

Research Overview of Clock Synchronization in Wireless Sensor Network

Hua Yuting*, Xin Yu, Lu Kangli and Wang shenghui

School of Information Engineering, Zhejiang A&F University, Hangzhou Zhejiang 311300, China

Received: April 17, 2018; Accepted: May 15, 2018; Published: June 04, 2018

*Corresponding author: Hua Yuting, school of Information Engineering, Zhejiang A&F University, Hangzhou Zhejiang 311300, China, E-mail: 794939362@qq.com

Abstract

Clock synchronization is the key technology of Wireless Sensor Networks, since inaccurate clocks have a significant impact on network accuracy and security. In this paper, an important application of clock synchronization in wireless sensor networks is introduced. It is concluded that the deterministic causes of clock out-of-synchronization are the effects of offset and drift rate. Five classic methods of improved synchronization method for comparative study. Including: the improved method of least squares, delay measurement time synchronization, sensor network timing synchronization, tiny synchronization, lightweight tree synchronization. And we summed up non deterministic factors such as precision, complexity, overhead and so on are summarized, and then we put forward the corresponding problems. Finally, the new development direction and trend of wireless sensor time synchronization algorithm are proposed.

Keywords: Wireless sensor network; Clock synchronization; Drift rate; Deviation;

Introduction

The wireless sensor network is composed of randomly distributed micro-nodes integrated with sensors, data processing units, and communication modules. It is composed of self-organizing methods and can accomplish tasks such as data collection, processing, and signal transmission through cooperation among nodes in the network. Wireless sensor network is an Ad-Hoc network, which is also a distributed sensor network that is highly integrated by various sensors and controlled by its own system. Therefore, in addition to the common features such as mobility and disconnection of wireless network, wireless sensor network has many other distinctive features: flexibility, strong anti-interference ability, and adaptability. They are widely used in homes and schools, medical, commercial, manufacturing, defense, and other aspects.

With more and more researchers discovering and applying sensors, more and more powerful functions have also been studied and used by people. Wireless sensor is a technology that every country pays attention to and the clock synchronization is also regarded as the most important part of the wireless sensor network. In a distributed system, it is impossible to provide a unified global clock for independent modules. Each module maintains its own local time which causes drift and deviation due

to the environment of each module, resulting in the observation time and the sensor node to record time is not the same, and there is no guarantee that time will always be synchronized after a calibration [1]. This will lead to low accuracy in the collection of information and delivery of information, and sensors need a common time stamp to work together to complete complex sensing tasks. Clock synchronization is also of great significance for many energy-saving schemes. For example, the sensor can sleep at the right time, when data arrives, it can be received in time, and nodes also need to sleep and wake up at the same time, of course, this also requires precise timing between sensor nodes. There are many wireless sensor nodes so the key to coordinating cooperation among various nodes is to have a unified clock which requires the synchronization method to have advantages of good scalability, low cost, and low complexity. Because the initialization or change of the environment factors will cause the node clock to lose synchronization and make the measurement clock inaccurate, resulting in inconvenience, clock synchronization is particularly important. Accurately calculating the deviation and drift rate of the node clock is the key to the formation of clock synchronization.

As early as in 1992, the inaccuracy and inconvenience of setting up the host time was discovered, on this basis "timeout algorithm" was proposed. In the commercial network at that time, clock synchronization was achieved through network nodes. Specifically, it means that in the past, when the destination node initiated the clock synchronization signal, it started timing, and then started the local clock timer. If it exceeds a certain period of time, this result was not counted in the record, in order to avoid the error caused by the dead loop and forced algorithm to end [2]. This was the first time in the country that some scholars studied the principle of clock synchronization and its drawbacks. The "timeout algorithm" has opened the course of synchronous development of domestic clock. With the application of traditional shortwave time, it was found that although the coverage area was wide but it was easily affected by natural conditions and the time synchronization is inaccurate. In 1997, the Global Positioning System (GPS) of the time ranging navigation satellite was used in the clock synchronization device. The application has gradually increased [3]. In 1999, Network Time Protocol (NTP) and direct connection time transmission were

widely used as two important methods for the synchronization of computer clocks. Combines direct connection time transmission with an accuracy of up to 1 μ s and NTP protocol with an accuracy of only 1 ms, it was not only improved the accuracy of clock synchronization, but also saved cost and increased the flexibility of use of the NTP protocol [4,5]. In 2001, as the research on wireless sensor clock synchronization became more and more in-depth, Domestic scholars no longer regard clock synchronization as overall synchronization, but instead divide the distributed system clock synchronization method into internal used relative clock synchronization and absolute time synchronization under external time reference conditions. As well as the absolute synchronization under external time reference conditions, under this premise, a new triangle mesh simplification method based on quadratic errors was proposed, which not only realized a simple and fast method, but also could maintain high precision, it caused the discussion of many scholars. At the beginning of the 21st century, domestic scholars improved and combined traditional clock synchronization algorithms [6,7]. In 2005, there appeared to be improved high-precision and low-cost synchronization algorithms by improving traditional GPS methods. This algorithm retained the averages of the low cost, long-distance, and stable signals of GPS, and the overall design concept of GPS was used to correct the time-frequency signal of atomic clocks on the ground stations [8,9]. Before this, GPS had been applied in various fields, including that digital communication network, traffic, electricity, financial data network, military, ionospheric characteristics research. In order to give full play to the advantages of GPS, domestic scholars began to study the low-cost and real-time high-precision clock synchronization method of GPS, and also achieved some achievements solved the problem of low synchronization accuracy and low stability [10]. However, the GPS usage environment was limited. More scholars chose to use NTP protocol to achieve network clock synchronization based on the advantages of GPS. Later, an algorithm that used the convergence algorithm for error analysis and error estimation has emerged, which was more innovative in that it was not only able to be called by the NTP client software in a custom manner, but also easy to use in the local hardware clock system [11]. Accuracy could reach below the second level, but at that time, technological development was limited to fail to realize the effect of testing the client software and multiple servers at the same time. This was also the biggest deficiency of the NTP protocol [12].

Since 2010, wireless sensors have been known and used by more and more people, and clock synchronization has been deeply studied and used in different aspects. Combining clock synchronization with traditional ranging, combined with a ranging mechanism based on Time Difference of Arrival (TDOA), a new algorithm was generated [13]. Simulation results showed that this algorithm has better robustness and message exchange capacity than the existing classical time synchronization algorithm. Overcoming the limitations of the traditional ranging positioning on clock synchronization, adopting reverse thinking and proposing new algorithms not only eliminated the precondition of TDOA on clock synchronization, but also greatly improved robustness, synchronization accuracy, etc [14]. In

the same year, domestic scholars started to study the rush of synchronization algorithms for traditional clock synchronization algorithms such as: Reference Broadcast Synchronization (RBS), Delay Measurement Clock Synchronization (DMTS), sensor network Timing Synchronization Protocol (TPSN), and Tiny Synchronization (Tiny-sync), Lightweight Tree Synchronization (LTS), etc. They have been improved and innovated, with improvements in accuracy, robustness, stability, overhead, etc, and the reduction of data packets in the level discovery phase through the position between nodes. The transmission, in order to reduce the clock synchronization energy consumption [15]. With the development of science and technology, the hardware and software environment was becoming more and more comprehensive, and the traditional clock synchronization method was known to everyone and applied to all aspects of life [16]. As the accuracy, efficiency, and simplicity of each algorithm have been improved, the fault tolerance rate has also become an important measure. Taking the FTSP method as an example, during the use of the traditional FTSP method, there were many received error messages and the clock drift rate fluctuated greatly. After combining it with the error tolerance rate, the new drift rate was calculated by weighted averaging the existing clock drift rate in the past, resulting in an improved FTSP method [17]. The weighting coefficient was still determined dynamically by the variance of the previous measurement values, and after many experiments. It was found that the drift rate of the error message was relatively stable, the influence of synchronization error was small, and the fault tolerance rate was strong [18].

In this paper, the error of wireless sensor clock synchronization is studied, and the influencing factors of deterministic and non-deterministic errors are summarized. After several observations and actual time experiments, the deterministic errors with the greatest impact include the drift rate and the amount of deviation, and relate figures are shown. Through experiments, we study the non-deterministic errors in the information transfer in each clock synchronization method, propose corresponding improvement measures, and finally propose new directions and prospects.

Sensor Clock Synchronization Problem Features

Clock synchronization as a problem to be solved in wireless sensor networks has drawn attention from scholars at home and abroad. Nowadays, in the Internet network, the clock synchronization scheme follows the NTP network, and it adjusts the clock according to the delay and time difference between one cycle and the next cycle, so as to achieve the purpose of the time consistency of the two routers. The NTP network protocol requires two-way communication to eliminate delays in the transmission of information to achieve clock synchronization. However, in the current sensor networks, it is more difficult to implement bidirectional communication symmetry.

Relatively speaking, the time-of-flight navigation satellite global positioning system is more capable of meeting sensor nodes with limited capabilities and calculations, so GPS is the most common means for clock synchronization. However, GPS is expensive and requires high environmental requirements.

It must be used in an unobstructed environment [19]. GPS was originally invented by the United States to meet military needs. The space is composed of several GPS satellites, each of which is the atomic clocks are equipped with helium and neodymium, which are periodically calibrated by the monitoring station. GPS receiving devices generally receive signals from four or more satellites. Not only can they measure the position on the earth, but they can also monitor the gap between them and the world time, and then calibrate it. Due to the special nature of wireless sensor networks, their transmission has untimely delays, strict requirements, and robustness. Therefore, NTP and GPS cannot be directly used for wireless sensor network clock synchronization.

There are two causes of the unsynchronized wireless sensor clock: deterministic and non-deterministic. The deterministic error includes many aspects. However, it has been found through experiments that the effects of drift rate and deviation on clock synchronization are the most. As shown in figure 1, clock skew and

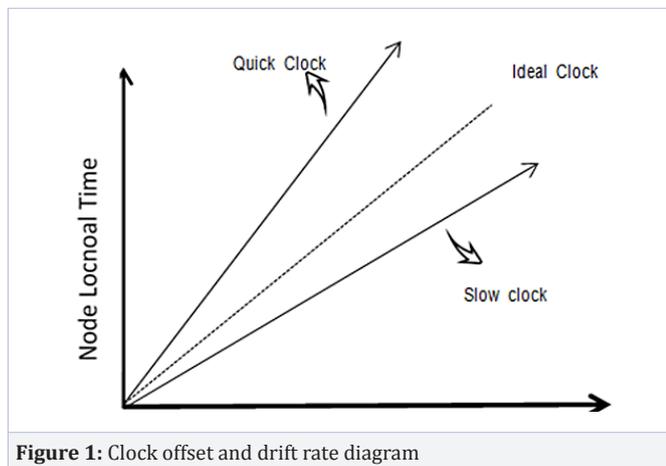


Figure 1: Clock offset and drift rate diagram

clock drift rates are different in different environments. Definition represents ω_i node clock drift rate, that is the rate of change of the observation time and the actual time when the signal reaches the node i , θ_i represents the deviation of the node i clock, that is the time difference between the observation time and the actual time at the same time. It can be seen that the observation time T and the actual time t have the following relationship:

$$T = \omega_i t + \theta_i \quad (1)$$

Clock Synchronization Method

Broadcast method

RBS Clock Synchronization and Its Improvement

(a) Overview of RBS methods

RBS is the most commonly used clock synchronization method in distributed sensors. The linear drift method is used to calculate the clock drift between two nodes. The clock crystal oscillator is used to adjust the time, so that the drift value is close to zero, achieving more accurate Clock synchronization. The specific operation is that without affecting the normal oscillation, the experiment adopts the method of measuring the frequency

directly from the crystal oscillator to measure the interruption period, and through continuous adjustment, an accurate interruption period is obtained, and then the true frequency of the crystal oscillator is reversed. Its mathematical model can be expressed as:

$$h(t) = \omega(t)d\tau + h(t) \quad (2)$$

The frequency is defined as the first derivative of the clock and is used to represent the timing rate of the clock. The frequency of the ideal clock is always 1, and the crystal frequency of the actual clock will fluctuate with changes in external environmental factors such as temperature, humidity, pressure, supply voltage, and electromagnetic interference. Frequency offset is time synchronization other than the uncertainty of message transmission delay algorithm error source and is expressed as:

$$\rho(t_h) = \frac{d_h h_{th}}{d_h t_h} \quad (3)$$

Under the premise of constant temperature, power supply load, and negligible stability, an accurate crystal frequency can be obtained, and multiple experiments are performed to compensate the crystal oscillator, so that the drift between nodes tends to zero. At the same time, according to the local timestamp between two nodes, the average relative offset is calculated, and the local clock of one of the nodes is corrected as a standard, so that the offset approaches zero. The RBS method principle is shown in figure 2. The reference node sends messages to node a and node b at the same time to record the standard clock. At the same time, node b sends a message to node a, and then synchronizes the clock of node a with node b. In order to verify whether the real clock synchronization is achieved, node a sends a synchronization message to node b again, which ingeniously excludes sending. The effect of the clock on the clock synchronization ensures that true clock synchronization is achieved.

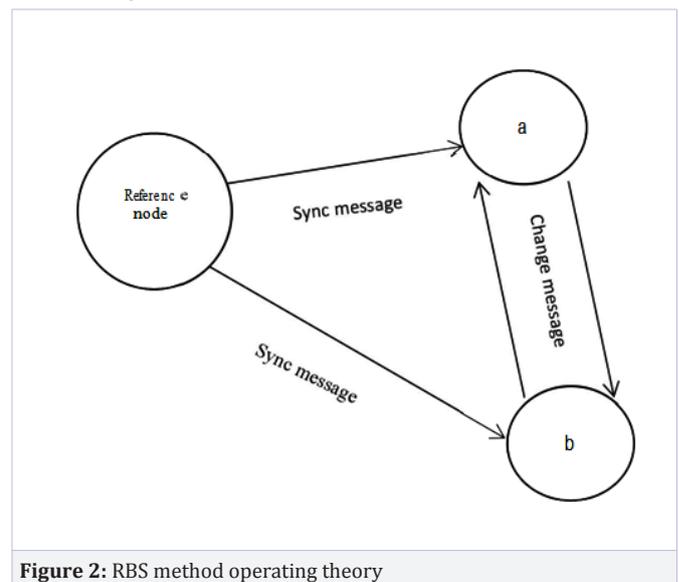


Figure 2: RBS method operating theory

(b) RBS method improvement

The data error due to using the RBS method is larger; many improvements have been made to the RBS method in recent years. For example, the RBS method combined with least squares method is to use the least squares method to identify the singularity point and effectively delete the singularity pointing the sample which is sensitive to the singularity point by the fault error or the sample point does not fall within the scope of the experimental design. It can effectively identify and delete singular points, avoid the existence of a large number of singularities and affect the true distribution of experimental data. Due to the traditional least squares method is easy to cause the clock synchronization convergence speed is too slow, it is proposed to combine the improved least squares method with the RBS method to accurately record the drift and deviation between the local clock, and constantly make the inter-node drift value tend to zero [20].

The RBS synchronization method has a requirement that the local clock of the synchronization node is subject to drift and offset compensation. The least squares method uses a linear fitting operation method to perform a fitting operation on the sampled time point set to obtain the node local clock relative to the reference. Node clock drift rate and clock offset value. Least squares method has the advantages of small data size, but individual singularities have a great influence on the calculation results. In the experimental process, there is no guarantee that there is no singularity. Therefore, an improved least squares method is proposed to calculate the clock. Drift

and deviation to improve the speed of clock synchronization. The specific methods are as follows: First, the data is imported; the data difference between the point and the next sample point is calculated. Second, it needs to determine whether this difference is positive. If it is negative, the data is a singular point, and the singularity point will be removed again. Repeat the above steps. If it is a positive number, use the least squares method to calculate. This network environment is the synchronous clock of the RBS algorithm. The RBS method is irreversible during the running process. Therefore, combining the least squares method and the RBS algorithm is beneficial to removing the singularity points and can truly reflect the real situation of the sample data.

DMTS Method and Its Improvement

(a) Method Overview

The DMTS method based on delay measurement becomes the most popular clock synchronization method for a long time because of its simple principle, low complexity and energy saving. DMTS is a unidirectional clock synchronization method based on the sender model. In order to avoid the processing delay of the sending node and the MAC layer access delay, the sending node adds a timestamp t_0 to the broadcast packet. Assume that the length of the sent message is N_A bits, and the time consumption of each bit is t , the recipient records the local time t_1 after receiving the synchronization, and record the time t_2 before adjusting its own local clock record. At this time, in order to achieve clock synchronization with the sending node, the receiver can adjust the time of the receiving node to $t_0 + t * N_A + (t_2 - t_1)$ shown in figure 3.

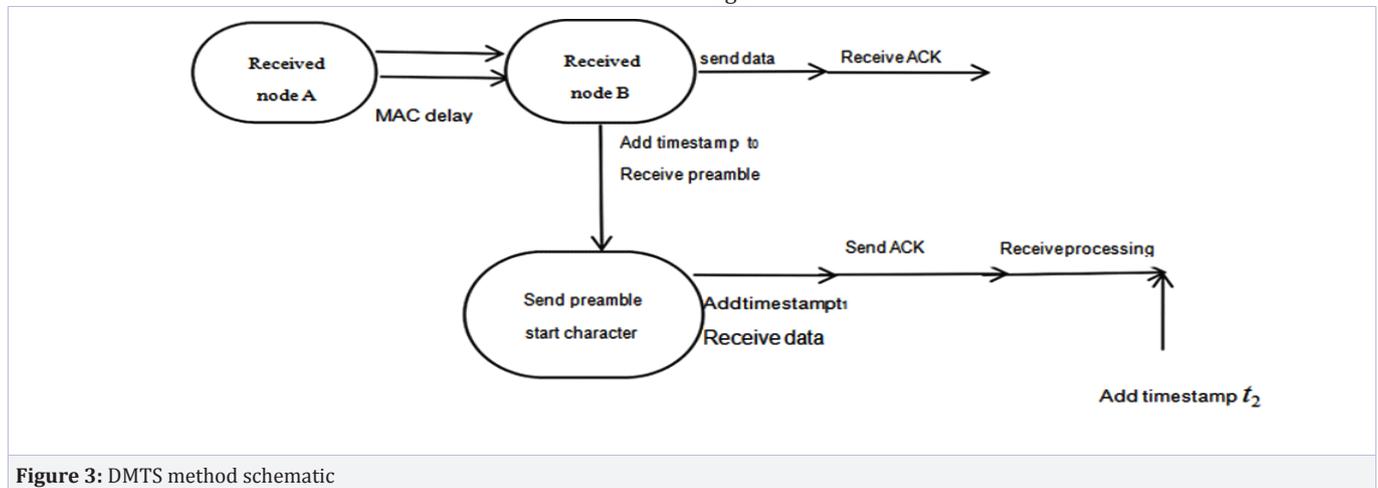


Figure 3: DMTS method schematic

(b) Method Optimization

Due to the DTMS method compromises the energy consumption and the synchronization accuracy, the synchronization accuracy is weak, and it is only suitable for wireless sensor networks with less precise requirements. Therefore, scholars in related fields have used DMTS method and TPSN method in recent years. Combine the advantages of both and optimize the traditional DMTS method. (Figure 4)

The specific process is shown in Fig. 4. There are two parts of the path in this method. The main path is a path consisting of three routing nodes A, B and C. The secondary path is composed of node B and two sensor nodes D and E. Subnetwork 1, and subnetwork 2 consisting of node C and two sensor nodes F and G. This improved DTMS method uses the TPSN method for the primary path of the router, and uses the DMTS method for the secondary path. This makes the algorithm simpler, the accuracy is greatly improved, and the network consumption is greatly reduced [21].

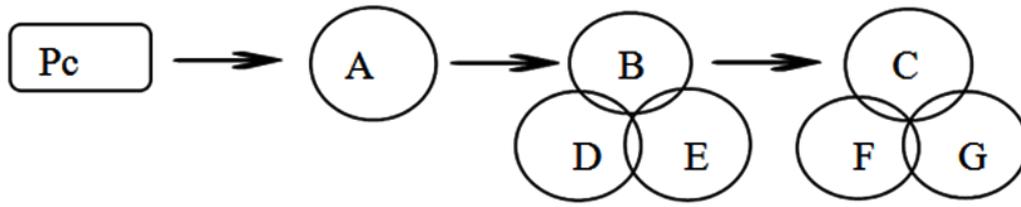


Figure 4: Fusion method experiment node graph

The specific implementation method is that on the primary path, the node A first sends a command to specify the node B as its next hop synchronization node. The number of node A is 0, and the ID of the node A is recorded and sent to the node B. When the node B receives the signal of the node A, it records its own layer number as 1, and the parent node ID is A, and adjust their own clock and node A clock synchronization, and so on, node B will send the group packet to the node C, and make node C as the next hop synchronization node, node C receives the signal, write down their own hierarchy number 2, and clock synchronization with the A node.

On the secondary path, the accuracy of the method is lower, so there is no major limitation on the period, and longer periods can be selected for synchronization to reduce network consumption [22]. In the sub-network formed by the router node B, the sensor nodes D and E, the next-hop nodes designated by the node B are nodes D and E. After receiving the signals, the D and E nodes modify their own local time to reach the nodes. In synchronization with B, the sub-network formed by router node C and F and G also uses the same method to achieve time synchronization.

Pairwise Synchronization Method

Bayesian theory combined with TPSN optimization method

(a) Method Overview

The TPSN proposer implements the TPsn and RBs mechanisms on the Berkeley Mica platform. For a pair of Mica nodes with a clock of 4 MHz, the TPSN time synchronization average error is 16.9 microseconds, while the RBS is 29.13 microseconds [23]. The TPSN method and the traditional NTP protocol use two handshakes to exchange time stamps to achieve clock synchronization. Compared with the RBS method, the TPSN method has a higher accuracy difference. The synchronization process is shown in figure 5. There are two nodes, node A to be synchronized and a reference node B. The node A to be synchronized sends synchronization request to the reference node B and records the local clock T1. Node B records the local time T2 after receiving the request. After a period of time, a synchronous reply message is sent, and the clock at the time T3 is recorded. The recorded and transmitted data T2 and T3 are transmitted to the node A to be synchronized. The node A receives the reply message and records the time T4. At this point, assuming that the fixed transmission delay and the relative drift are 1, the following formula can be obtained: the time delay of the

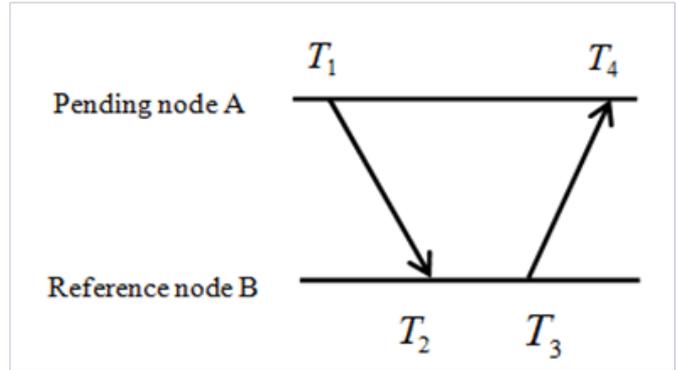


Figure 5: TPSN method node diagram

round trip d and the time offset θ , adjust the own clock, plus the time deviation, we can sum up: $t_1 = t_1 + \theta$, in which T_1 is the local clock of node A to be synchronized [24].

$$d = \frac{[(T_2 - T_1) + (T_4 - T_3)]}{2} \tag{4}$$

$$\theta = \frac{[(T_2 - T_1) - (T_4 - T_3)]}{2} \tag{5}$$

(b) Method Optimization

The TPSN method records time stamps for the sending and receiving signals of the computer. In this method, the synchronization error of any node depends on the number of hops from the root node and is irrelevant to the total nodes, so the TPSN method can be used for large networks. This method has good scalability, but when the path length grows, the errors caused by clock frequency instability and delays in information exchange between synchronous nodes will gather and affect the accuracy of the algorithm to a large extent. Therefore, we recently combined Bayesian estimation with the TPSN method to reduce the synchronization error through Bayesian estimation and optimize the TPSN method [25].

Bayesian estimation is a method that takes an unknown parameter α as a random variable of known distribution $\pi(\alpha)$, thereby mathematically prioritizing and using prior information. It will be called the prior probability, given the known sample space D, and the prior probability $\pi(\alpha)$ can be corrected to obtain the posterior probability. $P(\alpha/d)$ This can be rolled out:

$$P(\alpha/d) = \frac{P(\alpha/d)P(\alpha)}{P(\alpha)} = \frac{P(\alpha/d)P(\alpha)}{\int P(\alpha/\Phi)P(\Phi)d(\Phi)} \tag{6}$$

The posterior probability $P(\alpha_D)$ was calculated from Bayesian estimation and likelihood functions, on the basis of the prior probability, the preconditions are added, and the probability that the correction is passed after the new condition is obtained is a result that is closer to the actual situation.

In the TPSN method, node synchronization is achieved through progressive message exchanges, which inevitably leads to propagation delays. Therefore, synchronization errors are also generated. In multi-hop networks, synchronization errors are greater, and the TPSN method combined with Bayesian estimation can reduce Transfer delays in message exchanges improve time synchronization accuracy.

The error is generally divided into two types, deterministic and non-deterministic. The deterministic error refers to the clock drift rate and the clock deviation. The linear slope refers to the clock drift rate. The intercept of the straight line on the ordinate represents the clock skew of the two nodes. And the non-deterministic portion is caused by the delay in the message delivery. It can be seen as a normal distribution with an average value of 0. The optimization algorithm needs to solve is to gradually reduce the error caused by the message delivery delay, so as to improve Synchronization accuracy.

Improved Tiny-sync Method

(a) Method Overview

In the clock synchronization method, the Tiny-sync method is a common one. The method uses two-way synchronization between the nodes for multiple times, and the synchronization results are linearly fitted to calculate the frequency offset and phase offset between the two nodes according to the formula $T = \omega_i t + \theta_i$ can get the clock drift and deviation between two nodes.

In the method, node j sends a message to another node i. The time for sending the message is T_1 , and then the node i records its own clock T_2 . At this time, the message sent by node j is received, a message is returned, and node j receives and receives the node i. The news and record the time T_3 at this time.

Tiny-sync method is based on these three time data points to limit the value, the algorithm can continue to obtain data points through the exchange of messages to in order to improve the accuracy of the algorithm.

(b) Method Optimization

In order to prevent accidental deletion of useful data by the Tiny-sync method, the Mini-sync method was summarized and the principle of the Mini-sync method was consistent with the Tiny-sync method [26]. It was only based on the original design that a standard was ensured. Data deleted by the Tiny-sync method is definitely not needed. Due to the abundance of data points, these two methods have the following advantages over other time synchronization algorithms: high precision, strong fault tolerance and high degree of use, but the Mini-sync method has a long convergence time and a large overhead. They are not suitable for many occasions [27]. In order to test whether

there is noise data in the Tiny-sync method, we have done many simulation experiments. In the simulation experiment, the data in [27] is used. The experimental results show that as the sample data increases, the error rate gradually increases. When the sample data reaches 1500, the error rate is as high as 100%. Therefore, the use of the Tiny-sync method is also limited by the environment. Because in the Tiny-sync method, there is a large change in the range of data to be used, the error rate rises, and noise data is generated. In a huge data, each data has noise data and it will have a large impact on the final result. Deviation, therefore, the scholars in the near future have integrated the curve fitting method, and proposed an improved Tiny-sync method, which expands the application scope of the Tiny-sync method.

This improved Tiny-sync method is mainly to reduce the error rate in a noisy data environment, so an ITS method that is not affected by a noisy data environment is proposed. Using the same data for simulation analysis, the data shows that no matter how the data volume changes, the error rate of the ITS method is zero, the data sample is from 200-1500, or even higher, and the error rate is still zero. This is mainly due to the data considered by the ITS method. As a general trend, the impact of individual data points on the results has been minimized.

Using the data in [27] to compare the accuracy of the two algorithms, the conclusion is shown in figure 6: When the data amount is less than 1000, the curve representing the ITS method fluctuates around the curve of the Tiny-sync method and is consistent with the accuracy of the Tiny-sync method. When the amount of data is greater than 1000, the curve of the ITS method approaches the X axis, and sometimes the ordinate of the method accuracy curve is equal to zero. Therefore, it can be concluded that the ITS method and the Tiny-sync method. The accuracy is basically consistent [28].

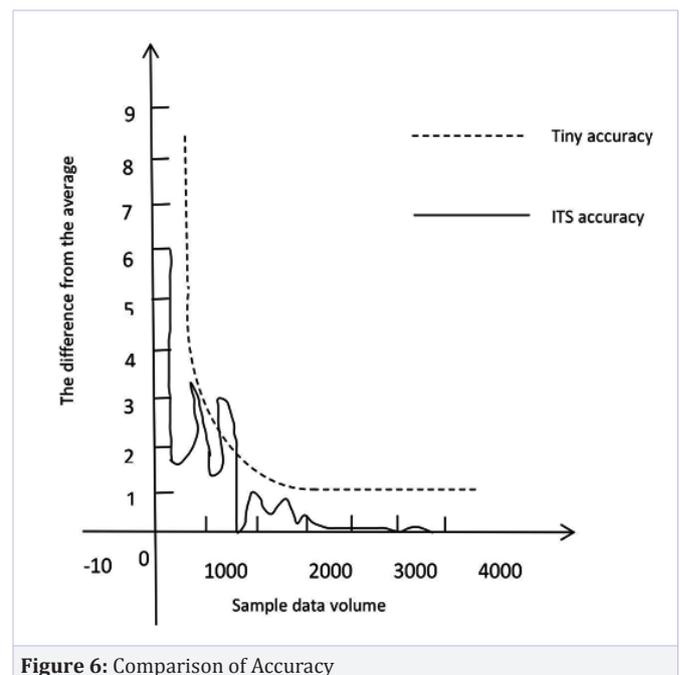


Figure 6: Comparison of Accuracy

LTS Method

(a) Method Overview

In wireless sensor networks, LTS is a more conservative clock synchronization method. It is a wireless sensor clock synchronization method that saves energy by sacrificing accuracy, and is also a relatively simple method. The LTS method is mainly used for global clock synchronization and is based on bidirectional pair synchronization. It proposes a distributed synchronization algorithm and a centralized synchronization method. Therefore, the LTS method is more suitable for environments where the accuracy is within the specified accuracy and the complexity is small.

The LTS method is divided into two parts. The first part is the centralized synchronization method. The centralized synchronization method needs to construct a spanning tree. Firstly, single-hop synchronization is performed along the edge of the spanning tree. There is a linear relationship between the number of pair synchronizations and the number of edges. The root of the spanning tree serves as a reference point and synchronization is sometimes required again. The time reception, transmission, and synchronization of the entire synchronous network are affected by the depth of the spanning tree. Therefore, any information of the spanning tree cannot be missed. This information should be analyzed and utilized.

The second part is the distributed synchronization method. In the distributed synchronization method, the clock synchronization is decided by each node. The spanning tree is only used in the centralized synchronization method. When one of the nodes needs to be synchronized, the node will send a synchronization message to the nearest reference node from the reference node. All nodes between the nodes to be synchronized must complete clock synchronization before the nodes to be synchronized.

The biggest advantage of the LTS method is that it saves network overhead, can save network resources to a great extent, and avoid network resource consumption caused by communication with other upper-layer nodes. However, the LTS method is of general precision and can only be used for accuracy requirements. Small environment.

Comparison

The clock synchronization methods of the above five kinds of wireless sensors cover the optimization application of the relatively basic clock synchronization method at present, and some existing clock synchronization optimization methods are also based on this improvement. There are many requirements in wireless sensor networks, and various applications have different characteristics. In order to meet these requirements, we need to compare different methods and understand their advantages and disadvantages. Therefore, through summary and experimentation; summary is shown in Table 1:

Table 1: Comparison of performance of various time synchronization methods

Agreement Type	Improved RBS Method	Bayesian and TPSN Methods	Combining DMTS and TPSN Methods	Improved Tiny-Sync Method	LTS Method
The way of synchronization	As required	Continuous	As required(Continuous)	Continuous	Continuous
Algorithm classification	Receiver-Receiver	Sender-Receiver	Sender-Receiver	Sender-Receiver	Sender-Receiver
Synchronization range	Section	Full network	Full network	Full network	Full network
Accuracy	500 μs	560 μs	550 μs	370 μs	260 μs
Overhead	Higher	Lower	Lower	Lower	Low
Algorithm convergence speed	Fast	General	Slower	General	Slow
Algorithm complexity	0.6	1.2	0.3	0.67	0.4
Energy utilization	36%	50%	79%	53%	68%
Fault Tolerance	Stronger	General	General	Strong	Weaker
Robustness	High	Higher	Higher	Higher	General

The improved RBS method has better handling of singular points than traditional RBS methods. The traditional RBS method has high precision, but it has lower processing power for odd-numbered points [29]. It is prone to the following errors: accuracy errors due to crystal frequency differences, accuracy errors due to deviations in data at the receiving node, beacon packets propagate in the medium. The improved RBS

method improves the convergence speed of the method on the basis of removing singular points. This makes the method more accurate and more robust and fault-tolerant than other improved methods. Therefore, the improved RBS method can be used in environments with high accuracy requirements.

The TPSN method has high accuracy and does not become

weaker as the number of nodes increases. However, as the length of the path and the number of hops increase, the clock frequency becomes unstable, and the error caused by the delay between the information of each pair of synchronous nodes is added. It will have a significant impact on clock synchronization accuracy. Under this premise, combined with Bayesian estimation, the problem of unstable precision can be solved by reducing the delay of message exchange, but the problem of failure of the root node is still not solved. Therefore, fault tolerance is generally applicable to low-cost and low-precision wireless sensor networks.

The complexity of the algorithm formed by the combination of the DMTS method with the lower accuracy of the algorithm and the higher precision TPSN method becomes lower. The accuracy of the algorithm is lower than that of the TPSN method but the energy consumption is greatly reduced. The structure is divided into main and sub-paths, the division of labor is clear, the structure is simple, and the robustness is relatively high. In general, the combination of the two is a complement to each other. It is applicable to small and medium-sized wireless sensor networks with low precision requirements, and can be used in systems such as access control and time attendance [30].

The Tiny-sync method is widely used as one of the classic wireless sensor clock synchronization methods, but due to its own algorithm, the noise data processing ability is poor so the influence of noise data cannot be avoided. Due to its low energy consumption and low accuracy, the improved Tiny-sync method places special emphasis on noise data. It has a very effective mitigation of noise data processing and effectively eliminates noise data, thereby alleviating the effect of noise data on algorithm accuracy. However, the accuracy of the improved Tiny-sync method is still very low. Using the data in [31], after several simulation experiments, it can be concluded that the delay from one node to another must be at least 3.08ms, the round-trip delay is 25.217ms, and the delay of the message probe transmission and return obeys the mean of 3.536. With a standard deviation of

1.312 ms, accuracy and energy consumption are lower than those of other methods, and the root node failure cannot be adjusted. Therefore, it is suitable for small and medium-sized wireless sensor networks that require low precision and high energy consumption [32].

The LTS method is divided into two methods: a centralized method and a distributed method. The centralized method is similar to the TPSN method. The spanning tree affects the accuracy of the algorithm. In the distributed method, only synchronization with the parent node is required, which greatly saves network overhead, saves network resources, and avoids resource waste caused by overly complicated network processes. Medium complexity is minimized, but it also results in less accurate methods. Compared with other improved methods, each performance is poor, which is suitable for small and medium-sized wireless sensor networks with low precision requirements, low complexity, and high energy consumption requirements.

By comparison, it can be seen that TPSN!LTS and TS use algorithms such as S to construct a hierarchical tree using flooding broadcast, and then use message bi-directional exchange model to synchronize time between upper and lower nodes [33]. The TPSN uses the time-stamping method at the MAC layer to achieve a time synchronization accuracy of 16.9 microseconds. Therefore, when designing an algorithm, the MAC layer can be used to mark the timestamp to improve the synchronization accuracy [34]. The RBS algorithm has good synchronization precision in short-distance single-hop, but it is not suitable for multi-hop synchronization. For clustering networks, this method can be considered to synchronize intra-cluster nodes." TSync algorithm adopts dual-channel the method combines the broadcast, synchronization mechanism and the message bidirectional exchange model algorithm, and the communication overhead is small. But it assumes that each node knows its underlying neighboring node ID number, which is a problem that needs to be solved in the actual synchronization [35]. "For some applications

Table 2: Comparison of patterns of various time synchronization methods

Algorithm	MAC Layer Timestamp	Synchronous Mode	Routing Method	Clock Model	Time Conversion Model
Improved RBS Method	×	r-r	Overlapping clusters	Fixed frequency offset	Linear fitting
Bayesian and TPSN Methods	√	s-r	Spanning Tree	Frequency deviation is zero	Bidirectional phase difference-Delay estimate
Combining DMTS and TPSN Methods	√	s-r	Spanning Tree	Frequency deviation is zero	Bidirectional phase difference-Delay estimate
Improved Tiny-Sync Method	×	s-r	Spanning Tree	Fixed frequency offset	Linear fitting
LTS Method	×	s-r	Spanning Tree	Fixed frequency offset	Linear fitting

that do not require high precision, but require low computational complexity, consider using DMTS. The algorithm “according to different application environments, it can also use a combination of methods to synchronize the network [36]. Under a variety of criteria, no one algorithm is optimal “For different application requirements can choose from a variety of tradeoffs to the right algorithm” For different design goals, each algorithm has better than other algorithms Can learn from each other to design a better synchronization algorithm.

In addition, the summary of other aspects is shown in Table 2:

The summary of existing theories and techniques can lay a good foundation for carrying out relevant research. There are no specific descriptions of the implementation details of various existing wireless sensor network time synchronization algorithms [37]. Instead, the time synchronization process is divided into three modules: “local time stamp”, “synchronous message exchange” and “reference time estimation”. The theoretical models and synchronization techniques associated with them. In the actual algorithm design, according to the specific situation of the wireless sensor network and the user’s synchronization requirements, the existing synchronization theories and technologies should be flexibly used and expanded, and an efficient time synchronization algorithm should be developed [38].

Future Outlook

Through the analysis and research on the improved methods of the above-mentioned several typical methods, it is found that each clock synchronization method has its advantages and disadvantages in the case of different application environments or applications, and the improved method is improved on the basis of the original traditional methods. The weaknesses can be weakened and the advantages can be more prominent. The development of wireless sensor network clock synchronization has been relatively mature from immature to nowadays. Constantly researching clock synchronization protocols applied in specific environments is the result of continuous learning and innovation by dome [39].

The research of the synchronization technology in the future wireless sensor network is a more comprehensive process. So far there is no more comprehensive and complete wireless sensor method, which requires each algorithm to learn from each other, through the mutual integration to draw a new more comprehensive method. Should make full use of data fusion technology, from multiple angles to explore the clock synchronization method optimization program, such as combining the maximum likelihood method, convex optimization methods to improve the accuracy of the algorithm, hierarchical operation to achieve cost savings [40]. For hardware, large-scale hardware is still not perfect. It depends on advanced technology and a powerful hardware platform [41]. Now domestic and foreign scholars have less research on this piece, so they need continuous improvement and innovation [42]. These will also be the research hotspots of future wireless sensor networks.

References

1. Kang GL, Wang FB, Duan YJ. Overview of time synchronization in wireless sensor networks. *Computer Measurement & Control*. 2005;13(10):1021-1023.
2. Fang X, Fu C, Chen GL. Network Time Synchronization Algorithm and Its Implementation. *Computer Engineering and Applications*. 1992;(3):23-29.
3. Gao AM. Analysis and solution of short wave timing failure in time synchronization system, *Telecommunication Technology*, 2014;54(7):946-948.
4. Feng BQ, Xu JM, Li B. Research on Computer Time Synchronization. *Journal of Xi’an Jiaotong University*. 1999;(05):106-107+112.
5. Mallat SG. A theory for multi resolution signal decomposition: The wavelet representation. *IEEE Trans PAMI*. 1989;11(7):674-693.
6. He P, Wu HT. Research and Application of Time Synchronization Algorithm for Distributed Systems. *Journal of Computer Applications*, 2001;(12):20-21+24.
7. Cristian F. Probabilistic Clock Synchronization. *Distributed Computing*. 1989;(3):146-158.
8. Bullock JB, King TM. Test results and analysis of a low cost core GPS receiver for time transfer application. *Proceedings of IEEE International Symposium on Frequency Control*. 1998:273-277.
9. He HB. Research design of GPS-based high precision time synchronization system. Chengdu: Sichuan University. 2005.
10. Thomas C. Real-Time Restitution of GPS Time and Through a Kalman Estimation. *Metrologia*. 2004;29(6):397-414.
11. Pan AM, Wang GY, David J. *Visual C++ Technology Insider (Fourth Edition)*. Tsinghua University Press. 2004;4(1):725-747.
12. Chen M. Research and Implementation of Network Time Synchronization System Based on NTP Protocol. Wuhan: Huazhong University of Science and Technology. 2005.
13. Li Z, Li RF, Wei YH. Time synchronization and ranging cooperative algorithm in wireless sensor networks. *Computer Research and Development*. 2010;47(4):638-644.
14. Sun LM, Li JZ, Chen Y, Zhu HS. *Wireless Sensor Networks*. Beijing: Tsinghua University Press. 2005.
15. Shen MY, Ai ZX. Research on Low Energy Time Synchronization of Wireless Sensor Networks. *Computer Engineering and Applications*. 2012;48(8):112-115.
16. Yu JA, Chen XH, Wang WD. A low overhead time synchronization algorithm for wireless sensor networks. *Computer Simulation*. 2009;26(1):124-129.
17. Dong SP, Chen W. Fault-tolerant time synchronization protocol algorithm. *Sensors and Microsystems*. 2017;36(08):149-151.
18. Ganeriwala S, Kumar R, Srivastava MB. Time-sync protocol for sensor networked. *Proc of 1st International Conference on Embedded Networked Sensor Systems*. 2003:138-149.
19. Yu SG. Integration of GPS Network and Wireless Sensor Network. *Network Communications and Information Technology*. 2015;(2):59-61.
20. Yan AB, Liu WY, Shi YL, Guan YM. RBS time synchronization algorithm based on improved least square method. *Automation Instrumentation*. 2015;36(7):8-11.

21. Yang ZK, Zhao DS, Wang YM, Cheng WQ, He IH. An Overview of Clock Synchronization Algorithms for Wireless Sensor Networks. *Computer Applications*. 2005;(05):1170-1172+1176.
22. Han CH, Li LH, Zhao EB. Synchronization Algorithm in Wireless Sensor Networks. *Information Security and Communications Security*. 2005;(05):54-56.
23. Yang CM. Research on Time Synchronization Algorithm for Wireless Sensor Networks. University of Science and Technology of China. 2008.
24. Xia XF, Yan BY, Liu FY, Lu W. Optimization Algorithm of Wireless Sensor Network TPSN. *Computer Measurement & Control*. 2010;18(6):1465-1467.
25. Fang HY, Cui XX, Liu Y. A Survey of Positioning Problems in Wireless Sensor Networks. *Computer and Information Technology*. 2005;(06):1-6+54.
26. Zhou XW, Wei W, Yan BP. Research on Time Synchronization Algorithm for Wireless Sensor Networks. *Journal of Transducer Technology*. 2006;(01):20-25+29.
27. Sichertiu ML, Veerarittiphan C. Simple accurate time synchronization for wireless sensor networks. *IEEE Wireless communications and Networking Conference*. 2003.
28. Ye JM, Wang ZY, Cao Y, Zhao H. LTS state tree generation algorithm based on CHAM model. *Journal of Harbin Engineering University*. 2003;(03):287-291.
29. Jun Y. Research on low energy consumption time synchronization algorithm for wireless sensor networks. Wuhu: Anhui University of Engineering.
30. Peng Y, Wang D. An Overview of Wireless Sensor Network Positioning Technology. *Journal of Electronic Measurement and Instrument*. 2011;25(05):389-399.
31. Sichertiu ML, Veerarittiphan C. Simple, accurate time synchronization for wireless sensor networks. *Wireless Communications & Networking, Wcnc*. 2003;2:1266-1273.
32. Tian XZ, Chen CJ, Yan JJ. An Improved Tiny-Sync Algorithm. *Computer Systems& Applications*. 2009;18(04):81-84.
33. Su W, Akyildiz IF. Time-Diffusion Synchronization Protocol for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*. 2005;13(2):384-397.
34. Kim H, Kim D, Yoo S. Cluster-based Hierarchical Time Synchronization for Multi-hoP Wireless Sensor Networks. *Proceeding of the 20th International Conference on Advanced Information Networking and Applications (AINA.06)*. 2006;2:318-322.
35. Ramanathan P, Shin KG, Butler RW. Fault Tolerant Clock Synchronization in Distributed Systems. *IEEE Computer*. 1990;23(10):33-42.
36. Su W, Akyildiz IF. Time-Diffusion Synchronization Protocol for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*. 2005;13(2):384-397.
37. Pang Y, Li S, Gong QC. An Overview of Wireless Sensor Network Time Synchronization Technology. *Computer Applications and Software*. 2016;33(12):1-5.
38. Xue SH. Development and Application of Time Synchronization Network Technology. *Informatization Construction*. 2016;(03):99.
39. Burrell J, Brooke T, Beckwith R. Vineyard computing: sensor networks in agricultural production. *IEEE Pervasive Computing*. 2004;3(1):38-45.
40. Mainwaring A, Culler D, Polastre J, et al. Wireless sensor networks for habitat monitoring. *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*. 2002:88-97.
41. Xiang L. Implementation of Mobile Node Access in Mobile-Wireless Sensor Networks. *Information Security and Communication Security*. 2005;(5):390-393.
42. Yang CM. Research on Time Synchronization Algorithm for Wireless Sensor Networks. University of Science and Technology of China. 2008.