Review of Time Synchronization in Wireless Sensor Networks

Fang WANG¹*, Yuan-qing QIN² and Hai-dong LEI¹

¹College of Physics and Information, Jianghan University, 430056, China
²College of Automation, Huazhong University of Science and Technology, 430074, China

*Corresponding author

Keywords: Time synchronization, Wireless sensor network, Consensus.

Abstract. Time synchronization is one of the vital technologies of WSNs. Three classes of time synchronization technologies were summarized in this article: traditional time synchronization technique, cooperation time synchronization technique and consensus time synchronization technique. The synchronization mechanism of the above-mentioned time synchronization methods were introduced, and their advantages and disadvantages were discussed in this paper. This paper also pointed out the research trends in time synchronization of WSNs.

Introduction

Wireless sensor networks (WSNs) have been widely used in military areas, enjoinment monitoring, agriculture, because of its low cost and easy to deploy. Time synchronization is a support technique of WSNs. It provides a common timestamp for nodes in network and is the premise of WSNs’ normal performs. The cooperative awareness of nodes, localization, data confusion and the protocols’ realization of WSNs all needs time synchronization. Therefore, it has great theoretical significance and application value to study time synchronization of WSNs. Many synchronization techniques have been proposed for sensor networks by far. Each node in the networks is required to estimate its clock skew and clock offset relative to the reference clock in these algorithms.

The WSN nodes depend on counting interrupts of crystal oscillator on nodes to keep local time [1]. The frequency of crystal oscillator may vary with the change of environment temperature and the aging of crystal oscillator. At the same time, the differences among power-on times of nodes will led to the different initial local times of nodes. All of these result in the non-synchronous of nodes’ local time. The model of time synchronization of WSNs is introduced in following.

Assuming the local time of node I, which is according to its local clock, can be expressed as:

\[ C_i(t) = \int_0^t f_i(t) dt + C_i(t_0) \]  

where \( f_0 \) is the nominal frequency of crystal oscillator on node; \( f_i(t) \) is the actual frequency of crystal oscillator on node; \( t_0 \) is the initial time of node; \( C_i(t_0) \) is the read of local clock at time \( t_0 \). The frequency of crystal oscillator is assumed constant during a short period.

The local clock of node i can be expressed as:

\[ C_i(t) = \alpha_i t + \beta_i \]  

where \( \alpha_i \) is the clock skew; \( \beta_i \) is the clock offset, defined as the difference between local clock of node i and absolute time.

It must be realized that the absolute time is unknown to nodes in WSN. Therefore, \( \alpha_i \) and \( \beta_i \) can’t be calculated from Eq. 2. However, the linear correlation between node i and node j can be obtained as following:

\[ C_j(t) = \alpha_j C_i(t) + \beta_j \]
where αij is the relative clock skew between node i and node j, αij=αj/αi; βij is the relative clock offset between node i and node j, βij=βj-(αjβi)/αi. If node i and node j is synchronized, then αij=1 and βij=0.

If all nodes in a WSN synchronize with the same clock, then the synchronization of the network is reached.

Large amount of researches in WSNs synchronization have been performed since Elson first proposed this issue in 2002. Up to now, researches in this field can be divided into three classes: traditional time synchronization techniques, cooperative synchronization technique and consensus synchronization technique.

**Traditional Synchronization Algorithms for WSNs**

The traditional synchronization algorithms for WSNs can be divided into three classes according to synchronize objects: (1) receiver-receiver based synchronization algorithm, such as RBS(Reference Broadcast Synchronization) [2]; (2) sender-receiver based bisynchronous algorithm, such as TPSN(Timing-Sync Protocol for Sensor Networks) [3]; (3) sender-receiver based one-way time synchronization algorithm, such as FTSP(Flooding Time Synchronization Protocol) [4], DMTS(Delay measurement Time Synchronization) [5].

**RBS**

A global reference clock is used in RBS, and other nodes synchronize according to the average offset between each other. Firstly, the reference node periodically broadcasts synchronization message to other nodes in the network. Then nodes in the propagation area receive the message and recode the receive time Tik (i=1,2,...,n; n is the number of receivers; k is the order of receiving synchronization message) according to their local clocks. Finally, the receivers exchange their receive times and calculate the offsets Tjk-Tik (the offset between node j and node i) between each other. Then each node can get its average offsets to other nodes [2]:

\[
\text{offset}_{ij} = \frac{1}{m} \sum_{k=1}^{m} (T_{jk} - T_{ik}) \quad \forall i \in n, j \in n
\]

where m is the number of reference broadcasts. Each node adjust its local clock according to the offset. Therefore, all the nodes receiving the same reference broadcasts are synchronized.

When extending to multi-hop networks, RBS divides the network into serval clusters, and sets a beacon node for each cluster [2]. All the nodes in a cluster using the beacon node to synchronize. Receivers that are in the same cluster (i.e., have heard the sync pulse from the same beacon node) can relate their clocks to each other. There are some particular nodes, which are in two or more than two clusters at the same time, called gateways. The gateway, which is illustrated in Figure 1 as node g, can heard from beacon node A and beacon node B and relate the clocks on its neighbours. So the synchronization between clusters is achieved.

**TPSN**

TPSN is a hierarchical structure time synchronization algorithm proposed by Ganeriwal. This algorithm has two phases [3]: level discovery phase and synchronization phase. In level discovery phase, a layered topological structure of the network is established, and each node in the network is given a sequence number of layer. The root node, which is the reference clock of the network and is the sponsor of this level discovery process, is assigned a level 0. Other nodes are layered according to their distances to root node. In order to avoid breakdown and fit networks’ topological structure.
change, TPSN algorithm selects the root node and establishes the topological structure of the network dynamically. In synchronization phase, the root node broadcasts synchronization information messages, the immediate neighbors of the root node receive this packet and synchronize with the root node. In this way, the i level nodes synchronize with the i-1 level nodes, and so on. Finally, all nodes synchronize with the root nodes.

![Figure 2. Two-way Exchange of Packets between Two Nodes.](image)

The TPSN algorithm adopts timestamps in MAC layer and two-way exchange of packets to diminish the affect of transmission delay and the delay uncertainty. The two-way exchange of packets mechanism is shown in Figure 2. The sender transmits synchronization request message at time T1. The receiver receives this message at time T2, records the receive timestamp T1, and transmits acknowledge message at time T3. The sender gets all timestamps of this exchange process (T1, T2, T3 and T4) at time T4, where T1 and T4 are local time of sender, while T2 and T3 are local time of receiver. The sender can calculate the clock drift $\theta$ and propagation delay $\delta$ as [3]:

\[
\theta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}
\]

\[
\delta = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}
\]

Then the sender can correct its clock with the drift, therefore, synchronize to the receiver. All nodes gradually synchronize to the lower level neighbors layer by layer according to the hierarchical structure established in level discovery phase, and the synchronization of the whole net is realized.

**DMTS**

![Figure 3. Flow of Time Synchronization Message Transmitting.](image)

As shown in Figure 3, senders in DMTS transmit preamble and start symbols before transmitting data [4]. The sender transmits a time synchronization message with its timestamp T, which is added after MAC delay and a clear channel is detected. On receiving the preamble and start symbols, the receiver recodes its local time T1, and local time T2 on starting to process received data. Ignoring the propagation delay, the whole delay is the sum of the time to transmit preamble and start symbols and the difference between T2 and T1. Assuming that the preamble and start symbols have n bits and each bit needs a period of rto transmit, then the local clock of the receiver can be set to T_r to synchronize to the sender [4].
\[ T_r = T + n\tau + (T_2 - T_1) \]  

DMTS also uses a hierarchical structure to implement multi-hop synchronization and its multi-hop synchronization mechanism is similar to TPSN.

**Advantages and Disadvantages of Traditional Time Synchronization Algorithms**

Receiver-receiver based synchronization technique synchronizes receivers with one another, removing the largest sources of nondeterministic latency from the critical path, which significantly improves the precision of synchronization. However, the nodes in the networks need to exchange local time messages, which enlarged the energy consumption. Moreover, receiver-receiver based synchronization technique has poor scalability.

Sender-receiver based bisynchronous algorithm, such as TPSN, converges quickly; but it requires increasement of energy consumption and computation complexity to obtain better precision of synchronization. Sender-receiver based one-way time synchronization algorithm, such as DMTS, has lower energy consumption due to the smaller number of message broadcasting required. However, its precision of synchronization is lower than RBS.

In addition, there exists an inherent problem in sender-receiver based synchronization algorithms: the accumulated synchronization error. Since each node estimates synchronization parameters with time messages from itself or its neighbors, there will be inherent errors in the estimation. So when a node multiple hops away from the reference node estimates its parameters from intermediate nodes that already have estimation errors, the synchronization error become larger. This will greatly affect the synchronization algorithms’ scalability.

**Cooperative Time Synchronization**

![Figure 4. Cooperation Synchronization Schematic Diagram.](image)

An-Swol and Servetto first put forward cooperative time synchronization technique in 2006 [6]. The main idea of cooperative time synchronization technique is simultaneously using information from all the neighbors to improve the quality of a timing observation made by a node. As shown in Figure 4, the reference clock node 1 cannot directly transmit time reference message to receive node n in the distance, because of limited transmit power. In cooperative time synchronization method, node 1 sends m synchronization pulses at equal time intervals, and the transmit times of these m synchronization pulses are recorded by the one-hop neighbors of node 1. Then the one-hop neighbors of node 1 estimate the transmit time of the m+1 synchronization pulse based on recorded m synchronization pulses, and transmit the m+1 synchronization pulse simultaneously with node 1. The superposition of signal power results in further propagation area for the composite synchronization pulses of node 1 and its one-hop neighbors. Therefore, node 3 can receive the composite synchronization pulses. In this way, all nodes in the network will transmit synchronization pulses simultaneously at last, which means synchronization is implemented.

The outstanding performance of cooperative time synchronization technique is that it eliminates the error accumulation phenomenon existing in traditional time synchronization methods, because nodes far away from time reference node can receive the synchronization pulses from reference node through the relay of intermediate nodes. At the same time, relaying allows lower transmit power of nodes, which reduces the energy consumption of nodes and prolongs the lifetime of network.
However, cooperative time synchronization technique uses spatial averaging to realize synchronization, so it needs high density of nodes, which may restrict its use in sparse networks. In addition, it is difficult to guarantee the perfect recombination of synchronization pulses from different nodes that have different delays.

Recently research in cooperative time synchronization techniques laid emphasis on algorithm implementation and the establishment of spanning tree for time synchronization [7, 8].

**Consensus Time Synchronization**

The study of consensus started in 1665. Physicist Huygens discovered that two clock plummets hanging on the beam swinging synchronized after a period of swinging. In 1935, Smith pointed out that thousands of fireflies flash in synchrony at night in certain parts of Southeast Asia. Recently, Reza Olfati-Saber’s research in consensus of multi agents introduced new interesting in consensus study [9].

Consensus time synchronization technique does not need to know the topology structure of network to synchronize. It depends on the exchange of time messages between nodes to converge the local clocks on nodes to the same reference clock. The consensus time synchronization technique is robust and has good scalability. The model for it is introduced in following.

Assuming exist a virtual clock $C_v(t)$, which is the reference clock, can be expressed as [1]:

$$C_v(t) = \alpha_t + \beta_v \quad (8)$$

And each node can get the estimation of virtual time using its own local clock:

$$\hat{C}_i(t) = \hat{\alpha}_iC_v(t) + \hat{\gamma}_i \quad (9)$$

Therefore the synchronization of WSNs changed to finding the appropriate $(\hat{\alpha}_i, \hat{\gamma}_i)$, such that:

$$\lim_{t \to \infty} \hat{C}_i(t) = C_v(t) \quad i = 1, 2, ..., N \quad (10)$$

The previous expression can be rewritten by substituting Eq. 2 into Eq. 9 to get:

$$\hat{C}_i(t) = \hat{\alpha}_i, \alpha_t + \hat{\alpha}_i, \beta_i + \hat{\gamma}_i \quad \text{(11)}$$

Then Eq. 10 is equivalent to:

$$\lim_{t \to \infty} \hat{\alpha}_i(t) = \frac{\alpha_v}{\alpha_i} \quad i = 1, 2, ..., N \quad (12)$$

$$\lim_{t \to \infty} \hat{\gamma}_i(t) = \beta_v - \hat{\alpha}_i(t)\beta_i = \beta_v - \frac{\alpha_v}{\alpha_i}\beta_i \quad i = 1, 2, ..., N \quad (13)$$

Therefore, if Eq. 12 and Eq. 13 is guaranteed, then the synchronization of WSNs is reached.

In consensus time synchronization technique, each node broadcasts its time message to the networks and neighbors in the propagation area receive the message. So after a period (considering propagation delays), each node has received the time messages from several neighbors. Then each node estimates the time parameters, such as $(\hat{\alpha}_i, \hat{\gamma}_i)$, using consensus algorithm with the received time messages. With these estimated values, each node updates its time parameters and broadcasts the new parameters to network. This process will be carried out periodically, until all the nodes have the same time parameters [10, 11].

To implement synchronization, the model of clocks on nodes should be studied. In addition, the hinge of synchronization is the design of control inputs of consensus controller, which affects the
convergence speed of synchronization process [12]. The delay of synchronization messages’ transmission and packets lost is the main factor of synchronization performance.

**Summary**

The time synchronization techniques were reviewed in this paper. Three classes of time synchronization techniques were summarized: traditional time synchronization techniques, cooperative time synchronization technique and consensus time synchronization technique. The synchronization mechanisms of these synchronization techniques were introduced and the advantages and disadvantages of them were compared.

As to finite-sized networks, traditional time synchronization techniques can quickly converge, and have less calculate complexity. Nevertheless, traditional time synchronization techniques adopt hierarchical structure or clusters to cope with the large-scale networks and suffer from an inherent scalability problem those synchronization errors grow with the incensement of hops across the network. This greatly restricts its use.

As to the networks with larger scale, cooperative time synchronization technique and consensus synchronization technique is more popular. However, cooperative time synchronization method uses spatial averaging to average out the errors inherent in each node, which means the node needs enough synchronization pulses to calculate. So cooperative time synchronization method is more suitable to dense wireless networks. Consensus synchronization technique provides a new method for time synchronization of WSNs. Its convergence speed needs improve. In addition, synchronization with delays and packets lost has not been perfectly solved. However, with the research progresses in consensus, consensus synchronization technique will be more practical.

**References**


