Extensible time synchronisation protocol for wireless sensor networks

Fuqiang Wang*
Shandong Computer Science Center (National Supercomputer Center in Jinan),
Shandong Academy of Sciences and
Shandong Provincial Key Laboratory of Computer Networks,
Jinan, 250014, China
Email: wangfq@sdas.org
*Corresponding author

Xiaoming Wu
College of Information Science and Engineering,
Shandong University of Science and Technology,
Qingdao, 266061, China
and
Shandong Computer Science Center (National Supercomputer Center in Jinan),
Shandong Academy of Sciences and
Shandong Provincial Key Laboratory of Computer Networks,
Jinan, 250014, China
Email: wuxm@sadas.org

Yinglong Wang
College of Information Science and Engineering,
Shandong University of Science and Technology,
Qingdao, 266061, China
Email: wangyl@sadas.org

Peng Zeng
Laboratory of Industrial Control Network and System,
Shenyang Institute of Automation,
Chinese Academy of Sciences,
Shenyang, 110016, China
Email: zp@sia.cn

Yang Xiao
Department of Computer Science,
The University of Alabama,
Tuscaloosa, AL 35487-0290, USA
Email: yangxiao@ieee.org

Abstract: One important aspect in many applications of wireless sensor networks (WSNs) is time synchronisation. There are still some problems to be solved for time synchronisation, such as error and imbalance of time synchronisation precision, causing the fluctuations among nodes’ time and being even worse for those with a larger number of hops, producing unstable communication links for the network. In this paper, we introduce an extensible time synchronisation protocol (ETSP) for WSNs and nodes get synchronised through time stamps from the root node directly or indirectly. We improve the least square method for linear regression to get better performance of time drift and spread the time information from the root node to the whole network. The proposed method reduces fluctuations in terms of time among nodes especially for the nodes far from the root, enhancing stability in large-scale networks. Simulation and experimental results show good performance of our method.
Keywords: WSNs; wireless sensor networks; time synchronisation; cumulative time error; unguen distribution.


Biographical notes: Fuqiang Wang holds a PhD in Mechatronic Engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, in 2011. He is an Associate Professor of Shandong Computer Science Center (National Supercomputer Center in Jinan). His research has involved wireless communication and mobile wireless sensor networks.

Xiaoming Wu is currently working towards his PhD at Shandong University of Science and Technology. He is also an Associate Professor of Shandong Computer Science Center (National Supercomputer Center in Jinan). His research has involved mobile wireless sensor networks.

Yinglong Wang is currently a Professor of College of Information Science and Engineering, Shandong University of Science and Technology. His research has involved wireless sensor networks.

Peng Zeng holds a PhD in Mechatronic Engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, in 2005. He is a Professor of Shenyang Institute of Automation, Chinese Academy of Sciences. His research has involved industrial communication and wireless sensor networks.

Yang Xiao is a Professor for The University of Alabama (Department of Computer Science, The University of Alabama, Tuscaloosa, USA).

1 Introduction

Recent achievements in the fields of microelectronics and wireless communication have led to the widespread deployments of the wireless sensor networks (WSNs) (Wu et al., 2005). A WSN can be described as a network consisting of multiple nodes that cooperatively perceive the environment and gather information via wireless links (Akyildiz et al., 2002; Wang and Xiao, 2006). WSNs have been widely used in many applications such as habitat monitoring, industrial process detection, wild environments monitoring, etc. (Medina et al., 2013; Liu et al., 2014a, 2014b; Ko et al., 2010; Xiao et al., 2010; Shen et al., 2015; Xie and Wang, 2014). There are many features for WSNs such as sensing locally, a large number of nodes, multi-hop wireless communication and dynamic topology (Keskin et al., 2016; Xiao et al., 2009). Time synchronisation plays an important role in WSNs (Galloway et al., 2010; Wang et al., 2013; Du et al., 2008) and is a basic technology for time division multiple access (TDMA) networks such as data fusion, node localisation, etc. (Villas et al., 2014; Zhao et al., 2014; Ergen and Varaiya, 2010; Chu et al., 2006; Hu et al., 2008).

Each sensor node has a control unit called hardware clock for the sensor's internal usage; it also has a logical clock controlled by time synchronisation algorithms based on the physical clock (Liu et al., 2007). The counting frequency of hardware clocks in a sensor node may change over time because of conditions such as temperature, voltage, etc. (Huang et al., 2014). The clock speed for sensor nodes works differently. In fact, the quartz oscillators exhibit various instabilities that can bring an error of 10–100 ppm, corresponding to 1 MHz clock with an error of 10–100 µs in 1 s. Moreover, the initial hardware clock value for each node is different. After a clock is manufactured, the initial time value is different, and a sensor node follows a cycle of sleep-wake up based on external events to save energy. Therefore, time synchronisation is essential and the inconsistencies of hardware clock error need to be considered.

Flooding the reference time to the whole WSN network is a typical way to achieve a network-wide time synchronisation. There are also some methods using pairwise communication to get synchronised. Many common used methods include reference broadcast synchronisation (RBS) (Elson et al., 2002), Timing-sync protocol for sensor networks (TPSN) (Ganeriwal et al., 2003), flooding time synchronisation protocol (FTSP) (Maroti et al., 2004), etc. Although these protocols can get time error of a few microseconds, there always exist exponentially growing errors with the network scale. Therefore, network-wide accumulative errors can be big and are sensitive to the dynamics of the network topology. The effects will be worse for large-scale networks and make the network unstable.

In the past few years, some distributed time synchronisation methods without requiring any certain reference node have been proposed. Such methods utilise consensus to achieve the robust and the scalability for time
synchronisation (Ahmed et al., 2014). They are distributed and able to compensate the difference of clocks such as clock instantaneous offset and clock skew by clock speed. However, protocols based on consensus are complex and slow to converge for a large network.

In this paper, we introduce an extensible time synchronisation protocol (ETSP) for WSNs and all nodes in a multi-hop networks get synchronised with the time stamp from the root node directly or indirectly. We improve the least square method for linear regression to get better performance of time drift and the time information is spread from the root node to the whole network. This indicates that all data for synchronisation are sourced from the root node, and therefore, it is easy to control the rhythm of time synchronisation. The approach provides stable links and reduces time fluctuation among nodes in for a network with a large scale. Since the synchronisation information depends on the hardware clock of the root node, there is no effect to the synchronisation of the other nodes in the network when failures occur for some nodes and bring less uneven error distribution.

The rest of the paper is organised as follows. Section 2 provides related work. We present our method and analysis in Section 3. Sections 4 and 5 provide simulation results and experimental results, respectively. Finally, we conclude our paper in Section 6.

2 Related work

There are many studies of time synchronisation in WSNs (Galloway et al., 2010; Wang et al., 2013; Du et al., 2008; Liu et al., 2007; Elson et al., 2002; Ganeriwal et al., 2003; Maroti et al., 2004; Ahmed et al., 2014; Sommer and Wattenhofer, 2009). Among these protocols, RBS was first proposed in the paper (Maroti et al., 2004). In RBS, a sending node sends messages with reference signals to its neighbours with physical-layer broadcasts, and then nodes receiving such signals record their local time at the receiving time point. Nodes with such receiving time will exchange with each other. The approach can achieve precision in terms of microseconds to an external timescale, for instance, universal time controller (UTC). Elson and Romer (2003) improve performance of RBS by adopting tunable and post-facto modes of synchronisation without a global timescale for the whole network and with domain knowledge.

RBS adopts a time critical path of the transmitter side to eliminate non-determinism time, but it needs quite a few of exchanging messages for timestamps from nodes.

Ren et al. (2008) propose self-correcting time synchronisation which is an optimal process to dynamically adjust the online process to simultaneously converse offset and drift compensation.

The starting point of TPSN (Ganeriwal et al., 2003) is an improvement design from existing time synchronisation protocol NTP and special developed for WSNs. The protocol consists of two phases: a hierarchical structure is established before synchronisation; a global timescale throughout the network is established by performing a pairwise synchronisation algorithm.

Maroti et al. (2004) present FTSP by employing hardware solutions and an efficient time-stamping mechanism. The protocol can dynamically readjust the reference node and be robust to link failure. By exploiting periodic flooding of time information and adapting dynamic topology to promote the robustness, the method can achieve high precision performance by using medium access control (MAC)-layer time stamping (Xiao, 2005, 2006). The authors in Gheorghe et al. (2010) extend the FTSP by using additional fault-tolerance features. The algorithm checks faults caused by transmission errors or by malicious nodes.

Ferrari et al. (2011) propose glossy, which is a flooding architecture for WSNs that utilises interference by concurrently transfer an interfered and same packet constructively so that the receiver can decode the packet. Accurate synchronisation and reliable flooding are needed for Glossy to achieve the probability that nodes getting the flooding packet is higher than 99.99%, and the duration time required for radio turning on is only a couple of milliseconds for a flood period.

Huang et al. (2011) focus on the security vulnerability for FTSP and they gave several methods for defending against attacks from malicious nodes. The effect of multiple reference nodes is reduced by a selecting method for the reference node, and the using of four filters to defend against global time attack, node replication attack and seqNum attack.

Slow-flooding is proposed in Yildirim and Kantarcı (2014) with a potential side effect on synchronisation accuracy and scalability, while rapid-flooding needs frequent communications with larger energy consumption but it is faster than slow flooding in term of synchronisation.

Schenato and Fiorentin (2011) propose the ATSP protocol by impelling all nodes to agree on a common clock speed. The process for time synchronisation for GTSP is similar to ATSP by minimising the clock skew between neighbouring nodes (Sommer and Wattenhofer, 2009), i.e., local skew. Every node in the network achieves synchronisation with all their neighbours for both ATS and GTSP. A node’s common clock value and clock speed are acquired by utilising the clock values receiving from neighbouring nodes. All nodes in the network synchronise with their neighbours without needing for a reference node for ATS and GTSP. However, most applications focus on the synchronisation with a reference node (Schmid et al., 2010), which may be important if synchronising a stable time source, but the above two studies are not. Therefore, in this paper, our work still need a reference node in the network and is more concentrated on the stable of multi-hop wireless network synchronisation.

3 Time synchronisation method

In this section, the clock model and the detail of our time synchronisation method in WSNs are presented.
3.1 Clock model

Clocks in nodes, in general, based on crystal oscillators which provide local time for the nodes. The clock time (software clock) in a node is just a counter that changes according to crystal oscillator's signal. The software clock is increased by the interrupt handler when an interrupt occurs. Most oscillators in nodes are not so precise because the frequency corresponding to physical time cannot be exactly right. Even a frequency deviation of only 0.001% would lead to a clock error of about 1 s per day (Wang et al., 2011). Considering the software clock in a distributed system that synchronises to UTC, the node's local time is \( C(t) \) that is not always the same as physical time \( t \).

For a perfect clock, the derivative \( dC(t)/dt \) (clock rate) should be equal to 1. The clock rate may vary over time due to environmental conditions, such as humidity and temperature, but the clock rate has the following limit:

\[
1 - \rho \leq \frac{dC(t)}{dt} \leq 1 + \rho. \tag{1}
\]

Define the clock rate \( dC(t)/dt \) as \( f(t) \). Then the clock value \( T(t) \) can be written as follows (Wang et al., 2011).

\[
T_i(t) = \int_{t_0}^{t} f_j(\tau) d\tau + \psi_i(t_0), \tag{2}
\]

where \( \psi_i(t_0) \) is the hardware clock offset of the node \( i \) at time \( t_0 \). Dealing with equation (2) a Taylor series expansion of \( T(t) \) can be achieved as in equation (3).

\[
T_i(t) = \beta_i + \alpha_i t + \gamma_i t^2 \ldots \tag{3}
\]

Here the subscript \( \beta \) is the offset, and \( \alpha \) is the skew and \( \gamma \) can be used to quadratic term model and detect time variation.

Some protocols use constant model \((\alpha = 0 \text{ and } \gamma = 0)\) that concern nothing more than time offset between two clocks. Moreover, linear model \((\gamma = 0)\) is popularly adopted for several protocols such as FTSP and estimation of \( \alpha \) and \( \beta \) is accomplished via linear regression (least squares).

If the error is Gaussian, the estimation can be a good choice for any error commonly. If the skew and offset are varying with time, a quadratic model can be better (Huang et al., 2014).

If not concerning the drift between two clocks, constant model needs to exchange messages from time to time, ceaselessly adjusting their clock values to maintain time synchronisation. A quadratic model will lead to higher computational complexity and it is not suitable for resource constrained WSNs. Linear model is widely used in WSNs, and many protocols can estimate the clock skew to get better time synchronisation performance in WSNs.

In the paper, the popular used linear model \((\gamma = 0 \text{ in equation (3)})\) is applied and the time model is described as follows:

\[
T_i(t) = \alpha_i t + \beta_i. \tag{4}
\]

3.2 Node-to-node time synchronisation

The goal of ETSP is to achieve a network-wide synchronisation of the local clocks for the relevant nodes. We suppose that each node has a local hardware clock exhibiting the typical timing errors of crystals and has an unreliable but error corrected wireless link to communicate with its neighbours.

The ETSP synchronises the time of a sender to potential receivers using a single radio message with MAC layer time stamping that can eliminate many of the errors, as observed in Woo and Culler (2001), at both the sender and the receiver sides. We assume that there is a time reference node in a network, the reference node broadcasts packets with time information periodically, and the nodes receive such data to get synchronised. However, accurate time synchronisation at discrete points is insufficient and compensation for the clock drift of the nodes is inevitable to achieve high precision and to keep the communication overhead low.

One typical method to compensate clock drift is using linear regression such as FTSP. We assume that the receiver has got several pairs of time stamp: \((x_i, y_i), (i = 1, 2, \ldots, m)\), and we employ the linear model as in formula (4). Therefore, there exists a line \( y = ax + b \) to express the relationship for the time between the sender and the receiver. The least square linear regression is to get a pair of \((a, b)\) to minimise the value as follows in formula (5).

\[
\phi(a, b) = \sum_{i=1}^{m} [y_i - (ax_i + b)]^2. \tag{5}
\]

The least square method mentioned above fits the data in \( y \)-axis that means using \( d_i' = |y_i - (ax_i + b)| \) to denote the length from the point \((x_i, y_i)\) to the regression line. However, \( d_i' \) is not the least length from the point to line no matter in terms of graphical or mathematical operation because the method ignores the impact from the slope of the regression line.

The angle between the coordinate axis and the fitting line is \( a \) as depicted in Figure 1, and then we have \( \tan a = a \) so that the relation between \( y \)-axis length \( d_i \) and the minimum length \( d_i' \) is as follows:

\[
d_i = d_i' \cos \alpha = d_i' \frac{1}{\sqrt{1 + a^2}}. \tag{6}
\]

Assume that \( \Delta = \left| d_i - d_i' \right| \), which is represented as the difference between the theoretical minimum distance and the simplified distance in \( x \)-axis. \( \Delta \) increases with \( a \) as depicted in equation (7). Therefore, if \( a \rightarrow 0 \), then \( \Delta \rightarrow 0 \), and if \( a \rightarrow +\infty \), then \( \Delta \rightarrow d_i' \). For the time drift \( a \) in WSNs is always positive, the deviation \( \Delta \) increases in the same trend with the slope of the line. When the line is perpendicular to the \( x \)-axis, the error will be maximised and when the slope of the fitted line decreases, the error will be smaller.
\[ \Delta = d' \cdot \left| 1 - \frac{1}{\sqrt{1 + a'^2}} \right| . \] (7)

From the above analysis, we can infer that when fitting the line in x-axis using the least square method, a similar phenomenon will occur as in Figure 2, but the conclusion is just the opposite. In other words, the error is decreased with the increase of the value of slope.

**Figure 1** Least square in one direction (x-axis) linear regression

\[ y = ax + b \]

**Figure 2** Least square in one direction (y-axis) linear regression

\[ y = ax + b \]

Therefore, we try to use the minimum distance square method to avoid these problems.

The minimum distance square employs \( d_i \) instead of \( d' \) to get the minimum distance in the true sense and reduce the fitting errors significantly to improve the accuracy of the fitted line. Compared with the method of least squares, the minimum distance square method is no longer a single direction (x direction or y direction) for data fitting, but considering the x and y directions deviation.

The minimum distance of a point \((x_i, y_i)\) \((i = 1, 2, \ldots, m)\) to the linear regression function \( y = ax + b \) can be represented by:

\[ d_i = \frac{|y_i - (ax_i + b)|}{\sqrt{1 + a'^2}}. \] (8)

Denote \( \phi(a, b) = \sum_{i=1}^{m} d_i^2 \) as the representation of the minimum distance square from the point to fitted line.

\[ \phi(a, b) = \frac{1}{1 + a'^2} \sum_{i=1}^{m} [y_i - (ax_i + b)]^2. \] (9)

For the best linear regression function, we need to take a minimum value of \( \phi(a, b) \).

Seeking pole value theory based on mathematics multivariate function, we need to find all the stationary points \((a_0, b_0)\) for \( \phi(a, b) \), and that means satisfying the following conditions:

\[ \frac{\partial \phi}{\partial a} = 0 \]

\[ \frac{\partial \phi}{\partial b} = 0 \] (10)

The equivalent formula of formula (9) can be reconstructed as follows:

\[ (1 + a'^2) \sum_{i=1}^{m} [y_i - (ax_i + b)]^2. \] (11)

\[ \frac{\partial \phi}{\partial a}, \frac{\partial \phi}{\partial a}, \frac{\partial \phi}{\partial a}, \frac{\partial \phi}{\partial b}, \frac{\partial \phi}{\partial b} \] and \( \frac{\partial^2 \phi}{\partial a^2} \) can be obtained by seeking partial derivatives and the second partial derivatives of function (9), and then combined with equation (10), the following equation can be obtained.

\[ \begin{cases} \frac{a(S - T) + (a^2 - 1)R - b(a^2 - 1)}{X} + 2abY - ab^2 = 0, \\ aX - Y + b = 0 \end{cases} \] (12)

where

\[ \begin{cases} X = \frac{1}{m} \sum_{i=1}^{m} x_i, & Y = \frac{1}{m} \sum_{i=1}^{m} y_i \\ R = \frac{1}{m} \sum_{i=1}^{m} x_i y_i, & S = \frac{1}{m} \sum_{i=1}^{m} x_i^2, & T = \frac{1}{m} \sum_{i=1}^{m} y_i^2 \end{cases} \] (13)

The relative clock drift of two nodes always be positive, and we can solve the quadratic equation (12) to obtain the variables can be as follows:
\begin{equation}
\begin{aligned}
a &= -M + \sqrt{M^2 + 4} \\
b &= Y + \frac{X}{2} \left( M - \sqrt{M^2 + 4} \right)
\end{aligned}
\end{equation}

where \( M \) can be given by:
\begin{equation}
M = \frac{(S - T) - \left( X^2 - Y^2 \right)}{R - XY}.
\end{equation}

3.3 Multi-hop time synchronisation

In this paper, we suppose that there is a unique node (referred to as the root node) in the network and its hardware clock plays the role of global clock of the network and its logical clock is always synchronised with its hardware clock. The node within the communication range of root node is called one-hop node and the node located beyond the range of the root node is called multi-hop node. Accordingly, there are multi-hop time synchronisation and one-hop time synchronisation.

In practical WSN applications, the network scale is more than one hop. ETSP can be used in a multi-hop network with time information flooding from the root of the network. Unlike traditional time synchronisation methods, which employ the time of potential parents’ logical clocks as time source to get synchronisation hop by hop, ETSP applies time information of root’s clock for all nodes including multi-hop nodes.

ETSP utilises reference points to perform synchronisation for multi-hop nodes. A reference point contains several parts: sending time stamp from the root node \( s_0 \), receiving time stamp from local \( r_j \), and calculating a serial of packet delays along data transmitting path \( \theta_m \) as shown in Figure 3. Reference points are generated by sending and receiving periodic broadcast messages sourced from the root node or any synchronised node in the network. The root node is a special node that contains the reference time for the whole network. A node that can receive messages directly from the root can collect reference points without considering the packet delays. Nodes over ranging of the broadcast radius of the root can gather reference points indirectly from those synchronised nodes. When a node collects enough reference points, it estimates the regression line as described in the previous section.

In this paper, we assume that every node in the network is equipped with several sensing units, a radio module compliant to IEEE 802.15.4 standard (Jin et al., 2014; Xiao et al., 2006), a microcontroller, and an energy source. When a sender sends a wireless packet, it has the ability to put the time stamp together with the packet during transmission. Then the receiver can save receiving time stamps according to sending time stamp at the same physical time, and even if it is not able to record the time stamp in the same physical time, the time difference can be obtained through transformation.

Part of the synchronisation packet structure is shown in Figure 4. The packet for synchronisation must contain two fields: the send-stamp and the delay time. The send-stamp field \( s_0 \) is sourced by the root node and it will remain unchanged when such packet broadcast to the whole network. The delay time field \( \theta_j \) is varied when the hop-distance increased. Therefore, \( \theta_j \) can be acquired from the following formula:
\begin{equation}
\theta_j = \sum_{i=1}^{n} (a_i \theta_i).
\end{equation}

An \( i \)-hop node can get the relevant time information \( (s_0 + \theta_{i1}, r_j) \), and such reference points are related to the root node’s current hardware clock value. The divergence of time information for linear regression is caused by the delay time \( \theta_i \), so that each node \( i \) should forward the synchronisation message originated by the root immediately. This means that each synchronised node needs to relay such message as soon as possible to minimise \( \theta_i \).

The algorithm about how to schedule the communication links is not discussed here. This paper uses the root’s sending time stamp \( s_0 \) and can greatly reduce the influence of the global clock diverse caused by hop distance between the node \( i \) and the root node. Therefore, ETSP would almost have no special accumulative effect.

**Figure 3** The principle of multi-hop time synchronisation

**Figure 4** Structure of a synchronisation packet

The convergence of the network to a globally synchronised state after a start-up or failure depends on the speed of time information message propagation in the network. Since the synchronisation information depends on the root node’s clock time, there is no effect for time synchronisation of other nodes in the network when failures occur for some nodes and the method brings less uneven error distribution.

3.4 Error analysis

The purpose of this section is to analyse the error of the proposed algorithm ETSP. Section 3.2 indicates that ETSP applies the minimum distance square to get the best linear regression for time synchronisation. In Section 3.3, the diversity of multi-hop nodes synchronisation is the delay
time \( \delta \). In this section, we will analyse how the delay time contributing to time error.

From the time model in equation (1), node \( i \)'s hardware clock drift \( C_i'(t) - 1 \) is always bounded. In other words, there exists a positive number \( \rho_i \) to satisfy that

\[
|C_i'(t) - 1| \leq \rho_i, \quad \forall t \geq 0.
\]

The drift of crystal oscillator that provides node clock running is given by manufacture and always no more than 30 ppm (parts per million). Therefore, the positive number \( \rho_i \) always have a restriction of \( \rho_i \leq 30 \times 10^{-6} \text{s}^{-1} \). A bounded drift variation model defined in Ferrari et al. (2010) to meet the gradually changed environmental conditions: that is

\[
|C_i''(t)| \leq \left| \left( C_i'(t) - 1 \right) \right|^4 \leq 10^{-4}, \quad \forall t \geq 0,
\]

and it seems reasonable to assume that

\[
|C_i''(t)| \leq \left| \left( C_i'(t) - 1 \right) \right|^2 \leq 10^{-10}, \quad \forall t \geq 0.
\]

Therefore, according to Taylor series expansion of \( \tau(t) = C(t) \) in equation (3), the higher order the terms, the less effect they bring.

There are many aspects of environmental conditions that can influence the accuracy of the clock in a node such as humidity, pressure, battery voltage and age of oscillator. However most of these fields will not significantly change in one synchronisation period (tens or hundreds of seconds). One of the most notable factors contributing to the hardware clock drift is temperature. In Lenzin et al. (2009), the hardware clock frequency of a Mica2 node for different ambient temperature is up to one microsecond per second when the temperature of the environment changes no more than 5°C. Therefore, the temperature-based drift model can be give as depicted in Huang et al. (2014) as follows:

\[
|C_i'(t_1) - C_i'(t_2)| \leq \frac{|t_1 - t_2|}{5\Omega} \times 10^{-6} \text{s}^{-1}, \quad \text{when } |t_1 - t_2| \leq \Omega.
\]

(17)

The variable \( \Omega \) denote for the time needed for the temperature to change 1°C. In fact, the change of environmental temperature is very slow: for instance, the lowest temperature for a day is always at 6 a.m. and the highest at 2 p.m., and the difference between these two points is commonly no more than 15°C. This means that the change of environmental temperature is no more than two degree in 1 h.

The difference for a one hop node and a multi-hop node is the delay time \( \delta \). For a one hop node, this value is zero in contrast to non-zero for a multi-hop node. According to formula (17), the diversification of time delay can be written as follows:

\[
\Delta \delta \leq \frac{10^{-6} \text{s}^{-1}}{5\Omega} \sum_{i=1}^{n} \theta_i.
\]

(18)

According to Maroti et al. (2004), the common time for sending a synchronisation packet between two adjacent nodes is 630 ms. However, we test a network with 100 nodes, 10 hops network and no more than 40 bytes' length packet in carrier access and multiple access (CSMA) mode (Ghaboosi et al., 2008; Xiao and Li, 2004), the delay time for two nodes is no more than 100 ms. This means that the duration of a synchronisation process is no larger than 1 s and the change of time delay will be always far less than 1 μs for all nodes in the network. Then there will have almost no accumulative effect for the time information that is used for time synchronisation process. Therefore, it is the more stable method for multi-hop networks.

4 Simulation

We analyse the performance of our algorithm ETSP in this section. For a more realistic reflection of the actual situation, the input data are obtained by the actual operating results of real nodes. Two nodes are placed within the communication range from each other, and the simulated data is obtained by the recording of receiving time stamp and sending time stamp during the data exchanging with each other. The data is recorded every 10 s to provide enough data for simulations. Therefore, we can obtain more meaningful test results with the simulation of different algorithms by using the same actual data.

A pair of sending time stamp and receiving time stamp is called a point and we have thousands of such points for simulation. By using the time information from real nodes, we can avoid to model the analogue of clock in a node (the actual time in a node contains too many uncertainties including environmental changes, ageing of components, etc.). Such a method makes our simulation results reflect the true effect of the actual operation of the algorithm.

The comparison of synchronisation results between the least square method and the minimum distance square method under different parameter settings is given in Figure 5. Thousands pairs of time stamps are acquired throughout days and nights, in order to get a true reflection of the time changes for nodes in a whole day. We simulate the time error for the two different regression methods and get a statistical result for contrasting: if the minimum distance method is superior to the least squares, error recorded as a value 1, otherwise –1. We accumulate the scores of different synchronisation periods with different fitting points. The curves in Figure 5 are always with positive effect and this indicates that the minimum distance method will always get better regression results than that of the least square method.

Figure 6 expresses the average error of the two methods with different fitting points. When the fitting points are less, the performance of the minimum distance method will be better and the decrease of fitting points will weaken the computational complexity of the algorithms. When using four fitting points, both of the two algorithms have good performance, i.e., the average error of different synchronisation period is the lowest. However, the result does not indicate that the regression with four fitting points is always the best for all kinds of synchronisation periods.
We will analyse the results of different fitting points with different synchronisation periods in the following part of the paper.

**Figure 5** The comparison of synchronisation result between the least square method and the minimum distance square method (see online version for colours)

![Graph showing comparison](image)

**Figure 6** The average error of time synchronisation with the minimum distance square method and the least square method of different fitting points (see online version for colours)

![Graph showing average error](image)

From the above analysis, the minimum distance square method will acquire better performance especially for less fitting points, so that the following simulation will focus on the minimum distance square method. Figure 7 exhibits the performance of the minimum distance square method in different periods with a group of fitting points. The diversities are becoming worse with the increasing of the fitting points, especially for the synchronisation periods over 300 s. However, the lines of lower three synchronisation periods show the opposite trend: time synchronisation effects is better according to the increasing of fitting points (displaying in the enlarged part in the chart). Therefore, we can conclude that when the synchronisation period is larger, the fewer fitting points are better and when the period is small, more fitting points are needed to get better linear regression results. But when considering all the effects of different periods, the choice of four fitting points without thinking about the synchronisation periods would be better as depicted in Figure 6, and it has the lowest average time error when using four fitting points.

Since energy is an important factor for WSNs and the synchronisation period will greatly affect energy consumption as depicted in Ganesan et al. (2009), it is essential to employ a period as large as possible to save energy. But this does not mean that the bigger the better for synchronisation period. The fact is that the greater periods bring the worse errors. We summarise the different synchronisation period using different amount of synchronisation points, and the regression results are plotted as shown in Figure 8. For the synchronisation period equal and below 120 s, the average time error is always less than 1 time tick. However, when the synchronisation periods grow more than 300 s, the fitting results would become worse with the increasing of fitting points. Therefore, our results can be used as a reference for selecting an optimal synchronisation period.

**Figure 7** The performance of the minimum distance square method in different periods with a group of fitting points (see online version for colours)

![Graph showing performance](image)

**Figure 8** The results of the minimum distance square method in different synchronisation periods (see online version for colours)

![Graph showing results](image)

When the synchronisation period is small, the regression results with more fitting points will get better performance but on the other side (with a larger period), a small number of fitting points will get better results. The statistical results of the maximum errors are the same case as depicted
in Figure 9. Therefore, there must be a balance for synchronisation periods and the amount of fitting points that can get optimal performance of time synchronisation for WSNs.

Figure 9 The maximum error distribution in different synchronisation periods using different number of fitting points (see online version for colours)

5 Experiment results

To demonstrate the feasibility of executing the proposed algorithm on real sensor nodes, we built a prototype of ETSP in the SCSC-RFA1 WSN platform (Wang et al., 2014). The main goal of the implementation was to show the stability and reliability of our algorithm for a multi-hop network.

For traditional methods, the time information is broadcasted progressively. This means that every synchronised node will send messages with its own logical time for its children to get synchronised, and such a method would bring more errors for larger hops because the logical time has larger accumulative error. We tested a ten-hop network using FTSP and the time errors for the first hop and the 10th hop are shown in Figure 10. The time error is ranged from −2 to 5 time ticks for the first hop in contrast with −22 to 29 time ticks for the 10th hop. Moreover, big error range fluctuation would contribute to the extreme instability for communication link.

Figure 10 Comparison of time error for different hops (see online version for colours)

To reveal the performance of our method in a multi-hop network, we implemented a WSN composed of tens of sensor nodes with a topology of tens of hops. We formed a 20-hop network in contrast with the FTSP under the same condition. The average time error and the maximum error for nodes in each hop with their parents were recorded and analysed statistically as shown in Figure 11. The average time error of ETSP is around 1 time tick although the maximum error of ETSP varies from 3 to 7 time ticks. However, the average error for FTSP is larger even when the number of hops increases and the maximum time error grows more sharply than that of ETSP. This implicates more fluctuations for time errors of network nodes. Therefore, we conclude that ETSP can acquire more stable time synchronisation for multi-hop WSNs even for such extensible networks with tens of hops.

Figure 11 Time error for different hops between FTSP and ETSP (see online version for colours)

The experimental results show that the proposed protocol has a very good application prospect in large-scale networks especially for multi-hop networks. This is important for a wireless network with thousands of nodes and even millions. The time synchronisation algorithm developed for large-scale networks could become the hot topic for the research of wireless synchronous technology.

6 Conclusions

This paper proposed an ETSP for WSNs. The improvement of linear regression method shows better performance than the least square for time synchronisation in wireless networks; time information spread from the root node, and this means that all synchronised process is sourced from the root node so that it is easy to control the rhythm of time synchronisation. Since the synchronisation information depends on the hardware clock of the root node, there would bring less effect to the time synchronisation when failures occur for some nodes and less uneven error distribution. Therefore, ETSP has no spatial accumulative effect when the scale of the network increases and is highly suitable for a large-scale network. The simulation and the experiments show that the performance of our method has markedly results for extensible WSNs.
Furthermore, the input data of our simulation is sourced from real nodes and the same data is used for the two methods with different parameters. This semi-real simulation can quickly obtain different results for comparison under the same conditions, and this is more real than the simulations and easier to achieve outcomes under the same condition than real experiments. Future work will focus on the balance between the time period and the time error to minimise the energy consumption for time synchronisation and even analysis together with the fitting points to get the optimal method for time synchronisation in WSNs.

Acknowledgements

The authors wish to thank project (61401257, 61501282) supported by National Natural Science Foundation of China, Basic Research foundation by Shandong Academy of Sciences and Director foundation by Shandong Provincial Key Laboratory of Computer Networks.

References


