Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.1062 – 1068

SURVEY ARTICLE

Analysis of Time Synchronization Protocols for Wireless Sensor Networks: A Survey

Gopi Krishna Yadav[1], Prof. Awadhesh Kumar[2], Akash Raghuvanshi[3]

[1]Research Scholar, M.Tech(CSE), Kamla Nehru Institute of Technology, Sultanpur, India

[2]Associate Professor CSE Department, Kamla Nehru Institute of Technology, Sultanpur, India

[3]Research Scholar, M.Tech(CSE), Kamla Nehru Institute of Technology, Sultanpur, India

[1] gopi.yadav2010@gmail.com, [2] awadheshkumar.knit@gmail.com, [3] akash.raghuvanshi@gmail.com

Abstract: – The time synchronization is a major problem for Wireless sensor networks. Wireless sensor networks have been changing and enhancing the way people perceive environment and technology. Many applications of sensor networks required local clocks to be synchronized for precision but sensor networks required different properties. In order to get fast and on-time response for different applications, it is necessary that wireless sensor nodes must co-operate with one other. The most important requirement to maintain the co-ordination among different sensor nodes is time synchronization. This paper surveys and analysis the existing clock synchronization protocols based on factors like precision, cost, accuracy and complexity. Also this paper compares the qualitative and quantitative evaluation for these WSN protocols. Finally, the survey provides a valuable framework by which researchers can compare new and existing synchronization protocols.

Keywords: - Time synchronization, Wireless sensor network, sensor nodes and wireless communication.

I. INTRODUCTION

A Wireless Sensor Network is a group of hundreds or thousands of sensor nodes that have capabilities of sensing the environment and communicate the information in wireless medium [2]. Wireless sensor network is the collection of sensor nodes with limited resources that collaborates in order to achieve a common goal. Sensor nodes are not only used for military applications, they have also used in *geographical monitoring, environmental monitoring and control, pollution monitoring, health and medical, target tracking , navigation, transport, emotion based computing and so on [9]***.**

To identify the correct event time, sensor nodes must be synchronized among themselves with universal time i.e. *global time*. Therefore, time synchronization is significant aspects in Wireless sensor networks. The performance of time synchronization is influenced by various factors such as *Precision, Network size, Complexity, Convergence time* and *Energy consumption***.** The limited energy resource is the drawback of the wireless sensor networks, therefore to save the energy, the nodes power consumption must be reduced, to reduce the consumption the nodes must turn their transceiver **on** and **off** at appropriate time, an accurate timing is required between the nodes[7]. Sensor nodes are very tiny instruments and running with a limited energy, so it is not easy to synchronize nodes effectively because an energy consumption. Also, they have not much processing power and this prevents any complex algorithms to run on WSN. There are three basic solutions for time synchronization in sensor networks:

1) Sender-Receiver Based Synchronization.

2) Receiver-Receiver Based Synchronization.

3) Delay Measurement Synchronization.

Receiver-Receiver based synchronization algorithms commonly use one-way message exchange such as in the Reference Broadcast Synchronization (RBS). On the other hand, Sender-Receiver based synchronization protocols used the two-way message exchange, such as the Timing-sync Protocol for Sensor Networks (TPSN). There is also some synchronization protocols based on one-way message exchange as well as the measurement of delay. Time synchronization plays a very important role because it allows the entire system to co-operate and function as a group. It is important for a many of tasks such ***as*** *event detection, data fusion* and *co-coordinating wake and sleep cycles*. Each node runs on its own local clock and sends (receives) packets to (from) other nodes at sampling rate of its own clock. Also, to conserve the power, the nodes can enter the sleep mode after sending/receiving the packet and wakeup just before its time for another packet. If the clock is not synchronized, the nodes can miss the packets sent by other nodes. The three basic methods of synchronization available in wireless sensor networks are:

1. Relative Ordering: The synchronization is based on the order of messages or events. In this method, clocks are not synchronized, only the order is maintained [2].

2. Relative Timing: In this method, a node keeps information about its offset and drift in correspondence to neighboring nodes. So, nodes have an able to synchronize its neighboring nodes.

3. Global Synchronization: In this method, All the network nodes to synchronize to this global clock.

Researchers have developed successful clock synchronization protocols for some wired networks over the few decades. These are not used in a wireless sensor network because the challenges posed by wireless sensor networks are different. The most important differences are:

1. Wireless sensor networks can contain thousands of sensors, and their wide deployment is enabled by the fact that sensors are becoming cheaper and smaller in size.
2. Self-configuration and robustness become a direct necessity in order to function under rapid deployment conditions and operation in inaccessible or dangerous environments.
3. Energy conservation is a very important concern. It is impossible to provide a power source to each sensor in such a vast network, and the small sizes of sensors restrict the amount of energy that can be stored and procured.

II. WHY NEED FOR SYNCHRONIZATION IN WSNs

There are many reasons to showing the synchronization problems in sensor networks. Some reasons are as following:

1. Sensor nodes are necessary required to co-ordinate their operations to perform a special task, e.g. Data fusion. In which data is collected at different nodes are combined into a meaningful result.
2. Life time of network is totally dependent on the power. So to increase the life of network, we need to used power saving methods. For example when using power-saving modes, the nodes should sleep and wake up at coordinated times.
3. To share the transmission medium we used scheduling algorithms like time-division multiple access (TDMA). It helps in the time domain to eliminate transmission collisions and conserve energy [2].

III. SYNCHRONIZATION PROTOCOLS FOR WSNs

There are many time synchronization protocols in wireless sensor networks. These protocols are not too much differing from each other. Today we have many protocols but basic idea behind them is always there, but improving in these protocols is doing there. Three basic protocols are Reference Broadcast Synchronization (RBS), Timing-sync Protocol for Sensor Networks (TPSN), and Flooding Time Synchronization Protocol (FTSP). Some main Protocols are as following:

Network Time Protocol: Network time protocol (NTP) is the traditional synchronization method which worked only in wired network. It is also known as Two-way message exchange in which messages are exchanged between a pair of nodes, In order to obtain a definitive relation between the two clocks with a single message exchange; two basic assumptions need to be made.

1. The offset between the clocks is constant in the small time period during the message exchange.
2. The propagation delay is the same in both directions.

***Reference Broadcast Synchronization*:** Elson et al. proposed a receiver to receiver synchronization protocol, is called Reference Broadcast Synchronization. Most of the time synchronization methods use a sender to receiver synchronization approach where sender transmits the timestamp information and the receiver receives the information and synchronize them. RBS is different because it is a receiver to receiver synchronization method. RBS is a synchronization method for a whole network. The basic synchronization idea is a reference broadcast to a set of client nodes in the one hop neighborhood of a beacon node. The idea is that a third party broadcasts a beacon to all the receiver nodes. The beacon does not contain any timing information, where the receivers compare their clocks to one another to calculate their relative phase-offsets and transmits the recorded times to each other. The timing is based on when the node receives the reference beacon. Using offset and rate difference, each client can transform a local clock reading to the timescale of other client. To extend this approach to multi-hop networks, the network is divided into clusters such that a single beacon can synchronize all nodes in its cluster. For different clusters, gateway nodes participate in two or more clusters take part in the Reference Broadcast Synchronization method to all their clusters. After knowing phase offsets and rate differences to all adjacent clusters, gateway nodes can transform timestamps from one cluster to another. In multi-hop time synchronization method, nodes send their timestamp data using their own local clocks [14]. When timestamps are exchanged among different nodes, the timestamps are transformed to the receiver's local time using phase offsets and rate difference. The sender cann't enter into the critical path because this protocol provides reciever to reciever synchronization.

In figure 1 (top of the diagram) the critical path in the traditional system, includes the sender. But in RBS, sender is removed from the critical path. The critical path contains the propagation and receiver time uncertainty. When the transmission range is small, we can eliminate the propagation time and the critical path only contains the uncertainty of the receiver.

Advantage: The main advantage of RBS is:

1. RBS is a reciever to reciever synchronization method so it eliminates the sender from the critical path. After eliminating the uncertainty of the sender, the only uncertainty is the propagation time and receive time.

2. When the range is small, the propagation time is negligible in the networks.

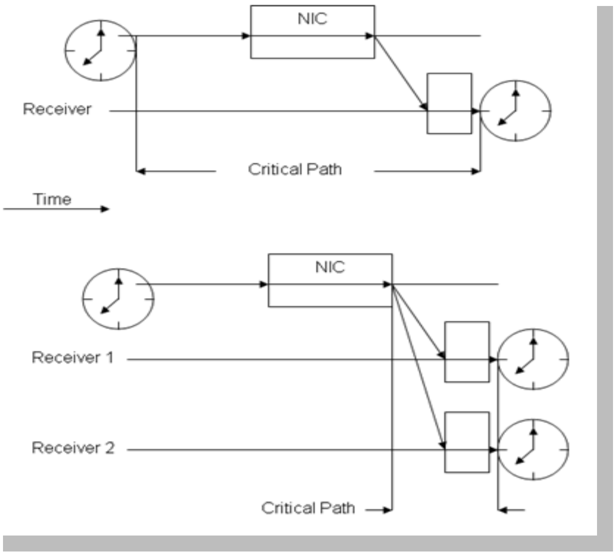

Figure 1 - Comparison of a traditional synchronization system with RBS [1]

Timing-Sync protocol for Sensor Networks (TPSN): Ganeriwal et al. proposed a time synchronization protocol for sensor networks which called the Timing-Sync Protocol for Sensor Network(TPSN). TPSN is a sender to reciever synchronization method which provides synchronization for a whole network. This protocol works in two phases:

1. Level Discovery Phase
2. Synchronization Phase

The level discovery phase creates the hierarchical topology of the network where each node assigned a level. Only one node, called root node resides on level zero. In the synchronization phase, all the i level nodes will synchronize with the i-1 level nodes, that means all nodes will synchronize with the root node.

Level Discovery Phase: In level discovery phase First, a node should be determined as the root node. In the sensor network, root node also be a sink node; and the sink may have a GPS reciever, in this case this algorithm will synchronize all nodes to time in the physical world. If sink is not available in the sensor network, then sensor nodes can periodically take over the functionality of the root node. Once the root node is determined, it will initiates the level discovery phase by broadcasting the *level discovery* packet. It contains the identity and level of the sender node. After receiving this packet, the neighbors of the root node will assigh themselves as level-1. Then each level 1 node broedcasts a *level discovery* packet with its identity and level of the sender node in the packet. After assigning a level for each node, it discards further incoming packets. This process will continue until all nodes have received the *level discovery* packet[15].

Synchronization phase: The basic concept of this phase is a two way communications between a pair of nodes. As like level discovery phase, synchronization phase is also begins at the root node and propagates through the network. The clock drift between a pair of nodes is constant in the small time period during a single message exchange and propagation delay is also assumed to be constant in both directions. Figure 2 illustrates the two way message exchange between a pair of nodes. Assume that the time T1,T2,T3 and T4 all are measured times. Node A send the synchronization packet at time T1 to node B. This packet will contain node A's level and identity and time T1 when it was sent. Node B will receive the packet at time T2. After receiving the packet, node B send the *acknowledgement packet* to node A at time T3. This packet contains the level number of node B and times T1, T2, T3. Then node A can calculate the clock drift and propagation delay and synchronize itself to B.

$$\Delta = \frac{(T1-T2) - (T4-T3)}{2}; \qquad d = \frac{(T2-T1) + (T4-T3)}{2}; \qquad [8]$$

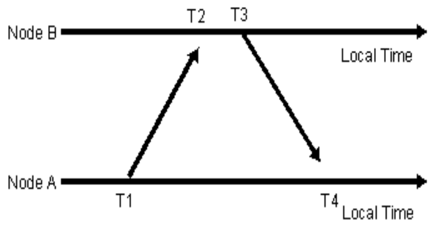

Figure 2. Two - way message exchange between a pair of nodes [1].

The synchronization process is again initiated by the root node. Root node broadcasts the *time-sync* packet to the neighbor nodes called *level 1 node*. Before initiating the two way message exchange, these nodes wait for a random amount of time. The root node sends the acknowledgement and the level 1 node adjust their clocks to be synchronized with the root node. Again, at least one level 1 node is neighbor to the level 2 node and level 2 nodes waits for a random amount of time before initiating the two ways messaging with the level 1 nodes. This process will continue until all nodes are synchronized to the root node. This communication propagates through the tree until all i-1 level nodes are synchronized with the level i nodes. At the last all nodes will be synchronized with the root node.

Advantage: **1.** TPSN is designed to be a multi-hop protocol; so transmission range is not an issue.

2. TPSN has a 2 to 1 better precision than RBS because TPSN is a sender to receiver synchronization method and it is better to receiver to receiver synchronization.

Flooding Time Synchronization Protocol (FTSP): The FTSP was designed at Vandebuilt University and implemented using Berkeley Mica2 motes. This protocol is similar to TPSN, but it works to improve the disadvantages of TPSN. It works like TPSN that it starts with a root node and that all nodes are synchronized to the root. The goal of the FTSP is to achieve a network-wide synchronization of the local clocks of the participating nodes. The FTSP synchronizes the time of the sender to multiple receivers utilizing a single radio message time- stamped at the both the sender and the receiver sides. After receiving the information, the receiver notes its local time. To achieve high precision and to keep the communication overhead low, clock drift is needed. Linear regression is used in FTSP to compensate for clock drift. FTSP provides multi-hop synchronization. The root of the network is elected dynamically called master node–maintains the global time and all other nodes synchronize their clocks to that of the rootand will organize in an ad-hoc manner to communicate the timing information among all slave nodes. In FTSP, network structure is Mesh type topology where in TPSN, it is tree topology[11].

Advantages: There are many advantages of FTSP which has improved on TPSN.

1.TPSN did not handle topology changes well but FTSP provides the ability for dynamic topology changes. The protocol specifies the root node will be periodically re-elected, so a dynamic topology is necessary.

2. Like TPSN, FTSP also provides MAC layer time stamping which increasing the accuracy and reducing jitter. It utilizes the multiple time stampings and linear regression to estimate clock drift and offset.

3. FTSP is robust in flooding of synchronization messages and node failure and MAC layer time stamping for precision are the major advantages of FTSP.

Time- Diffusion Synchronization Protocol (TDP): Time –Diffusion Synchronization Protocol is used as a network-wide time synchronization protocol. To provide the network-wide time synchronization, the time differences among the sensor nodes are need to be minimized before protocols requiring time-stamps, e.g, security applications, voicefusion, vedio fusion and environmental data fusion, are realizable. It allows the sensor networks to find an equilibrium time and maintains a small time deviation from the equilibrium time. The TDP is used to maintain the time across the whole network within a certain tolerance. The tolerance level may be adjusted based on the application of the sensor networks. The TDP enables the sink to detect the time difference between multiple sources, so that we can say that the temporal differences may be adjusted. TDP allows the sink to issue a start time to the sensor nodes allowing interactive sensing and monitoringy . The time in the sensor network reaches an equilibrium value, it still may drift overtime and has fluctuation throughout the sensor network. Although the time variation across the whole network may be small, it is necessary to translate the time in the sensor network to a common time, e.g UTC[5].

*Advantage***:** The advantage of TDP is that the performance of voice and video applications can be improved when multiple sources are sending data back to the sink through directed diffusion.

Consensus Clock Synchronization (CCS): Maggs et. Al, proposed Consensus Clock Synchronization Protocol in Wireless Sensor Networks that provides internal synchronization to a vertual consensus clock. Each node keeps a vertual clock which maps the physical clock time to a vertual consensus time generated from the vertual nodes running the CCS algorithm. The consensus clock synchronization protocol trying to synchronize to an external reference, the CCS protocol aims to achieve an internal consensus within the network on what time is, and how fast it travels. In CCS algorithm for each synchronization round, updates the compensation parameters for each node and overtime the network clocks converge to consensus. The consensus vertual clock has its own skew rate and offset relative to the physical clock. Consensus clock is a new vertual clock that is generated from the network of nodes running the CCS algorithm. To achieve these compensation paarameters in synchronization rounds, nodes repeat the CCS algorithm, are divided into two mail phases: Offset Compensation and Skew Compensation[9].

In the Offset Compensation phase, nodes exchange local clock values with each other which are used to synchronize nodes to reach to a common consensus time. The goal of offset compensation is to remove the offset error from all the clocks in the network. The skew rates of the nodes, clocks cause their offset errors to gradually increase until the next synchronization round. In Skew compensation, nodes iteratively compares the results of previous and current synchronization rounds to adjust skew compensation parameters of the vertual consensus clock. The goal of skew compensation is to ensure all compensated clocks in the network tick at the same time.

Advantage: Consensus Clock Synchronization provides the internal synchronization for the vertual clocks. This algorithm is fully decentralized and robust for node failures and mobility and there is no requirement for a reference clock.

Clustered Based Consensus Time Synchronization Algorithm (CCTS): Clustered based consensus time synchronization algorithm is the extension of the Distributed Consensus Time Synchronization algorithm. To obtain the better energy efficient and find faster convergence in the clock synchronization of the node, the clustering technique is used into the algorithm.The basic idea of CCTS is to use local informationto achieve a global agreement. This algorithm includes two parts: *Intracluster time synchronization and Intercluster time synchronization*. In the intracluster time synchronization, the improved version of distributed consensus time synchronization is used. In this algorithm, the cluster head is responsible for exchanging messages within the cluster. The average value of intracluster vertual clocks are used to update the offset compensation parameterand the average value of skew compensation parameters of intracluster vertual clock are used to update the skew compensation parameters. In the intercluster time synchronization, cluster heads exchange messages via gateway nodes. Clock compensation parameters of intracluster vertual clocks of every cluster head are assigned with corresponding weights based on the size of each cluster is used to update the clock compensation parameters of the network vertual clocks[12].

*Advantage***:** Clustered based consensus clock synchronization algorithm improves the convergence rate due to the combination of clustering topologies and reduces the communication traffic in comparision to distributed consensus algorithm.

IV. QUALITATIVE AND QUANTITATIVE COMPARISON Of TIME SYNCHRONIZATION PROTOCOLS

We compare and evaluate the various synchronization protocols on the basis of qualitative and quantitave criteria.

***Qualitative comparison*:** Here we compares the protocols on the basis of overall quality criteria like accuracy, energy efficiency, scalability, overall complexity and fault tolerance. These quality criterias are play a vert important role in time synchronization protocols. Accuracy is used to measure of the precision of synchronization. The meaning of scalability is any synchronization techniquemust work well with any number of nodes in the network. In wireless sensor network, if delivery of a message is poor then fault tolerance protocols solved message loss problemto some levels. Table 1 shows the comparision of various protocols in terms of qualitative criteria:

Protocols	Accuracy	Energy-efficiency	Complexity	Scalability	Fault tolerance
RBS	High	High	High	Good	No
TPSN	High	Average	Low	Good	Yes
FTSP	High	High	High	N/A	N/A
TDP	High	Average	High	Good	Yes
CCS	High	High	N/A	Good	Yes
Clustered CCS	Unknown	High	High	Good	Yes

Table 1: Qualitative Performance of Synchronization Protocols

***Quantitative Comparision*:** The time synchronization protocols differ in their precision of their synchronization results, energy consumption, computational and communicational results. High precision clearly a desirable feature of a synchronization protocol. In terms of algorithmic complexity, computational cost increases according to varying higher synchronization precision. Convergence time is the total time required to synchronize a whole network. A protocol that requires a large number of message exchanges per synchronization will gives the result in a longer convergence time. Network size states that how many nodes are working in that protocol. Network size gives the maximum number of nodes which are used in the network-wide synchronization. Table 2 gives the comparison of various protocols in terms of quantitative criteria:

Protocols	Precision	Complexity	Convergence time	Network size	Sleep mode
RBS	1.85 ±1.28 µs	High	N/A	2-20 nodes	Yes
TPSN	16.9 µs	Low	Unknown	150-300 nodes	Yes
FTSP	>14 µs	N/A	High (multihop)	60 nodes	Unknown
TDP	100 µs	High	High (multihop)	200 nodes	Yes
CCS	unknown	N/A	Average	100 nodes	Yes
Clustered CCS	29.4±30.2 µs	High	High	200 nodes	Yes

Table 2: Quantitative Performance of Synchronization Protocols

V. CONCLUSION

In wireless sensor network**,** the sensor nodes are heavily resource-constrained because of limited power. Wireless sensor network protocols require handling sensor mobility and scalability. This survey paper presents the clock synchronization protocols for WSN's and also gives the comparison among different existing synchronization protocols based on the qualitative and quantitative approach. The comparison among different synchronization protocols presents based on the various factors like precision, accuracy, complexity and cost will provide basic guidelines to the researchers in integrating various solution features to create a successful clock synchronization scheme for the application.

REFERENCES

1. Fikret, Sivrikaya and Bulent Yener, Rensselaer Polytechnic Institute 2004**, "***Time Synchronization in Sensor Networks: A Survey***"** IEEE Network July/August 2004.
2. *Baljinder Kaur, Amandeep Kaur,"* A Survey of Time Synchronization Protocols for Wireless Sensor Networks*",* IJCSMC, Vol. 2, Issue. 9, September 2013, pg.100 – 106.
3. Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani March 22, 2005, "Clock Synchronization for Wireless Sensor Networks: A Survey" Department of Computer Science University of Illinois at Chicago.
4. Salim el khediri1, Nejah Nasri1, Mounir Samet1, Anne Wei2 and Abdennaceur Kachouri3,"Analysis Study of Time Synchronization Protocols in Wireless Sensor Networks",.
5. Weilian Su, Member, IEEE, and Ian F. Akyildiz, Fellow, IEEE, 2005,"Time-Diffusion Protocol for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 13, No. 2, April 2005.
6. Michael Kevin Maggs, Steven G. O'Keefe, and David Victor Thiel, 2012,"ConsensusClock Synchronization for Wireless Sensor Networks", IEEE Sensor Journal, vol. 12, No. 6, June 2012.
7. Gopal Chand Gautam, Teek Parval Sharma, 2011," A Comparative Study of Time Synchronization Protocols in Wireless Sensor Networks", International Journal of Applied Engineering Research, Dindigul, Volume 1, No Networks4, 2011**.**
8. Deeksha Verma, Arun Kumar Tripathi, 2014," Time Synchronization in Wireless Sensor Network based on Level Discovery Phase", International *Journal of Computer Applications (0975 – 8887) Volume 104 – No 10, October 2014.*
9. Dhyanendra S. Rathore1, Dibyahash Bordoloi1, Emmanuel S. Pilli2 and M. C. Govil, 2014," Virtual Clock Rate Based Time Synchronization for Wireless Sensor Networks", Proceedings of the Second International Conference on "Emerging Research in Computing, information, Communication and Applications." ERCICA 2014.
10. Michael Roche, "Time Synchronization in Wireless Networks", April 2015.
11. Miklós Maróti, Branislav Kusy, Gyula Simon, Ákos Lédeczi,2004, "The Flooding Time Synchronization Protocol", SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
12. Jie Wu, Liyi Zhang, Yu Bai, and Yunshan Sun, 2015, "Cluster-Based Consensus Time Synchronization for Wireless Sensor Networks", IEEE Sensor Journal, vol. 15, No. 3, March 2015.
13. J. Elson, L. Girod, and D. Estrin, "Fine-Grained Time Synchronization using Reference Broadcasts, "Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002.
14. Hyojung Lee, Wonpil Yu, Youngmi Kwon, "Efficien RBS in Sensor Networks," in IEEE ITNG, 2006.
15. S.Ganeriwal, R. Kumar and Srivastava, M. B, 2003 "Timing-Sync Protocol for Sensor Networks", the First ACM Conference on Embedded Networked Sensor System (SenSys), p. 138-149.

