International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 2, No. 2, 2015, pp. 249-257.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

Recent Time Synchronization Protocol's in Wireless Sensor Networks

Seyed Kazem Kazeminezhad[a]

[a] *Islamic Azad University of Tabriz*

Abstract

Time synchronization is essential for any kind of network due to network software requirements. Since WSNs have particular constraints and limitations'; synchronizing the physical times for these networks is considered to be a complex task. In this paper we study newly presented time synchronization protocols in wireless sensor networks. There are some synchronous and asynchronous methods. Some are based on remote clock reading and others are based on source clock frequency recovery. Energy consumption is critical in WSN. Minimizing the number of message transmissions from sensor nodes to the head node or from the head to the source nods is the best way to save energy. We will discuss some of the methods of time synchronization in WSN. We will discuss TFTS, ISTS, Algorithm Based on SFD, Double-Broadcasting for WSN and Indirect-Broadcasting Synchronization Protocol for WSN. Each method has its certain characteristics such as power consumption and accuracy and so on. Study results shows that each network should have its own operational time synchronization protocol due to its own needs. In this paper, the researcher studies recent time synchronization protocols.

Keywords: WSN, Power Consumption, TSP, Non Deterministic

Introduction:

The Time synchronization is one of critical components in WSN operation, as it provides a common time frame among different nodes. It supports functions such as fusing data from different sensor nodes, time-based channel sharing and media access control (MAC) protocols, and coordinated sleep wake-up node scheduling mechanisms [1].

Recent developments in the field of micro-electro-mechanic systems, wireless communication and digital electronics have simplified developing cheap, low consumption and tiny sensors which are capable of multi task monitoring and communication functions in uncontrolled environments. These tiny sensors are able to sense, process data and transmit information to the sink[8-11].

Synchronization, as an emerging phenomenon of a population of dynamically interacting units, has fascinated humans from ancestral times. Synchronization processes are ubiquitous in nature and play a very important role in many different contexts such as biology, ecology, climatology, sociology, technology, or even in arts [13-15].

- **Master–slave versus peer-to-peer synchronization**

  In Master–slave synchronization A master–slave protocol assigns one node as the master and the other nodes as slaves. The slave nodes consider the local clock reading of the master as the reference time and attempt to synchronize with the master. In general, the master node requires CPU resources proportional to the number of slaves, and nodes with powerful processors or lighter loads are assigned to be the master node[17,18].In peer-to-peer synchronization Any node can communicate directly with every other node in the network. This eliminates the risk of master node failure, which would prevent further synchronization. Peer-to-peer configurations offer more flexibility but they are also more difficult to control[16,19-21]

- **Clock correction versus untethered clocks**

  Most methods in practice perform synchronization by correcting the local clock in each node to run on par with a global timescale or an atomic clock, which is used to provide a convenient reference time. The local clocks of nodes that participate in the network are corrected either instantaneously or continually to keep the entire network synchronized[17,18].

- **Internal synchronization versus external synchronization**

  In Internal synchronization approach, a global time base, called real-time, is not available from within the system and the goal is to minimize the maximum difference between the readings of local clocks of the sensors. [17]

In external synchronization, a standard source of time such as Universal Time (UTC) is provided. Here, we do not need a global time base since we have an atomic clock that provides actual real-world time, usually called reference time. The local clocks of sensors seek to adjust to this reference time in order to be synchronized. Protocols like NTP[12] synchronize in this fashion because external synchronization is better suited to loosely coupled networks like the Internet. Most protocols in sensor networks do not perform external synchronization unless the application demands it, because energy efficiency is a primary concern and employing an external time source typically induces high-energy requirements. Internal synchronization usually leads to a more correct operation of the system, while external synchronization is primarily used to give users convenient reference time information. Note that internal synchronization can be performed in a master–slave or peer-to-peer fashion. External synchronization cannot be performed in a peer-to-peer fashion; it requires a master node which communicates with a time service like GPS to synchronize the slaves and itself to the reference time. [17]

- **Probabilistic versus deterministic synchronization**.

The former provides a probabilistic guarantee on the maximum clock offset with a failure probability that can be bounded or determined. But, the latter guarantees an upper bound on the clock offset with certainty.

- **Sender-to-receiver versus receiver-to-receiver synchronization**

Receiver-to-receiver lets a reference node transmit once and other nodes exchange messages independently of the reference node. Reference Broadcast Synchronization (RBS) [4], which is in this category, lets a sender send a broadcast beacon for receivers' reference, and receivers except the sender participate in synchronization. RBS increases the accuracy by eliminating sender side's delay uncertainty even though it can't transmit exact reference time (or global clock). *Sender-to-reciever* indicates that one node send a message while the other receive it. Timing-sync Protocol for Sensor Networks (TPSN) [5] and Flooding Time Synchronization Protocol (FTSP)[6] are classified into this synchronization category which can transmit global reference clock through the network. It is a contrast to RBS. Receiver Only approach was added later than previous two methods in [3] and Pair-wise Broadcast Synchronization (PBS) [7], where a group of nodes are synchronized by listening to the message exchanges of a pair of nodes.

Power consumption model:

We assume a simple model for radio hardware energy dissipation, where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics, as shown in Figure 1.

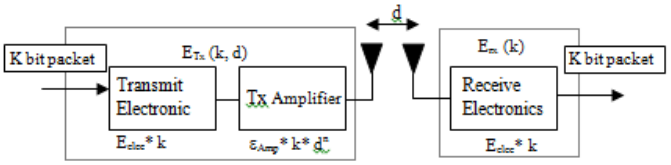

Figure 1. Radio energy dissipation model

For the experiments described here, both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models were used, depending on the distance between the transmitter and receiver [43].Thus, the energy consumption for transmitting a packet of *l* bits over distance d is calculated by equation 7.

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\varepsilon_{amp}d^4 & d \geq d_0 \end{cases} \quad (14)$$

Power control can be used to invert this loss by appropriately setting the power amplifier. If the distance is less than a threshold d_o, calculated by equation 8, the free space (*fs*) model is used; otherwise, the multi-path (*mp*) model is used.

$$d_0 = \sqrt{\frac{\varepsilon fs}{\varepsilon mp}} \quad (15)$$

Energy consumption to receive a packet of *l* bits is calculated according to equation 9.

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \quad (16)$$

Recently Presented Methods:

TFTS:

A Novel Triple Factor Time Synchronization for Effective Routing in Large Scale WSN. This method presents 3 different algorithms for solving various issues of time synchronization. While majority of the studies previous have adopted the assumption of propagation delay and noise etc. On a uniform topology, we choose to perform our experiments in both random as well as uniform topology considering majority of the real-time constraints pertaining to large scale wireless sensor network. Although our assumptions for Hypothetical Time Synchronization algorithm is purely impractical, still it is recommended to use it for preliminary analysis of the synchronization errors for obtaining some of the real-time error evaluation. For precise evaluation, Empirical Time Synchronization Algorithm is recommended, while for better evaluation, it is recommended to use our Optimized Time Synchronization Algorithm.[2]

The outcome of the study is compared with the most recent and standard work to find that the proposed system has outperformed the existing one with minimal time synchronization errors with respect to synchronization period. However, there are few constraints and limitation of the proposed system as it was not evaluated how the proposed time synchronization algorithm has its influence on energy depletion as well as routing performance. Hence, our future direction of work will be focused on establishing a relationship between the proposed time synchronization algorithms with energy factor as well as extended routing performance in large scale wireless sensor network. Moreover, as majority of the existing literatures uses synchronization error as the performance parameter, this becomes easier and justified to compare proposed TFTS with some existing literatures. The proposed system performs evaluation of scalability by considering number of transmission occurring on increasing numbers of nodes. With nodes increasing its size, the number of communications and control messages too increases, which should lead to more complexities to time synchronicity. Hence, the proposed system uses numbers of transmissions with increasing number of sensor nodes are the scalability parameters to evaluate effectiveness of time synchronous in wireless sensor network. Usually, effective time synchronous operation is highly demanded in large scale wireless sensor network owing to the scalability issues in it. In the area of the wireless sensor network, it is required that there are no false alarm being raised while capturing the events by the sensor nodes and such fault tolerant factors can be ensured by minimized observation of synchronous error computed in seconds unit.[2]

Intragroup Secure Time Synchronization (ISTS):

This proposal is based on RBS protocol commented on section 2. Our objective is that our algorithm exchanges fewer timing messages between sensor nodes, which means saving energy. RBS protocol requires a super node, which broadcasts a time reference message, so it is a centralized protocol. In contrast, in our proposal all nodes send a Secure Intragroup Time Synchronization Technique 413 messages about timing information. So, there is not a super node which manages the synchronization. We propose a receiver-receiver synchronization scheme, where each message exchanged between nodes inside a group is encrypted. The encryption keys are explained in the previous section. Each sensor node encrypts its message with its key and any node of logic branch that joins with central node can decode this message. Moreover, all nodes can decode messages encrypted by central node. For example, sensor node in Fig. with the key (Kg(4,1)) encrypts a message. This message can be decoded by all nodes which connect it to the central node. In contrast, when this message arrives to this one and it should be encrypted with its own key. [20]

WSN Time Synchronization Algorithm Based on SFD:

This method uses a third-party node to verify the accuracy of the two sensor nodes which have been time synchronized. The advantage of using third party is to avoid the SFD delay between sender and receiver. After the two nodes synchronized, the third-party node sends a abroad message to the two synchronized nodes. And the two receive the message at the same time, showing their local time stored in SFD register, and then record the time difference between the two nodes. The process is shown in Fig.5. To stat 50 experimental results, we found that the average of the two synchronized nodes' time difference is about 5.82 microseconds. The statistical results are based on the above results, also we can see that, in 50 of the experimental results, the min error is 4.84 microseconds, and the max error is 10.25

microseconds. The number of error which is less than 6 microseconds accounts for 72% of the total number. Therefore, the experimental results are stable at 4 to 6 microseconds. By analyzing the synchronization error is caused by the change of crystal frequency and the difference of the crystal frequency between the nodes. Also there is SFD delay between the sender and receiver, and now we do not know the value of SFD delay. If we can get the value of SFD delay, we will eliminate the SFD delay. The above time synchronization algorithm is used for two nodes. For the whole network, you can use the same principle to synchronize. At first we can make the nodes networking, and then sink node send time synchronization messages to all its son nodes. After the son nodes synchronized, they send time synchronization messages to their son nodes, and finally achieve the whole network time synchronization. Through the above experiment data and experimental results, it is proved that WSN time synchronization algorithm based on SFD in this paper has been achieved. Time synchronization error is about 5.82 microseconds, and it has realized higher precision time synchronization. Compared with other traditional WSN time synchronization method, for example, TPSN and RBS, the algorithm proposed in this paper has the advantages of simpler, more accurate. Precision of 11 microseconds has been reached using RBS on Berkeley mote, which is a popular platform. TPSN achieves 16.9 microseconds for the same experimental setup. TPSN and RBS both algorithms were applied on Mica platform. RBS, which is receiver-receiver algorithm, needs 4 messages sent and 3 messages received to synchronize two nodes. TPSN needs 2 messages sent and 2 messages received to synchronize two nodes. But the algorithm proposed in this paper needs only 1 message sent and 1 message received, and it is not depend on some platform, so it is simpler. The precision of it is about 5.82 microseconds less then TPSN and RBS, so it is more accurate. Above all, the algorithm proposed in this paper is better than the traditional WSN time synchronization method[22].

Time Synchronization by Double-Broadcasting for Wireless Sensor Networks:

DBS uses a different approach to synchronize and measure delay other than existing schemes. Synchronization is closely related to delay. So, measuring exact delay leads to successful synchronization. However, accurate estimation of delay is very hard because it varied unexpectedly. In DBS, a node, called Leader node, broadcast a packet too all receivers within the same hop to notify the reference time point and measure arriving time of the packet it sent by itself. The arriving time of the packet is almost the same as the arriving ones at the other receivers if it is assumed that propagation time among near nodes is small enough to be ignored and receive delays at all the node in a hop are the same. A leader node can measure forward delay from the arrival and the send time of a broadcast packet easily, which is different from estimated one with round trip delay used in the exiting schemes [25], [5].

DBS informs multiple receivers of synchronization information by double broadcasting by itself, instead of exchanging messages with receivers. The number of messages exchanged and synchronization convergence time are minimized and accuracy of delay estimation iss improved by measuring forward delay directly, not the round trip delay.

DBS Operation: Single-hop Operation. DBS scheme has 2 phases; Setup and Synchronization Phase .Setup phase constructs cluster hierarchy using a clustering algorithm. It is not the purpose of this paper to propose efficient clustering algorithm other than a new method of synchronization. Some clustering algorithms showing different performance have been proposed. Construction of the least

clusters in a network results in better performance for synchronization. If a cluster tree having the least hop number is made by a good clustering algorithm it can be used for this scheme. A simple clustering algorithm [5] is used h ere. It is assumed that clusters be constructed when sensor nodes are deployed and need not to be changed frequently .In Synchronization phase time synchronization for nodes in a network is performed .Synchronization phase is performed with two steps. In first step, a leader node broadcasts a probe packet, called Ready packet, to all nodes within a range and measure some parameters. In second step the leader node broadcasts time information Ready Time and estimated delay) to all nodes within a range. The broadcasts in the first and second step have different purposes and packet formats.[23]

A leader node at first step broadcasts a Ready packet to all nodes within a range. All nodes including a leader node in a range record the time called Ready Time when Ready packet arrives .All Ready Times at all nodes are considered to be the same instant, assumed that transit delay from the leader node to other nodes are the same and the propagation delays are ignorable because distance among nodes in communication range is small. In addition, the leader node can estimates forward delay with send time and receive time .Ready packet contains {sender address: sequence #: level: send time .{In the second step the leader node broadcasts Go packet containing time information to all nodes within a range. The format of Go packet is {sender address: sequence level: Ready Time: delay estimate: send _time}. All nodes except the leader node calculate 2 types of offsets with time information, called "Offsets" and "Offsetd ."Offsets is calculated from Ready time at its node, Ready time of the leader node in Fig. 3), Offsets = T2'' – T2. Offsetd is calculated from packet's arrival time, the leader node's send time and delay estimation of the leader node, Offsetd = T4'' – destimate - T3 .Delay estimations are calculated by weight-averaging method to smooth the fluctuations of sample delay. Either Offsetd or Offsets is used. Offsets may be used usually because that Offsets is expected to be more accurate than Offsetd [23].

Enhanced Indirect-Broadcasting Synchronization Protocol for WSN:

This method proposes enhanced version of IBS (called EIBS) which saves whole energy and prolongs network lifetime by re-constructing partial cluster tree locally. This scheme provides the capability of tree reconstruction algorithm which is not only simpler, but also more efficient in the light of overall power consumption and network lifetime, compared with other cluster construction approaches. The remainder of this method is organized as follows. Surveying existing clustering schemes proposes cluster-based time synchronization scheme with the capability of partial cluster reconstructing[24].

Conclusion:

So many synchronization protocols are presented till now. They are designed and presented for different goals and operations in different ways. For time synchronization protocols in WSN, constraints of energy as well as synchronization accuracy should be considered, especially. Time synchronization protocols are presented in many details. But the specifications of each are different due to their usage. Some act better in wired networks and others provide better performance in wireless sensor networks. No algorithm has been presented to do the exact time synchronizations because of non-deterministic errors. In this paper we tried to discuss newly presented methods for those who aim to use TSPs for WSN in their projects. As future works we will merge some of the methods and try to eliminate non-deterministic errors to achieve a new and exact method.

References:

Akyildiz, I. F., Su,W., Sankarasubramaniam,Y., Cayirci, E. (2002).Wireless sensor networks: a survey, jornul of computer network, Vol38pp 393-422.

and in Space, 2nd edn. CRC Press (2011).

Cristian, F., (1989). Probabilistic clock synchronization Distributed computing3, 148-153.

D. Estrin, R. Govindan, J. Heidemann, S. Kumar.(1999). Next century challenges:scalable coordination in sensor networks, ACM MobiCom99, Washingtion, USA, pp.263-270.

Elson, J., et al.: Fine-grained network time synchronization using reference broadcasts. In:ACM OSDI (2002)

Ganeriwal, S., et al.: Timing-Synch Protocol for Sensor Networks. In: ACM Sensys, USA (2003)

K. Nagarathna, Jayashree D. Mallapur in WirelessPersonal Communications (2015)

Li, Q., Rus, D.,(2004). Global clock synchronization in sensor networks, in proceedings of IEEE Conference on computer communications, Vol. 1, Hong Kong, china,pp. 564-574.

Maróti, M., et al.: The Flooding Time Synchronization Protocol. In: ACM SenSys 2004 (2004)

Miguel Garcia, Diana Bri, Jaime Lloret, Pascal Lorenz in Wireless Networks and Security (2013).

Mill, D.: Computer Network Time Synchronization: the Network Time Protocol on Earth

Mock, M., Frings, R., Nett, E., Trikaliotis, S., (2000). Continuous Clock Synchronization in Wireless real-time applications, in proceedings of 19th IEEE symposium on reliable distributed system, pp. 125-133.

Noh, K., et al.: A New Approach for Time Synchronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization. IEEE Tran. on Wireless Comm. 7(9) (September2008)

Osipov, G.V., Kurths, J., Zhou, C.,(2007). Synchronization in oscillatory networks, Springer, Berlin, Germany.

PalChaudhun, S., Saha, A., Johnson, D. B., (2003). Probabilistic clock synchronization service in sensor networks, Technical Report TR 03-418, Department of computer science, Rice University.

Paradis Lilia, Han Qi, 2007, Fault Management in Wireless Sensor Networks: a survey. Journal of networks and systems management, Vol. 15, PP. 171-190

Pikovsky, Rosenblum, A., Kurths M. J.,(2001). Synchronization, Cambridge University Press, Cambridge, UK.

Ping, S., (2003). Delay measurement time synchronization forwireless sensor networks, Intel Research, IRB-TR-03-013.

Rhee, I., et al.: Clock Synchronization in Wireless Sensor Networks_ An Overview:Sensors, 9 (2009), http://www.mdpi.com/journal/sensors

Shanshan Song, Lili He, Yu Jiang, Chengquan Hu… in Advances in Wireless Sensor Networks (2013).

Shi-Kyu Bae in Embedded and Multimedia Computing Technology and Service (2012).

Shi-Kyu Bae in Ubiquitous Information Technologies and Applications (2014).

Stojmenovic, I., (2005 by John Wiley & Sons). Handbook of Sensor Networks Algorithm And Architectures, Handbook, University of Ottawa.

Sundararaman, B., Buy, U. and D.Kshemkalyani, A., (2005). Clock Synchronization for Wireless Sensor Networks: a survey, ELSEVIER, ad hoc network3, 281-323.

Yick J, Mukherjee B, Ghosal D. Wireless Sensor Network Survey. Comput Netw 2008; 52:2292-2330.

Yik-Chung Wu et al., "Clock synchronization of wireless sensor networks," IEEE Signal Process. Mag., vol.28, no.1, pp.124-138, Jan. 2011.