Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Survey of Time Synchronization Protocols in Wireless Sensor Networks

Manish Kumar, Katyayni Rastogi, Aparna Tiwari, Apoorvi Dwivedi
Department of Computer Science Engineering, Galgotias College of Engineering and Technology
Greater Noida (U.P.), India

***Abstract*— Sensors are tiny battery operated devices which detect and respond to the changes in the environment. Wireless sensor networks are a collection of wirelessly connected sensors and are used in real world. The aggregated result of a network is produced by analyzing the local results of all the nodes. The nodes communicate through sending messages among each other. Hence time synchronization is an important aspect of wireless networks. In this paper, we study the different existing time synchronization protocols and compare them on different criteria to prove the need for a new protocol which is scalable, secure, energy efficient, fast convergent and less complex.**

Keywords*— Wireless sensor networks, time synchronization, clock offset, message delay,

I. INTRODUCTION

In the recent times, the number of low-cost sensors capable of wireless sensing and processing has been increased significantly. These sensors can be used in dangerous and inaccessible areas to provide results that would be very difficult for a centralized system to produce. The sensors are scattered within the network and are mobile. So they need to be configured in a communication network. The result by the network is generated by aggregating the individual results sensed by each sensor node. This fusion of results is possible only by exchanging messages that are time stamped by each sensor's local clock [3]. This requires that each sensor node is synchronized with every other node. Hence time synchronization becomes very important.

Examples of existing sensor network applications where precise time is needed include: integrating multi-sensor data, coordinating on future action, distributing an acoustic beam forming array, suppressing redundant messages by detecting duplicate detections of the same event by different sensors [1] etc. Hence, synchronization of nodes such that they give a correct aggregate result after data fusion is very important. This requires that the sensors must work according to some common notion of time. Protocols that provide such synchronization are called synchronization protocols. Various time synchronization protocols are defined such as RBS, TPSN, FTSP etc [2]. In this study, we try to compare different synchronization protocols based on the selected parameters. We will show that a particular protocol is better in some criteria while the other is better in some other. The different parameters we have selected to measure the performance of a protocol are efficiency, accuracy, energy usage, scalability etc.

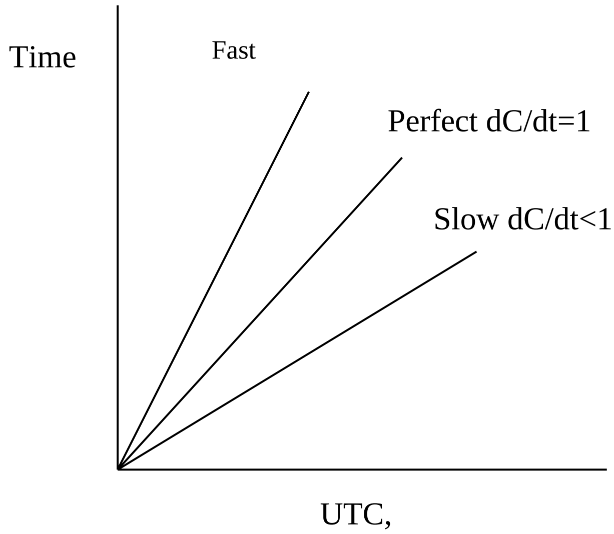
The outline of the paper is as follows: In section II we describe the problem formulation and discuss the clock system used in distributed systems. Section III will explore the existing solutions for the time synchronization problem. Here we discuss various sender-receiver and receiver-receiver protocols. The next section IV will compare these solutions. In the final section V, we conclude our study and discuss its future scope.

II. PROBLEM FORMULATION

In a distributed system, there are several nodes and each node has its local clock i.e. it will record the time of message arrival and sending according to its local clock. Also there is no global clock in the system. Hence, the local clocks of all the nodes cannot be synchronized using some common value. This presents a big problem as there can be clock offsets and drifts between different nodes and thus a problem occurs during communication. This problem is known as synchronization problem. To have a clear understanding of the synchronization problem, let us understand the clock system of the nodes.

A. General Clock Model

The term software usually refers to the computer clock to emphasize that it is only a counter, which is increased due to the quartz oscillator. When each interrupt occurs, the interrupt handler of the software clock must increment the oscillator by one. [3]. Even if the frequency deviation of the oscillator is 0.001%, it can lead to errors in a day about 1 second. Consider the local clock time C (t) for a node. For the perfect clock, dC (t) / dt is equal to 1. This is called clock skew. However, due to environmental factors such as temperature and humidity, the value of the clock skew will change. [3]



For fast clock, $dC / dt > 1$, and for slow clock $dC / dt <1$.

The following terms are commonly used in a generic clock system [1]:

• *Clock offset*: Offset is the difference between the global time and the actual time reported by a clock. This value of a clock is given by Ca (t) -t.

• *Skew*: Deviation is the difference in clock frequency and perfect clock frequency. At time t, the skew of clock C(b) with respect to clock C(a) is the (C'a (t) -C'b (t)). If skew is restricted by δ, the clock is allowed to the value 1-δ to 1 + δ divergence.

• *Drift*: This clock drift of the clock value is the second derivative with respect to time, namely C"a (t).

B. Requirement for a Time synchronization protocol

A good time synchronization protocol for WSN must comply and trade-off the following requirements: low-cost, accurate, precise, secure and periodically-scheduled [1, 2].

An ideal time synchronization protocol should be able to satisfy the given problems. Firstly, the sensor nodes are battery operated devices. Sending and receiving messages consume a lot of energy. Hence, synchronization protocols should be able to synchronize the nodes with exchanging only a minimum number of messages. Secondly, the synchronization should be precise to a microsecond level because a small error may cause a lot of problems in the network. Thirdly, the protocol should guarantee that the synchronization between two nodes will remain precise even after successive re-synchronizations. All these requirements must be fulfilled so that an error free aggregated output can be produced by the network.

III. TIME SYNCHRONIZATION PROTOCOLS

A number of protocols for time synchronization have been proposed in the WSN. These can be divided into two types: sender receiver protocol and receiver - receiver protocol. Synchronization takes place using the timestamp of the message and the delay in the message.

- *Receiver-Receiver Protocol*: In this type, any sender sends a message to one or more recipients, who are then synchronized with each other. Here, the sender does not participate in the synchronization process.
- *Sender - Receiver Protocol:* In this type, the sender sends a message to the recipient and the recipient sends an acknowledgment to the sender. Therefore, the transmitter now calculates the delay between the local clocks of the transmitter and receiver nodes and synchronizes them.

We have a number of time synchronization protocols available for wireless sensor networks. But here we focus on three main protocols: the Reference Based Synchronization Protocol (RBS), the Time Synchronization Based Protocol (TPSN), and the Flood Time Synchronization Protocol (FTSP).

A. Reference Based Protocol

The Reference Based Protocol (RBS) was proposed by Elson and his team. It is an example of the receiver-receiver protocol. The fundamental property of RBS is that a broadcast message is only use to synchronize a set of receivers with one another.[3]

The protocol is based on a idea that a third party will broadcast a synchronization message to all the receivers. This broadcasted message is not time stamped; the receivers will synchronize themselves by comparing the time at which they received the message and calculate their phase offsets with respect to each other. The synchronization is done by calculating the offset between the receivers.

The most common form of RBS can be understood with the help of two receivers (Node X and Node Y) and a sender. The sender sends a synchronization message to both the receivers. The message will be received by both the receivers at the same time. But they record the receiving time of the message according to their local clock. Now Node X and Node Y will exchange messages between them. This message contains the time when the synchronization message was received by them respectively. Assuming the message should be received by both the nodes at the same time, the receivers calculate their clock offsets with each other.

The Reference Broadcast Synchronization (RBS) protocol utilizes the concept of broadcast nature of wireless communication. According to this property, two receivers located within listening distance of the same sender will receive the same message at approximately the same time.[4] It is based on the assumption that when a message is sent from a node to two different nodes they will reach the physical layers of the receivers at approximately same time. Upon receiving the reference message by the sender both the nodes note down their local times and exchange it between them. Now both the nodes have the times when the message was received by the other node. Thus they compare the local times and calculate their clock offset based on that. Hence, the nodes are able to synchronize with their neighbouring nodes with high precision. Similarly, this reference message is sent to all the nodes of the network which then send messages to their neighbours and synchronization is done.

RBS eliminates the uncertainty of the sender by removing it from the critical path. The only uncertainties are the propagation and receive time. We can compare the critical path of traditional protocols and RBS.[4]

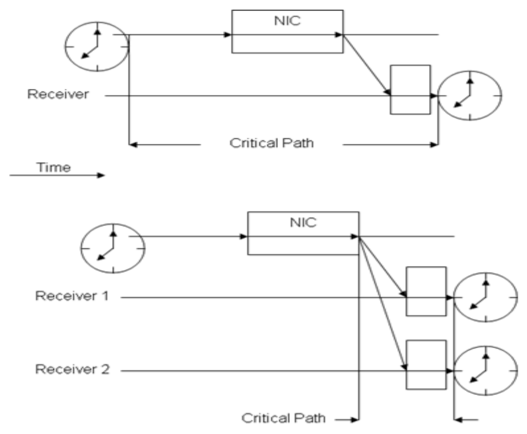


Fig 2. Comparison of critical paths of sender-receiver and RBS

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

B. Time Synchronization Protocol for Sensor Networks(TPSN)

TPSN is an example of sender-receiver based synchronization. TPSN is based on a tree where one node is elected as the root node and other nodes are synchronized with the root. There is a root discovery phase where the root is elected. Any node that has the GPS equipped may be selected to be the root otherwise any random node may be selected as the root node. Also, periodically root may be re-elected to reduce the burden on the root node. There are two phases in the TPSN protocol: level-discovery phase and the synchronization phase. In the first phase, root is elected and based on the distance with the root, all the other nodes are assigned levels. In the synchronization phase, actual synchronization is done by exchanging messages with the nodes with the next level nodes.

- *Level Discovery Phase:* In this phase, all the nodes are assigned a level. The root resides at level 0.[5] The neighbouring nodes of root are at level 1 and so on. After the discovery of the root node, the root will initiate assigning levels to all the other nodes. The root sends a *level_discovery* packet to its neighbours which reside on level 1. This packet contains the id and level of the sender. All the neighbours receiving this message will then assign themselves the level 1 and they further forward the *level_discovery* packet with their id and level to their neighbours respectively. This process is repeated until all the nodes have received a *level_discovery* packet and are assigned a level.
- *Synchronization Phase:* This phase is based on a two-way communication between the nodes. Similar to the level discovery phase, it starts with the root node and propagates to the other nodes of the network.[3]

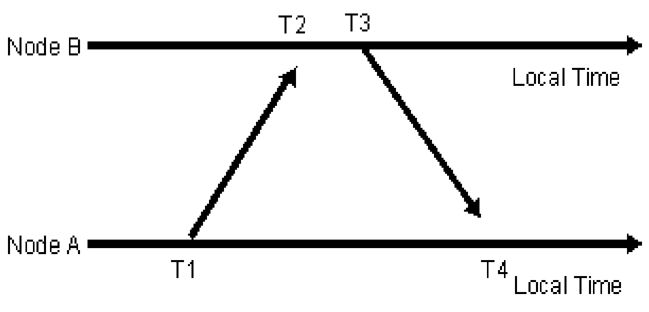


Fig 3: Two way communication between nodes

The figure shows the synchronization between nodes A and B.T1 and T4 are the time measured by the local clock of node A, and T2 and T3 represent the time measured by the local clock of the node B. When node A sends a message to node B, the synchronization phase begins. Time T1. This message contains the level of node A and time T1. Node B receives the packet in T2, where T2 is equal to T1 + D + d. Here, D and d represent the clock drift and propagation time between the two nodes, respectively. At time T3, B returns an acknowledgment packet to A. The confirmation packet contains the values of level B and T1, T2 and T3. Node A receives the packet at T4. Assuming that the clock drift and propagation delay do not change during this short time interval, A can calculate the clock drift and propagation delay as follows [4].

$$\text{Drift} = ((T2-T1)-(T4-T3))/2$$
$$\text{Delay} = ((T2-T1)+(T4-T3))/2$$

Any synchronization packet is associated with four delays: transmission time, access time, propagation time, and reception time. Eliminating one of these delays would be a good thing. Unlike RBS, TPSN does not completely eliminate the transmission time. But this can reduce it. In addition, the TPSN is designed as a multi-hop protocol; therefore, the transmission range is not an issue. The main disadvantage of the TPSN is that the root node represents an extra load because it starts and controls all operations.

C. Flooding Time Synchronization Protocol(FTSP)

FTSP is another example of transceiver synchronization. It is similar to TPSN because it is also based on a tree topology where all other nodes are synchronized with the root node. [1]
Synchronization begins when the root node sends a synchronization message to all participating nodes. The message contains the sender's timestamp indicating the global time of the transfer [4]. When the recipient receives the message, he will check the local time. This time is called the time of the recipient. The recipient now has twice the link to the sender who sent the message and the time the message was received. It now uses two values to calculate the clock offset. The message is time stamped on the MAC layer. For example, in the TPSN, there are timestamps on both the transmitting side and the receiving side. [3] In order to maintain high precision, it is necessary to compensate for clock drift. FTSP uses linear regression.

The FTSP is designed for large multi-hop networks. Here, root is dynamically elected and will be re-elected periodically to minimize the load on a single node.

FTSP is very powerful because it uses a lot of synchronization messages to combat link and node failures. Flooding also allows dynamic modification of the topology. The agreement states that the root node will be re-elected on a regular basis. Therefore a dynamic topology is required. Like TPSN, FTSP also provides MAC timestamps to greatly improve accuracy and reduce jitter. In addition to propagation delay errors, this will eliminate everything. It uses multiple timestamps and linear regression to estimate drift and clock skew.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

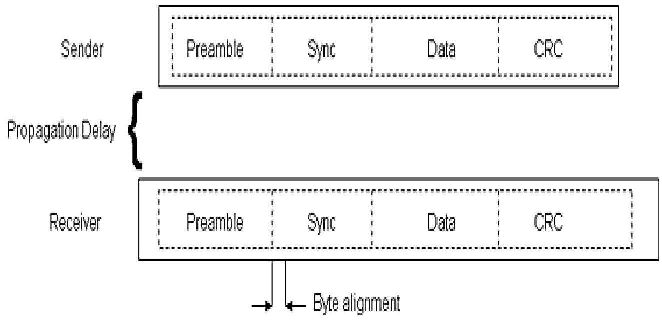


Fig 4: Packet transmitted with FTSP

IV. A COMPARATIVE STUDY

To compare the performance of different protocols, we studied them under different situations and evaluated their performances on the basis of selected criteria.

The system model for the evaluation was as follows:

- The network consists of n number of nodes which are connected wirelessly.
- Each node has its own local clock
- There is a root node and other nodes are on different levels than the root node. Root is at level 0.
- Each node can send messages to its neighbouring nodes. Nodes cannot send messages directly to the distant nodes. They require to send it through other nodes.
- The range for delays was chosen with distribution over the range of 10 to 600.

From studying the previous works on time synchronization protocols, we can say that FTSP is a good multi-hop protocol but its efficiency decreases as the number of hops increases[3]. FTSP shows better results as compared to the other protocols but it is efficient to use this protocol when the number of hops is less. TPSN has better efficiency for large number of node but it increases the message load on the network since there are two phases and each phase require messages to be sent. Root discovery phase of TPSN generates more messages. As a result energy consumption of TPSN is more than FTSP. RBS is a receiver-receiver based protocol. It is relatively simple and accurate than the other two protocols but the main disadvantage of RBS is that it is not a multi hop protocol. Also RBS is very vulnerable to the pulse-delay attack.

The detailed analysis of the protocols under different parameters is given below;

A. *Number of Messages*

The number of messages denotes the number of messages exchanged between the nodes for synchronization in the network. It is maximum for TPSN as there are two phases. In the first phase the root sends the level discovery message to all the other nodes while in the second phase there is two way communication between the nodes to achieve synchronization among them.[5].

In the FTSP protocol, root is elected dynamically and periodically, so there is no root discovery phase. Hence number of message exchanges required is less. In RBS, number of messages exchanged is more than FTSP but less than TPSN. It is because a sender will send beacon message to the different nodes which will then synchronize by exchanging their timestamps.

B. *Synchronization Error Time/Accuracy*

The smaller the error between the two clocks, the more accurate they are [3]. The accuracy of the RBS system increases as the resynchronization period increases. FTSP keeps its clock with as little error as possible; therefore, FTSP performs better than other protocols. Due to the small number of hops in our simulation, the accuracy of the FTSP is not greatly affected. As the number of hops in a multi-hop sensor network increases, the accuracy of the FTSP decreases [4]. FTSP provides good accuracy in models with fewer nodes. As the resynchronization increases, the accuracy of the RBS is improved. The TPSN gives the average accuracy.

C. *Energy Consumption*

Since sensors run on battery, it is very important for the synchronization protocols to be energy efficient. We assume that when a message is exchanged in a network, its energy decreases. Hence energy consumed in a network can be linked to the number of messages exchanged among the nodes. FTSP consumes least energy as it requires only one message to be sent in the network. RBS requires more energy than FTSP due to its nature of receiver-receiver based synchronization. TPSN consumes the most energy as number of messages exchanged is very high.

After comparison on so many parameters the results can be summarized in the following table:

TABLE I
RESULTS OF COMPARISON AMONG EXISTING PROTOCOLS

Criteria	Time Synchronization Protocols		
	RBS	TPSN	FTSP
Year	2002	2003	2004
Nature	Receiver-Receiver Synchronization	Sender-Receiver Synchronization	Sender-Receiver Synchronization
Clock correction	No	Yes	Yes
Energy efficiency	High	Medium	High
Mobility	No	No	Yes
Complexity	High	Low	High
Accuracy	Low	High	High
Multi-hop	No	Yes	Yes
Scalability	Poor	Good	Average

V. CONCLUSION AND FUTURE SCOPE

A study of the most commonly used time synchronization protocols is made and they are compared on various parameters. The main outcome of the study is that no existing protocol is best in all the parameters. Wireless network size is increasing daily to cater to the needs of the complex applications. Hence time synchronization protocols must be made scalable enough so that they are able to synchronize a significant amount of nodes since the node size may grow upto thousands in number. FTSP is a good protocol and the same is proved in many studies but its accuracy and efficiency declines when the number of hops is increased. Hence it is important to make progress in a time synchronization protocol which combines the advantages of the existing protocols and eliminates their shortcomings. We are hopeful that the data presented in the paper will be useful for the future work in the field of time synchronization.

ACKNOWLEDGMENT

We are thankful to our department faculty for providing us the opportunity to work with them and to guide us to conduct this study. We are also thankful to the authors of the papers which we have used in our study. This paper is only a comparison of the synchronization protocols and we thank all the study papers which have been published on these protocols.

REFERENCES

[1] Ill-Keun Rhee ,Jaehan Lee, Jangsub Kim ,Erchin Serpedin and Yik-Chung Wu *Clock Synchronization in Wireless Sensor Networks: An Overview,* 2009

[2] Salim el khediri1, Nejah Nasri1, Mounir Samet,Anne Wei and Abdennaceur Kachouri *Analysis Study of Time Synchronization Protocols in Wireless Sensor Networks,* 2009

[3] Amit Nayyar, Meenakshi Nayyar and Lalit Lumar Awasthi *A comparative study of Time Synchronization Protocols in Wireless Sensor Networks* in International Journalof Computer Applications,2011

[4] Prakash Ranganathan,and Kendall Nygard *Time Synchronization in Wireless Sensor Networks:A Survey* in International Journal of Obicomp,2010

[5] Saurabh Ganeriwal , Ram Kumar and Mani B. Srivastava *Timing Sync Protocol for Sensor Networks,*2011

[6] Hill, J.; Horton, M.; Kling, R.; Krishnamurthy, L. *The platforms enabling wireless sensor networks.* Commun. ACM 2004, 6, 41-46.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, *Wireless sensor networks: a survey, Computer Networks*, The International Journal of Computer and Telecommunications Networking, Vol.38, No.4, pp.393422, 15 March 2002.

[8] Miklos Maroti , Branislav Kusy , Gyula Simon , Akos Ledeczi, *The flooding time synchronization protocol,* Proceedings of the 2nd international conference on Embedded networked sensor systems, November 03-05, 2004, Baltimore,

[9] Chafika Benzaid ,Amin Saiah and Nadjib Badache *Secure Pairwise Broadcast Time Synchronization in Wireless Sensor Networks* IEEE 2011.