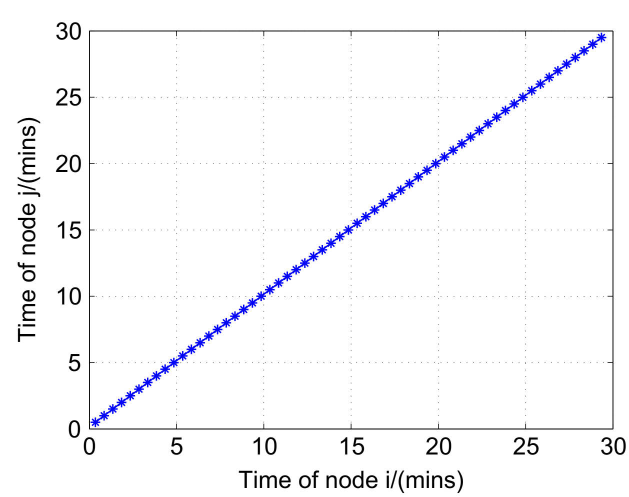

Fig. 1. Experiment to verify the linear model.

Then, a logical clock $L_i(t)$ is developed to represent the synchronization time, which is given as

$$L_i(t)=\hat{\alpha}_i\tau_i(t)+\hat{\beta}_i=\hat{\alpha}_i\alpha_i t+\hat{\alpha}_i\beta_i+\hat{\beta}_i, \tag{3}$$

where $\hat{\alpha}_i\alpha_i$ is the logical clock skew and $\hat{\alpha}_i\beta_i+\hat{\beta}_i$ is the logical clock offset. $\hat{\alpha}_i$ and $\hat{\beta}_i$ are two software parameters for time synchronization.

The goal of time synchronization design is to find a pair of parameters $(\hat{\alpha}_i(k), \hat{\beta}_i(k))$ for each node i, such that

$$\begin{cases} \lim\limits_{k\to\infty} \hat{\alpha}_i(k)\alpha_i=\alpha_v, \\ \lim\limits_{k\to\infty} \hat{\alpha}_i(k)\beta_i+\hat{\beta}_i(k)=\beta_v, \end{cases} \tag{4}$$

where both α_v and β_v are constants, and k is the iteration number. Eq. (4) guarantees that all nodes will have the same logical clock.

3.2. ATS and MTS preliminaries

3.2.1. Average time synchronization (ATS) [16]

The main idea of ATS is that each node averages its logical clock with that of neighbor node at each iteration, and then all logical clocks of nodes will converge to a common logical clock.

Let α_{ij} be the relative skew, which satisfies $\alpha_{ij}=\alpha_j/\alpha_i$. Under ATS, the relative skew α_{ij} is initially obtained from

$$\alpha_{ij}=\frac{\tau_j(t_1)-\tau_j(t_0)}{\tau_i(t_1)-\tau_i(t_0)}, \tag{5}$$

where $(\tau_i(t_1),\tau_j(t_1))$ and $(\tau_i(t_0),\tau_j(t_0))$ are two pairs of hardware clock readings. Then, at each iteration

$$\alpha_{ij}\leftarrow\rho_\eta\alpha_{ij}+(1-\rho_\eta)\frac{\tau_j(t_1)-\tau_j(t_0)}{\tau_i(t_1)-\tau_i(t_0)}, \tag{6}$$

and the skew and offset compensations are given as

$$\begin{aligned} &\hat{\alpha}_i\leftarrow\rho_v\hat{\alpha}_i+(1-\rho_v)\alpha_{ij}\hat{\alpha}_j, \\ &\hat{\beta}_i\leftarrow\hat{\beta}_i+(1-\rho_o)(\hat{\tau}_j(t_1)-\hat{\tau}_i(t_1)), \end{aligned} \tag{7}$$

respectively, where ρ_η, ρ_v, $\rho_o\in(0,1)$ are three constants. The updating of $\hat{\alpha}_i$ and $\hat{\beta}_i$ are the clock skew and offset compensation for node i.

3.2.2. Maximum time synchronization (MTS) [17]

The main idea of MTS is that each node selects the maximum logical clock among its neighbor nodes as the reference node at each iteration, thus all logical clocks of nodes will converge to the maximum logical clock among all nodes.

After obtaining α_{ij} by (5), node i can calculate $q_{ij}=\alpha_{ij}\hat{\alpha}_j/\hat{\alpha}_i=\alpha_j\hat{\alpha}_j/\alpha_i\hat{\alpha}_i$, which denotes the ratio of two nodes' logical clock skews. If node i receives a message from its neighbor node j at time t, compute q_{ij}, and if $q_{ij}>1$

$$\begin{aligned} &\hat{\alpha}_i\leftarrow\alpha_{ij}\hat{\alpha}_j, \\ &\quad\hat{\beta}_i\leftarrow\hat{\alpha}_j\tau_j(t)+\hat{\beta}_j-\alpha_{ij}\hat{\alpha}_j\tau_i(t); \end{aligned} \tag{8}$$

if $q_{ij}=1$

$$\hat{\beta}_i\leftarrow\max_{l=i,j}\{\hat{\alpha}_l\tau_l(t)+\hat{\beta}_l\}-\hat{\alpha}_i\tau_i(t). \tag{9}$$

3.3. Problems of interest

The theoretical analysis and simulations provided in [17] show the advantages of MTS over ATS. However, both theoretical and simulation results of MTS are obtained under the assumption that the communication delay and measure noises are ignored. Therefore, in the following part, we first investigate the performance of ATS and MTS under uncertainties through theoretical analysis. Then, we conduct extensive experiments on our sensor network testbed to validate the theoretical results. Based on the experimental result, we will discuss the advantages and disadvantages of ATS and MTS, and propose a revised MTS protocol, MMTS, which is able to achieve a more desirable synchronized clock.

4. Performance of ATS and MTS under uncertainty

The convergence of ATS and MTS has been proved in [16] and [17], respectively. However, the theoretical results cannot be guaranteed if various uncertainties e.g., communication noise and delay, which are unavoidable in practice, are taken into account, and the time synchronization process may even fail under certain uncertainties [25]. Therefore, it is necessary and meaningful to check how the time synchronization would be affected by various kinds of uncertainties.

Consider two neighboring nodes denoted by i and j, respectively. Assume $\tau_i(t)$ is the hardware clock information that each node i broadcasts at time t. Taking the uncertainty into consideration, we have

$$\tau_i(t)=\alpha_i t+\beta_i-\Delta_i(t), \tag{10}$$

where $\Delta_i(t)\in[-\Delta_0,\Delta_0]$ denotes the uncertainty, which can be viewed as the communication delay or measurement noise, etc, and Δ_0 is a constant and its value diverges for different nodes and environment changes. Then, each relative skew estimation (5) is changed as

$$\alpha_{ij}(t_1)=\frac{\alpha_j}{\alpha_i}+\theta_{ij}(t_1), \tag{11}$$

where $\theta_{ij}(t_1)=(\Delta_j(t_0)-\Delta_j(t_1))/(\tau_i(t_1)-\tau_i(t_0))$ is the estimation error introduced by communication and measurement uncertainties. Assume that $\theta_{ij}(t_1)\in[-\Delta_1,\Delta_1]$, where Δ_1 is a constant depending on Δ_0 and it can be adjusted by the broadcast period. Let $x_i=\alpha_i\hat{\alpha}_i$ and $y_i=\hat{\alpha}_i\beta_i+\hat{\beta}_i$ be the logical clock skew and offset, respectively.

We first consider the performance of ATS under uncertainty. The relative skew estimation based on (6) is re-written as

$$\begin{aligned} \alpha_{ij}^A(t_k)&=\frac{\alpha_j}{\alpha_i}+\rho_\eta^{k-1}\theta_{ij}(t_1)+\sum_{l=2}^{k}(1-\rho_\eta)\rho_\eta^{k-l}\theta_{ij}(t_l) \\ &=\frac{\alpha_j}{\alpha_i}+\delta_{ij}(k), \end{aligned}$$

where $\delta_{ij}(k)$ is the estimation error of relative skew at the kth iteration. Since each $\theta_{ij}(t_l)\in[-\Delta_1,\Delta_1]$, we have $\delta_{ij}(k)\in[-\Delta_1,\Delta_1]$ for $k\geq1$. Hence, under the uncertainty, the skew compensation of ATS is given as

$$\hat{\alpha}_i^A(t_k)=\rho_v\hat{\alpha}_i^A(t_{k-1})+(1-\rho_v)\left(\frac{\alpha_j}{\alpha_i}+\delta_{ij}(k)\right)\hat{\alpha}_j^A(t_k). \tag{12}$$

By multiplying a_i on both sides of (12), the error of skew compensation under ATS at time t_k denoted by $e_s^A(t_k)$ satisfies

$$\begin{aligned} e_s^A(t_k)&=|x_i(t_k)-x_j(t_k)| \\ &=|\rho_v(a_i\hat{\alpha}_i^A(t_{k-1})-\alpha_j\hat{\alpha}_j^A(t_k))+(1-\rho_v)\alpha_i\hat{\alpha}_j^A(t_k)\delta_{ij}(k)|. \end{aligned}$$

Note that $\hat{\alpha}_j^A(t_{k-1})=\hat{\alpha}_j^A(t_k)$, we have

$$\begin{aligned} e_s^A(t_k)&\leq\rho_v e_s^A(t_{k-1})+(1-\rho_v)|\alpha_i\hat{\alpha}_j^A(t_k)\Delta_1| \\ &\leq e_s^A(t_{k-1})-(1-\rho_v)[e_s^A(t_{k-1})-|\alpha_i\hat{\alpha}_j^A(t_k)\Delta_1|]. \end{aligned} \tag{13}$$

From (13), it is found that if the value of $e_s^A(t_{k-1})>|\alpha_i\hat{\alpha}_j^A(t_k)\Delta_1|$, we have $e_s^A(t_k)<e_s^A(t_{k-1})$. Hence, $e_s^A(t)$ can converge to a steady state, i.e.,

$$e_s^A(t)\leq|\alpha_i\hat{\alpha}_j^A(t)\Delta_1|, \tag{14}$$

when $t\to\infty$. Note that for ATS it is difficult to give an error bound for offset compensation, as the iterations of offset compensation

$\hat{\beta}_i^A(t_k)$ is related to $\hat{\beta}_i^A(t_{k-1})$ and $\hat{\alpha}_i^A(t_k)$, i.e., the error bound of offset compensation depends on all historical offset and skew compensation. Therefore, we will further evaluate the performance of ATS under uncertainties through extensive experiments in later section.

Then, we analyze the performance of MTS under uncertainty. Let $\hat{\alpha}_i^M(t)$ be the software parameter for the skew compensation of MTS. Since the relative skew estimation is also calculated by (11), the skew compensation is re-written as

$$\hat{\alpha}_i^M(t_1) = \frac{\alpha_j \hat{\alpha}_j^M(t_1)}{\alpha_i} + \hat{\alpha}_j^M(t_1)\theta_{ij}(t_1). \tag{15}$$

By multiplying a_i on both sides of (15) and defining $e_s^M(t_1)$ as the error of skew compensation at time t_1, we have

$$\begin{aligned} e_s^M(t_1) &= |x_i(t_1) - x_j(t_1)| \\ &= |\alpha_i \hat{\alpha}_i^M(t_1) - \alpha_j \hat{\alpha}_j^M(t_1)| \\ &\le |\alpha_i \hat{\alpha}_j^M(t_1) \Delta_1|. \end{aligned} \tag{16}$$

Let $e_o^M(t_1)$ be the error of offset compensation under MTS at time t_1. In the offset compensation under MTS, the uncertainty generates two extra parts expressed as

$$e_o^M(t_1) = |y_i(t_1) - y_j(t_1)| = |e_{od}^M(t_1) + e_{os}^M(t_1)|, \tag{17}$$

where $e_{od}^M(t_1) = \hat{\alpha}_j^M(t_1)\Delta_j(t_1)$ and $e_{os}^M(t_1) = \alpha_i \hat{\alpha}_j^M(t_1)\theta_{ij}(t_1)t_1$. Considering the common period T, the maximum error of logical time $e^M(t_1)$ is given as

$$e^M(t_1) = |L_i(t_1) - L_j(t_1)| \le \frac{T e_s^M(t_1)}{\alpha_{min}} + |e_{od}^M(t_1)|, \tag{18}$$

where $\alpha_{min} = \min\{\alpha_i, \alpha_j\}$.

According to the above analysis, both ATS and MTS cannot achieve perfect synchronization when various uncertainties exist, and the synchronization accuracy depends on the bound of uncertainty. Taking skew compensation as an example, the synchronization accuracy of ATS and MTS under uncertainties is bounded by (14) and (16), respectively. In the following sections, we will verify and compare the performance of both ATS and MTS through experiments.

5. Testbed setup

Sensor Node Type: Our established testbed is composed of Micaz sensor nodes. Typically, the Micaz node has an ATmega128L micro-controller, and provides an 8-MHz crystal oscillator and a 32-KHz crystal oscillator. We synchronize the clock which is generated by 32-KHz crystal oscillator, and each clock tick is $1/32\,K = 30.5\ \mu s$, i.e., 1 tick equals to 30.5 μs. The Micaz sensor node also provides a CC2420 chip for IEEE802.15.4/Zigbee by Chipcon. The chip works in 2.4-GHz and the packet is transmitted at the maximum rate of 250-Kbps.

Global Reference Time: The sensor nodes in our experiments are divided into normal sensor node, reference node and sink node, respectively. The normal nodes run the protocols and the reference node and sink node are used to depict the synchronization accuracy, which is the same as [12]. During the experiments, the reference node sends a query for the global time periodically to all the normal sensor nodes, and the sink node collects the responses to the query from all the normal sensor nodes by reporting the logical clocks when they receive the query. Since all of the normal nodes receive the query almost at the same time, we can compute the maximum error of all normal nodes' logical clocks by

$$e(t) = \max_{i,j \in \mathcal{V}} [L_i(t) - L_j(t)], \tag{19}$$

where $\mathcal{V}$ is the node set of all normal nodes. $e(t)$ is exploited to measure the synchronization accuracy at time t. Clearly, all nodes achieve time synchronization completely if and only if $e(t) = 0$.

Communication Process: According to the protocols, each normal sensor node will broadcast messages periodically with a common period T based on its own hardware clock, where the messages include the nodes' identity (ID), the current two software parameters of logical clock and hardware clock reading. Once the neighbors receive the messages, they update the two parameters of their logical clocks according to ATS and MTS protocols. In the experiments, MAC-layer time-stamping is used for each node to read the local hardware clock accurately. The CC2420 chips generate an interrupt at the first bit of a transmission or reception message, which is called Start Frame Delimiter (SFD), to get the local hardware clock reading immediately. As the transmission time is extremely short, the hardware clock readings we get from the transmission node and reception nodes are nearly the same. However, the transmission message includes the hardware clock reading when last transmission message instead of itself is transmitted. Therefore, the reception nodes should store the local hardware clock readings for the next iteration.

6. Experimental results

There are 22 sensor nodes in our experiments, where 20 nodes as normal sensor nodes are required to synchronize the clocks, and the other two nodes serve as reference node and sink node. Different network topologies, 5*4 grid topology, ring topology and linear topology shown in Fig. 2, are considered to compare the performance of ATS and MTS. The common broadcast period T is set to 30 s, which is the same as [16]. For ATS, the protocol parameters are set as $\rho_v = \rho_o = \rho_\eta = 0.5$.

6.1. Convergence rate: the transient performance

Convergence rate is defined as the number of cycles after which the time synchronization error is below a certain bound. Such a metric is critical to demonstrate the transient property of time synchronization protocols. We conduct the experiments to compare the convergence rate of ATS and MTS. Note that the synchronization cannot be achieved completely because of the uncertainties, we consider that it has synchronized when the maximum error $e(t)$ keeps fluctuating around a certain level, which is called at the steady states.

The experimental results for grid topology on convergence rate of both ATS and MTS are shown in Fig. 3(a) and (b). We also conduct the simulation with the results shown in Fig. 3(c) and (d) as reference, which makes the comparisons more clearly. It is observed from Fig. 3(a) that MTS takes about 5 synchronization cycles to reach the steady states while ATS does not achieve the steady states before 50 cycles observed from Fig. 3(b), which means that the convergence rate of MTS is much faster than that of ATS. The results show that ATS converges asymptotically while MTS is with finite convergence rate. Fig. 3(b) shows that between cycle 40 to 50 MTS achieves the accuracy of around 6.144 ticks on average while the synchronization error of ATS is still around 130 ticks after 40 cycles. After another 10 cycles, the error of ATS can be reduced to less than 30 ticks. For the simulation results in Fig. 3 (c) and (d), similar initial settings are employed as those in the experiment. It is clear that the performance of ATS and MTS in simulation is similar to that in experiment. However, in the simulation, as the uncertainties are neglected, MTS can converge completely.

The results for the ring and linear network topologies are provided in Fig. 4. It is observed that MTS has a much faster

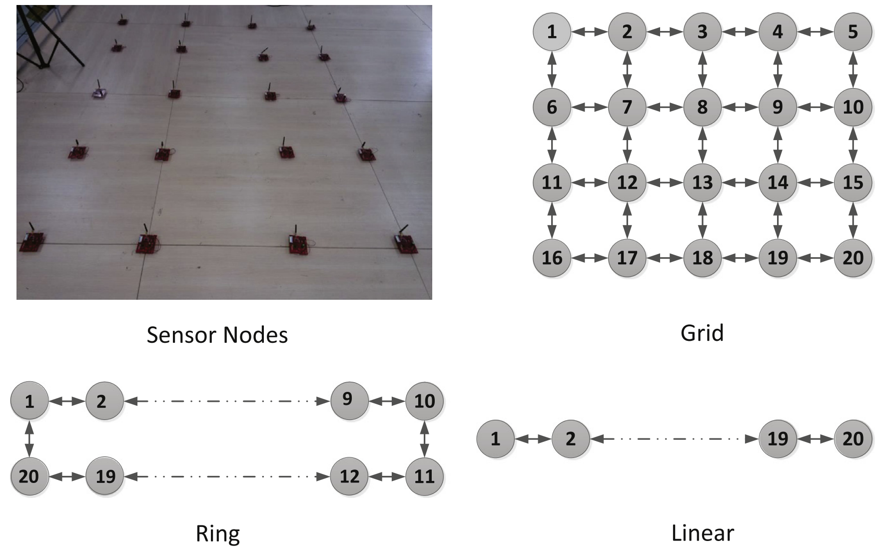

Fig. 2. Sensor nodes and three topologies of 20 nodes.

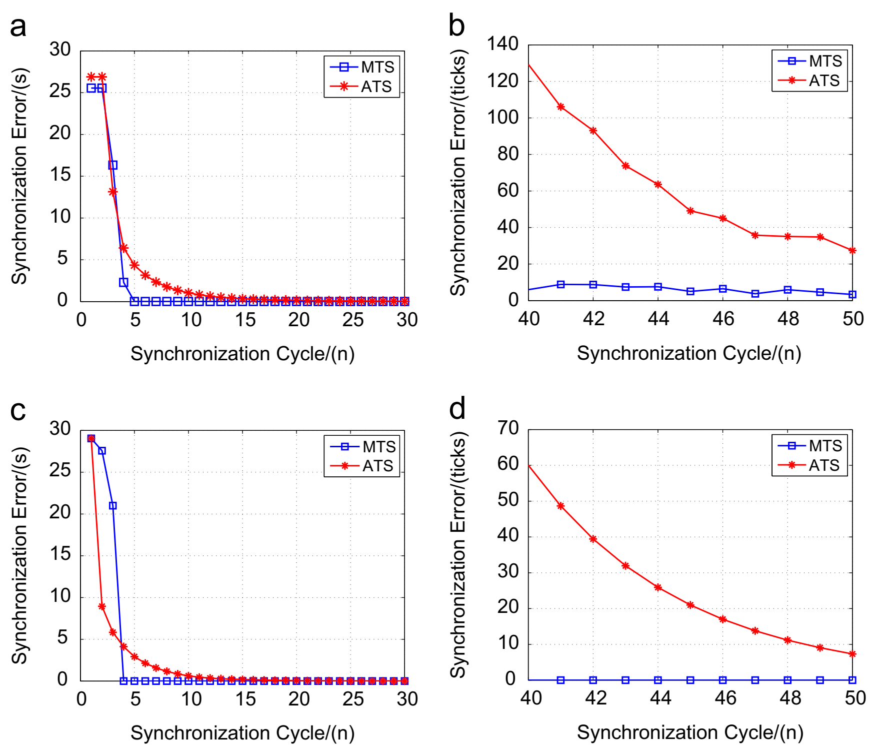

Fig. 3. Comparisons between ATS and MTS in 5⁎4 grid topology: (a) experiment: cycle 0–30; (b) experiment: cycle 40–50; (c) simulation: cycle 0–30; and (d) simulation: cycle 40–50.

convergence rate than ATS in both ring and linear network topologies. Specifically, Fig. 4(a) shows that in the ring topology, MTS just takes 7 synchronization cycles to get synchronized with 13.817 ticks mean synchronization accuracy observed from Fig. 4(b), while the synchronization error of ATS is still about 1 s after 50 cycles. Fig. 4(d) shows that in linear topology MTS takes only 9 synchronization cycles to reach the steady states and the mean synchronization accuracy is equal to around 17.816 ticks, which is shown in Fig. 4(e). Meanwhile, the synchronization error of ATS after 70 cycles is about 1.3 s. Also, we give the simulation results in Fig. 4(c) and (f). Comparing the simulation results with experimental results, it can be seen that MTS converges much faster than ATS.

From the above experimental results and corresponding simulations, it is found that MTS has a much faster convergence rate than ATS, which is the main advantage of MTS over ATS, and both ATS and MTS have the fastest convergence rate in grid topology and slowest convergence rate in linear topology.

6.2. *Synchronization accuracy: the stationary performance*

Synchronization accuracy is defined as the time synchronization error where the time synchronization error always keeps fluctuating around. Thus it is an important metric to judge the time synchronization performance at the steady state. In this subsection, we compare the synchronization accuracy of ATS and

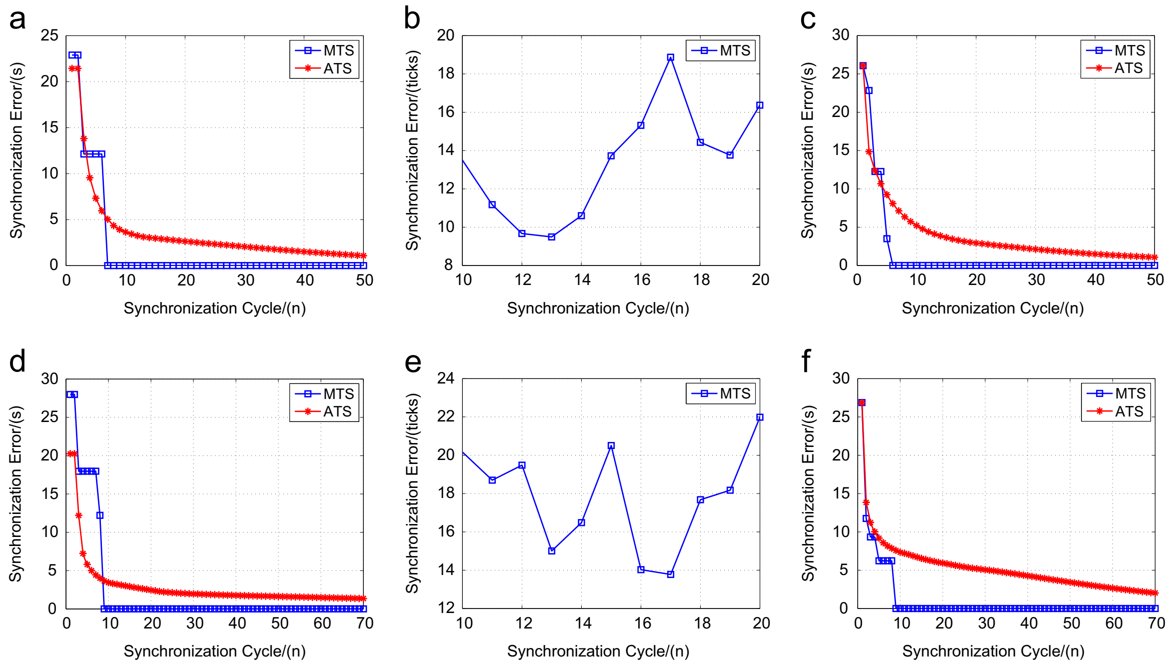

Fig. 4. Comparisons between ATS and MTS in ring and linear topologies: (a) experiment: ring; (b) error of MTS in ring; (c) simulation: ring; (d) experiment: linear; (e) error of MTS in linear; and (f) simulation: linear.

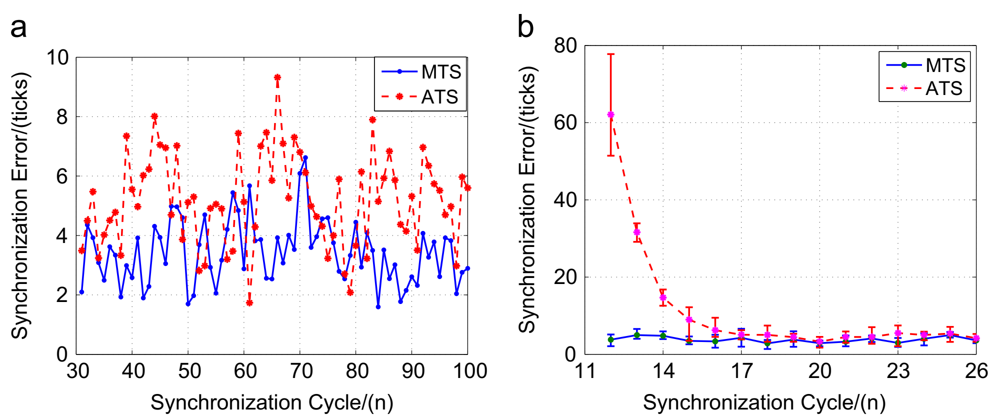

Fig. 5. Comparisons on synchronization accuracy in 3*3 grid topology: (a) synchronization accuracy and (b) synchronization accuracy in statistics.

Table 1
Comparison results of convergence and accuracy.

Protocol	Topologies	Grid	Ring	Linear
MTS	Cycles/(n)	5	5	8
	Accuracy/(ticks)	3.7	4.1	7.4
ATS	Cycles/(n)	16	42	122
	Accuracy/(ticks)	5.5	9.5	18.6

MTS, where the maximum error $e(t)$ is used to describe the accuracy, and 3*3 grid topology is considered.

Fig. 5(a), which begins from the 30th cycle, shows that both ATS and MTS have a high synchronization accuracy, which is below 10 ticks. Another result which has statistical significance is shown in Fig. 5(b). It also shows that MTS has similar synchronization accuracy with ATS. The figure begins from the 12th cycle and there are average, maximum and minimum values of $e(t)$ in each cycle for five times experiments.

Table 1 summaries the convergence rate and synchronization accuracy of ATS and MTS for different topologies composed of 9 sensor nodes. It can be observed that the convergence rate of MTS is much smaller than that of ATS. Moreover, the convergence rate of MTS is less affected by the topology compared with ATS. Meanwhile, the synchronization accuracy of MTS outperforms ATS for different topologies, and the advantage becomes more obvious when the connectivity becomes worse, e.g., linear topology.

6.3. Effect of Hop number

In this subsection, we investigate how the relative hop number will affect the relative time synchronization difference between each pair of nodes. We select node 1 as the reference node and define $e_{i1} = |L_i - L_1|, \ i = 1, 2\ldots20$, as the logical clock error between node i and node 1, i.e., the difference of logical clock values between nodes i and 1. For the default 5*4 grid topology, we obtain e_{i1} by averaging the steady state values of 20 trials. The experimental results are shown in Fig. 6, where the node in position $(1, 1)$ is node 1, and the average synchronization error of each node from node 1 as a function of grid location is displayed under both ATS and MTS. It is observed from the figure that the average synchronization error of each node from node 1 gradually

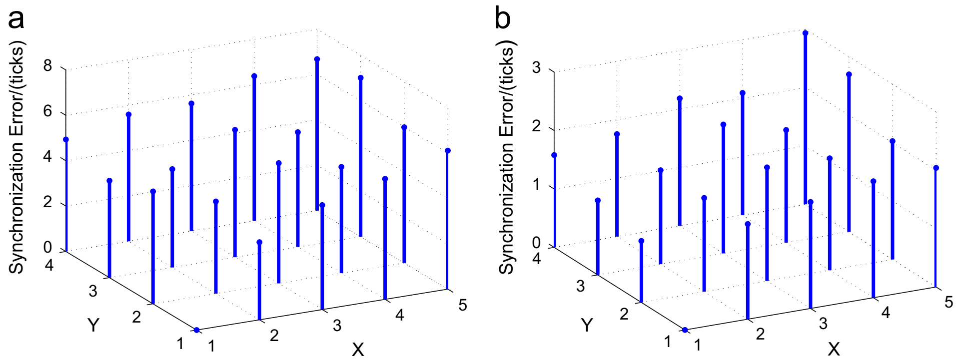

Fig. 6. Average relative synchronization error to node 1 of each node: (a) ATS and (b) MTS.

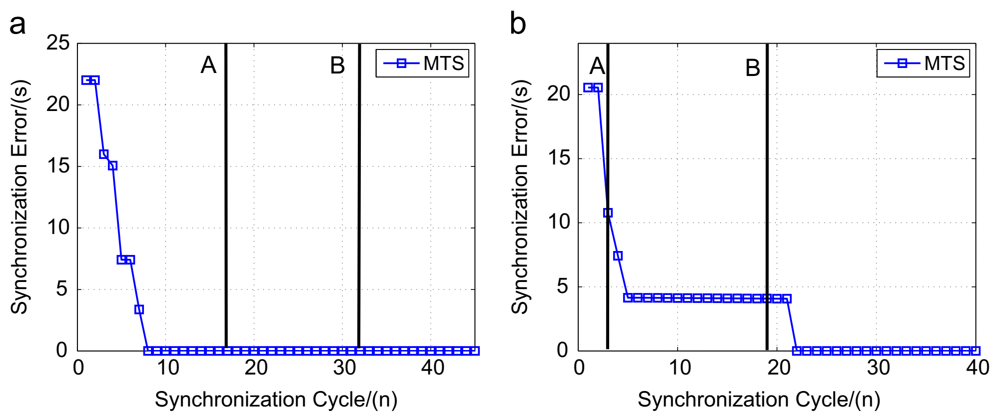

Fig. 7. Effect of turning off the radio: (a) radios are turned off at the 17th cycle and turned on again at the 32nd cycle and (b) radios are turned off at the 3rd cycle and turned on again at the 19th cycle.

increases with the number of communication hops for both ATS and MTS. Thus, it illustrates that the synchronization error may accumulate with the increase of hop number.

6.4. *Effect of topology change*

The effect of dynamic topology under ATS has been well studied in [16], while that under MTS has not been validated via experiments. Here, the effect of topology change under MTS is studied in terms of turning off the radio, node failure and restart and new node joining, which are three common events in WSNs and will render the network topology to change. Turning off the radio of one node will cut off the communication with its neighbor nodes, but the power of the node is still on and the clock is working. Some situations, e.g., node mobility causing the communication break, are also simulated by turning off the radio of the nodes. Node failure and restart, new node joining also make the network topology change since they change the node number of the network. For the following experiments in this part, the network is initially ring topology with 20 normal sensor nodes, and then it changes to linear with 19 sensor nodes, and finally it turns back to ring.

First, turning off the radio is considered. For the experimental result shown in Fig. 7(a), the network gets synchronized from the initial condition after a few cycles. Then, the radios of two nodes are turned off after 17 cycles. At last, the two nodes turned on again the radios after another 15 cycles. It can be found that turning off the radio does not influence the convergence of clocks. We also consider that the radios of two nodes are turned off at the 3rd cycle before the synchronization having achieved. The

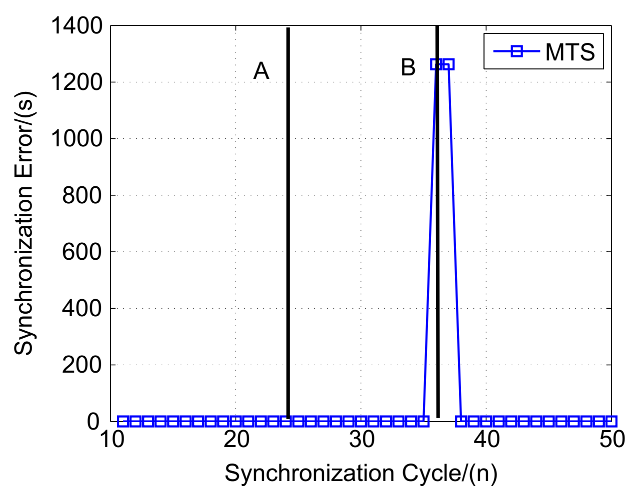

Fig. 8. Node fails at the 24th cycle and restarts at the 36th cycle.

experimental result is displayed in Fig. 7(b). Clearly, MTS cannot converge because of the disconnected network when the radios of two nodes are turned off. When the two nodes turned on again the radios at the 19th cycle, the time synchronization of the whole network completes after several cycles, which again supports that MTS is robust to turning off the radio.

Then, we consider the node failure and restart in the following experiments. As shown in Fig. 8, at the 24th cycle, one node is shut down, and it is switched on again after another 12 cycles, and then it is observed that the network gets synchronized 2 cycles later.

This result demonstrates that node failure and restart does not affect the synchronization under MTS.

Finally, new node joining is discussed. Considering that one node fails first which changes the topology from ring to linear, and then one new node joins and the topology becomes a ring again. If the new node is not special which means that it is not the fastest one, the network get synchronized 2 cycles later after new node joins. As in Fig. 9(a), one node fails at the 20th cycle and new node joins after another 12 cycles. It is found that MTS converges when new node joins. If the new node has the maximum clock, the logical clock of nodes will be synchronized to that of the new one. Thus, it takes more cycles to reach re-synchronization, which is confirmed in the experimental result Fig. 9(b). In this figure, at the 21st cycle, one node fails, and after another 8 cycles one new node joins the network. It is observed that MTS still converges when a new node with the maximum clock among all nodes joins.

From these experiments, it is found that MTS can always achieve time synchronization when the network topology changes in terms of turning off the radio, node failure and restart and new node joining, unless the necessary conditions for time synchronization are not satisfied, e.g., the network is divided into two disconnect parts. Therefore, MTS has the same robustness for topology change as ATS, as both of them are fully distributed.

6.5. *Summary of performance comparison*

Based on all the experimental results, we summarize the comparison results between ATS and MTS in Table 2 in order to show their properties better. Basically, both ATS and MTS can achieve accurate time synchronization under different topologies. And both protocols show good performance considering different uncertainties. However, MTS is able to achieve accurate time synchronization by a much faster rate which makes it more applicable for the time-critical and energy-constrained sensor network applications.

It should be noticed that there are still some aspects which may be improved. For example, as shown in Fig. 6, the synchronization error will increase with the hop number gradually for both ATS and MTS. Hence, ATS and MTS need to be improved to achieve a better synchronization error between the nodes which are non-adjacent. Meanwhile, since the uncertainties will affect the performance of both ATS and MTS, e.g., the synchronization accuracy, the uncertainty compensation, such as delay compensation,

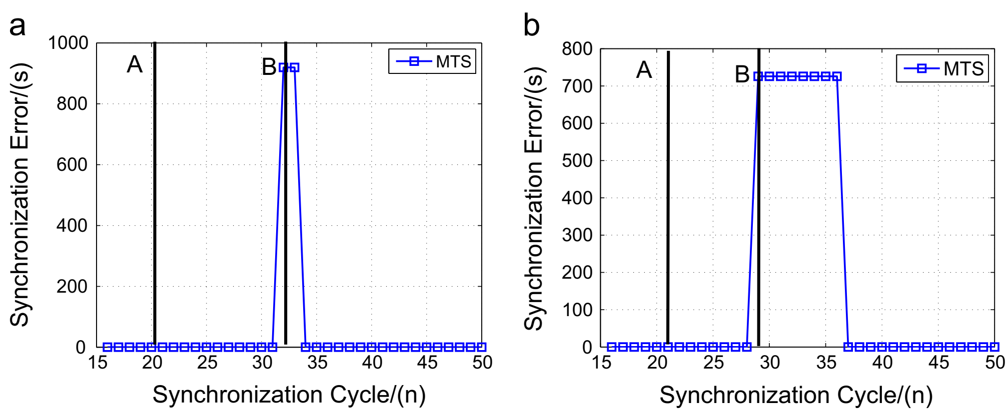

Fig. 9. Effect of new node joining: (a) node fails at the 20th cycle and new node joins at the 32nd cycle and (b) node fails at the 21st cycle and new node joins at the 29th cycle.

Table 2
Summary of the comparisons.

Protocol	ATS VS MTS
Convergence rate	MTS is much faster than ATS
Synchronization accuracy	Both ATS and MTS have a high accuracy
Effect of hop number	ATS and MTS are almost the same
Effect of topology change	ATS and MTS are both convergent

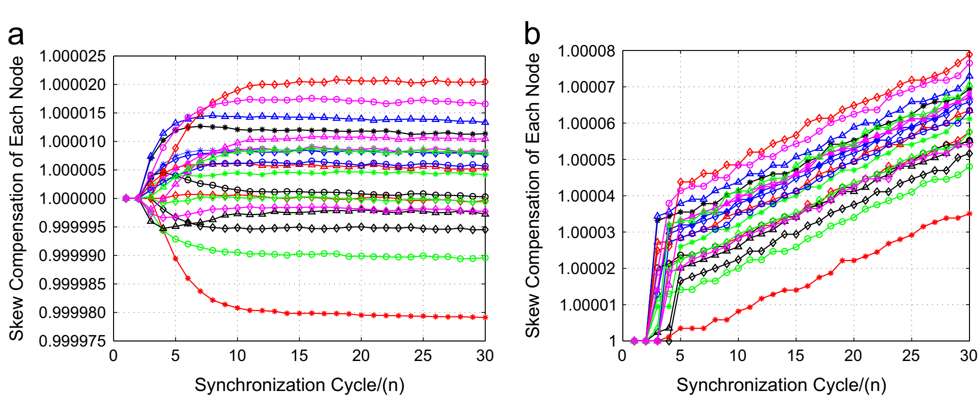

Fig. 10. The dynamics of software parameters $\hat{\alpha}_i(t)$, $i \in V$: (a) under ATS and (b) under MTS.

should be taken into consideration to decrease the time synchronization error caused by the uncertainties. In addition, due to the uncertainties, the estimation of logical clock skew will increase with the iterations under MTS, which will render the logical clocks of nodes increasing. Thus, in the following subsection, we will propose a revised protocol of MTS to achieve a better performance, and also provide experiments to verification.

6.6. Discussion

According to the above performance comparisons between ATS and MTS, one sees that MTS is a more promising consensus-based time synchronization protocol as it has much faster convergence rate than ATS. However, it is observed that under ATS the logical clock skew is stable and close to 1, while that under MTS is slightly larger than 1 and increases with iterations. For example, considering 20 sensor nodes with 5*4 grid topology, the dynamics of software parameters $\hat{\alpha}_i(t), i \in \mathcal{V}$ under ATS and MTS are shown in Fig. 10. It is observed that after convergence the software parameters $\hat{\alpha}_i(t), i \in \mathcal{V}$ under ATS are stable and close to 1 (each $\hat{\alpha}_i(t)$ satisfies $|\hat{\alpha}_i(t)-1| < 2.5 \times 10^{-5}$). While under MTS, all the software parameters are larger than 1, and increase with iterations due to the uncertainties. Since the hardware clock skew of each node i satisfies $|a_i - 1| \le 10^{-4}$ [8,30] and the maximum logical clock is usually with larger clock skew than 1, the logical clock skews obtained from MTS are slightly larger than 1. Note that for a time synchronization protocol, it prefers to obtain stable synchronized clocks as well as close to clocks with clock skew equal to 1, as clocks with clock skew equal to 1 are more desirable in most applications. Hence, we revise MTS by adopting minimum consensus process so that the synchronized clocks can be more stable and closer to the desirable clock.

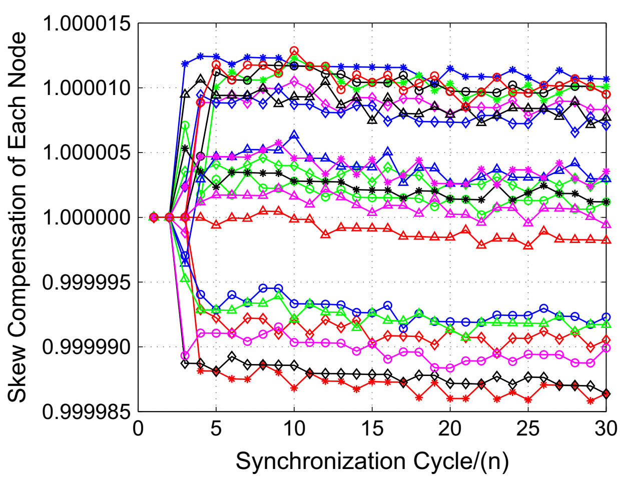

Fig. 11. The dynamics of software parameters $\hat{\alpha}_i(t),\ i \in V$ under MMTS.

We briefly state the idea of revising MTS. Note that if only the minimum consensus is utilized for time synchronization, all nodes will synchronize their logical clocks to the minimum clock among nodes and each logical clock skew will decrease with iterations when uncertainties are considered. Intuitively, if the average of maximum and minimum clocks is selected as the reference clock, we can obtain that the logical clocks of nodes have stable clock skews which are closer to 1. Therefore, by combining maximum and minimum consensus, we propose a revised synchronization protocol, Maximum Minimum Time Synchronization (MMTS). The details of MMTS protocol are given in Appendix A.

In order to validate the analysis mentioned above, the experiments are conducted in a network composed of 20 sensor nodes with 5*4 grid topology which show that MMTS can maintain the advantages of MTS and obtain stable logical clocks with their skews close to 1. First, the experimental result in Fig. 11 shows that each $\hat{\alpha}_i(t)$ under MMTS will be stable and close to 1 after several cycles. By comparing the associate result under MTS, which is shown in Fig. 10(b), it is found that unlike MTS, under MMTS each $\hat{\alpha}_i(t)$ keeps fluctuating around a level much closer to 1, and we have $|\hat{\alpha}_i(t)-1| < 1.5 \times 10^{-5}$. Since the hardware clock skew of each node i is also very close to 1 [11], the value of each logical clock skew $\hat{\alpha}_i\alpha_i$ under MMTS will be stable and close to 1. Then, the experimental result shown in Fig. 12(a) compares the convergence rate of ATS, MTS and MMTS. Meanwhile, it is observed that MMTS has a similar convergence rate compared with MTS, which is also much faster than ATS. Additionally, we compare the synchronization accuracy of these three consensus based protocols between cycle 40 and 50, which is shown in Fig. 12(b). The mean synchronization accuracy under MTS is about 6.144 ticks and under MMTS it is about 3.190 ticks, while under ATS the synchronization error is still larger than 20 ticks and the steady states are not arrived. Thus, according to the above experiments, the revised protocol MMTS maintains the advantages of MTS and obtains better logical clocks than MTS.

7. Conclusion

Considering various uncertainties in the real implementations, we investigate two typical consensus-based time synchronization protocols, ATS and MTS, via experimental studies in this paper. We first analyze how the uncertainty will affect the synchronization accuracy under both ATS and MTS. Then we implement both protocols on one

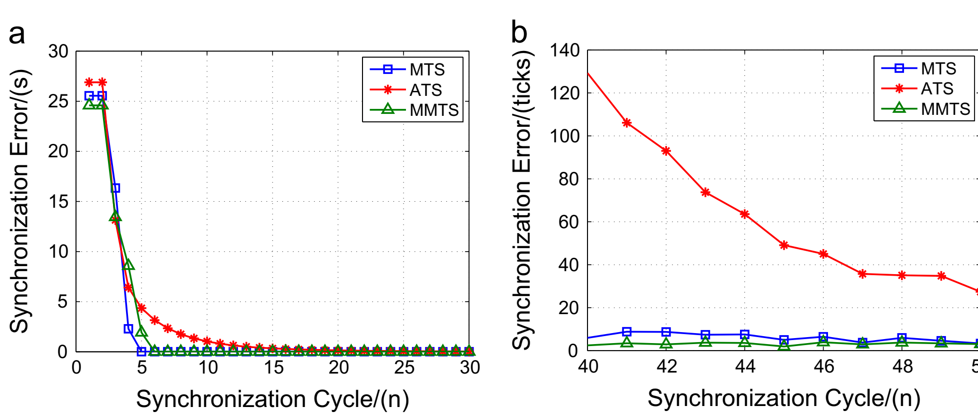

Fig. 12. The performance of ATS, MTS and MMTS: (a) cycle 0–30 and (b) cycle 40–50.

testbed with more than 20 Micaz nodes. We conduct extensive experiments to evaluate the time synchronization performance and robustness for both ATS and MTS. The results validate the theoretical analysis of both ATS and MTS. In order to obtain more desirable synchronized logical clocks, we propose a revised protocol MMTS by adopting both maximum consensus and minimum consensus processes. The experiments demonstrate the effectiveness of MMTS.

Acknowledgement

This paper was partially supported by the 863 High-Tech Project under Grants 2011AA040101-1, 2012AA041709-04, and the Fundamental Research Funds for the Central Universities under Grants 2013QNA5013 and 2013FZA5007.

Appendix A. MMTS protocol

By a similar method in [31], the relative skew can be estimated by

$$\alpha_{ij}(k)=\frac{\frac{\tau_j(t_k)-\tau_j(t_{k-1})}{\tau_i(t_k)-\tau_i(t_{k-1})}+(k-1)\alpha_{ij}(k-1)}{k}, \tag{A.1}$$

where $\alpha_{ij}(0)=1$. From (A.1), $a_{ij}(k)$ is the average of k times estimation of each relative skew a_{ij} for $i,j\in\mathcal{V}$. Thus, increasing the communication time between the nodes will decrease the estimate error of each relative skew [31]. Then, by combining maximum and minimum consensus, we propose the following MMTS protocol.

Algorithm 1. Maximum minimum time synchronization (MMTS).

1: Given the initial conditions $\hat{\alpha}_i=1,\hat{\beta}_i=0,\mu_i=0$ and $\nu_i=0$ for $i\in V$, and set a common broadcast period T to each node.
2: If $\tau_i(t)/T\in N^+$, node i broadcasts packet including parameters $\tau_i(t),\hat{\alpha}_i,\hat{\beta}_i,\mu_i$ and ν_i to its neighbors.
3: If node i receives the packet from node j at time $t_k,k\in\{0,N^+\}$, then records its current hardware clock reading $\tau_i(t_k)$, and calculates $\alpha_{ij}(k)$ by (A.1) when $k\geq 1$.
4: Compute $\hat{\alpha}_{imax}$, $\hat{\beta}_{imax}$, $\hat{\alpha}_{imin}$ and $\hat{\beta}_{imin}$ respectively by $\hat{\alpha}_{imax}=\hat{\alpha}_i+\mu_i$, $\hat{\beta}_{imax}=\hat{\beta}_i+\nu_i$; $\hat{\alpha}_{imin}=\hat{\alpha}_i-\mu_i$, $\hat{\beta}_{imin}=\hat{\beta}_i-\nu_i$. Then, compute $p_{ij}(k)=\frac{\alpha_{ij}(k)(\hat{\alpha}_j+\mu_j)}{\hat{\alpha}_{imax}}$ and $q_{ij}(k)=\frac{\alpha_{ij}(k)(\hat{\alpha}_j-\mu_j)}{\hat{\alpha}_{imin}}$.
5: **Maximum consensus:** If $p_{ij}(k)>1$, then $\hat{\alpha}_{imax}=\alpha_{ij}(k)(\hat{\alpha}_j+\mu_j)$ and $\hat{\beta}_{imax}=(\hat{\alpha}_j+\mu_j)\tau_j(t_k)+\hat{\beta}_j+\nu_j-\alpha_{ij}(k)(\hat{\alpha}_j+\mu_j)\tau_i(t_k)$; if $p_{ij}(k)=1$, then $\hat{\beta}_{imax}=\max_{l=i,j}\{(\hat{\alpha}_l+\mu_l)\tau_l(t_k)+\hat{\beta}_l+\nu_l\}-(\hat{\alpha}_i+\mu_i)\tau_i(t_k)$.
6: **Minimum consensus:** If $q_{ij}(k)<1$, then $\hat{\alpha}_{imin}=\alpha_{ij}(k)(\hat{\alpha}_j-\mu_j)$ and $\hat{\beta}_{imin}=(\hat{\alpha}_j-\mu_j)\tau_j(t_k)+\hat{\beta}_j-\nu_j-\alpha_{ij}(k)(\hat{\alpha}_j-\mu_j)\tau_i(t_k)$; if $q_{ij}(k)=1$, then $\hat{\beta}_{imin}=\min_{l=i,j}\{(\hat{\alpha}_l-\mu_l)\tau_l(t_k)+\hat{\beta}_l-\nu_l\}-(\hat{\alpha}_i-\mu_i)\tau_i(t_k)$.
7: Update parameters $\hat{\alpha}_i,\hat{\beta}_i,\mu_i$ and ν_i respectively by $\hat{\alpha}_i=\frac{\hat{\alpha}_{imax}+\hat{\alpha}_{imin}}{2},\hat{\beta}_i=\frac{\hat{\beta}_{imax}+\hat{\beta}_{imin}}{2}$; $\mu_i=\dfrac{\hat{\alpha}_{imax}-\hat{\alpha}_{imin}}{2},\nu_i=\dfrac{\hat{\beta}_{imax}-\hat{\beta}_{imin}}{2}$.
8: Replace $[\tau_i(t_{k-1}),\tau_j(t_{k-1}),\alpha_{ij}(k-1)]$ by $[\tau_i(t_k),\tau_j(t_k),\alpha_{ij}(k)]$.

In MMTS, $\hat{\alpha}_{imax}$ and $\hat{\beta}_{imax}$ are two software parameters respectively for clock skew and offset compensation of node i based on maximum consensus, while $\hat{\alpha}_{imin}$ and $\hat{\beta}_{imin}$ are those based on minimum consensus. Let $\alpha_{max}=\max_{i\in\mathcal{V}}\alpha_i$, $\alpha_{min}=\min_{i\in\mathcal{V}}\alpha_i$, $\beta_{max}=\max_{i\in\mathcal{V}_{max}}\beta_i$, $\beta_{min}=\min_{i\in\mathcal{V}_{min}}\beta_i$, where $\mathcal{V}_{max}=\{i|\alpha_i=\alpha_{max},i\in\mathcal{V}\}$, $\mathcal{V}_{min}=\{i|\alpha_i=\alpha_{min},i\in\mathcal{V}\}$. As proved in [17], based on maximum consensus, the synchronization can be achieved in a finite time, and thus after converging

$$\hat{\alpha}_{imax}\tau_i(t)+\hat{\beta}_{imax}=\alpha_{max}t+\beta_{max},$$

holds true for $\forall i\in\mathcal{V}$. Similarly, based on minimum consensus, after algorithm converges

$$\hat{\alpha}_{imin}\tau_i(t)+\hat{\beta}_{imin}=\alpha_{min}t+\beta_{min},$$

holds true for $\forall i\in\mathcal{V}$. Therefore, MMTS will also converge in a finite time when the uncertainties are ignored since it just combines the maximum and minimum consensus. Moreover, after MMTS converges

$$\begin{aligned}\hat{\alpha}_i\tau_i(t)+\hat{\beta}_i&=\frac{\alpha_{imax}+\alpha_{imin}}{2}\tau_i(t)+\frac{\beta_{imax}+\beta_{imin}}{2}\\&=\frac{\alpha_{max}+\alpha_{min}}{2}t+\frac{\beta_{max}+\beta_{min}}{2},\end{aligned}$$

holds true for $\forall i\in\mathcal{V}$.

References

[1] Chaudhari Q, Serpedin E, Wu Y. Improved estimation of clock offset in sensor networks. In: Proceedings of ICC, 2009. p. 33–6.
[2] Clark J, Rafael Fierro F. Mobile robotic sensors for perimeter detection and tracking. ISA Trans 2007;46(1):3–13.
[3] He S, Chen J, Yau D, Shao H, Sun Y. Energy-efficient capture of stochastic events by global- and local-periodic network coverage. In: Proceedings of MobiHoc, 2009. p. 155–64.
[4] Raghavendra CS, Sivalingam KM, Znati TF. Wireless sensor networks. 2nd editionSpringer; 2004.
[5] Howitt I, Manges W, Kuruganti P, Allgood G, Gutierrez J, Conrad J. Wireless industrial sensor networks: framework for QoS assessment and QoS management. ISA Trans 2006;45(3):347–59.
[6] Cao X, Chen J, Zhang Y, Sun Y. Development of an integrated wireless sensor network micro-environmental monitoring system. ISA Trans 2008;47(3):247–55.
[7] Bian T, Venkatesan R, Li C. Adaptive time synchronization for wireless sensor networks with self-calibration. In: Proceedings of ICC, 2009. p. 5031–5.
[8] Sundararaman B, Buy U, Kshemkalyani A. Clock synchronization for wireless sensor networks: a survey. Ad Hoc Netw 2005;3(3):281–323.
[9] Freris NM, Kowshik H, Kumar PR. Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation. Proc IEEE 2010;98(11):1828–46.
[10] Elson J, Girod L, Estrin D. Fine-grained network time synchronization using reference broadcasts. In: Proceedings of OSDI, 2002.
[11] Ganeriwal S, Kumar R, Srivastava MB. Timing-sync protocol for sensor networks. In: Proceedings of SenSys, 2003.
[12] Maroti M, Kusy B, Simon G, Ledeczi A. The flooding time synchronization protocol. In: Proceedings of SenSys, 2004.
[13] Olfati-Saber R, Fax JA, Murray RM. Consensus and cooperation in networked multi-agent systems. Proc IEEE 2007;95(1):215–33.
[14] Hajar A, Haeri M. Average consensus in networks of dynamic multi-agents with switching topology: infinite matrix products. ISA Trans 2012;51(4):522–30.
[15] Carli R, Zampieri S. Networked clock synchronization based on second order linear consensus algorithms. In: Proceedings of IEEE CDC, 2010. p. 7259–64.
[16] Schenato L, Fiorentin F. Average TimeSynch: a consensus-based protocol for clock synchronization in wireless sensor networks. Automatica 1886;47(9):1878–86.
[17] He J, Cheng P, Shi L, Chen J. Time synchronization in WSNs: a maximum value based consensus approach. In: Proceedings of CDC, 2011. p. 7882–7.
[18] Maggs M, O'Keefe SG, Thiel D. Consensus clock synchronization for wireless sensor networks. IEEE Sens J 2012;12(6):2269–77.
[19] Solis R, Borkar V, Kumar PR. A new distributed time synchronization protocol for multihop wireless networks. In: Proceedings of CDC, 2006. p. 2734–9.
[20] Philipp S, Roger W. Gradient clock synchronization in wireless sensor networks. In: Proceedings of IPSN, 2009.
[21] Zhong Z, Chen P, He T. On-demand time synchronization with predictable accuracy. In: Proceedings of INFOCOM, 2011. pp. 2480–8.
[22] Ferrari F, Zimmerling M, Thiele L, Saukh O. Efficient network flooding and time synchronization with Glossy. In: Proceedings of IPSN, 2011. p. 73–84.

[23] Chen J, Yu Q, Zhang Y, Chen H, Sun Y. Feedback-based clock synchronization in wireless sensor networks: a control theoretic approach. IEEE Trans Veh Technol 2010;59(6):2963–73.
[24] Huang Y, Wu S. Time synchronization protocol for small-scale wireless sensor networks. In: Proceedings of WCNC, 2010. p. 1–5.
[25] Freris NM, Graham SR, Kumar PR. Fundamental limits on synchronizing clocks over networks. IEEE Trans Autom Control 2011;56(2):1352–64.
[26] Leng M, Wu Y-C. On clock synchronization algorithms for wireless sensor networks under unknown delay. IEEE Trans Veh Technol 2010;59(1):182–90.
[27] Leng M, Wu Y-C. Low-complexity maximum-likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays. IEEE Trans Signal Process 2011;59(10):4860–70.
[28] Chaudhari QM, Serpedin E, Qaraqe K. Minimal cost clock synchronization using a sender-receiver protocol in wireless sensor nets. In: Proceedings of SPECTS, 2008.
[29] Chaudhari QM, Serpedin E, Qaraqe K. Some improved and generalized estimation schemes for clock synchronization of listening nodes in wireless sensor networks. IEEE Trans Commun 2010;58(1):63–7.
[30] Choi B, Liang H, Shen X, Zhuang W. DCS: distributed asynchronous clock synchronization in delay tolerant networks. IEEE Trans Parallel Distrib Syst 2012;23(3):491–504.
[31] He J, Cheng P, Shi L, Chen J. Clock synchronization for random mobile sensor networks. In: Proceedings of IEEE CDC, 2012. p. 2712–7.