International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 12ǁ December 2013 ǁ PP.24-31

Survey on Clock Synchronization in WSN

Devdatta Bagul[1], Prof. Suresh Kurumbanshi[2], Prof. Upendra Verma [3]
[1]*(MPSTME, SVKM's NMIMS, Maharashtra, India)*
[2]*(MPSTME, SVKM's NMIMS, Maharashtra, India)*
[3]*(MPSTME, SVKM's NMIMS, Maharashtra, India)*

ABSTRACT: *Wireless ad-hoc sensor network are becoming popular to collect data from the area of interest. But for data collection purpose nodes in the network must be synchronized with each other or with the global clock. Synchronization is needed for media access control (MAC) protocol such as TDMA. Wireless sensor networks have limited battery energy, because they are small in size. All these requirements of wireless sensor networks demand an accurate and energy efficient synchronization protocol. This survey paper reviews clock synchronization, need of synchronization and explains basic synchronization methods and protocols in detail.*

KEYWORDS: *Clock synchronization, Multi-hop, Sensor nodes, Sink node, Wireless Network.*

I. INTRODUCTION

The research in microelectronics has resulted in small size, low cost, low weight, and low power sensors node. A sensor node senses the physical world around it and transmits the collected data to sink node. At sink node data from all sensor nodes is collected and analyzed to obtain certain result and conclusion. For this purpose many sensors nodes are deployed in an area of interest and they all together form the wireless sensor network (WSN). The network formed is ad-hoc network, because nodes in this network communicate with each other without any infrastructure. Generally a sensor node performs following function. The transducer in it senses the physical world around it and collects the data. CPU of sensor and operating system processes the data collected by sensor and form network with other nodes. Radio transmitter of node transmits data to sink node. Generally large number of nodes is deployed in area of interest and they use multi-hop routing protocol for transmitting information to sink node.

There are many applications of wireless sensors network:
1) The behaviors of animals can be observed using sensor nodes. For this purpose sensor nodes can be deployed in fixed position or they can be attached with the animal body [2].
2) WSNs are used to measure temperature and pressure at different locations at the same time to draw some conclusion about weather.
3) In agriculture WSNs are used to monitor the parameters such as temperature, soil moisture, light and humidity. All these parameters affect the growth of plants [18].
4) For military WSNs are used to track the path military vehicles. To detect the enemy intrusion, and for all other surveillance application [17] [18].
5) They are used for traffic control. Driving becomes safer if WSNs are deployed in cars.

1.1 Clock Synchronization

Clock synchronization is very important for wireless sensor networks. Clock synchronization in WSNs means local clocks of all sensor nodes in the network should provide common timescale [3]. Local clocks of sensor nodes use quartz crystal oscillator. Frequency of quartz crystal oscillator changes because of change in ambient conditions. Hence observed times of different nodes may differ. However, for many applications and network protocols common time scale is essential. Consider the example of target tracking using WSN. In this application nodes are deployed in fixed positions and report position and time to sink node. At sink node position and time data from different nodes is combined to find position of target [4]. Consider another example of forest fire monitoring. In this application, sensor nodes at different positions report about fire when fire enters into their range. Sensor readings about fire and times are reported to sink node where data is combined to get a result. In this case also clock synchronization of different nodes is essential. Clock synchronization is also essential for medium access control (MAC) protocol such as TDMA [8]. Sensors use battery for electric power. Because of small size they have limited battery power. Packet transmission in sensor networks, use much more energy than local computation [12]. To save energy all sensor nodes can sleep and wake-up at the same time [1][3]. For this purpose also synchronization of nodes in network is essential. Along with above mentioned needs of synchronization the single synchronization method is not useful for all applications; sensors should have different method of synchronization for different requirements.

1.2 Methods of synchronization

1. Relative Ordering: In this method the synchronization is according to the order of messages or events. In this method clocks are not actually synchronized but only order of messages or events is maintained.
2. Relative Timing: In this method a node calculates drift and offset of its own clock as compared with neighboring node and synchronizes its clock with the neighboring clock.
3. Global Synchronization: In this method one node is connected to GPS system, and it has global clock. All other nodes in the network synchronize their clocks with this global clock.

II. CLOCK MODEL [8][9]

Each sensor node has a local clock. In this clock a quartz crystal oscillator is used. Frequency of quartz crystal oscillator changes with change in ambient conditions, hence local clocks run faster or slower as compared to global clock (real time).

Let h(t) be reading of local clock of node at real time t. The rate at which the node's clock runs is defined as,

$$\alpha(t) = \frac{dh(t)}{dt}$$

Then, local time of any node is,

$$h(t)=\alpha(t).t + \beta(t)$$

Where $\beta(t)$ is the offset of that node. Offset is difference between times of local clock and global clock.

Above defined clock rate $\propto (t) = \frac{dh(t)}{dt}$ has value equal to 1 for a perfect clock at all times. Such clock will not run faster or slower. Its shows correct time always. But frequency of quartz oscillator of node's clock changes because change in supply voltage, ambient temperature or humidity. Hence local clock of node runs faster or slower and clock rate is always different from 1.

The clock drift of a node is defined as the deviation of the clock rate from value 1. The drift for any node is,

$$\rho(t) = \alpha(t) - 1$$

Any value of drift $\rho(t)$ is not possible. $\rho(t)$ is always within limit such that,

$$0\leq \rho(t) < 1$$

And therefore,

$$1-\rho(t) \leq \propto (t) \leq 1 + \rho(t)$$

III. COMMON CHALLENGES FOR SYNCHRONIZATION METHODS

For clock synchronization in network all protocols use the same method of exchanging the messages. A node which needs to synchronize sends a message to another node or reference node in the network. Generally a sender node includes send time of its own clock in the message which is called time stamping. When a node generates a request message with time stamp to send to another node, the request message faces various delays while reaching to receiver node. The main and dominating delays are the propagation delay and physical channel access delays. These delays cannot be determined accurately, because they have different values in different situations. Therefore these delays are called nondeterministic delays. This non determinism makes the job of synchronization most challenging. Because of these delays two nodes in the network cannot compare their clocks accurately for synchronization purpose. Therefore it is necessary to understand these delays in a better way.

[1] Send time: The time needed to construct a message at sender side is called send time. This time depends on system call overhead of the operating system, the current processor load, and the time needed to transfer the message to the network interface.

[2] Access time: Media access control (MAC) layer takes some time before actual transmission of the message. This time is called access time. This time depends on type of MAC scheme used. For example in TDMA scheme some time is spent in waiting for time slot for transmission.

[3] Propagation time: When a message goes from sender to receiver it goes through wireless medium. Some time is required for the message to travel through this actual medium. This time is called propagation delay.

[4] Receive time: Receiver's network interface also needs some time to receive the message and transfer it to the host. Physical layer of the receiver receives the message in the form of bits. Some time is needed to receive all these bits and then constructing the packet and in sending it to the application layer [2].

IV. DESIGN PRINCIPLES OF CLOCK SYNCHRONIZATION SCHEMES FOR WIRELESS SENSOR NETWORKS

The synchronization scheme should meet the following requirements. A synchronization scheme can be evaluated on the basis of these requirements. A synchronization scheme can meet very few of these requirements.

1. Scalability: In wireless sensor network any number of new nodes at any time is deployed in the network. A synchronization scheme should suite with increasing number of nodes.
2. Precision / Accuracy: The need of accuracy or precision depends on for what purpose the sensor network is used. For some applications such as ordering of message less accuracy is sufficient but for application such position and velocity measurement of a vehicle more accuracy of synchronization is needed.
3. For some applications sensor nodes are deployed in remote areas. If a node fails, the immediate repairing of it is impossible. In such failure of a sensor node, the synchronization scheme should continue to work for rest of the network.
4. Lifetime: In some applications synchronization among nodes should last for longer duration of it may last for shorter duration in other application. Duration of being synchronized with other nodes depends on applications, and synchronization scheme should be according to it.
5. Cost and Size: Because of advanced technology, wireless sensor nodes are becoming very small in size and decreasing in cost. Therefore, hardware of large size or of more cost should not be attached to sensor node. The synchronization method should be less in cost and size.
6. Sensor nodes operate on battery power and because small size available battery power is less. The protocol designed for synchronization should use less battery power [16].

V. DIFFERENT PROTOCOLS FOR SYNCHRONIZATION

Many clock synchronization protocols have been investigated. The synchronization schemes which were investigated initially include the GPS system. But this scheme is costly, energy-inefficient and cannot work properly if obstacles are present [6].

5.1 Network Time Protocol (NTP):

NTP protocol is used for time synchronization in internet. It is designed by Mills [11]. In NTP there is hierarchy of nodes in the network. Some nodes are connected to GPS system for synchronization, these are called “Stratum 0” time servers. Next “stratum 1” computers are connected to “stratum 0” node.”Stratum 2” computers are connected to “Stratum 1” computers. In general “Stratum L” computers are connected to “stratum L-1” computers. “Stratum L” computers are called time servers and “Stratum L-1”computers clients. To synchronize with the remote server, a client sends a request packet containing time stamps. Upper level computers can work as time servers for lower level computers. Time server sends a reply packet to client containing information about global time. By measuring round trip delay for message exchange and by addingit toglobal time, client synchronizes itself with time server. NTP has accuracy of milliseconds only. In WSN, media access control (MAC) layer can introduce a delay of several hundreds of milliseconds at each intermediate node in message passing process. Therefore NTP can be used in WSNs for which more accuracy of synchronization is not needed. Second disadvantage is that, NTP protocol does not save energy. For these reason NTP is not used

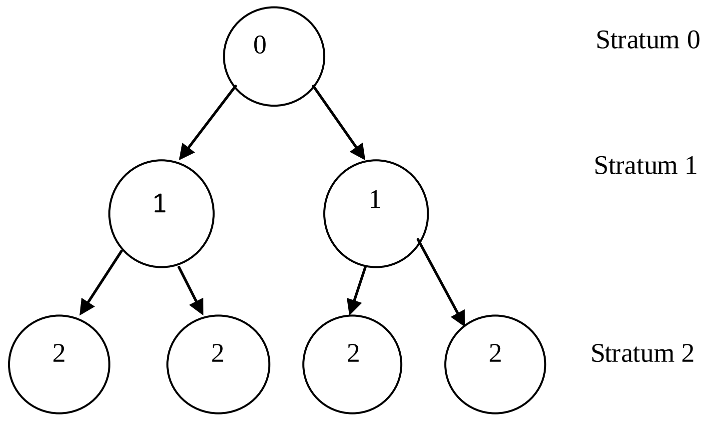

Fig.1 NTP Hierarchy

5.2 Precision Time Protocol (PTP):

The precision time protocol (PTP) is specified by IEEE 1588 for synchronization in network and control systems. It is a master-slave synchronization method. The network is divided into network segments. There are two types of clocks, ordinary clock and boundary clock.

Ordinary clock synchronizes clocks in one sub-network, while boundary clock has connections to two or more sub-networks. A boundary clock synchronizes two or more ordinary clocks [13].PTP also uses time stamping of messages. The master sends the synchronization message (sync) to many slaves. The master clock measures the actual time t_{m1} of placing the message on the network. It is not possible to send this time t_{m1} is sync message because message packet is created before placing it on network. Therefore time t_{m1} is sent in next message, follow-up message. When slave receives sync massage they all note the receiving time ts_1 of messages. After receiving sync massages they again receive the follow-up message which contains the value of t_{m1}. After this all slaves send Delay_Req message and record the sending time t_{s2}. After receiving Delay_Req message, masters sends Delay_Rep message. Delay_Rep message contains the reception time t_{m2} of Delay_Req message. Before sending Delay_Rep message matching of messages is done by master, because it receives many Delay_Req messages.

Slave calculates master to slave delay for message transfer,

$$d_{ms} = t_{s1} - t_{m1}$$

and then slave to master delay in message transfer,

$$d_{sm} = t_{m2} - t_{s2}$$

using value of d_{ms} and d_{sm}one way delay is calculated

$$d_w = \frac{(d_{ms}+d_{sm})}{2}$$

the offset between master and slave clocks is calculated as,

$$\text{offset} = d_{ms} - d_w = \frac{(d_{ms}-d_{sm})}{2}$$

Slave synchronizes its clock with master's clock by using this offset value.

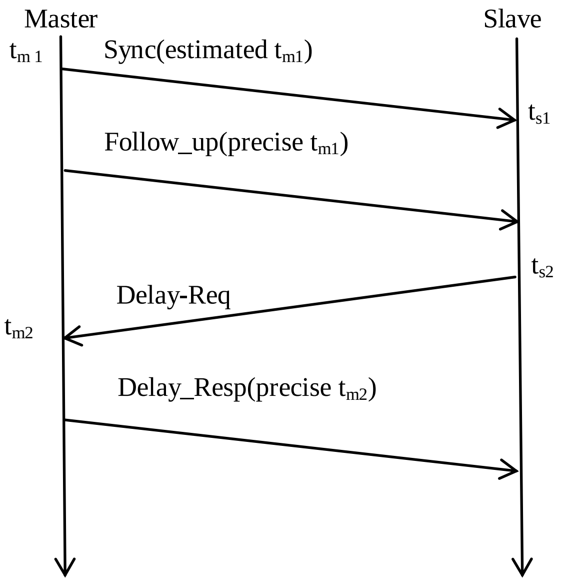

Fig.2 PTP messaging timing diagram [13]

5.3 Reference Broadcast Synchronization (RBS):

This scheme of synchronization was proposed by Elson, Girod and Estin [5]. There are four types of challenging delays for synchronization of clocks in sensor networks. These delays are send time, accesses time, propagation time and receive time. Out of these four delays two delays, send time and accesses time are removed by RBS scheme. In most of the synchronization schemes a sender sends a message packet to the receiver and in response receiver also sends a replay packet. These both request and reply packet contains time information. Using this timing information delay and offset are calculated and receiver corrects its clock. But RBS scheme uses a different method of synchronization. In RBS, one node broadcast a message to two receivers (Receiver 1 and Receiver 2). Broadcasted message does not contain time information. Both receivers of message note the receive time t_1 and t_2 of the message. Means Receiver 1 and receiver 2 receive the message at times t_1 and t_2 respectively. Receiver 1 and Receiver 2 send message packets to each other and inform each other about times t_1 and t_2. Means there is a exchange of messages. By knowing arrival time's $t_1$and t_2 of same message in two clocks, nodes can compare their clocks and synchronize with each other. Same scheme can be used for any number of nodes to synchronize their clocks. As receiver nodes synchronize their clocks using receiving time of same message there is no effect of two delays produced by sender on its side (sand time and access time).Therefore more accuracy in synchronization can be archived. Figure show that critical path is longer on time scale for traditional schemes of synchronization, while critical path is shorter in RBS scheme. The RBS protocol provides average accuracy of 29.1 micro-seconds for a single hop network.

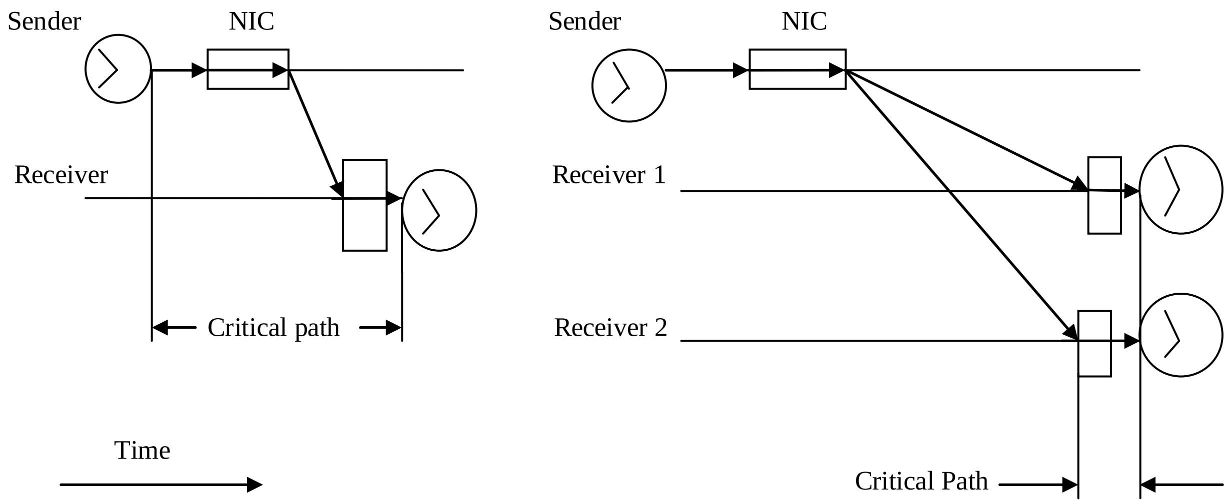

Fig.3 Critical path for ordinary scheme and RBS scheme

5.4 Timing-Sync Protocol for Sensor Networks (TPSN):

Network-wide time synchronization protocol was proposed by Ganeriwal et.al. [7]. TPSN protocol can be used for spanning tree based networks. In sensor network any number of new nodes is deployed at any time according to the need.This changes network topology and can affect synchronization scheme. TPSN can handle this problem very easily. TPSN is a sender-initiated method of synchronization. A special node called root node initiates the process of synchronization in the whole network. There are two phases of this protocol, level discovery phase and synchronization phaseLevel discovery phase:In level discovery phase, levels are assigned to the nodes in the whole network. One node in the network is connected to GPS system for its own synchronization. This node is called root node. Level discovery phase is started by root node .Root node has level 'o'. It sends a level discovery packet to its all neighbors (i. e. broadcasting oflevel discovery message). The level- discovery packet contains identification number and level of sender. It is assumed that all the nodes in the network have their own identification number. When immediate neighbors of root node receive this level-discovery packet, they can decide their own level. They decide their level which is grater by 1 than the root node. That means immediate neighbors of root node will have level 1. After this level '1' nodes send level-discovery packets to their neighbors. Neighbors of level '1' node decide their level as '2'. This process continues and levels are assigned up to leaf nodes. When a level is assigned to any node, it will discard further incoming level-discovery packets. Synchronization phase: Synchronization phase is initiated by root node. For that purpose root node sends a time-sync packet to its neighbors. Time-sync packet is used only to give

intimation of synchronization. When a level '1' node receives this packet, it waits for some time before starting synchronization process. This waiting is to avoid contention in the medium access. Level '1'node performs synchronization with level '0' node by two way message exchange process. When synchronization of level '1' node is completed, level '2' node starts synchronization process with level '1' node. This process of synchronization continues up to leaf node. Synchronization process is explained by taking example of node 'A' synchronizing with node 'B'. When node 'A' receives time-sync packet, it waits for some time and, sends synchronization pulse packet to node 'B'. The synchronization –pulse packet contain s A's level number and sending time T_1 of this packet. When node B receives this packet at time T_2, it sends back a response packet. Suppose B sends response packet at time T_3. This response packet contains B's level number and all the time values T_1, T_2 and T_3. Node A receives response packet at time T_4.
If drift between clocks of A and B is Δ andd is the propagation delay then,

$$T_2 = T_1 + \Delta + d$$

Node A has all the time values T_1, T_2, T_3 and T_4 with it. It calculates Δ and d by using equations.

$$\Delta = [(T_2 - T_1) - (T_4 - T_3)] / 2$$

$$d = [(T_2 - T_1) + (T_4 - T_3)] / 2$$

Node A synchronizes itself with node B using above values. Using same procedure all the nodes in network synchronize with upper level nodes. Means node of level i synchronizes with node of level i-1. Time-sync packet is sent only by root node to its immediate neighbors. For nodes having level values greater than 1 this message is not needed for synchronization. They listen the messages exchanged by upper level nodes for synchronization and stat synchronization after the synchronization of upper level node is completed. For this purpose lower level node waits for some time, so that synchronization of upper level node is completed.

Disadvantage:

TPSN provides an average accuracy of 16.9 microseconds. If any node fails, other nodes depending for synchronization on this node cannot synchronize their clocks.

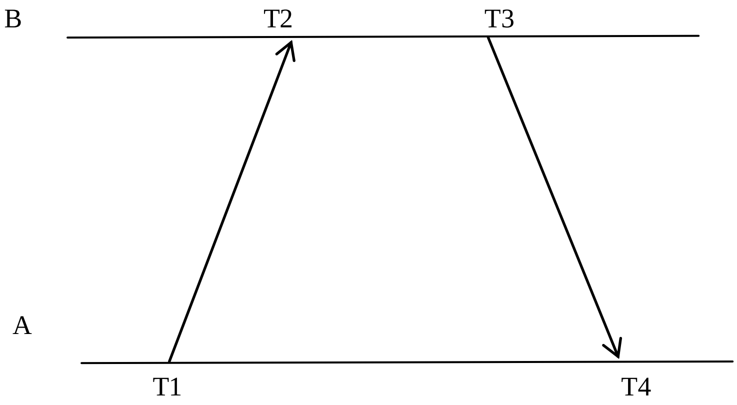

Fig.4 TPSN [7]

5.5 Flooding Time Synchronization Protocol (FTSP):

The FTSP protocol was designed by Miklos Moroti, Branisllarkusy and Simon and Akos [2]. FTSP protocol synchronizes all the nodes in the network with single synchronization message. A single node called the root node keeps the information of global time. This root node synchronizes all the nodes in the network with a single broadcasted message. An ad-hoc network is formed for synchronization, therefore there is no effect of node failure or link failure on synchronization. MAC layer time stamping is used in message for synchronization which reduces many errors. The root node or master node sends a broadcast message to other nodes in the network for synchronization. Broadcasted message contains the master's (sender's) time stamp which is a global time, at the transmission of message. When a receiver node receives this message it notes the receiving time of message in its local clock. Thus a receiver has two times, global time and local time. By calculating the offset between master and receiver clocks, the receiver node can correct its local clock. All the nodes receive the same message at the same time; therefore they can correct their clocks at the same time. At the boundary of each SYNC byte in message time-stamp is included, when message is sent or received. Therefore multiple time stamps are available for calculation. The data packet containing SYNC bytes is shown. The parts of the data packet are Preamble, SYNC, Data and Crc.

FTSP provides an accuracy of 1.48 micro-sec for single hop network and o.5 micro-sec for multi-hop network.

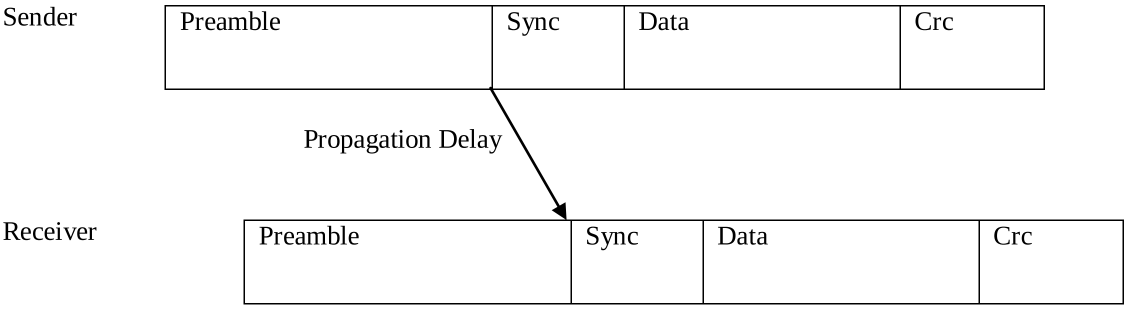

Fig.5 FTSP [2]

5.6 Post – Facto Synchronization:

Post-facto synchronization scheme was designed by Elson and Estrin [16]. Wireless sensor nodes have limited battery power. Therefore wireless sensor nodes are designed to save battery power. Sensor nodes have pre-processor and a general purpose processor. General purpose processor sleeps when not in use. Pre-process works on low power and it continues to work and powers on the general purpose processor for message communication and local computation. If general purpose processor sleeps for longer time, then synchronization will not last for longer time. To solve this problem post-facto synchronization scheme was designed. In post-facto scheme nodes are not synchronized for all the time.

When certain event occurs, each node notes the time of occurrence of that event using its own clock.Immediately after this event a third node called a beacon node, broadcasts a synchronization pulse, to all nodes in its radio range. Nodes which receive this pulse record the time of arrival of pulse. Receiver nodes calculate time between occurrence of event and time of arrival of synchronization pulse. Using this calculation these nodes synchronize for that instant only. Post-facto synchronization led Elson and Estrin to their RBS scheme.

5.7 Tiny-Sync and Mini-Sync Protocol:

Tiny-Sync and Mini-Sync are two lightweight synchronization protocols created for sensor networks by Sichitin and Veerarittiphan [14]. In this method also there is exchange of messages between nodes for time-stamping but do not correct clock directly. In this method a relation between clocks of nodes is obtained. Relation between clocks is obtained by equation

$$C_1(t) = a_{12}. C_2(t) + b_{12}$$

Where $C_1(t)$ and $C_2(t)$ are times in two clocks '1' and '2', a_{12} is relative drift and, b_{12} is relative offset between clocks.

Two nodes '1' and '2' send multiple messages to each other to measure $a_{12}$and b_{12}i.e. relative drift and relative offset. Using this information two lines are plotted with minimum and maximum slops. Upper and lower bound values of relative drift and relative offset are obtained from slopes and intercepts of two lines. Then average relative drift and relative offset is obtained by plotting average line of above two lines. Three data points are decided to apply tight bounds on relative drift and relative offset between two nodes. To create these data points a node 1 sends a probe message with time stamp to node 2. When node 2 receives this message it timestamps it with time t_b and sends back an acknowledgment to node1 which timestamps it with time t_r. If upper and lower bounds are tight, more accuracy of synchronization is obtained.

5.8 Lightweight Tree- Based Synchronization (LTS):

To minimize the complexity of synchronization algorithm Greunen and Rabaey [15] proposed lightweight Tree-based synchronization (LTS). Some wireless sensor networks need more accuracy of synchronization while others need less accuracy. Therefore they designed two algorithms working together according to the need of accuracy. When less accuracy is needed first algorithm works while for more accuracy second algorithm takes the charge of synchronization. First algorithm works in two phases, phase '1'and phase '2'.In phase '1' a spanning tree of nodes in the network is created. There is one master node or sink node in the whole network. Master node initiates the synchronization procedure in phase 2 of algorithm. Pair wise synchronization is performed along n-1 edges of spanning tree. It is a sender to receiver type of synchronization. If more accuracy of synchronization is needed instead of first algorithm, the second algorithm takes the charges of synchronization. In second algorithm synchronization is done in distributed fashion. For this purpose there are two or more master nodes. When synchronization is needed for node 'i', it sends a synchronization request to nearest master node. All the nodes in the path of synchronization request are synchronization before node 'i' synchronizes. The advantage of this scheme is that overhead of synchronization is reduced, because one request synchronizes many nodes on the requestpath.Again collection of synchronization requests is also done.For this collection node ichecksif there are any pending requests of neighboring nodes for synchronization.

VI. CONCLUSION AND FUTURE WORK

The basic synchronization methods for wireless sensor networks are discussed in this paper. RBS, TPSN and FTSP are the basic methods. Other methods with little variation are also discussed. Energy saving and accuracy are measure issues while designing synchronization protocols. With this intention in mind we presented basic methods of synchronization. As per demand of society more accurate and energy saving protocols should be designed. This survey will be helpful to me to design such a protocol.

VII. ACKNOWLEDGMENT

This research paper is made possible through the help and support from everyone, including: parents, teachers, family, friends, and in essence, all sentient beings. Especially, please allow me to dedicate my acknowledgment of gratitude toward the Following significant advisors and contributors: First and foremost, I would like to extend my gratitude to Hon. Shri Amrish Patel, Chancellor SVKMs NMIMS for his most support and encouragement by providing excellent infrastructure and research faculties in Tribal area like Shirpur. I would like to extend my gratitude to Dr Tapan Bagchi, Director Shirpur Campus, Dr M V Deshpande Associate Dean, MPSTME Shirpur and Dr N S Choubey, Head of the department for constant and unconditional support. Finally, I sincerely thank to my parents, family, and friends, who provided the advice and financial support. The product of this research paper would not be possible without all of them.

REFERENCES

[1] Li-Ming He, "Time Synchronization Based on Spanning Tree for Wireless Sensor Networks", IEEE 2008.

[2] Miklos Maroti, Branislav Kusy, Gyula Simon, Akos Ledeczi, "The Flooding Time Synchronization Protocol", SenSys 04, Novemmber 3-5,2004, Baltimore, Maryland, USA.

[3] Khuram Shahzad, Arshad Ali, N.D.Gohar, "ETSP:An energy-efficient Protocol For Wireless Sensor Networks", 22nd International Conference On Advance Information Networking and Applications, IEEE, 2008.

[4] Suarabh Ganeriwal, Ram Kumar, Sachin Adlakha, Mani Srivastava, "Network-wide Time Synchronization in Sensor Networks", Technical Report, NESL 01-01-2003.

[5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," ACM SIGOPS Oper. Syst. Rev., vol. 36,no. SI, pp. 147–163, Winter 2002.

[6] C. Dunn, D. Jefferson, S. Lichten, J. Thomas, Y. Vigue, and L. Young, "Time and position accuracy using codeless GPS," in Proc. 25th PTTIConf., Marina Del Rey, CA, 1993, pp. 169–179.

[7] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in Proc. 1st Int. Conf. SenSys, 2003, pp. 138–149.

[8] Philipp Sommer, Roger Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks",ISPN'09, April 13-16, 2009, San Francisco, California, USA.

[9] Phillp Blum, Lennart Meier, Lothar Thiele, "Improved Interval Based-clock Synchronization Wireless Sensor Networks" ISPN'04, April 26-27, 2004, Berkely, California, USA.

[10] VilgotClaesson, HenrikLonn, NeerajSuri, "An Efficient TDMA Synchronization Approachfor Distributed EmbeddedSystems"Volvo Research Foundation, Proj.F99/7.

[11] D. L. Mills, "Internet time synchronization: The network time protocol", IEEE Trans. Commun., vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[12] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Andrea Passarella, "Energy conservation wireless sensor networks: A Survey", Ad Hoc Networks 7 (2009) 537-568.

[13] Todor Cooklev, John C. Eidson, Afshaneh Pakdaman, "An Implementation of IEEE 1588 OverIEEE 802.11b for Synchronization of Wireless Local Area Network Nodes" IEEE Trans. Instrum. Meas., vol. 56, no. 5, pp. 1632–1639, Oct. 2007.

[14] M.L. Sichitiu and C. Veerarittiphan, "Simple, Accurate Time Synchronization for Wireless Sensor Networks" 2003 IEEE.

[15] Jana van Greunen, Jan Rabaey, "Lightweight Time Synchronization for Sensor Networks", WSNA'03, September 19, 2003, San Diego, California, USA.

[16] Jeremy Elson, Deborah Estrin, "Time Synchronization for Wireless Sensor Networks" Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing,April 2001, San Francisco, CA, USA.

[17] Jagannathn Sarangpani, "Wireless Ad-Hoc and Sensor Networks", CRC Press, Taylor and Francis Group.

[18] Edgar H. Callaway, "Wireless Sensor Networks: Architectures and Protocols", CRC Press 2004 ISBN: 0849318238.