Computer Communications 37 (2014) 77–91

Contents lists available at ScienceDirect

Computer Communications

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

Long term and large scale time synchronization in wireless sensor networks

Ge Huang [a], Albert Y. Zomaya [a], Flávia C. Delicato [b,*], Paulo F. Pires [b]

[a] Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, NSW 2006, Australia
[b] UFRJ, Federal University of Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 16 September 2011
Received in revised form 24 June 2013
Accepted 7 October 2013
Available online 17 October 2013

Keywords:
Quadratic Taylor synchronization
Time synchronization protocols
Wireless sensor networks

ABSTRACT

Time synchronization is very important in wireless sensor networks (WSNs). Many applications, for example natural disaster monitoring and structural health monitoring of huge buildings, require a highly accurate, long-term and large-scale time synchronization among the sensor nodes that compose the network. In this paper, we propose a new time synchronization protocol, named *2LTSP* (long term and large scale time synchronization protocol), which aims at addressing such requirements. Theoretical analysis and simulation results show that when the synchronization period is less than 100 s, the error of *2LTSP* is within 0.6 ms, no matter how large the size of the network is. Besides, when the required synchronization error limit is 9 ms, the communication cost of *2LTSP* is less than 3 packets per hour in networks of any size. Therefore, *2LTSP* is highly accurate and energy-efficient even for large-scale and long-term running networks.

1. Introduction

For most WSN applications, a scalable time synchronization service responsible for maintaining a common notion of time among the network nodes is often required to enable sleep scheduling, localization, data fusion, in-network signal processing and nodes' coordination, among others WSN essential functions.

Each node in the WSN is equipped with a hardware clock used as reference for controlling its internal signal processing functions. On top of the hardware clock, each node has a logical clock whose value and frequency can be changed by time synchronization algorithms. The rates of the hardware clocks of sensor nodes depend on conditions like the temperature, the supply voltage, or crystal quartz aging, all changing over time. Even under perfectly stable and identical environmental conditions, the hardware clocks of different nodes will run at different speeds. Moreover, each node has different initial hardware clock value. On one hand, the initial hardware clock value is set to be different during manufacturing. On the other hand, in order to save energy, the node goes through a cycle of sleep and wake up based on externally detected events. Thus, hardware clock error is inevitable, and time synchronization is essential.

Time synchronization algorithms are essentially functions to synchronize local clocks (clocks at individual sensor nodes) with the WSN global clock. These algorithms are based on exchanging messages time-stamped with the values of the hardware clock among nodes. The goal of a time synchronization algorithm is to ensure, by adjusting local clocks to the global clock at regular intervals, that logical clocks of any two nodes in the WSN are as closely synchronized as possible at all times. Each time synchronization algorithm will change, in its own way, the logical clocks of the synchronized nodes in its synchronization process. Different approaches for synchronizing nodes are prone to different drawbacks and errors. Moreover, most of current time synchronization algorithms for WSNs meet requirements of specific application domains. Our proposed algorithm aims at covering a broad range of application requirements while trying to overcome several limitations of existing approaches. In the next paragraphs, we briefly discuss the current main issues and challenges for time synchronization algorithms in WSNs. First, we discuss the issue of spatial cumulative error in WSN time synchronization algorithms. Second, we analyze the accuracy provided by such algorithms. Finally, we discuss the different approaches in the light of energy consumption.

If a synchronization algorithm can only achieve its goal by performing multiple steps (i.e. multiple exchanges of messages among nodes), we call it an *accumulative one*, otherwise a *non-accumulative one*. Since WSN usually span many hops, the synchronization error introduced in *accumulative* algorithms by instability of the clock frequency and variable delays of message exchange between pair wise synchronizing nodes accumulates as the path length grows [30]. In other words, as we prove in Section 2.3 and demonstrate in our simulation experiments, the error of any accumulative

* Corresponding author.
E-mail address: fdelicato@gmail.com (F.C. Delicato).

0140-3664/$ - see front matter Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.10.003

synchronization algorithm has spatial accumulative effect. Designing of non-accumulative synchronization algorithms is a challenging work. The authors in [1] have shown that even popular clock synchronization algorithms have several problems in this context. For instance, the state-of-the-art clock synchronization algorithm FTSP [2] exhibits an error that grows exponentially with the size of the network. To the best of our knowledge, the PulseSync synchronization algorithm [1] is the only non-accumulative algorithm in the literature. Since the error of PulseSync has no spatial accumulative effect, it is suitable for large-scale networks. However, the PulseSync is not suitable for long-term running networks due to its low accuracy (refer to Section 3).

For each node in a WSN, there exists, in theory, a unique function that maps, at any physical time, the local (hardware) clock to the global clock (refer to Section 4.3), which we call the *clock transfer function*. Time synchronization algorithms can be classified into two categories according to the type of *clock transfer function*: *linear* [1–7] and *quadratic* [8], which respectively use linear polynomials and quadratic polynomials to compute the clock transfer function. Moreover, a time synchronization algorithm is called a *quadratic Taylor one* if it uses the quadratic Taylor expansions to compute the clock transfer function. Quadratic Taylor synchronization algorithms are much better, in term of synchronization accuracy, than linear synchronization algorithms (refer to Section 2.2). Most existing synchronization algorithms have a low accuracy because they approximate the global clock using linear polynomials rather than quadratic Taylor expansions. To satisfy the application's demand for synchronization accuracy, synchronization algorithms with low accuracy must be executed frequently. For example, FTSP [2] has to resynchronize the nodes in the network every minute in order to achieve 90 ms synchronization error (an acceptable synchronization error limit for several applications). Consequently, synchronization algorithms with a low accuracy are only useful for short-term applications such as surveillance and object tracking and are unsuitable for efficient duty cycling and other applications that require continuous time synchronization such as synchronized sampling, because they spend a lot of energy due to their frequent resynchronization [8]. Despite the higher accuracy achieved with quadratic synchronization algorithms, their design is a challenging work. To the best of our knowledge, the synchronization algorithm proposed in [8] is the only quadratic algorithm currently reported in the literature. Its major shortcoming is to be unsuitable for large-scale networks (refer to Section 3).

When a synchronization process ends, the logical clock of each synchronized node is close to the global clock. However, over time, the logical clocks of the synchronized nodes gradually differ from the global clock. To obtain acceptable synchronization accuracy, the synchronization process has to be started at regular intervals. If the synchronization process is started once every T seconds, we say the *synchronization period* is T seconds. It is clear that a shorter synchronization period leads to higher synchronization accuracy. However, shorter synchronization periods also lead to larger energy consumption in the network nodes. Since energy saving is a crucial requirement in WSNs, the balance between synchronization accuracy and energy consumption needs to be maintained. For a given time interval T (unit: second) and time synchronization algorithm f, let's assume that $E_T(f)$ (unit: microsecond) represents the f's synchronization error for a synchronization period equal to T. For a given synchronization error limit ε (unit: microsecond) and time synchronization algorithm f, we use $T_\epsilon(f)$ (unit: second) to denote the maximal time interval T such that $E_T(f) \leqslant \varepsilon$, that is, $T_\varepsilon(f) = \max\{T|E_T(f) \leqslant \varepsilon\}$. We call $T_\varepsilon(f)$ the f's *immanent period* in the condition that the synchronization error limit is equal to ε (for short, *ε-period*). The longer the immanent period of a time synchronization algorithm is, the more energy-efficient it should be. The immanent period is used to compute the communication cost of time synchronization algorithms (refer to Section 2.4). The most representative time synchronization algorithms existing in the literature have all a short immanent period. For example, the immanent period considering $\varepsilon = 9\mu s$-periods of FTSP [2], RSP [4] and PulseSync [1] are, respectively, 100 s, 300 s and 100 s. Therefore, they are not suitable for long-term running networks.

Considering all the aforementioned issues and challenges, in this paper we present a new non-accumulative, quadratic Taylor time synchronization algorithm called *2LTSP*. Since *2LTSP* is non-accumulative, unlike most existing time synchronization algorithms, its error has no spatial accumulative effect as the size of the network increases. Therefore, it is suitable for large-scale networks. Moreover, as *2LTSP* approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as most existing schemes do, it is highly accurate. Theoretical analysis and simulation results show that when the synchronization period is less than 100 s, the error of *2LTSP* is within 0.6 ms, no matter how large the size of the network is. Moreover, when the required synchronization error limit (ε) is 9 microseconds, the immanent period of *2LTSP* is larger than or equal to 1300 s in networks of any size. Since only a packet is flooded to the entire network in a synchronization process of *2LTSP* (see Section 4) and its immanent period is long, then it is an energy-efficient protocol. This means *2LTSP* is suitable for long-term running networks.

The rest of this paper is organized as follows. In Section 2, we will give some definitions and assumptions that are important to understand the formulation of our proposal. In Section 3, we review the existing clock synchronization protocols that are directly related to our approach. The proposed synchronization scheme is presented in Section 4. Error analysis is given in Section 5. Section describes the performed simulation experiments. Section 7 describes the implemented prototype of the proposed algorithm. Finally, Section 8 draws the conclusion. In addition, some proofs were included in Appendixes A and B.

2. Concepts and assumptions

This section is organized as follows. In Subsection 1, we present the WSN model and communication frame format adopted in this work. In Subsection 2, we discuss the Quadratic Taylor synchronization. In Subsection 3, we discuss the spatial accumulative effect in WSN. In Subsection 4, we define communication cost model used in this work.

2.1. WSN model and communication frame format

In this work, a WSN is modeled as a graph. The *network graph* $G(N)$ of a WSN N is defined as follows: (1) the vertex set of $G(N)$ is the set of all nodes in network N; (2) for any two vertices A and B, the sufficient and necessary condition for the existence of an edge connecting them in $G(N)$ is that they are within radio range of each other. Hereafter, we will use the terms from graph theory without explaining them. For any two nodes A and B, the distance between them in the network graph is called the *hop distance*. In this paper, we suppose that there exists a unique node (referred to as the *reference node*) such that its hardware clock is the global clock of the network and its logical clock is always synchronized with its hardware clock. The *network radius* is defined as the hop distance between the reference node and the farthest node from it. A network whose radius is equal to *one* is called a *one-hop network*, while a network whose radius is larger than *one* is called a *multi-hop network*. A time synchronization in a one-hop network

is called a *one-hop synchronization*, while a time synchronization in a multi-hop network is called as a *multi-hop synchronization*.

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 [9–12], a microcontroller and an energy source. Fig. 2.1.1 shows a schematic view of the IEEE 802.15.4 frame format.

During the transmission process, when the required number of preamble bytes have been transmitted, the radio module will automatically transmit the one byte long *SFD* (Start of frame Delimiter). The *SFD* is fixed and it is not possible to change this value by software. Once the *SFD* has been transmitted, the radio modulator will start reading data from the *TX FIFO*. It expects to find the frame length field followed by MAC header and MAC payload. The frame length field is used to determine how many bytes are to be transmitted. Bytes may be added to the *TX FIFO* during transmission. Frame reception starts with detection of the *SFD*, followed by the length byte, which determines when the reception is complete. Once the *SFD* has been received, the radio demodulator will start writing received data to the *RX FIFO*. Most of current WSN radio transceivers are endowed with a packet sniffer. Packet sniffing is a non-intrusive way of observing data that is either being transmitted or received by the radio module.

Fig. 2.1.2 shows the structure of a *synchronization packet*. The *synchronization packet* has 8 fields. Both fields F_1 and F_2 are flags consisting of 8 bytes of zeros, and the other fields all have 4 bytes. Field ID_1 is the ID number of the reference node, while field ID_2 is the ID number of the sender node. In the procedure that the sender transfers a *synchronization packet* to the receiver, we need to obtain three pairs of time stamps: (i) the pair consisting of the time that the *SFD* has just sent and the time it was received; (ii) the pair consisting of the time which the field F_1 has just sent and the time that such field was received; and (iii) the pair consisting of the time that the field F_2 has just sent and the time it was received.

The fields $T_{0,V}$ ($V = 3$–5) are respectively the reference node's current hardware clock values when it has just sent the *SFD*, the first flag and the second flag. Note that the node microcontroller can know, by observing the output of the packet sniffer, when the node radio has just sent/received the *SFD*, the first flag and the second flag. The field $\dot{L}$ is the sender's current logical clock value when it has just sent the first flag. Since the logical clock of the reference node is the same as its hardware clock, $\dot{L} = T_{0,4}$ when the reference node is the sender.

ID_1	F_1	ID_2	F_2	$T_{0,3}$	$T_{0,4}$	$T_{0,5}$	$\dot{L}$

Fig. 2.1.2. Structure of a *synchronization packet*.

2.2. Quadratic Taylor synchronization

Since the third derivative of the hardware clock function is very small [14], we know from Lagrange's remainder theorem that its quadratic Taylor expansion is a very good approximation. In addition, because the third derivative of the hardware clock function is much smaller than its second derivative [14], we know from Lagrange's remainder theorem that its quadratic Taylor expansion is a much better approximation than its linear Taylor expansion. Moreover, we know from [14] that the linear Taylor expansion of the hardware clock function is a better approximation than other linear polynomials. Therefore, the quadratic Taylor expansion of the hardware clock function is a much better approximation than linear polynomials. The above argument shows that quadratic Taylor synchronization algorithms are much better, in term of synchronization accuracy, than linear synchronization algorithms.

2.3. Spatial accumulative effect

By our definition, the hardware clock of the reference node is just the global clock. In order to adjust local clocks to the global clock, synchronization algorithms need the hardware clock information of the reference node. Suppose that R is the reference node and A is a node. The precondition that A can synchronize with R is that there exist nodes $R = A_1, ..., A_n = A$ such that for each i ($1 < i \leqslant n$), A_i is within the radio range of $A_{i\text{-}1}$.

Accumulative synchronization algorithms achieve the synchronization between A and R in the following way. At physical time t_1, R ($=A_1$) obtains an estimation $\hat{M}_1$ of its own current hardware clock information M_1 by using its own hardware time stamps and sends A_2 a message including $\hat{M}_1$. At physical time t_2 ($t_2 > t_1$), A_2 obtains an estimation $\hat{M}_2$ of the R's current hardware clock information M_2 by using $\hat{M}_1$ and sends A_3 a message including $\hat{M}_2$. At physical time t_3 ($t_3 > t_2$), A_3 obtains an estimation $\hat{M}_3$ of the R's current hardware clock information M_3 by using $\hat{M}_2$ and sends A_4 a message including $\hat{M}_3$. Since the data used by A_2 contains an intrinsic error, and the data used by A_1 ($=R$) do not contain any error, the estimation error of A_2 is larger than that of A_1. The estimation methods used by both A_2 and A_3 are certainly the same. However, the data used by A_3 have larger error than the data used by A_2. Therefore, the estimation error of A_3 is larger than that of A_2. Such behavior continues until at physical time t_n ($t_n > t_{n\text{-}1}$) A ($=A_n$) receives the message (which includes the approximate value $\hat{M}_{n-1}$ of the R's hardware clock information $M_{n\text{-}1}$ at physical time $t_{n\text{-}1}$ estimated by $A_{n\text{-}1}$ using $\hat{M}_{n-2}$) from $A_{n\text{-}1}$ and adjusts its own logical clock according to the approximate value $\hat{M}_n$ of the R's current hardware clock information M_n estimated by A using $\hat{M}_{n-1}$. From the above description, we know that the estimation error $\hat{M}_i - M_i$ in the ith step affects the estimation error $\hat{M}_{i+1} - M_{i+1}$ in the $(i+1)$th step.

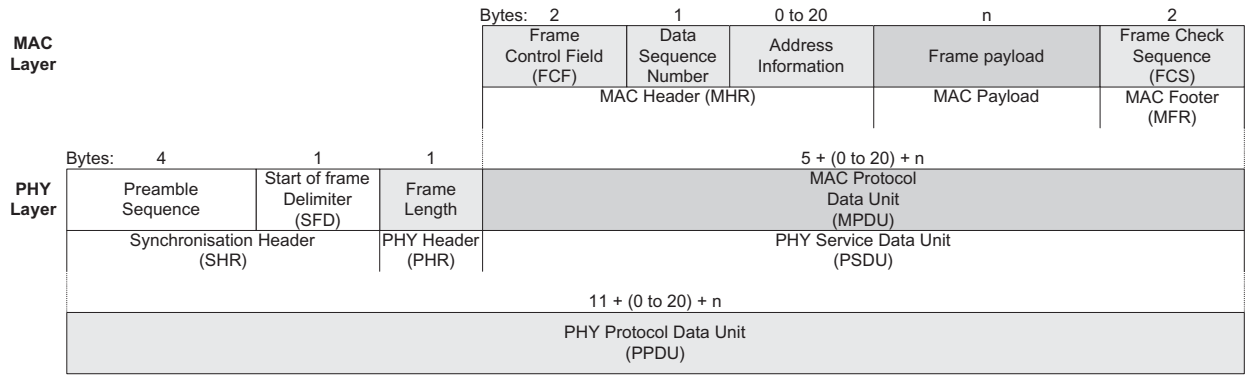

Fig. 2.1.1. Schematic view of the IEEE 802.15.4 frame format [13].

It is clear that the larger the hop distance between R and A is, the larger is the estimation error $\hat{M}_n - M_n$ of A. Since the estimation error for the global clock dramatically affects synchronization accuracy, the A's synchronization error will increase as the hop distance between R and A grows. This phenomenon is called the *spatial accumulative effect* of accumulative synchronization algorithms. We know from computational mathematics that when the argument has an error, the higher the degree of a polynomial is, the larger its error is. Therefore, accumulative quadratic synchronization algorithms have stronger spatial accumulative effect than accumulative linear ones.

Non-accumulative synchronization algorithms achieve the synchronization between A and R in the following way. At physical time t_1, R (=A_1) obtains an estimation $\hat{M}_1$ of its own current hardware clock information M_1 by using its own hardware time stamps and sends A_2 a message including $\hat{M}_1$. Each A_i ($1 < i > n$) only forwards the message sent by R. At physical time t_n ($t_n > t_1$), A (=A_n) receives, from A_{n-1}, the message sent by R and adjusts its own logical clock according to the approximate value $\hat{M}_n$ of the R's current hardware clock information M_n estimated by A using $\hat{M}_1$. Since the A's estimation error $\hat{M}_n - M_n$ for the global clock is not related with the hop distance between R and A, the A's synchronization error is not related with the hop distance between R and A. Therefore, non-accumulative synchronization algorithms have no spatial accumulative effect.

2.4. Communication cost

We define the *communication cost* in the condition that synchronization error limit is equal to ε as the number of packets flooded to the entire network in time unit to ensure that synchronization error is not larger than to ε. It is not difficult to see that the communication cost in the condition that synchronization error limit is equal to ε is equal to the quotient of the number of packets flooded to the entire network in one synchronization process and the ε-period.

3. Related work

Since 2002 time synchronization in WSNs has been widely researched [1–7,15–24]. However, most existing time synchronization algorithms are application related, which means the algorithms only work effectively under certain conditions and for specific application domains. In contrast to such approaches, 2LTSP algorithm aims to be as agnostic as possible to the target application domain and therefore it seeks to cover a wide range of application requirements such as high accuracy, energy efficient (long-term) and large-scale WSN.

The Flooding Time Synchronization Protocol (FTSP) [2], which uses a rooted ad hoc tree to organize the network topology, is an accumulative synchronization algorithm. FTSP synchronizes the time of a sender to possibly multiple receivers making use of a single radio message time-stamped at the MAC layer at both the sender and the receiver sides. Linear regression is used in FTSP to compensate for clock drift as suggested in [16]. Because the error of FTSP grows exponentially as the size of the network grows [1] and FTSP has a very short immanent period [8], it is not suitable for large-scale networks or long-term running networks.

Ratio-based Time Synchronization Protocol (RSP) [4] uses two continuous (in the sense that one message is sent after the other) synchronization messages to synchronize the clock of the receiver with that of the sender. By marking the time-stamp of the message in MAC layer, some delay time in handling the transmission can be eliminated and the accuracy is enhanced. As reported in the literature and observed in our simulations (refer to Section 6.1), the 9μs-period of RSP is less than 300 s. This means that only when synchronization period is not larger than 300 s, the synchronization error of RSP can be less than 9 ms. Thus, RSP is unsuitable for long-term running networks. Moreover, simulated experiments show that its error grows exponentially as the size of the network grows. Therefore, it is also not suitable for large-scale networks.

The Pulse Synchronization Protocol (PulseSync) [1], which uses a rooted breadth-first search tree to organize the network topology, is a non-accumulative synchronization algorithm. Since the error of PulseSync has no spatial accumulative effect, it is suitable for large-scale networks. However, the 9μs-period of PulseSync is less than 100 s (the same as FTSP's). Therefore, PulseSync is not suitable for long-term running networks.

The synchronization algorithm proposed in [8] approximates the global clock through a quadratic polynomial rather than a linear polynomial as used in previous researches. Our conducted simulated experiments show that its 9 μs-period is less than 100 s and its error grows exponentially as the size of the network grows. Therefore, it is not suitable for both large-scale networks and long-term running networks. In addition, since this algorithm has very high computational complexity, it is unsuitable for use in the resource constrained WSN nodes.

Zheng et al. [6] propose a unified framework to jointly solve the problems of localization and time synchronization in WSNs from the signal processing perspective. Some pioneer researchers have noticed the similarities between the problem of source localization and time synchronization [21,22]. However, they only pointed out the links between the two problems and explored the possibility of jointly implementing localization and synchronization at the protocol level. The algorithm is the first work that mathematically models and analyses the joint localization and synchronization problem from signal processing point of view. They first study the maximum likelihood method and least square method for joint localization and time synchronization, and then use this method to solve the problems of localization and time synchronization. As this algorithm does not achieve clock drift compensation, its synchronization accuracy is very low. Since this algorithm is accumulative, it has strong spatial accumulative effect.

Rahamatkar et al. [7] propose a reference-based, tree structured time synchronization algorithm called TSRTS (Tree Structured Referencing Time Synchronization). The TSRTS is based on the protocol proposed by [23]. Its aim is to minimize the complexity of the synchronization. The TSRTS works on two phases. First phase used to construct an ad hoc tree structure and second phase used to synchronize the local clocks of sensor nodes. The goal of the TSRTS is to achieve a network wide synchronization of the local clocks of the participating nodes. The TSRTS synchronizes the time of a sender to possibly multiple receivers utilizing a single radio message time-stamped at both the sender and the receiver sides. MAC layer time stamping can eliminate many of the errors, as observed in [24]. Linear regression is used in TSRTS to compensate for clock drift as suggested in [16]. Since TSRTS is accumulative, it has strong spatial accumulative effect.

4. Long term and large scale time synchronization protocol (2LTSP)

Our synchronization algorithm *2LTSP* is an unstructured time synchronization scheme, meaning it has no special demand on the topology structure of the network. In contrast to tree based approaches, unstructured approaches do not maintain any global configuration with regard to the network topology, and, consequently, node mobility and failure do not cause particular problems. The node with the lowest ID is elected as the reference node, whose hardware clock will serve as the global clock, using

any leader election algorithm such as that in [2]. If this node fails, then the node with the lowest ID in the remaining network is elected as the new reference node. The reference node periodically floods the network with a *synchronization packet*. After receiving the packet, the nodes within the radio range of the reference node will correct their logical clocks and become synchronized nodes. All synchronized nodes forward the packet after updating its field $\dot{L}$. All non-synchronized nodes, upon receiving the synchronization packet will do the same actions. The aforementioned steps will repeat until each node has performed the same actions. A non-synchronized node can receive packets from multiple synchronized nodes. In such a case, the non-synchronized node only considers the first received packet. From the above description, we know that only a *synchronization packet* is flooded to the entire network in a single *2LTSP* synchronization process. Since at least a packet is flooded to the entire network in any single synchronization process, *2LTSP* has the minimum communication cost in a single synchronization process.

For simplicity and without loss of generality, in this paper we will consider linear networks as shown in Fig. 4.1.

Node 0 is the reference node. Only node 1 is within the radio range of node 0. For any $0 < i < n$, only nodes $i-1$ and $i+1$ are within the radio range of node i. Only node $n-1$ is within the radio range of node n. For any node i, in this paper we use $C_i(t)$ and $L_i(t)$ to represent respectively the hardware clock value and logical clock value of node i at physical time t.

The rest of this section is organized as follows. In Subsections 1 and 2, we will describe in details our algorithm *2LTSP*. The theoretical basis for *2LTSP* is given in Subsection 3.

4.1. Starting a synchronization process

Node 0 starts a synchronization process by preparing a *synchronization packet M*. Following we describe how node 0 transmits *M*. Firstly, node 0 sets fields ID_1 and ID_2 of *M* to its own ID number. Then it begins to send *M*. When node 0 has just sent the *SFD*, it records its current hardware clock value T_a and sets field $T_{0,3}$ of *M* to T_a. As soon as node 0 has sent the first flag, it records its current hardware clock value T_b, and sets fields $T_{0,4}$ and $\dot{L}$ of *M* to T_b. Once node 0 has sent the second flag, it records its current hardware clock value T_c and sets field $T_{0,5}$ of *M* to T_c. Each node can know, by observing the output of the packet sniffer, when its radio module has just sent/received the *SFD*, the first flag and the second flag. Some type of nodes (for example CC2520 nodes [13]) support the packet sniffer in hardware, and others (for example CC2420 nodes [25]) do not. The type of nodes that do not support the packet sniffer in hardware can realize this by programming.

4.2. Completing one synchronization step

In the procedure during which node i ($1 \leqslant i \leqslant n$) receives *M* from node $i-1$, node i can obtain its own hardware time stamps $T_{i,0}$, $T_{i,1}$ and $T_{i,2}$. They are respectively the node i's current hardware clock values when it has just received the *SFD*, the first flag and the second flag. When the procedure that node $i-1$ sends *M* to node i ends, node i obtains the reference node's hardware time stamps $T_{0,V}$ ($V = 3$–5) and its own hardware time stamps $T_{i,V}$ ($V = 0$–2) as well as the node $i-1$'s current logical clock value $\dot{L}_{i-1}$ when node $i-1$ has just sent the first flag, as shown in Fig. 4.2.1.

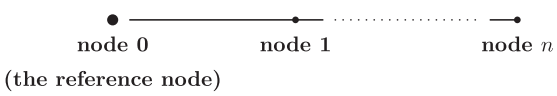

Fig. 4.1. Network logical topology.

Note that in Fig. 4.2.1, $T_{i\text{-}1,V}$ ($V = 0$–2) are respectively the node $i-1$'s current hardware clock values when it has just received the *SFD*, the first flag and the second flag from node $i-2$, while $T_{i\text{-}1,V}$ ($V = 3$–5) are respectively the node $i-1$'s current hardware clock values when it has just sent the *SFD*, the first flag and the second flag to node i.

As soon as *M* has been completely received, node i modifies field ID_2 of *M* to its ID number and begins to forward *M*. When node i has just sent the *SFD*, it records its current hardware clock value $T_{i,3}$. As soon as node i has sent the first flag, it accomplishes in its turn the following four tasks. The first task is to record its current hardware clock value $T_{i,4}$. The second task is to set its current logical clock value $\dot{L}_i$ by the following formula:

$$\dot{L}_i = \dot{L}_{i-1} + \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}(T_{i,4} - T_{i,1}) + \frac{1}{2} \times \frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{(T_{i,2} - T_{i,1})(T_{i,1} - T_{i,0})}(T_{i,4} - T_{i,1})^2. \quad (4.2.1)$$

From the aforementioned discussion, we know that $\dot{L}_0 = T_{0,4}$. Hence, the recursive formula (4.2.1) can be successfully computed. From the next subsection, we know that Eq. (4.2.1) is a quadratic Taylor expansion approximation to the global clock. The third task is to update the calculation formula of its logical clock $L_i(t)$ as follows:

$$L_i(t) = \dot{L}_i + \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \cdot (C_i(t) - T_{i,4}). \quad (4.2.2)$$

From the next subsection, we know that Eq. (4.2.2) computes the node i's logical clock by imitating the global clock after the synchronization process has ended. The fourth task is to modify field $\dot{L}$ of *M* to its current logical clock value $\dot{L}_i$

Once node i has sent the second flag, it records its current hardware clock value $T_{i,5}$. From Eqs. (4.2.1) and (4.2.2), we know that *2LTSP* directly estimates an approximate formula from the node i's hardware clock to the hardware clock of the reference node (node 0), rather than firstly estimates an approximate formula from node i to node $i-1$. Then it composes an approximate formula from node i to the reference node using the approximate formula from node i to node $i-1$ and the approximate formula from node $i-1$ to the reference node. This means *2LTSP* is a non-accumulative time synchronization algorithm. From the above argument, we know that *2LTSP* approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as most existing schemes do. We can see from Eq. (4.2.1) that *2LTSP* only uses 15 arithmetic operations and consumes 28 bytes of memory – 4 bytes for each one of the seven variables occurring in $\dot{L}_i$. Hence, *2LTSP* is suitable to resource constraint systems such as WSN nodes. In the next subsection, we will discuss the theoretical basis for *2LTSP*.

4.3. Theoretical basis for 2LTSP

The goal of this subsection is to describe the theoretical base for the proposed algorithm *2LTSP*. A crystal oscillator provide the hardware clock of each node. Most crystal oscillators are not very precise because the frequency, which makes the time increase, is never exactly right [26].

The problem to synchronize the node i's logical clock to the node 0's hardware clock can be reduced to finding the time transformation function $g_{0\leftarrow i}(x)$ such that $C_0(t) = g_{0\leftarrow i}(C_i(t)), \forall t \geqslant 0$. In theory, $g_{0\leftarrow i}(x)$ can be constructed from the following parametric equations:

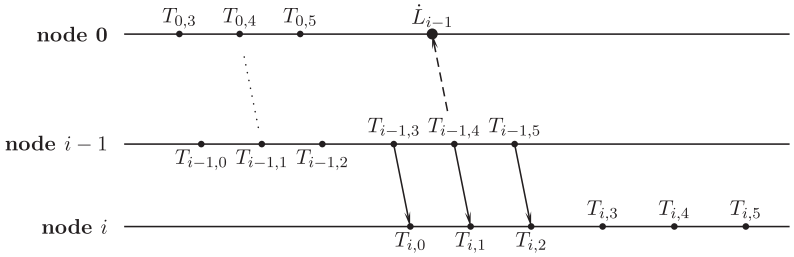

Fig. 4.2.1. Multi-hop synchronization.

$$\begin{cases} y = C_0(t) \\ x = C_i(t) \end{cases}, \quad 0 \leqslant t < +\infty. \tag{4.3.1}$$

As $x = C_i(t)$ is a strictly increasing function of t, $t = C_i^{-1}(x)$. Thus $y = C_0(C_i^{-1}(x))$, that is, $g_{0\leftarrow i}(x) = C_0(C_i^{-1}(x))$ Since we do not know the exact expressions of $C_0(t)$ and $C_i(t)$, we cannot derive the exact expression of $g_{0\leftarrow i}(x)$. Therefore, in practice we usually can only derive the approximation of $g_{0\leftarrow i}(x)$. To our convenience of continuing discussion, we first give the following lemmas. Their proofs are included in Appendix B.

Lemma 4.3.1. *For any $0 < i \leqslant n$,*

$$\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \approx g'_{0\leftarrow i}(T_{i,1}).$$

Lemma 4.3.2. *For any $0 < i \leqslant n$,*

$$\frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{(T_{i,2} - T_{i,1})(T_{i,1} - T_{i,0})} \approx g''_{0\leftarrow i}(T_{i,1}).$$

Lemma 4.3.3. *For any $0 < i \leqslant n$, $\dot{L}_{i-1} \approx g_{0\leftarrow i}(T_{i,1})$*

Currently most of time synchronization algorithms use a linear polynomial to approximate $g_{0\leftarrow i}(x)$. Our algorithm *2LTSP* uses a quardratic Taylor expansion $\tilde{g}_{0\leftarrow i}(x)$ to approximate $g_{0\leftarrow i}(x)$. In the following we will describe how to use the seven-tuple $(T_{0,3}, T_{0,4}, T_{0,5}, T_{i,0}, T_{i,1}, T_{i,2}, \dot{T}_{i-1})$ obtained by node i to derive $\tilde{g}_{0\leftarrow i}(x)$. From Taylor's theorem we know that $g_{0\leftarrow i}(x)$ is approximately equal to the first 3 terms of its Taylor series in a neighborhood of $T_{i,1}$, that is,

$$g_{0\leftarrow i}(x) \approx g_{0\leftarrow i}(T_{i,1}) + g'_{0\leftarrow i}(T_{i,1})(x - T_{i,1}) + \frac{1}{2} g''_{0\leftarrow i}(T_{i,1})(x - T_{i,1})^2. \tag{4.3.2}$$

From lemmas 4.3.1, 4.3.2, and 4.3.3 we know that by substituting $g_{0\leftarrow i}(T_{i,1}) g'_{0\leftarrow i}(T_{i,1})$ and $g''_{0\leftarrow i}(T_{i,1})$ in approximate Eqs. (4.3.2) with $\dot{L}_{i-1}$, $\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}$ and $\frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{(T_{i,2} - T_{i,1})(T_{i,1} - T_{i,0})}$ respectively, we can obtain the following approximation to $g_{0\leftarrow i}(x)$:

$$\tilde{g}_{0\leftarrow i}(x) = \dot{L}_{i-1} + \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \cdot (x - T_{i,1}) + \frac{1}{2} \cdot \frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{(T_{i,2} - T_{i,1})(T_{i,1} - T_{i,0})} \cdot (x - T_{i,1})^2. \tag{4.3.3}$$

It is clear from Eqs. (4.2.1) and (4.3.3) that for any $0 < i \leqslant n$, $\dot{L}_i = \tilde{g}_{0\leftarrow i}(T_{i,4})$. From lemmas 4.3.1, 4.3.2, and 4.3.3 and formula (4.3.3), we know that $\tilde{g}_{0\leftarrow i}(x)$ is a quadratic Taylor expansion approximation of $g_{0\leftarrow i}(x)$. Hence from Eq. (4.3.1) we know that $\tilde{g}_{0\leftarrow i}(C_i(t))$ is a quadratic Taylor expansion approximation of the global clock $C_0(t)$. This means that *2LTSP* approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as most existing schemes do.

From Taylor's theorem we know that $g_{0\leftarrow i}(x)$ is also approximately equal to the first 2 terms of its Taylor series in a neighborhood of $T_{i,4}$, that is,

$$g_{0\leftarrow i}(x) \approx g_{0\leftarrow i}(T_{i,4}) + g'_{0\leftarrow i}(T_{i,4})(x - T_{i,4}). \tag{4.3.4}$$

Since $T_{i,4} - T_{i,1}$ is shorter than the time needed for node i to forward a *synchronization packet*, $T_{i,4} - T_{i,1}$ is very short. Therefore from the continuity of $g'_{0\leftarrow i}(x)$ and $\tilde{g}'_{0\leftarrow i}(x)$ we know that $g'_{0\leftarrow i}(T_{i,4}) \approx g'_{0\leftarrow i}(T_{i,1})$ and $\tilde{g}'_{0\leftarrow i}(T_{i,4}) \approx \tilde{g}'_{0\leftarrow i}(T_{i,1})$. From Eq, (4.3.3) we know that $\tilde{g}'_{0\leftarrow i}(T_{i,1}) = \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}$. Because $\tilde{g}_{0\leftarrow i}(x) \approx g_{0\leftarrow i}(x)$, $g'_{0\leftarrow i}(T_{i,4}) \approx \tilde{g}'_{0\leftarrow i}(T_{i,4})$. Furthermore $g'_{0\leftarrow i}(T_{i,4}) \approx \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}$. Since $\dot{L}_i = \tilde{g}_{0\leftarrow i}(T_{i,4}) \approx g_{0\leftarrow i}(T_{i,4})$ and $g'_{0\leftarrow i}(T_{i,4}) \approx \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}$, by substituting $g_{0\leftarrow i}(T_{i,4})$ and $g'_{0\leftarrow i}(T_{i,4})$ in approximate Eq. (4.3.4) with $\dot{L}_i$ and $\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}}$ respectively, we can obtain the following approximation to $g_{0\leftarrow i}(x)$

$$\hat{g}_{0\leftarrow i}(x) = \dot{L}_i + \frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \cdot (x - T_{i,4}) \tag{4.3.5}$$

It is clear from Eqs. (4.2.2) and (4.3.5) that for any $0 < i \leqslant n$, $L_i(t) = \hat{g}_{0\leftarrow i}(C_i(t))$. As $\hat{g}_{0\leftarrow i}(x) \approx g_{0\leftarrow i}(x)$, $C_0(t) = g_{0\leftarrow i}(C_i(t))$ and $L_i(t) = \hat{g}_{0\leftarrow i}(C_i(t))$, so after the synchronization process has ended, node i computes its logical clock by imitating the global clock. In the next section, we will analyze the error of *2LTSP* and prove it has high accuracy.

5. Error analysis

The purpose of this section is to analyze the error of the proposed algorithm *2LTSP*. From Section 4, we know that *2LTSP* is a non-accumulative time synchronization algorithm, and approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as most existing schemes do. From Section 2, we know that any non-accumulative synchronization algorithm has no spatial accumulative effect, and any quadratic Taylor synchronization algorithm is much better, in term of synchronization accuracy, than linear synchronization algorithms. Therefore, we have proved that *2LTSP* has a high accuracy no matter how large the size of the network is. In this section, we will give the upper limit of the *2LTSP*'s synchronization error. Let $X(t)$ be a continuous random variable, and let $E(X)$ be the mean of X. If $X(t_0) = E(X)$, then we call t_0 a *mean point* of X. For sake of comprehension, Table 5.1 summarizes some variables and constants used from now on in this paper.

For practical applications, a WSN with radius of 100 hops is fairly large. For example, the physical radius of a network with radius 100 hops and consisting of Mica2 nodes can reach 15 km because the radio range of a Mica2 node is 152.5 m [2]. According to [2], sending a synchronization packet between two adjacent nodes needs at most 0.62 s. Therefore, in a WSN whose radius is no larger than 100 hops, the duration of a synchronization process is no larger than 62 s. Thus, we can suppose that T is no larger than 100 s.

Table 5.1
Definitions of variables and constants used in the paper.

Variable/ constant	Description
$C_i'(t)$	$C_i'(t)$, the derivative of $C_i(t)$, is the node i's *hardware clock frequency* at physical time t. $C_i'(t)-1$ is called the node i's *hardware clock drift* at physical time t
$C_i''(t)$	$C_i''(t)$, the second derivative of $C_i(t)$, is called the node i's *hardware clock drift variation* at physical time t
$C_i'''(t)$	$C_i'''(t)$, the third derivative of $C_i(t)$, is called the node i's *hardware clock quadratic drift variation* at physical time t
$t_{i,V}$	For any $1 \leqslant i \leqslant n$, $t_{i,V}$ (V = 0–2) are, respectively, the physical times in which node i has just received the *SFD*, the first flag and the second flag. For any $0 \leqslant i \leqslant n$, $t_{i,V}$ (V = 3–5) are, respectively, the physical times that node i has just sent the *SFD*, the first flag and the second flag. Hence $T_{i,V} = C_i(t_{i,V})$ ($i = 1\sim n$, $V = 0\sim 5$) and $T_{0,V} = C_0(t_{0,V})$ (V = 0, 4, 5).
ρ	the upper limit of hardware clock drifts, not larger than 30×10^{-6} s^{-1}
θ	the upper limit of hardware clock drift variations, not larger than 10^{-8} s^{-2}
	the upper limit of hardware clock quadratic drift variations, no larger than 10^{-10} s^{-3}
Δ	the time needed for the environmental temperature to change 1 °C
T	synchronization period
τ	synchronization duration of a synchronization process, that is, the elapsed time from the starting of the process, when the reference node sends a *synchronization packet*, to the ending of the process, when all nodes in the WSN have already received this packet.
ω	In any time interval with lengthω, each node's hardware clock frequency has mean points.

Since T is usually far less than T, we can suppose that $T \leqslant T \times 10^{-2}$ and $T^2 \leqslant T$. According to the literature [1,27], node i's *hardware clock drift* $C_i'(t)-1$ is always bounded, that is, there exists a positive number ρ_i such that $|C_i'(t)-1| \leqslant \rho_i, \forall t \geqslant 0$. The hardware device manufacturer can provide ρ_i, and normally $\rho_i \in$ [5,30] ppm (parts per million) [1,27]. Therefore it is reasonable to assume that hardware clocks satisfy the *bounded drift model* defined in [1,27]: there exists a positive number $\rho \leqslant 30 \times 10^{-6}$ s^{-1} such that for any node i, $|C_i''(t)-1| \leqslant \rho, \forall t \geqslant 0$. Since environmental conditions change gradually, it is reasonable to assume that hardware clocks satisfy the *bounded drift variation model* defined in [28]: there exists a positive number $\theta \leqslant 10^{-8}$ s^{-2} such that for any node i, $|C''i(t)| \leqslant \theta, \ \forall t \geqslant 0$. Since for any node i, $|C'i(t)-1| \leqslant 30 \times 10^{-6}$ and $|(C'i(t)-1)'| = |C''i(t)| \leqslant 10^{-8}$ in the usual condition, it seems reasonable to assume that $|C'''i(t)| = |(C'i(t)-1)''| \leqslant 10^{-10}$. Thus, it is also reasonable to assume that hardware clocks satisfy the following *bounded quadratic drift variation model*: there exists a positive number $\varphi \leqslant 10^{-10}$ s^{-3} such that for any node i, $|C'''i(t)| \leqslant \varphi$, $\forall t \geqslant 0$.

Since the time interval between two consecutive synchronization processes is generally quite short [8], environmental conditions such as humidity, pressure, battery voltage, and age of oscillator will not significantly change in the same synchronization period. Note that though the *2LTSP*'s synchronization period can reach 1300 s, this time interval is still rather short for changing these environmental conditions. Hence, the most prominent factor causing the hardware clock drift is temperature [1]. By the conclusion reported in [1], the change of the hardware clock frequency of a Mica2 node is no greater than one microsecond per second in the environment whose temperature difference is no greater than 5 °C. Therefore, it is reasonable to assume that hardware clocks satisfy the following *temperature-based drift model*: for each node i,

$$|C'i(t_1) - C'i(t_2)| \leqslant \frac{|t_1 - t_2|}{5\Delta} \times 10^{-6} \text{ s}^{-1} \quad \text{when } |t_1 - t_2| \leqslant \Delta \tag{4.3.6}$$

The authors of [1] do not indicate the time needed for the temperature to change 5 °C, but the change of environmental temperature is usually slow, for example, the temperature difference during noon to 3 P.M is commonly less than 5 °C. Because the change of the environmental temperature is usually slow, we can assume that $\tau + T \leqslant \Delta$ and $\Delta \geqslant 1300$ s. Suppose that $X(t)$ is a continuous random variable in time interval $[t_0, t_0 + \Delta]$. Let $e = E(X)$ In the usual condition, $P(X(t) \leqslant e) \approx P(X(t) \geqslant e)$. Therefore there exists a very long time sequence $t_1 < \cdots < t_m$ such that $X(t_1) < e, X(t_2) > e, X(t_3) < e, \cdots$. According to the zero-point theorem for continuous functions, there exist $\lambda_k \in (t_k, t_{k+1})(k = 1, \ldots, m-1)$*suchthat*$X(\lambda_k) = e$. Thus $X(t)$ usually has many mean points. Hence, it is reasonable to assume that hardware clocks satisfy the following *mean-based drift model*: there exists a positive number ω (not larger than 10 s) such that each node's hardware clock frequency has mean points in any time interval with length ω.

Now we summarize the hypotheses used throughout the paper:

(1). $\tau \leqslant 100$ s, $\tau \leqslant T \times 10^{-2}$, $\tau^2 \leqslant T$, $\omega \leqslant \Delta \times 10^{-2}$, $\tau + T \leqslant \Delta$ and $\Delta \geqslant 1300$ s.
(2). [bounded drift model] There exists a positive number $\rho \leqslant 30 \times 10^{-6}$ s^{-1} such that for any node i, $|C_i'(t)-1| \leqslant \rho$, $\forall t \geqslant 0$.
(3). [bounded drift variation model] There exists a positive number $\theta \leqslant 10^{-8}$ s^{-2} such that for any node i, $|C'''i(t)| \leqslant \theta, \forall t \geqslant 0$.
(4). [bounded quadratic drift variation model] There exists a positive number $\varphi \leqslant 10^{-10}$ s^{-3} such that for any node i, $|C_i'(t)| \leqslant \varphi, \forall t \geqslant 0$.
(5). [temperature-based drift model] For any node i,

$$|C_i'(t_1) - C'i(t_2)| \leqslant \frac{|t_1 - t_2|}{5\Delta} \times 10^{-6} \text{ s}^{-1} \quad \text{when } |t_1 - t_2| \leqslant \Delta.$$

(6). [mean-based drift model] There exists a positive number $\omega \leqslant 10$ s such that each node's hardware clock frequency has mean points in any time interval with length ω.

Lemma 5.1 gives the upper limit of error between the logical clocks of synchronized nodes and the hardware clock of the reference node (node 0). From the above hypotheses, Lagrange's mean value theorem, Taylor's formula and the intermediate value theorem for continuous function, we can prove Lemma 5.1. The proof detail of Lemma 5.1 is given in Appendix A. In this paper, we use marker "□" represents the end of the proofs.

Lemma 5.1. *For any* $1 \leqslant i \leqslant n$,

$$|L_i(t) - C_0(t)| \leqslant 8.1 \times \frac{\tau + T}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2 \text{ when } t \in [t_{i,4}, t_{i,4} + T)$$

From Lemma 5.1, *we can prove the following key result.*

Theorem 5.1. *For any* $1 \leqslant i \leqslant n$, $|L_i(t) - C_0(t)| \leqslant 9 \times 10^{-6}$ s when $t \in [t_{i,4}, t_{i,4} + T)$.

Table 6.1
Definitions of variables used in this section.

Variable	Description	Assumption
R	The radius of the network	
g_i	The changing period of the clock frequency of node i. Every g_i second, node i changes its clock frequency	The initial value of g_i is an integral between 18 s and 54 s; g_i follows the temperature-variation model
$\dot{r}_i$	The last clock frequency of node i, that is, the node i's frequency before g_i seconds	The initial value of $\dot{r}_i$ is 1
r_i	The current clock frequency of node i	$r_i \sim N\left(\dot{r}_i, \left(\frac{g_i}{25T} \times 10^{-6}\right)^2\right)$

Proof. From Lemma 5.1, we know that when $t \in [t_{i,4}, t_{i,4} + T)$,

$$|L_i(t) - C_0(t)| \leqslant 8.1 \times \frac{\tau + T}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2.$$

From the hypotheses (1) and (4), we know that $\tau \leqslant 100$, $\tau + T \leqslant \Delta$ and $\varphi \leqslant 10^{-10}$. Thus $|L_i(t) - C_0(t)| \leqslant 9 \times 10^{-6}$ s is clear. Hence, we have proved this theorem. □

From the above theorem, we know that when synchronization period is no larger than the time needed to change the environmental temperature in 1 °C, the error of *2LTSP* is within 9 ms no matter how large the size of the network is. This means that *2LTSP* has no spatial accumulative effect and its 9μs-period is larger than or equal to the time needed for the environmental temperature to change 1 °C. Because the change of the environmental temperature is usually slow, *2LTSP* is suitable for both large-scale networks and long-term running networks. Since we have assumed that the time Δ needed for the environmental temperature to change 1 °C is larger than or equal to 1300 s, it is clear that the 9μs-period of *2LTSP* is larger than or equal to 1300 s under conditions assumed in this paper. The following simulation results also corroborate to such conclusions.

6. Simulations

Our goal in this section is to analyze the performance of our algorithm *2LTSP* experimentally. We will analyze the synchronization error of *2LTSP* compared to three representative time synchronization algorithms, namely FTSP [2], RSP [4], and PulseSync [1]. It is important to recall that in this paper we only consider linear networks as shown in Fig. 4.1. All nodes need to report their current global times at the same physical time. It is also important to note that the hardware clock value of the reference node is the global time of the network, while global times of non-reference nodes are their logical clock values. To implement the task of reporting the current global times of the nodes at the same physical time, we added two new nodes, which do not participate in the synchronization, to the topology in Fig. 4.1. One of such node is a query node, while the other one is a sink node. The query node is responsible for broadcasting a query message periodically. Whenever a node participating in the synchronization receives a query message, it immediately sends a packet time-stamped by using its current global time to the sink node. The sink node is responsible for collecting the packets sent by the nodes participating in the synchronization and analyzing synchronization errors. The query node and the sink node can directly communicate with all nodes attending synchronization. The query intervals are set to 10 s. In each experiment, the system running duration is 36 h physical time. For sake of comprehension, Table 6.1 lists some variables used in this section.

In this section, we will assume that the hardware clocks satisfy the following temperature-variation model and discrete normal distribution model:

(1) *temperature-variation model*: for every node i, g_i is a conditional random variable described as follow. If $g_i = 6$, it increases by 1. If $6 < g_i < 18$, then the probability that it increases by 1 is 70%, and the probability that it decreases by 1 is 30%. If $18 \leqslant g_i \leqslant 54$, both the probability that it increases by 1 and the probability that it decreases by 1 are 50%. If $54 < g_i < 180$, then the probability that it increases by 1 is 30%, and the probability that it decreases by 1 is 70%. If $g_i = 180$, it decreases by 1;

(2) *discrete normal* distribution *model*: for every node i, r_i is a normal random variable with mean $\dot{r}_i$ and standard deviation $\frac{g_i}{25T} \times 10^{-6}$, that is, $r_i \sim N(\dot{r}_i, (\frac{g_i}{25T} \times 10^{-6})^2)$.

Since we have assumed that synchronization period is not larger than the time needed for the environmental temperature to change 1°C, $T \leqslant \Delta$. Let t_2 be current physical time. By the definitions of r_i and $\dot{r}_i$, $r_i = C_i'(t_2)$ and $\dot{r}_i = C_i'(t_2 - g_i)$. As we have assumed that the hardware clock of node i satisfies the temperature-based drift model described in Section 5, so $|r_i - \dot{r}_i| \leqslant \frac{g_i}{5\Delta} \times 10^{-6}$. Furthermore, $|r_i - \dot{r}_i| \leqslant \frac{g_i}{5T} \times 10^{-6}$. By the central limit theorem, many random phenomena in practice follow or approximately follow normal distributions. Therefore it is very reasonable assuming that the node i's frequency of hardware clock follows a normal distribution, that is, $r_i \sim N(\dot{r}_i, \sigma_i^2)$. Because $r_i \sim N(\dot{r}_i, \sigma_i^2)$ and $|r_i - \dot{r}_i| \leqslant \frac{g_i}{5T} \times 10^{-6}$, it easily follows from probability theory that $r_i \sim N(\dot{r}_i, (\frac{g_i}{25T} \times 10^{-6})^2)$. Thus, the rationality of the discrete normal distribution model follows. The rationality of the temperature-variation model can follows from the result in [1].

As far as we know, there are no public available time synchronization simulators for WSNs that satisfy the temperature-variation model or the discrete normal distribution model at present. Therefore, to analyze the performance of *2LTSP* experimentally, we developed an event driven simulator at the packet level in C++ language. Our simulator is an extension of the simulator presented in [9], and satisfies both the temperature-variation model and the discrete normal distribution model. The simulator described in [9] is based on the time character of Mica2 platform [29], and it decomposes packet delay into 6 delay components, modeling them separately. In such simulator, the frequency of crystal oscillator is modeled as Gaussian so that clock drift can be setup.

In the next subsections, we analyze the performance of 2LTSP through a set of experiments. In all experiments, the radio range is 152.5 m.

6.1. Impact of synchronization period on accuracy

The purpose of the first conducted experiment is to analyze the impact of changing synchronization period on the synchronization accuracy. The network in this experiment consists of two nodes. In this experiment, we observe 28 one-hop synchronization processes in which network radius R always takes the value of 1 hop. The time synchronization algorithms used are *2LTSP*, FTSP, RSP and PulseSync. The synchronization period T takes values of 100, 300, 500, 700, 900, 1100 and 1300 s. We use $X_{i,j}(i = 1, 2, 3, 4; j = 1, \cdots, 7)$ to represent the average absolute errors in microsecond

Table 6.2
90% Confidence intervals of average absolute errors in microsecond for experiment 1.

T algorithm	100 s	300 s	500 s	700 s	900 s	1100 s	1300 s
2LTSP	[0.59, 0.62]	[1.13, 1.17]	[1.52, 1.56]	[1.86, 1.90]	[2.25, 2.31]	[2.59, 2.67]	[2.93, 3.02]
FTSP	[10.12, 11.27]	[18.19, 20.95]	[23.23, 26.19]	[27.58, 31.28]	[28.78, 32.49]	[32.75, 37.81]	[36.77, 41.63]
RSP	[5.98, 6.72]	[10.51, 12.06]	[14.34, 15.94]	[16.42, 18.36]	[17.93, 20.43]	[20.92, 23.66]	[23.95, 27.09]
PulseSync	[8.93, 10.24]	[15.74, 18.80]	[21.22, 24.49]	[24.85, 29.00]	[25.25, 29.17]	[29.32, 34.94]	[32.76, 38.01]

of these synchronization processes, where i corresponds to one of the considered algorithms (*2LTSP*, FTSP, RSP and PulseSync), and j corresponds to one of the synchronization periods (100, 300, 500, 700, 900, 1100 and 1300 s). Since the frequency of the hardware clock of every node is random, each $X_{i,j}$ is a random variable. We use $E(X_{i,j})$ to denote the mean of $X_{i,j}$, and use $I(X_{i,j})$ to denote the 90% confidence intervals of $X_{i,j}$. We repeated this experiment 121 times. We use $x_{i,j}^{(k)}(k = 1, \cdots, 121)$ to represent the kth sample value of $X_{i,j}$. By probability and statistics, we know that

$$E(X_{i,j}) = \frac{1}{121} \times \sum_{k=1}^{121} x_{i,j}^{(k)} \text{ and } I(X_{i,j}) = [E(X_{i,j}) - y_{i,j}, E(X_{i,j}) + y_{i,j}]$$

where $y_{i,j} = \frac{1.658}{\sqrt{121}} \times \sqrt{\frac{1}{120} \times \left(\sum_{k=1}^{121} (x_{i,j}^{(k)})^2 - 121 \times E(X_{i,j})^2\right)}$. The 90% confidence intervals (namely, confidence intervals of 90%) of average absolute errors for each algorithm are drawn in Table 6.2.

The elements of the '*2LTSP*' row are $I(X_{1,j}), j = 1, \cdots, 7$, and the other rows have the similar meaning for the remainder algorithms. We draw means of average absolute errors in microsecond for each algorithm in Fig. 6.1. The points on the '2LTSP' curve are, in turn, $(100, E(X_{1,1}))$, $(300, E(X_{1,2}))$, $(500, E(X_{1,3}))$, $(700, E(X_{1,4}))$, $(900, E(X_{1,5}))$, $(1100, E(X_{1,6}))$ and $(1300, E(X_{1,7}))$, and the other curves have the similar meaning.

Results from Table 6.2 and Fig. 6.1 show that the *2LTSP* achieved synchronization error much smaller than the other assessed algorithms for all tested synchronization periods. For synchronization periods larger than 100 s, the achieved synchronization errors are higher than 5 μs for FTSP, RSP and PulseSync algorithms, while the error for *2LTSP* remains low even when the synchronization period is increased up to 1300 s (varying from 0.6 μs to a maximum of 3.0 μs). Such results demonstrate that *2LTSP* is suitable for WSN applications with a long running lifetime. We can also see from Fig. 6.1 that the accuracy of *2LTSP* is at least 11 times higher than that of PulseSync; the accuracy of *2LTSP* is at least 13 times higher than that of FTSP; and the accuracy of *2LTSP* is at least 8 times higher than that of RSP. In addition, we can conclude from Table 6.2 and Fig. 6.1 that the 9 μs-periods of both FTSP and PulseSync are less than 100 s; the 9 μs-period of RSP is less than 300 s; and the 9 μs-period of *2LTSP* is larger than 1300 s. Regarding the communication costs of the evaluated algorithms, FTSP, PulseSync and *2LTSP* flood one packet to the entire network in a synchronization process, while RSP floods two packets. Thus, the communication costs of both FTSP and PulseSync are larger than 36 packets/hour; the communication cost of RSP is larger than 24 packets/hour; and the communication cost of *2LTSP* is less than 3 packets/hour. Therefore, the communication costs of both FTSP and PulseSync are at least 12 times than that of *2LTSP*; and the communication cost of RSP is at least 8 times than that of *2LTSP*.

6.2. Accuracy comparison in a two-hop network

The purpose of the second experiment is to compare synchronization accuracy in a two-hop network. The network in this experiment consists of three nodes. In this experiment, we observe 28 two-hop synchronization processes in which network radius R always takes value of 2 hops. Again, the time synchronization algorithms used in the comparison are *2LTSP*, FTSP, RSP and PulseSync. The synchronization period T is always assigned a value of 700 s. We repeated this experiment 121 times. The mean and 90% confidence interval of average absolute errors in microsecond for each algorithm are drawn in Table 6.3.

The first column and the second column of the '*2LTSP*' row are $E(X_{1,1})$ and $I(X_{1,1})$ respectively, and the other rows have similar

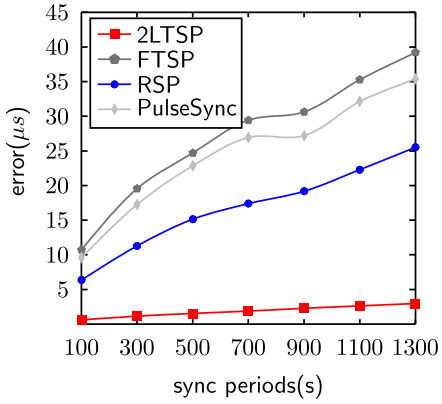

Fig. 6.1. Impact of synchronization period on accuracy.

Table 6.3
Means and 90% confidence intervals of average absolute errors in microseconds for experiment 2.

Algorithm	Mean of average absolute errors	90% Confidence interval of average absolute errors
2LTSP	1.86	[1.85, 1.88]
FTSP	116.10	[106.54, 125.66]
RSP	207.93	[186.87, 228.99]
PulseSync	25.87	[24.12, 27.63]

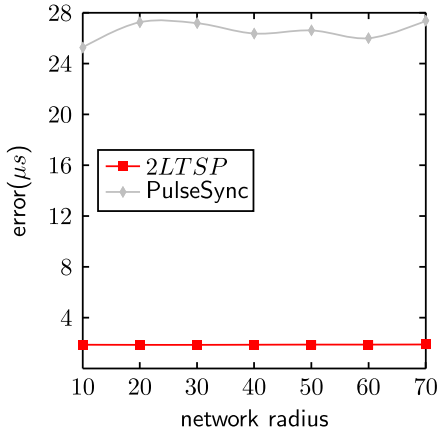

Fig. 6.2. The impact of network radius on accuracy.

Table 6.4
90% Confidence intervals of average absolute errors for experiment 3.

R algorithm	10	20	30	40	50	60	70
2LTSP	[1.85, 1.88]	[1.84, 1.87]	[1.84, 1.87]	[1.85, 1.88]	[1.87, 1.89]	[1.87, 1.90]	[1.88, 1.90]
PulseSync	[24.26, 26.29]	[25.98, 28.54]	[26.06, 28.30]	[25.33, 27.41]	[25.66, 27.55]	[24.98, 27.02]	[26.09, 28.66]

meaning. From Fig. 6.1 and Table 6.3, we can observe that the errors of FTSP and RSP in a one-hop network are much larger than their errors in a two-hop network. Therefore, they have very strong spatial accumulative effect as the size of the network increases. Furthermore, they are not suitable for large-scale networks.

6.3. Impact of network radius on accuracy

The purpose of experiment 3 is to analyze the impact of changing the network radius on the synchronization accuracy. In this experiment, we observe 14 multi-hop synchronization processes in which synchronization period T always takes value of 700 s. The evaluated time synchronization algorithms are *2LTSP* and PulseSync, and the network radius R takes values of 10, 20, 30, 40, 50, 60 and 70 hops. We repeated this experiment 121 times. The 90% confidence intervals of average absolute errors in microsecond for each algorithm are drawn in Table 6.4.

The rows in Table 6.4 have the same meaning as the corresponding rows in Table 6.2. We draw means of average absolute errors in microsecond for each algorithm in Fig. 6.2. The curves in Fig. 6.2 have the same meaning as the corresponding curves in Fig. 6.1.

From Fig. 6.2 we can conclude that synchronization error of both *2LTSP* and PulseSync does not have spatial accumulative effect. From Fig. 6.2 we can also conclude that the accuracy of *2LTSP* is at least 13 times higher than that of PulseSync. This is because *2LTSP* approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as PulseSync does.

7. *2LSTP* implementation on arduino sensor nodes

In order to demonstrate the feasibility of executing the proposed algorithm on real sensor nodes, we built a prototype of *2LTSP* in the Arduino[1] WSN platform. The main goal of the implementation was to show that current WSN hardware, although limited in resources (as memory and processing), is able to execute all the steps required for *2LSTP* operation, including the following tasks, without preventing the packet correct transmission: catch an interrupt, read the hardware clock and update the packet under transmission in time.

We implemented a WSN composed of two sensor nodes with a logical topology as shown in Fig. 4.1. Each node in the network is equipped with a temperature sensing unit, an Arduino Uno microcontroller, a XBee module and a lithium battery as energy source. The XBee Module is an IEEE 802.15.4 compliant radio transceiver.

We implemented a *synchronization packet* as shown in Fig. 4.1 and the steps required to a sender node transfers the *synchronization packet* to a receiver node (refer to Section 2.1), obtaining the three pairs of time stamps needed for *2LTSP* operation. For any given *synchronization packet*, the first node that sends it is called its *originator*, while the other nodes that send it are called its *forwarders*. A different code image was developed for the *originator* node and the *forwarder* node. Each image was then deployed in one of the implemented nodes and then we run the *2LTSP* synchronization process. From the 32,256 bytes of total available memory in the WSN nodes, the image binary size for the *originator* node was 3,538 bytes, while the image size for the *forwarder* node was 6,808 bytes. After performing a set of 30 rounds of execution of the *2LTSP* synchronization process in the real nodes, we observed that the correct packet transmission was not compromised. The implemented code for the Arduino platform is available at http://labnet.nce.ufrj.br/cia2/2LTSP.

[1] http://arduino.cc/en/Main/ArduinoBoardUno/.

Table A.1
Variables and constants used in this appendix.

Variable/ constant	Description
$\Delta_i(h,t)$	$\Delta_i(h,t) = C_i(t+h) - 2C_i(t) + C_i(t-h)$ is the second-order difference of $C_i(t)$ with step length h
α_0	$T_{0,5} - T_{0,4} = C_0'(\alpha_0)(t_{0,5} - t_{0,4})$, where $\alpha_0 \in (t_{0,4}, t_{0,5})$
β_0	$T_{0,5} - 2T_{0,4} + T_{0,3} = C_0''(\beta_0)(t_{0,5} - t_{0,4})^2$, where $\beta_0 \in (t_{0,3}, t_{0,5})$
$\zeta_i(t)$	$C_0(t) = C_0(t_{i,1}) + C_0'(\zeta_i(t))(t - t_{i,1})$, where $\zeta_i(t) \in (t_{i,1}, t)$
$\tilde{\zeta}_i(t)$	$C_0(t) = C_0(t_{i,4}) + C_0'(\tilde{\zeta}_i(t))(t - t_{i,4})$, where $\tilde{\zeta}_i(t) \in (t_{i,4}, t)$
$\xi_i(t)$	$C_0(t) = C_0(t_{i,1}) + C_0'(t_{i,1})(t - t_{i,1}) + \frac{1}{2}C_0''(\xi_i(t))(t - t_{i,1})^2$, where $\xi_i(t) \in (t_{i,1}, t)$
$\eta_i(t)$	$C_i(t) - C_i(t_{i,1}) = C_i'(\eta_i(t))(t - t_{i,1})$, where $\eta_i(t)$ is between $t_{i,1}$ and t
$\tilde{\eta}_i(t)$	$C_i(t) - C_i(t_{i,4}) = C_i'(\tilde{\eta}_i(t))(t - t_{i,4})$, where $\tilde{\eta}_i(t)$ is between $t_{i,4}$ and t

8. Conclusion

This paper proposed a time synchronization algorithm, named *2LTSP*, for wireless sensor networks. Theoretical analysis shows that *2LTSP* is a non-accumulative time synchronization algorithm and it approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial, as most existing schemes do. Therefore, *2LTSP* has no spatial accumulative effect as the size of the network increases and is highly accurate. Theoretical analysis and simulation results show that when synchronization period is not larger than the time needed for the environmental temperature to change 1 °C, the error of *2LTSP* is within 9 ms no matter how large the size of the network is. Because the change of the environmental temperature is usually slow, *2LTSP* is suitable for both large-scale networks and long-term running networks. A prototype of *2LTSP* was implemented in the Arduino WSN platform in order to demonstrate the feasibility of its execution in real, resource constrained sensor nodes.

Acknowledgments

We would like to express our gratitude to José R. Silva and the team from UBICOMP lab at Federal University of Rio de Janeiro for helping with the implementation. This work was partially supported by Brazilian Funding Agencies FAPERJ and CNPq (under grants 470586/2011-7 and 311363/2011-3 for Flavia Delicato and 310661/2012-9 for Paulo Pires). Professor Zomaya's work was supported by an Australian Research Council Grant number DP130104591.

Appendix A

The purpose of this appendix is to prove Lemma 5.1. Table A.1 summarizes the variables and constants used in the appendix.

To our convenience, we first give the following lemmas A.1, A.2, A.3, A.4, and A.5.

Lemma A.1. The following results are true:

(1) For each $0 < i \leqslant n$, $t_{i,2} - t_{i,1} = t_{i,1} - t_{i,0}$.

(2) For each $0 \leqslant i \leqslant n$, $t_{i,5} - t_{i,4} = t_{i,4} - t_{i,3}$.

(3) For each $0 < i \leqslant n$, $t_{i-1,5} - t_{i-1,4} = t_{i,2} - t_{i,1}$ and $t_{i-1,4} - t_{i-1,3} = t_{i,1} - t_{i,0}$.

(4) For each $0 < i \leqslant n$, $t_{i-1,3} = t_{i,0}$, $t_{i-1,4} = t_{i,1}$ and $t_{i-1,5} = t_{i,2}$.

(5) For each $0 < i \leqslant n$, $t_{i,4} - t_{i,3} = t_{i,2} - t_{i,1}$.

Proof. For each $0 < i \leqslant n$, as both $t_{i,2} - t_{i,1}$ and $t_{i,1} - t_{i,0}$ are the physical time needed for node i to receive 12 bytes, so $t_{i,2} - t_{i,1} = t_{i,1} - t_{i,0}$ follows from [2]. For each $0 \leqslant i \leqslant n$, since both $t_{i,5} - t_{i,4}$ and $t_{i,4} - t_{i,3}$ are the physical time needed for node i to send 12 bytes, $t_{i,5} - t_{i,4} = t_{i,4} - t_{i,3}$ follows from [2]. For each $0 < i \leqslant n$, because node $i-1$ has just sent the *SFD* and the first flag at physical times $t_{i-1,3}$ and $t_{i-1,4}$ respectively, and node i has just received the *SFD* and the first flag from node $i-1$ at physical times $t_{i,0}$ and $t_{i,1}$ respectively, $t_{i-1,4} - t_{i-1,3} = t_{i,1} - t_{i,0}$. In the same way, we can prove that $t_{i-1,5} - t_{i-1,4} = t_{i,2} - t_{i,1}$. For each $0 < i \leqslant n$, since node $i-1$ has just sent the *SFD* at physical time $t_{i-1,3}$, and node i has just received the *SFD* from node $i-1$ at physical time $t_{i,0}$, $t_{i-1,3} - t_{i,0}$ is less than one microsecond by the result in [2]. Because the resolution of the hardware clock is not greater than one microsecond in the WSN, $t_{i-1,3} = t_{i,0}$. Similarly, we can prove that $t_{i-1,4} = t_{i,1}$ and $t_{i-1,5} = t_{i,2}$. For each $0 < i \leqslant n$, since $t_{i,4} - t_{i,3}$ is the physical time needed for node i to send 12 bytes, and $t_{i,2} - t_{i,1}$ is the physical time needed for node i to receive 12 bytes, $t_{i,4} - t_{i,3} = t_{i,2} - t_{i,1}$ follows from [2]. Hence we have proved this lemma. □

Lemma A.2. *For any $0 \leqslant i \leqslant n$, there exists $t - h < \xi < t + h$ such that $\Delta_i(h,t) = C_i'(\xi)h^2$.*

Proof. By the Taylor's formula, there exist $t < \xi_1 < t + h$ and $t - h < \xi_2 < t$ such that

$$C_i(t+h) = C_i(t) + C_i'(t)h + \frac{1}{2}C_i''(\xi_1)h^2 \text{ and } C_i(t-h) = C_i(t) - C_i'(t)h + \frac{1}{2}C_i''(\xi_2)h^2.$$

Thus $\Delta_i(h,t) = C_i(t+h) - 2C_i(t) + C_i(t-h) = \frac{1}{2}(C_i''(\xi_1) + C_i''(\xi_2))h^2$. By the intermediate value theorem for continuous function, we know that there exists $\xi_2 \leqslant \xi \leqslant \xi_1$ such that $C_k''(\xi) = \frac{1}{2}(C_k''(\xi_1) + C_k''(\xi_2))$. Therefore $\Delta_i(h,t) = C_i''(\xi)h^2$. Hence we have proved this lemma. □

Lemma A.3. *The following conclusions hold:(1) There exists $\alpha_0 \in (t_{0,4}, t_{0,5})$ such that $T_{0,5} - T_{0,4} = C_0'(\alpha_0)(t_{0,5} - t_{0,4})$;(2) There exists $\beta_0 \in (t_{0,3}, t_{0,5})$ such that $T_{0,5} - 2T_{0,4} + T_{0,3} = C_0''(\beta_0)(t_{0,5} - t_{0,4})^2$;(3) For any $1 \leqslant i \leqslant n$, there exists $\zeta_i(t) \in (t_{i,1}, t)$ such that*

$$C_0(t) = C_0(t_{i,1}) + C_0'(\zeta_i(t))(t - t_{i,1});$$

(4) For any $1 \leqslant i \leqslant n$, there exists $\xi_i(t) \in (t_{i,1}, t)$ such that

$$C_0(t) = C_0(t_{i,1}) + C_0'(t_{i,1})(t - t_{i,1}) + \frac{1}{2}C_0''(\xi_i(t))(t - t_{i,1})^2;$$

(5) For any $1 \leqslant i \leqslant n$, there exists $\eta_i(t)$ between $t_{i,1}$ and t such that

$$C_i(t) - C_i(t_{i,1}) = C_i'(\eta_i(t))(t - t_{i,1});$$

(6) For any $1 \leqslant i \leqslant n$, there exists $\tilde{\zeta}_i(t) \in (t_{i,4}, t)$ such that

$$C_0(t) = C_0(t_{i,4}) + C_0'(\tilde{\zeta}_i(t))(t - t_{i,4});$$

(7) For any $1 \leqslant i \leqslant n$, there exists $\tilde{\eta}_i(t)$ between $t_{i,4}$ and t such that

$$C_i(t) - C_i(t_{i,4}) = C_i'(\tilde{\eta}_i(t))(t - t_{i,4}).$$

Proof. From the definitions of $T_{0,v}$ and $t_{0,v}$, we know that $T_{0,v} = C_0(t_{0,v})(v = 4, 5)$. By Lagrange's mean value theorem, there exists $\alpha_0 \in (t_{0,4}, t_{0,5})$ such that

$$C_0(t_{0,5}) - C_0(t_{0,4}) = C_0'(\alpha_0)(t_{0,5} - t_{0,4}).$$

Hence we have proved conclusion (1). From Lemma A.1, we know that $t_{0,5} - t_{0,4} = t_{0,4} - t_{0,3}$. Hence it is very easy to check the following equation:

$$C_0(t_{0,5}) - 2C_0(t_{0,4}) + C_0(t_{0,3}) = \Delta_0(t_{0,5} - t_{0,4}, t_{0,4}).$$

From Lemma A.2, we know that there exists $t_{0,3} < \beta_0 < t_{0,5}$ such that $\Delta_0(t_{0,5} - t_{0,4}, t_{0,4}) = C_0''(\beta_0)(t_{0,5} - t_{0,4})^2$.

From the definitions of $T_{0,v}$ and $t_{0,v}$, we know that $T_{0,v} = C_0(t_{0,v})(v = 3, 4, 5)$. Hence we have proved conclusion (2). For any $1 \leqslant i \leqslant n$, by Taylor's formula, there exists $\zeta_i(t) \in (t_{i,1}, t)$ such that

$$C_0(t) = C_0(t_{i,1}) + C_0'(\zeta_i(t))(t - t_{i,1}).$$

Hence we have proved conclusion (3). For any $1 \leqslant i \leqslant n$, by Taylor's formula, there exists $\xi_i(t) \in (t_{i,1}, t)$ such that

$$C_0(t) = C_0(t_{i,1}) + C_0'(t_{i,1})(t - t_{i,1}) + \frac{1}{2}C_0''(\xi_i(t))(t - t_{i,1})^2.$$

Hence we have proved conclusion (4). For any $1 \leqslant i \leqslant n$, by Lagrange's mean value theorem, there exists $\eta_i(t)$ between $t_{i,1}$ and t such that

$$C_i(t) - C_i(t_{i,1}) = C_i'(\eta_i(t))(t - t_{i,1}).$$

Hence we have proved conclusion (5). Conclusion (6) can be proved similarly as conclusions (3) and (7) can be proved similarly as conclusion (5). Hence we have proved this lemma. □

Lemma A.4. *For any $0 \leqslant i \leqslant n$, $|C_i'(t_1) - C_i'(t_2)| \leqslant \frac{4\times10^{-6}}{\Delta}$, $\forall t_1, t_2 \geqslant 0$.*

Proof. Suppose that $t_1 \geqslant 0$ and $t_2 \geqslant 0$ are two physical times. For every $0 \leqslant i \leqslant n$, by the hypothesis (6) in Section 5, there exists t_3 such that $C_i'(t_3) = E(C_i'(t))$ and $|t_3 - t_1| \leqslant \omega$. From the hypothesis (1) in Section 5, we know that $\omega < \Delta$. Therefore $|t_1 - t_3| < \Delta$. By the hypothesis (5) in Section 5,

$$|C_i'(t_1) - E(C_i'(t))| = |C_i'(t_1) - C_i'(t_3)| \leqslant \frac{|t_1 - t_3|}{5\Delta} \times 10^{-6} \leqslant \frac{\omega \times 10^{-6}}{5\Delta}.$$

Since $\omega \leqslant 10$, $|C_i'(t_1) - E(C_i'(t))| \leqslant \frac{2\times10^{-6}}{\Delta}$. Similarly, $|C_i'(t_2) - E(C_i'(t))| \leqslant \frac{2\times10^{-6}}{\Delta}$. Thus

$$|C_i'(t_1) - C_i'(t_2)| \leqslant |C_i'(t_1) - E(C_i'(t))| + |C_i'(t_2) - E(C_i'(t))| \leqslant \frac{4 \times 10^{-6}}{\Delta}.$$

Hereto we have proved this lemma. □

Lemma A.5. *For any $0 \leqslant i \leqslant n$, $|\dot{L}_i - C_0(t_{i,4})| \leqslant 8.1 \times \frac{\tau}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2$.*

Proof. By the definitions of $\dot{L}_0, T_{0,4}$ and $t_{0,4}$, we know that $\dot{L}_0 = T_{0,4}$ and $T_{0,4} = C_0(t_{0,4})$. Hence, when i is zero, this lemma is valid. Now we assume that $i > 0$. By the conclusion (4) in Lemma A.3, there exists $\xi_i(t_{i,4}) \in (t_{i,1}, t_{i,4})$ such that

$$C_0(t_{i,4}) = C_0(t_{i,1}) + C_0'(t_{i,1})(t_{i,4} - t_{i,1}) + \frac{1}{2}C_0''(\xi_i(t_{i,4}))(t_{i,4} - t_{i,1})^2.$$

From the conclusion (4) in Lemma A.1, we know that $t_{i,1} = t_{i-1,4}$. Therefore,

$$C_0(t_{i,4}) = C_0(t_{i-1,4}) + C_0'(t_{i,1})(t_{i,4} - t_{i,1}) + \frac{1}{2}C_0''(\xi_i(t_{i,4}))(t_{i,4} - t_{i,1})^2. \quad (A.1)$$

From the definitions of $T_{i,v}$ and $t_{i,v}$, we know that $T_{i,v} = C_i(t_{i,v})(v = 0, 1, 2, 4)$. Hence from Eq. (4.2.1), we know that

$$\dot{L}_i = \dot{L}_{i-1} + \frac{T_{0,5} - T_{0,4}}{C_i(t_{i,2}) - C_i(t_{i,1})} \cdot [C_i(t_{i,4}) - C_i(t_{i,1})] + \frac{1}{2} \times \frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{[C_i(t_{i,2}) - C_i(t_{i,1})] \cdot [C_i(t_{i,1}) - C_i(t_{i,0})]} \cdot [C_i(t_{i,4}) - C_i(t_{i,1})]^2.$$

By the conclusions (1) and (2) in Lemma A.3, there exist $\alpha_0 \in (t_{0,4}, t_{0,5})$ and $\beta_0 \in (t_{0,3}, t_{0,5})$ such that $T_{0,5} - T_{0,4} = C_0'(\alpha_0)(t_{0,5} - t_{0,4})$ and $T_{0,5} - 2T_{0,4} + T_{0,3} = C_0''(\beta_0)(t_{0,5} - t_{0,4})^2$. Thus,

$$\dot{L}_i = \dot{L}_{i-1} + \frac{C_0'(\alpha_0)(t_{0,5} - t_{0,4})}{C_i(t_{i,2}) - C_i(t_{i,1})} \cdot [C_i(t_{i,4}) - C_i(t_{i,1})] + \frac{1}{2} \times \frac{C_0''(\beta_0)(t_{0,5} - t_{0,4})^2}{[C_i(t_{i,2}) - C_i(t_{i,1})] \cdot [C_i(t_{i,1}) - C_i(t_{i,0})]} \cdot [C_i(t_{i,4}) - C_i(t_{i,1})]^2.$$

By the conclusion (5) in Lemma A.3, there exist $\eta_i(t_{i,0}) \in (t_{i,0}, t_{i,1})$, $\eta_i(t_{i,2}) \in (t_{i,1}, t_{i,2})$ and $\eta_i(t_{i,4}) \in (t_{i,1}, t_{i,4})$ such that $C_i(t_{i,v}) - C_i(t_{i,1}) = C_i'(\eta_i(t_{i,v}))(t_{i,v} - t_{i,1})(v = 0, 2, 4)$. Therefore,

$$\dot{L}_i = \dot{L}_{i-1} + \frac{C_0'(\alpha_0)(t_{0,5} - t_{0,4})}{C_i'(\eta_i(t_{i,2}))(t_{i,2} - t_{i,1})} C_i'(\eta_i(t_{i,4}))(t_{i,4} - t_{i,1}) + \frac{1}{2} \times \frac{C_0''(\beta_0)(t_{0,5} - t_{0,4})^2}{C_i'(\eta_i(t_{i,2}))C_i'(\eta_i(t_{i,0}))(t_{i,2} - t_{i,1})(t_{i,1} - t_{i,0})} C_i'(\eta_i(t_{i,4}))^2(t_{i,4} - t_{i,1})^2.$$

By the conclusion (1) of Lemma A.1, $t_{i,2} - t_{i,1} = t_{i,1} - t_{i,0}$. From Lemma A.1, $t_{i,2} - t_{i,1} = t_{0,5} - t_{0,4}$ can be easily proved. Hence we have prove that

$$\dot{L}_i = \dot{L}_{i-1} + \frac{C_i'(\eta_i(t_{i,4}))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0)(t_{i,4} - t_{i,1}) + \frac{1}{2} \times \frac{C_i'(\eta_i(t_{i,4}))^2}{C_i'(\eta_i(t_{i,2}))C_i'(\eta_i(t_{i,0}))} C_0''(\beta_0)(t_{i,4} - t_{i,1})^2.$$

From the above equation and Eq. (A.1), we can obtain the following recursive equation:

$$\dot{L}_i - C_0(t_{i,4}) = [\dot{L}_{i-1} - C_0(t_{i-1,4})] + \left(\frac{C_i'(\eta_i(t_{i,4}))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) - C_0'(t_{i,1})\right)(t_{i,4} - t_{i,1}) + \frac{1}{2}\left(\frac{C_i'(\eta_i(t_{i,4}))^2}{C_i'(\eta_i(t_{i,2}))C_i'(\eta_i(t_{i,0}))} C_0''(\beta_0) - C_0''(\xi_i(t_{i,4}))\right)(t_{i,4} - t_{i,1})^2$$

Since we have proved that $\dot{L}_0 = C_0(t_{0,4})$, from the above recursive equation we can easily prove that

$$\dot{L}_i - C_0(t_{i,4}) = \sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1}) + \frac{1}{2}\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))} C_0''(\beta_0) - C_0''(\xi_j(t_{j,4})) \times (t_{j,4} - t_{j,1})\right)^2. \quad (A.2)$$

We first prove the following inequality:

$$\left|\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1})\right| \leqslant \frac{8}{1-\rho} \times \frac{\tau}{\Delta} \times 10^{-6}.$$

From the conclusion (4) in Lemma A.1, we know that $t_{j,1} = t_{j-1,4}$. Therefore,

$$\sum_{j=1}^{i}(t_{j,4} - t_{j,1}) = \sum_{j=1}^{i}(t_{j,4} - t_{j-1,4}) = t_{i,4} - t_{0,4}. \quad (A.3)$$

Let $a = \min\{C_0'(t_{j,1}) | j = 1 \sim i\}$ and $b = \max\{C_0'(t_{j,1}) | j = 1 \sim i\}$. From Eq. (A.3), we can see that $a \leqslant \frac{1}{t_{i,4} - t_{0,4}} \cdot \sum_{j=1}^{i} C_0'(t_{j,1})(t_{j,4} - t_{j,1}) \leqslant b$. Hence, by the intermediate value theorem for continuous function, we know that there exists λ_i such that

$$\frac{1}{t_{i,4} - t_{0,4}} \sum_{j=1}^{i} C_0'(t_{j,1})(t_{j,4} - t_{j,1}) = C_0'(\lambda_i). \quad (A.4)$$

From Eqs. (A.3) and (A.4), we know that

$$\begin{aligned}
&\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1}) \\
&= \sum_{j=1}^{i}\left(\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} - 1\right) C_0'(\alpha_0) + C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1}) \\
&= C_0'(\alpha_0)\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} - 1\right)(t_{j,4} - t_{j,1}) + \sum_{j=1}^{i}[C_0'(\alpha_0) - C_0'(t_{j,1})](t_{j,4} - t_{j,1}) \\
&= C_0'(\alpha_0)\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} - 1\right)(t_{j,4} - t_{j,1}) + C_0'(\alpha_0)\sum_{j=1}^{i}(t_{j,4} - t_{j,1}) - \sum_{j=1}^{i} C_0'(t_{j,1})(t_{j,4} - t_{j,1}) \\
&= C_0'(\alpha_0)\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} - 1\right)(t_{j,4} - t_{j,1}) + [C_0'(\alpha_0) - C_0'(\lambda_i)](t_{i,4} - t_{0,4}) \\
&= C_0'(\alpha_0)\sum_{j=1}^{i}\frac{C_j'(\eta_j(t_{j,4})) - C_j'(\eta_j(t_{j,2}))}{C_j'(\eta_j(t_{j,2}))}(t_{j,4} - t_{j,1}) + [C_0'(\alpha_0) - C_0'(\lambda_i)](t_{i,4} - t_{0,4})
\end{aligned} \quad (A.5)$$

From the hypothesis (2) in Section 5, we can easily prove that $|C_0'(\alpha_0)| \leqslant 1 + \rho$ and $|C_j'(\eta_j(t_{j,2}))| \geqslant 1 - \rho$. Thus, from Eq. (A.5) we know that

$$\begin{aligned}
&\left|\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1})\right| \\
&\leqslant \frac{1+\rho}{1-\rho}\sum_{j=1}^{i}|C_j'(\eta_j(t_{j,4})) - C_j'(\eta_j(t_{j,2}))|(t_{j,4} - t_{j,1}) + |C_0'(\alpha_0) - C_0'(\lambda_i)| \\
&|(t_{i,4} - t_{0,4}).
\end{aligned} \quad (A.6)$$

From Lemma A.4, Eq. (A.3) and inequality (A.6) we know that

$$\begin{aligned}
&\left|\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))}{C_j'(\eta_j(t_{j,2}))} C_0'(\alpha_0) - C_0'(t_{j,1})\right)(t_{j,4} - t_{j,1})\right| \\
&\leqslant \frac{1+\rho}{1-\rho}\sum_{j=1}^{i}\left(\frac{4}{\Delta} \cdot 10^{-6} \cdot (t_{j,4} - t_{j,1})\right) + \frac{4}{\Delta} \cdot 10^{-6} \cdot (t_{i,4} - t_{0,4}) = \\
&\frac{2}{1-\rho} \cdot \frac{4}{\Delta} \cdot 10^{-6} \cdot (t_{i,4} - t_{0,4}) \leqslant \frac{8}{1-\rho} \times \frac{\tau}{\Delta} \times 10^{-6}
\end{aligned} \quad (A.7)$$

Next, we prove the following inequality:

$$\left|\frac{1}{2}\sum_{j=1}^{i}\left(\frac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}C_0''(\beta_0)-C_0'(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\right| \leqslant 2.48\theta\frac{1+\rho}{(1-\rho)^2}\frac{\tau}{\Delta}10^{-6}+0.31\varphi\tau^2.$$

Let $a=\min\{C_0''(\xi_j(t_{j,4}))|j=1\sim i\}$ and $b=\max\{C_0''(\xi_j(t_{j,4}))|j=1\sim i\}$. We can easily see that $a\leqslant\frac{1}{\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2}\sum_{j=1}^{i}C_0''(\xi_j(t_{j,4}))(t_{j,4}-t_{j,1})^2\leqslant b$. Hence, by the intermediate value theorem for continuous function, we know that there exists $\mu_i\in[t_{0,4},t_{i,4}]$ such that

$$\frac{1}{\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2}\sum_{j=1}^{i}C_0''(\xi_j(t_{j,4}))(t_{j,4}-t_{j,1})^2=C_0''(\mu_i). \tag{A.8}$$

From Eq. (A.8), we know that

$$\begin{aligned}
&\tfrac{1}{2}\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}C_0''(\beta_0)-C_0''(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\\
&=\tfrac{1}{2}\sum_{j=1}^{i}\left(\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right)C_0''(\beta_0)+C_0''(\beta_0)-C_0''(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\\
&=\tfrac{1}{2}C_0''(\beta_0)\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right)(t_{j,4}-t_{j,1})^2\\
&\quad+\tfrac{1}{2}\sum_{j=1}^{i}C_0''(\beta_0)(t_{j,4}-t_{j,1})^2-\tfrac{1}{2}\sum_{j=1}^{i}C_0''(\xi_j(t_{j,4}))(t_{j,4}-t_{j,1})^2\\
&=\tfrac{1}{2}C_0''(\beta_0)\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right)(t_{j,4}-t_{j,1})^2\\
&\quad+\tfrac{1}{2}C_0''(\beta_0)\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2-\tfrac{1}{2}\sum_{j=1}^{i}C_0''(\xi_j(t_{j,4}))(t_{j,4}-t_{j,1})^2\\
&=\tfrac{1}{2}C_0''(\beta_0)\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right)(t_{j,4}-t_{j,1})^2\\
&\quad+\tfrac{1}{2}[C_0''(\beta_0)-C_0''(\mu_i)]\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2
\end{aligned} \tag{A.9}$$

From the hypothesis (2) in Section 5, we can easily prove that $1-\rho\leqslant|C_j'(\eta_j(t_{j,\nu}))|\leqslant1+\rho$ $(\nu=0,2,4)$.

Thus, it is clear that

$$\begin{aligned}
&\left|\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right|=\left|\tfrac{C_j'(\eta_j(t_{j,4}))^2-C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}\right|\\
&=\left|\tfrac{C_j'(\eta_j(t_{j,4}))[C_j'(\eta_j(t_{j,4}))-C_j'(\eta_j(t_{j,0}))]+[C_j'(\eta_j(t_{j,4}))-C_j'(\eta_j(t_{j,2}))]C_j'(\eta_j(t_{j,0}))}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}\right|\\
&\leqslant\tfrac{1+\rho}{(1-\rho)^2}(|C_j'(\eta_j(t_{j,4}))-C_j'(\eta_j(t_{j,0}))|+|C_j'(\eta_j(t_{j,4}))-C_j'(\eta_j(t_{j,2}))|)
\end{aligned} \tag{A.10}$$

From Lemma A.4 and inequality (A.10), we know that

$$\begin{aligned}
\left|\frac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right|&\leqslant\frac{1+\rho}{(1-\rho)^2}\left(\frac{4}{\Delta}\cdot10^{-6}+\frac{4}{\Delta}\cdot10^{-6}\right)\\
&\leqslant\frac{1+\rho}{(1-\rho)^2}\frac{8}{\Delta}10^{-6}.
\end{aligned} \tag{A.11}$$

From Eq. (A.9) and inequality (A.11), we know that

$$\begin{aligned}
&\left|\tfrac{1}{2}\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}C_0''(\beta_0)-C_0''(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\right|\\
&\leqslant\tfrac{1}{2}|C_0''(\beta_0)|\sum_{j=1}^{i}\left|\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}-1\right|(t_{j,4}-t_{j,1})^2\\
&\quad+\tfrac{1}{2}|C_0''(\beta_0)-C_0''(\mu_i)|\cdot\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2\\
&\leqslant\tfrac{1}{2}|C_0''(\beta_0)|\tfrac{1+\rho}{(1-\rho)^2}\tfrac{8\times10^{-6}}{\Delta}\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2\\
&\quad+\tfrac{1}{2}|C_0''(\beta_0)-C_0''(\mu_i)|\cdot\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2.
\end{aligned} \tag{A.12}$$

By Lagrange's mean value theorem, there exists ν between β_0 and μ_i such that

$$C_0''(\beta_0)-C_0''(\mu_i)=C_0'''(\nu)(\mu_i-\beta_0). \tag{A.13}$$

Since $\beta_0\in(t_{0,3},t_{0,5})$ and $\mu_i\in[t_{0,4},t_{i,4}]$, from the definition of $t_{i,4}(t=1,2,\ldots,n)$ and the conclusion (4) in Lemma A.1 we know that they are both in $[t_{0,3},t_{i,4}]$. Therefore from Eq. (A.13) and the hypothesis (4) in Section 5, we know that

$$|C_0''(\beta_0)-C_0''(\mu_i)|\leqslant\varphi\times(t_{i,4}-t_{0,3}). \tag{A.14}$$

From the hypothesis (3) in Section 5, inequalities (A.12) and (A.14), we know that

$$\begin{aligned}
&\left|\tfrac{1}{2}\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}C_0''(\beta_0)-C_0''(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\right|\\
&\leqslant\tfrac{1}{2}\theta\tfrac{1+\rho}{(1-\rho)^2}\tfrac{8\times10^{-6}}{\Delta}\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2+\tfrac{1}{2}\varphi\cdot(t_{i,4}-t_{0,3})\cdot\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2
\end{aligned} \tag{A.15}$$

By the conclusion in [2], $t_{j,4}-t_{j,1}\leqslant0.62$ s, $\forall1\leqslant j\leqslant i$. Therefore, from Eq. (A.3) we know that

$$\sum_{j=1}^{i}(t_{j,4}-t_{j,1})^2<0.62(t_{i,4}-t_{0,4}) \tag{A.16}$$

Since $t_{i,4}-t_{0,4}<t_{i,4}-t_{0,3}\leqslant\tau$, from inequalities (A.15) and (A.16) we know that

$$\begin{aligned}
&\left|\tfrac{1}{2}\sum_{j=1}^{i}\left(\tfrac{C_j'(\eta_j(t_{j,4}))^2}{C_j'(\eta_j(t_{j,2}))C_j'(\eta_j(t_{j,0}))}C_0''(\beta_0)-C_0''(\xi_j(t_{j,4}))\right)(t_{j,4}-t_{j,1})^2\right|\\
&\leqslant0.31\theta\tfrac{1+\rho}{(1-\rho)^2}\tfrac{8\times10^{-6}}{\Delta}(t_{i,4}-t_{0,4})+0.31\varphi(t_{i,4}-t_{0,3})(t_{i,4}-t_{0,4})\\
&\leqslant2.48\theta\tfrac{1+\rho}{(1-\rho)^2}\tfrac{\tau}{\Delta}10^{-6}+0.31\varphi\tau^2.
\end{aligned} \tag{A.17}$$

From Eq. (A.2), inequalities (A.7) and (A.17) we know that

$$|\dot{L}_i-C_0(t_{i,4})|\leqslant\frac{8}{1-\rho}\cdot\frac{\tau}{\Delta}\cdot10^{-6}+2.48\theta\frac{1+\rho}{(1-\rho)^2}\frac{\tau}{\Delta}10^{-6}+0.31\varphi\tau^2.$$

From the above inequality and Table 1, we can easily check that

$$|\dot{L}_i-C_0(t_{i,4})|\leqslant8.1\times\frac{\tau}{\Delta}\times10^{-6}+\frac{1}{3}\varphi\tau^2.$$

Hence, we have proved this lemma. □

Considering the above preparations, now we can prove the key result of this appendix.

Lemma 5.1. *For any* $1 \leqslant i \leqslant n$, $|L_i(t) - C_0(t)| \leqslant 8.1 \times \frac{\tau+T}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2$ when $t \in [t_{i,4}, t_{i,4} + T)$

Proof. By the definitions of $\dot{L}_i$ and $t_{i,4}$, they are respectively the node i's logical time and the physical time when node i has just sent the first flag. Therefore $\dot{L}_i = L_i(t_{i,4})$. Furthermore, from the hypothesis (1) in Section 5 and Lemma A.5 we know that

$$|L_i(t_{i,4}) - C_0(t_{i,4})| = |\dot{L}_i - C_0(t_{i,4})| \leqslant 8.1 \times \frac{\tau}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2 < 8.1 \times \frac{T}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2.$$

Thus, the lemma holds when $t = t_{i,4}$. Below we assume that $t > t_{i,4}$. By the conclusion (6) in Lemma A.3, there exists $\tilde{\zeta}_i(t) \in (t_{i,4}, t)$ such that

$$C_0(t) = C_0(t_{i,4}) + C_0'(\tilde{\zeta}_i(t))(t - t_{i,4}). \tag{A.18}$$

From the definitions of $T_{i,v}$ and $t_{i,v}$, we know that $T_{i,v} = C_i(t_{i,v})(v = 1, 2, 4)$. Hence from Eq. (4.2.2), we know that

$$L_i(t) = \dot{L}_i + \frac{T_{0,5} - T_{0,4}}{C_i(t_{i,2}) - C_i(t_{i,1})} \cdot [C_i(t) - C_i(t_{i,4})].$$

By the conclusion (1) in Lemma A.3, there exists $\alpha_0 \in (t_{0,4}, t_{0,5})$ such that

$$T_{0,5} - T_{0,4} = C_0'(\alpha_0)(t_{0,5} - t_{0,4}).$$

Thus $L_i(t) = \dot{L}_i + \frac{C_0'(\alpha_0)(t_{0,5} - t_{0,4})}{C_i(t_{i,2}) - C_i(t_{i,1})} \cdot [C_i(t) - C_i(t_{i,4})]$. By the conclusions (5) and (7) in Lemma A.3, there exist $\eta_i(t_{i,2}) \in (t_{i,1}, t_{i,2})$ and $\tilde{\eta}_i(t) \in (t_{i,4}, t)$ such that

$$C_i(t_{i,2}) - C_i(t_{i,1}) = C_i'(\eta_i(t_{i,2}))(t_{i,2} - t_{i,1}) \text{ and } C_i(t) - C_i(t_{i,4}) = C_i'(\tilde{\eta}_i(t))(t - t_{i,4})$$

Therefore $L_i(t) = \dot{L}_i + \frac{C_0'(\alpha_0)(t_{0,5} - t_{0,4})}{C_i'(\eta_i(t_{i,2}))(t_{i,2} - t_{i,1})} \cdot C_i'(\tilde{\eta}_i(t)) \cdot (t - t_{i,4})$. rom Lemma A.1, we can easily prove that $t_{i,2} - t_{i,1} = t_{0,5} - t_{0,4}$. Hence we have prove that

$$L_i(t) = \dot{L}_i + \frac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0)(t - t_{i,4})$$

From the above equation and Eq. (A.18), we can obtain the following equation:

$$L_i(t) - C_0(t) = [\dot{L}_i - C_0(t_{i,4})] + \left(\frac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right)(t - t_{i,4}) \tag{A.19}$$

From the hypothesis (2) in Section 5, we can easily prove that $|C_0'(\alpha_0)| \leqslant 1 + \rho$ and $C_j'(\eta_j(t_{j,2}))| \geqslant 1 - \rho$. Thus,

$$\begin{aligned}
&\left|\tfrac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right| \\
&= \left|(\tfrac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} - 1) C_0'(\alpha_0) + C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right| \\
&= \left|\tfrac{C_i'(\tilde{\eta}_i(t)) - C_i'(\eta_i(t_{i,2}))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) + C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right| \\
&\leqslant \left|\tfrac{C_i'(\tilde{\eta}_i(t)) - C_i'(\eta_i(t_{i,2}))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0)\right| + \left|C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right| \\
&\leqslant \left|\tfrac{1+\rho}{1-\rho} |C_i'(\tilde{\eta}_i(t)) - C_i'(\eta_i(t_{i,2}))|\right| + \left|C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right|.
\end{aligned} \tag{A.20}$$

From Lemma A.4 and inequality (A.20), we know that

$$\left|\frac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))\right| \leqslant \frac{1+\rho}{1-\rho} \times \frac{4}{\Delta} \times 10^{-6} + \frac{4}{\Delta} \times 10^{-6} = \frac{8}{1-\rho} \times \frac{1}{\Delta} \times 10^{-6}.$$

From the above inequality, Eq. (A.18) and Lemma A.5, we can obtain the following inequality:

$$|L_i(t) - C_0(t)| \leqslant |\dot{L}_i - C_0(t_{i,4})| + |\frac{C_i'(\tilde{\eta}_i(t))}{C_i'(\eta_i(t_{i,2}))} C_0'(\alpha_0) - C_0'(\tilde{\zeta}_i(t))|(t - t_{i,4}) \leqslant 8.1 \times \frac{\tau}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2 + \frac{8}{1-\rho} \times \frac{T}{\Delta} \times 10^{-6}.$$

From the above inequality, the hypothesis (1) in Section 5 and Table 1, we can easily check that

$$[|L_i(t) - C_0(t)| \leqslant 8.1 \times \frac{\tau+T}{\Delta} \times 10^{-6} + \frac{1}{3}\varphi\tau^2$$

Hence we have proved this lemma. □

Appendix B

The purpose of this appendix is to prove the lemmas 4.3.1, 4.3.2, and 4.3.3 in Subsection 3 and Section 4.

Lemma 4.3.1. *For any* $0 < i \leqslant n$, $\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \approx g_{0 \leftarrow i}''(T_{i,1})$.

Proof. From the definition of the *synchronization packet* we know that $T_{i,2} - T_{i,1}$ is the time needed for node i to receive 12 bytes. Therefore $T_{i,2} - T_{i,1}$ is very short. Since $T_{i,2} - T_{i,1}$ is very short and $\lim_{h \to 0} \frac{g_{0 \leftarrow i}(T_{i,1}+h) - g_{0 \leftarrow i}(T_{i,1})}{h} = g_{0 \leftarrow i}'(T_{i,1}), \frac{g_{0 \leftarrow i}(T_{i,2}) - g_{0 \leftarrow i}(T_{i,1})}{T_{i,2} - T_{i,1}} \approx g_{0 \leftarrow i}'(T_{i,1})$. Because $T_{i,1} = C_i(t_{i,1})$ and $T_{i,2} = C_i(t_{i,2})$, $\frac{C_0(t_{i,2}) - C_0(t_{i,1})}{T_{i,2} - T_{i,1}} = \frac{g_{0 \leftarrow i}(T_{i,2}) - g_{0 \leftarrow i}(T_{i,1})}{T_{i,2} - T_{i,1}}$. From Lemma A.1, we can easily prove that $t_{i,2} - t_{i,1} = t_{0,5} - t_{0,4}$. As the time synchronization procedure (from the starting that the reference node sends a *synchronization packet* to the ending that all nodes have received this packet) is very short, so $t_{i,2} - t_{0,3}$ is very short. Thus on physical time interval $[t_{0,3}, t_{i,2}]$ the change of the reference node's hardware clock frequency is very small. By Lagrange's mean value theorem, there exist $\xi_{0,2} \in (t_{0,4}, t_{0,5})$ and $\eta_{0,i} \in (t_{i,1}, t_{i,2})$ such that $C_0(t_{i,2}) - C_0(t_{i,1}) = C_0'(\eta_{0,i})(t_{i,2} - t_{i,1})$ and $C_0(t_{0,5}) - C_0(t_{0,4}) = C_0'(\xi_{0,2})(t_{0,5} - t_{0,4})$. Since we have proved that $t_{i,2} - t_{i,1} = t_{0,5} - t_{0,4}$ and $C_0'(\eta_{0,i}) \approx C_0'(\xi_{0,2})$, it is very clear that $C_0(t_{0,5}) - C_0(t_{0,4}) \approx C_0(t_{i,2}) - C_0(t_{i,1})$. As $\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} = \frac{C_0(t_{0,5}) - C_0(t_{0,4})}{T_{i,2} - T_{i,1}}$, so we have proved that $\frac{T_{0,5} - T_{0,4}}{T_{i,2} - T_{i,1}} \approx \frac{C_0(t_{i,2}) - C_0(t_{i,1})}{T_{i,2} - T_{i,1}} \approx g_{0 \leftarrow i}'(T_{i,1})$. □

Lemma 4.3.2. *For any* $0 < i \leqslant n$, $\frac{T_{0,5} - 2T_{0,4} + T_{0,3}}{(T_{i,2} - T_{i,1})(T_{i,1} - T_{i,0})} \approx g_{0 \leftarrow i}''(T_{i,1})$.

Proof. From the definition of the *synchronization packet* we know that both $T_{i,1} - T_{i,0}$ and $T_{i,2} - T_{i,1}$ are the time needed for node i to receive 12 bytes. Therefore $T_{i,2} - T_{i,1}$ is very short and $T_{i,1} - T_{i,0} \approx T_{i,2} - T_{i,1}$. Using L'Hospital's Rule, it can be easily proved

$$\lim_{h \to 0} \frac{g_{0 \leftarrow i}(T_{i,1} + h) - 2g_{0 \leftarrow i}(T_{i,1}) + g_{0 \leftarrow i}(T_{i,1} - h)}{h^2} = g_{0 \leftarrow i}''(T_{i,1}). \tag{A.21}$$

Since $T_{i,2} - T_{i,1}$ is very short and $T_{i,1} - T_{i,0} \approx T_{i,2} - T_{i,1}$, the following approximate equation holds:

$$\frac{C_0(t_{i,2})-2C_0(t_{i,1})+C_0(t_{i,0})}{(T_{i,2}-T_{i,1})(T_{i,1}-T_{i,0})} = \frac{g_{0\leftarrow i}(C_i(t_{i,2}))-2g_{0\leftarrow i}(C_i(t_{i,1}))+g_{0\leftarrow i}(C_i(t_{i,0}))}{(T_{i,2}-T_{i,1})(T_{i,1}-T_{i,0})}$$
$$= \frac{g_{0\leftarrow i}(T_{i,2})-2g_{0\leftarrow i}(T_{i,1})+g_{0\leftarrow i}(T_{i,0})}{(T_{i,2}-T_{i,1})(T_{i,1}-T_{i,0})} \tag{B.1}$$
$$\approx \frac{g_{0\leftarrow i}(T_{i,1}+(T_{i,2}-T_{i,1}))-2g_{0\leftarrow i}(T_{i,1})+g_{0\leftarrow i}(T_{i,1}-(T_{i,2}-T_{i,1}))}{(T_{i,2}-T_{i,1})^2} \approx g''_{0\leftarrow i}(T_{i,1})$$

From Lemma A.1, we know that $t_{i,2}-t_{i,1}=t_{i,1}-t_{i,0}$. Hence it is very easy to check the following equation: $C_0(t_{i,2})-2C_0(t_{i,1})+C_0(t_{i,0})=\Delta_0(t_{i,2}-t_{i,1},t_{i,1})$. From Lemma A.2, we know that there exists $t_{i,0}<\xi<t_{i,2}$ such that

$$\Delta_0(t_{i,2}-t_{i,1},t_{i,1})=C''_0(\xi)(t_{i,2}-t_{i,1})^2.$$

In the same way, we can prove that $T_{0,5}-2T_{0,4}+T_{0,3}=\Delta_0(t_{0,5}-t_{0,4},t_{0,4})$ and there exists $t_{0,3}<\eta<t_{0,5}$ such that

$$\Delta_0(t_{0,5}-t_{0,4},t_{0,4})=C''_0(\eta)(t_{0,5}-t_{0,4})^2.$$

From Lemma A.1, $t_{i,2}-t_{i,1}=t_{0,5}-t_{0,4}$ can be easily proved. Therefore we have proved that there exist $t_{i,0}<\xi<t_{i,2}$ and $t_{0,3}<\eta<t_{0,5}$ such that

$$\begin{cases} C_0(t_{i,2})-2C_0(t_{i,1})+C_0(t_{i,0})=C''_0(\xi)(t_{0,5}-t_{0,4})^2 \\ T_{0,5}-2T_{0,4}+T_{0,3}=C''_0(\eta)(t_{0,5}-t_{0,4})^2 \end{cases}. \tag{B.2}$$

As the time synchronization procedure is very short, so $t_{i,2}-t_{0,3}$ is very short. Hence on physical time interval $[t_{0,3},t_{i,2}]$, the change of the reference node's hardware clock drift variation is very small. From this we know that $C''_0(\xi)\approx C''_0(\eta)$. Thus, from approximate Eqs. (B.1) and (B.2), we know that $\frac{T_{0,5}-2T_{0,4}+T_{0,3}}{(T_{i,2}-T_{i,1})(T_{i,1}-T_{i,0})}\approx g''_{0\leftarrow i}(T_{i,1})$. □

Lemma B.1. *For any $0\leqslant i\leqslant n$, $\dot{L}_i\approx C_0(t_{i,4})$.*

Proof. We prove this lemma by using mathematical induction on node numbers. From the definition of $\dot{L}_0$, we know that $\dot{L}_0=T_{0,4}$. Since $T_{0,4}=C_0(t_{0,4})$, $\dot{L}_0=C_0(t_{0,4})$. Therefore the conclusion of this lemma is true for node 0. Now suppose that the lemma is true for nodes whose numbers are less than i. By the induction hypothesis, $\dot{L}_{i-1}\approx C_0(t_{i-1,4})$. From Lemma A.1, we know that $t_{i-1,4}=t_{i,1}$. As $T_{i,1}=C_i(t_{i,1})$, so $\dot{L}_{i-1}\approx C_0(t_{i-1,4})=C_0(t_{i,1})=g_{0\leftarrow i}(C_i(t_{i,1}))=g_{0\leftarrow i}(T_{i,1})$. Hence, from Eq. (4.2.1), lemmas 4.3.1 and 4.3.2, we know that

$$\dot{L}_i\approx g_{0\leftarrow i}(T_{i,1})+g'_{0\leftarrow i}(T_{i,1})(T_{i,4}-T_{i,1}) + \frac{1}{2}g''_{0\leftarrow i}(T_{i,1})(T_{i,4}-T_{i,1})^2. \tag{B.3}$$

By the Taylor's formula, there exists $\Gamma_{0,i}\in(T_{i,1},T_{i,4})$ such that

$$g_{0\leftarrow i}(T_{i,4})=g_{0\leftarrow i}(T_{i,1})+g'_{0\leftarrow i}(T_{i,1})(T_{i,4}-T_{i,1}) + \frac{1}{2}g''_{0\leftarrow i}(\Gamma_{0,i})(T_{i,4}-T_{i,1})^2. \tag{B.4}$$

Since $\Gamma_{0,i}-T_{i,1}$ is shorter than the time needed for node i to forward a *synchronization packet*, $\Gamma_{0,i}-T_{i,1}$ is very short. Therefore by the continuity of $g''_{0\leftarrow i}(x)$, $g''_{0\leftarrow i}(\Gamma_{0,i})\approx g''_{0\leftarrow i}(T_{i,1})$. Furthermore, from approximate Eqs. (B.2) and (B.3), we know that $\dot{L}_i\approx g_{0\leftarrow i}(T_{i,4})$. Since $T_{i,4}=C_i(t_{i,4})$, $\dot{L}_i\approx g_{0\leftarrow i}(T_{i,4})=g_{0\leftarrow i}(C_i(t_{i,4}))=C_0(t_{i,4})$. □

Lemma 4.3.3. For any $0i\leqslant n$, $\dot{L}_{i-1}\approx g_{0\leftarrow i}(T_{i,1})$.

Proof. From lemmas A.1 and B.1, we know that $t_{i-1,4}=t_{i,1}$ and $\dot{L}_{i-1}\approx C_0(t_{i-1,4})=C_0(t_{i,1})$. Since $T_{i,1}=C_i(t_{i,1})$, $C_0(t_{i,1})=g_{0\leftarrow i}(C_i(t_{i,1}))=g_{0\leftarrow i}(T_{i,1})$. Thus, we have proved that $\dot{L}_{i-1}$ is approximately equal to $g_{0\leftarrow i}(T_{i,1})$. □

References

[1] C. Lenzen, P. Sommer, R. Wattenhofer, Optimal clock synchronization in networks, in: SenSys 2009: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2009, pp. 225–238.
[2] M. Maroti, B. Kusy, G. Simon, A. Ledeczi A, The flooding time synchronization protocol, in: Proc. of the 2nd Int'l Conf. on Embedded Networked Sensor Systems, 2004, pp. 39–49.
[3] S. Ganeriwal, R. Kumar, M.B. Srivastava, Timing-sync protocol for sensor networks, in: Proc. of the 1st ACM Conf. on Embedded Networked Sensor Systems (SenSys), ACM Press, 2003, pp. 138–149.
[4] Jang-Ping Sheu, Hu Wei-Kai, Jen-Chiao Lin, Ratio-based time synchronization protocol in wireless sensor networks, J. Telecommun. Syst. 39 (1) (2008) 25–35.
[5] Pai-Han Huang, Maulik Desai, Xiaofan Qiu, Bhaskar Krishnamachari, On the multihop performance of synchronization mechanisms in high propagation delay networks, IEEE Trans. Comput. 58 (5) (2009) 577–590.
[6] Jun Zheng, Yik-Chung Wu, Localization and time synchronization in wireless sensor networks: a unified approach, in: Proceedings of 2008 IEEE Asia Pacific Conf. on Circuits and Systems (APCCAS 2008), Macao, China, November 2008, pp. 594–597.
[7] Surendra Rahamatkar, Ajay Agarwal Dr., A reference based, tree structured time synchronization approach and its analysis in WSN, Int. J. Ad hoc Sensor Ubiquitous Comput. (IJASUC) 2 (1) (2011).
[8] Qasim M. Chaudhari, Erchin Serpedin, Clock estimation for long-term synchronization in wireless sensor networks with exponential delays, EURASIP J. Adv. Signal Process. 2008 (2008) 6. Article ID 219458.
[9] Xu Chao-Nong, Zhao Lei, Xu Yong-Jun, Li Xiao-Wei, Simsync: A Time Synchronization Simulator for Sensor Networks, Acta Autom. Sin. 32 (6) (2006).
[10] Sun Juan, Ma Zhongmei, Liu Jiawei, et al., ZigBee wireless networking technology and application based on LM3S9B96+CC2520, Microcontrollers Embedded Syst. 11 (12) (2011) 47–50.
[11] Feng Yong Ping, Zhang Di, Luo Rong, Design of a low-power wireless sensor network node, Appl. Electron. Tech. 39 (4) (2009) 33–36.
[12] Peng Yu, Luo Qinghua, Peng Xiyuan, The design of low-power wireless sensor node, in: 2010 IEEE Instrumentation and Measurement Technology Conference (I2MTC), 2010, pp. 917–922.
[13] http://focus.ti.com/lit/ds/symlink/cc2520.pdf.
[14] Ge Huang, Albert Y. Zomaya, Flávia C. Delicato, Paulo F. Pires, An accurate on-demand time synchronization protocol for wireless sensor networks, J. Parallel Distrib. Comput. 72 (10) (October 2012) 1332–1346.
[15] J. Elson, K. Römer, Wireless sensor networks: a new regime for time synchronization, in: Proc. of the 1st Workshop on Hot Topics in Networks (HotNets-I), 2002, pp. 28–29.
[16] J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcasts, in: Proc. of the 5th Symp. on Operating Systems Design and Implementation, ACM Press, 2002, pp. 147–163.
[17] Christoph Lenzen, Thomas Locher, Roger. Wattenhofer, Tight bounds for clock synchronization. PODC (2009) 46–55.
[18] F. Sivrikaya, B. Yener, Time synchronization in sensor networks: a survey, Network IEEE 18 (4) (2004) 45–50.
[19] L. Schenato, G. Gamba, A distributed consensus protocol for clock synchronization in wireless sensor network, in: Decision and Control, 2007 46th IEEE Conference on, Dec. 2007, pp. 2289–2294.
[20] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, Robust multi-hop time synchronization in sensor networks, in: Int. Conf. on Wireless and Mobile Computing (ICWN), Las Vegas, USA, 2004, pp. 454–460.
[21] K. Romer, F. Mattern, Towards a unified view on space and time in sensor networks,, Comput. Commun. 28 (2005).
[22] H. Oliveira, E. Nakamura, A. Loureiro, Localization in time and space for sensor networks, in: Proc. 21st International Conf. on Advanced Info. Net. and App., May 2007, pp. 539–546.
[23] S. Rahamatkar, Ajay Agarwal, V. Sharma, Tree structured time synchronization protocol in wireless sensor network, UbiCC J. 4 (2009) 712–717.
[24] Woo, Culler, D., (2001) A Transmission Control Scheme for Media Access in Sensor Networks", Mobicom, pp. 221–235.
[25] http://www.ti.com/lit/ds/symlink/cc2420.pdf.
[26] Ill-Keun Rhee, Jaehan Lee, Jangsub Kim, Erchin Serpedin, Wu Yik-Chung, Clock synchronization in wireless sensor networks: an overview, J. Sens. 9 (1) (2009) 56–85.
[27] F.Y. Ren, S.Y. Dong, T. He, C. Lin, A time synchronization mechanism and algorithm based on phase lock loop, J. Softw. 18 (2) (2007) 372–380.
[28] Federico Ferrari, Andreas F. Meier, Lothar Thiele, Accurate clock models for simulating wireless sensor networks, in: 3rd International Workshop on OMNeT++ (OMNeT++2010), ICST, Malaga, Spain, March 2010, pp. 1–4.
[29] MENSIC web site. Available at: http://www.memsic.com.
[30] Q. Gao, K.J. Blow, D.J. Holding, Simple algorithm for improving time synchronisation in wireless sensor networks, Electron. Lett. 40 (14) (2004) 889–891, http://dx.doi.org/10.1049/el:20045270.