A Novel Method on Clock Synchronization in WSN

Devdatta Bagul[1], Prof. Suresh Kurumbanshi[2], Prof. Upendra Verma[3]

[1,2&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. The RTSP algorithm is designed to achieve network wide clock synchronization with good accuracy and to save battery energy. In the proposed paper a change is suggested in the self-synchronization part of the RTSP algorithm. With this change RTSP algorithm will become more energy efficient, more accurate and will become simple to implement self-synchronization.

Keywords: Global time, Nodes clock, Offset and Synchronization

1. Introduction

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 [10]. 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.

In this paper, we have suggested a change in RTSP algorithm. The RTSP algorithm is better for network wide clock synchronization, saves battery energy, has better accuracy and robust to link and node failures. We have suggested a simple but better self-synchronization mechanism in this algorithm. With this change the algorithm will save more battery energy and improve accuracy further.

The rest of this paper is organized as follows. In section 2 clock models and common challenges for synchronization methods are presented. Section 3 describes Literature survey. Section 4 describes RTSP algorithm in details. The proposed change in RTSP algorithm is described in section 5.

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. G`lobal 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.

2. Clock Model [8, 9, 15]

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. It 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,
$1-\rho(t) \leq \propto (t) \leq 1 + \rho(t)$

3. LITERATURE SURVEY

Accuracy or error in two clocks is the difference between times shown by two clocks [16]. There are certain challenges to improve this accuracy in all synchronization methods. In all clock synchronization methods in WSN messages are exchanged between sender and receiver nodes. When a sender sends a synchronization message to receiver, the message faces various delays while reaching to receiver node and same thing happens with reply message also [2]. Some delays cannot be determined accurately because their values depend on situation at that time. This non-determinism of delays complicates the job of cock synchronization. Four main delays are discussed here [2].
1. Send time: The time needed to construct a message at sender side is called send time. This time depends on overhead of the operating system, and the time needed to transfer the message to the network interface.
2. Access time: Time wasted in waiting for access to the transmit channel is called access time. This time depends on the type of MAC scheme used.
3. Propagation time: Time taken by the message to travel from sender to receiver through wireless medium is called propagation delay.
4. Receive time: Time taken by receiver's interface to receive message and transfer it to host is called receive time. This is the time needed for physical layer of receiver to receive all bits of message and to construct a packet form it.
Clock of a sensor node can be synchronized with GPS receiver. GPS based clock synchronization can provide accuracy of 1 µs. But GPS system is costly and energy consumption is more. Also, the line of sight to the GPS satellite is needed. Because of these limitations GPS system is not feasible for every sensor node, as their number is large in a sensor network [6].
NTP protocol is used for time synchronization in internet which is designed by Mills [11]. Some nodes are connected to GPS system for synchronization, which are called time servers. To synchronize with the remote time server a client sends a request packet. Time server sends a reply packet to client containing information about global time. By measuring round trip delay for message exchange, client can synchronize itself with time server. NTP has less accuracy, of the order of milliseconds. 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 is not suitable for WSN because for WSN more accuracy of synchronization is important. Second disadvantage is that, NTP protocol does not save energy.
The Precision Time Protocol (PTP) is specified by IEEE 1588 for synchronization in network and control systems. It is master-slave synchronization method [13]. 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. PTP uses time stamping of messages. For synchronization master and slave perform message exchange containing time-stamps. By calculating delay in message transfer and offset between master and slave clocks, slave synchronizes its clock with master clock. Accuracy of PTP is in sub-microseconds. But PTP is suitable only for hierarchical master-slave architecture.
Sensor node clocks are simple clocks and they drift significantly from the global clock. The multi-hop nature of WSN complicates the problem because media access control (MAC) layer can introduce a delay of several hundreds of milliseconds at each intermediate node in message passing process. Therefore master-slave architecture is not feasible for WSNs. Different applications have different synchronization requirements and according to those different algorithms are developed. The synchronization field still under research. Different algorithms feasible for WSN for different needs are being developed.
Reference Broadcast Synchronization (RBS) scheme was proposed by Elson, Girod and Estin [5]. There are four types of challenging delays, send time, access time, propagation time and receive time. Out of these four delays two delays, send

time and access time are removed by RBS scheme. RBS scheme uses a different approach of synchronization. In RBS, one sender node broadcasts a message to two or more nodes in the broadcast range. Pairs of receiving nodes exchange information about time of reception of broadcast message which was sent by single sender node. The pairs of receiver nodes get information about offset between their clocks and they synchronize with each other. The RBS protocol provides average accuracy of 29.1 micro-second for a single hop network. This accuracy is less for certain applications.

The Timing-Sync Protocol for sensor networks, (TPSN) was proposed by Ganeriwal et.al. [7]. TPSN provides network-wide time synchronization. There are two phases of this protocol, level discovery phase and synchronization phase. In level discovery phase levels are assigned to the nodes in the whole network and spanning tree structure is formed. Level ‘0’ node is connected to GPS system. The synchronization phase is initiated by root node ‘0’. In synchronization phase nodes synchronize with their parent nodes in the spanning tree structure by a two-way message exchange method. TPSN provides an average accuracy of 16.9 microseconds which is again less value of accuracy.

Flooding Time Synchronization Protocol (FTSP) was designed by Miklos Moroti, Branislav Kusy, Simon and Ledeczi [2]. FTSP protocol synchronizes all the nodes in the network with a single broadcast synchronization 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 broadcasts message containing its time stamp of global time. 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 root node and receiver node clocks, the receiver node can correct its local clock. FTSP provides an accuracy of 1.48 μs for single hop network and 0.5 μs for multi-hop network.

4. EXISTING SYSTEM OF CLOCK SYNCHRONIZATION

An Accurate and Energy-Efficient Protocol for clock synchronization (RTSP) was developed M.Akhlaq and Terek Shetami [14]. The proposed algorithm is modification of RTSP; therefore RTSP is discussed in little detail.

In the RTSP scheme of synchronization, there is a reference node having identification number ‘0’ which has global time information. Reference node obtains global time information from GPS system other nodes in the network also have identification number storing from 1.

When a sensor node boots up, it asks its neighbors for identification of a reference node by sending ERN enquiry message. The enquirer node waits for a time T, if it does not get any answer during this time, it enters into the contest for election of new reference node by broadcasting the ERN message and declares itself to be reference node. The other nodes receiving this broadcast message, take any one action stated below.

1. When other nodes receive this broadcast message, they check identification of reference node in their memory. If their memory contains identification number for reference node greater than the ID of reference node in broadcast message, they update their memory and rebroadcast the same message.

2. If ID of receiving node is smaller than the ID of reference node in broadcast message, and it is a cluster heads it enters into contest for election of reference node by declaring itself to be reference node and broadcasting the message.

3. If a cluster head or a gateway node knows the same ID of the reference node, it will just be rebroadcast the message.

It is possible that more than one node can declare themselves to be the reference node. If such problem arises the reference node becomes ordinary node, if it finds that its ID is greater than the ID in message for reference node. The elected reference node takes the responsibility of broadcasting the global time information to other nodes periodically. The frequency of broadcasting such time informing messages is kept less to save battery energy.

When a node in WSN receives this broadcasted message containing information about global time, it compares global time with its local time and if there is any difference between the two times it sends a request (REQ) message to reference node and asks for global time. After receiving this request (REQ) message reference node sends a reply (REP) message containing global time Tr to the requesting node. The requesting node synchronizes its clock with clock of reference node. Intermediate nodes in the message path of REP message also synchronize themselves with the global clock, so that there will be no need for them to send REQ message for synchronization. Further these intermediate nodes can also answer to REQ message if they have recent global time information.

Because of multi-hop structure of WSN there will be a delay in the process of forwarding a REQ message and REP message. The delay produced should be compensated at every node which forwards the REQ message and REP message.

RTSP tackles this delay compensation problem in a better way. In figure1 node 44 sends REQ message to reference node ‘0’ at its local time T_1. The intermediate node 5 receives this message at its local time T_2. The node 5 stores ID of node 44 in its memory and also stores the values of T_1 and T_2 in T_{1old} and T_{2old} respectively. Then it forwards this message at its its local time T_1 to node 4. Node 4 also stores values of T_1, T_2 and ID of node 5. Finally REQ message reaches to node ‘0’ by recursive forwarding. Reference node o‘0’ receives this message at its local time T_2 and sends a reply message at its local time T_3. In reply (REP) message node ‘0’ sends values at T_1, T_2, T_3 and global time T_r. REP message is sent by the same path by which REQ message travelled. Ode 4 receives this REP message at its local time T_4. Node 4 calculates the delay d using values of T_1, T_2, T_3 and T_4 by using equation (1) and corrects value of T_r by adding d in it. It synchronizes its own clock and forwards this REP message to node 5 at time T_3. But before forwarding message node ‘5’ once again corrects the value of T_r by using equation (3) because some time elapses since T_r. Every receiving node of REP message performs same actions and final message is forwarded to node 44. Node 44 corrects its clock finally.

$$d = \frac{(T_2-T_1)+(T_4-T_3)}{2} \quad (1)$$

$$T_r = T_r + d \quad (2)$$

$$T_r = T_r + (T_3-T_4) \quad (3)$$

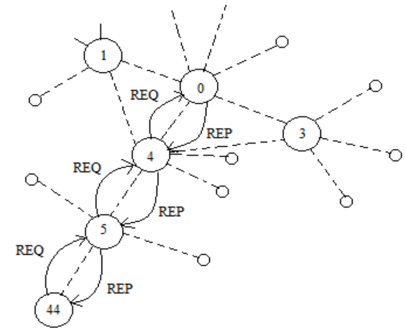

Figure 1

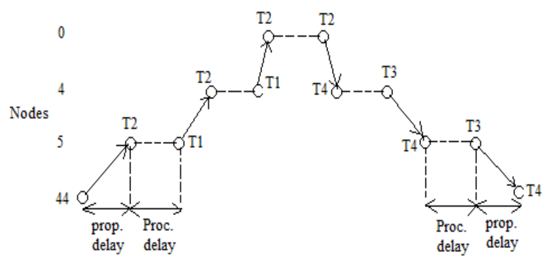

Figure 2

5. PROPOSED SYSTEM

In original RTSP algorithm self-synchronization mechanism is used to correct time in node's clock. For this self-synchronization purpose node collects time information from two data points and then calculates offset and drift. With these values of offset and drift node updates it's clock and becomes self-synchronized.

We are suggesting two changes in this process of self-synchronization. First change is that there is no need to calculate drift and offset at the same time for self-synchronization. The reason is that offset and drift calculation at the same time becomes essential when we compare two clocks of two different nodes in a network. For such two different clocks offset calculation becomes essential because there may be initial time difference between two clocks. If we are comparing two times of the same node clock at two different instants then only drift calculation is sufficient for self-synchronization. Because once the node clock is synchronized with global clock, by knowing average drift of node's clock we can calculate what offset Δt will be produced after certain predetermined time interval of self-synchronization as shown in figure 3. By adding or subtracting this offset Δt in nodes clock, its clock can be self-synchronized. Means in our scheme we are calculating drift first of a node clock by noting time at two different instants of the same node's clock and then we are using this drift value to calculate offset Δt of the same node's clock. This gives better self-synchronization. It is very simple to verify this by taking certain time values of the same node's clock.

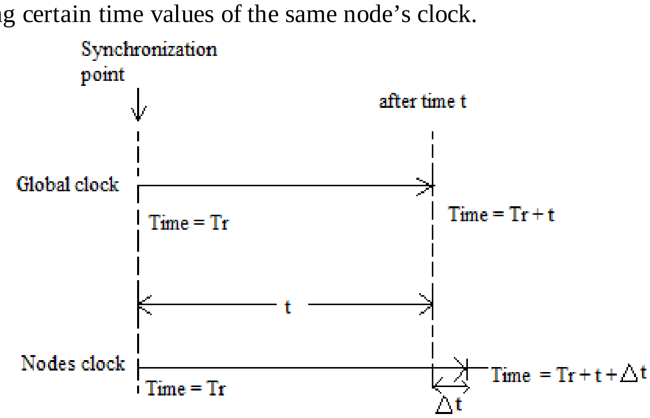

Figure 3

Second change we are making is that instead of two data points of a node clock we suggest to take three data points so that average drift can be calculated to perform better self-synchronization. Figure 3 shows that after synchronization both clocks show same time. Both clocks are allowed to run for same time t. As node's clock is fast, it shows additional time Δt. This offset Δt can be subtracted from nodes clock time and it can be corrected.

5.1 Self-Synchronization

When "RTSP" reply "REP" message is received by a node, the node notes the receiving time T_4 of this message. "REP" message contains global time T_r and node synchronizes itself with global time. A node can be self-synchronized by using these values of T_4 and T_r. But for this purpose at least three values of T_4 and T_r are to be stored in memory. These three values should be noted from three different "REP" messages received successively.
Let us assume that,
Time in global clock (Real time) = t
Time in node's clock = h(t)

$$\text{Clock rate, } \alpha = \frac{dh(t)}{dt}$$

If node's clock is correct, (perfect clock), then

$$dh(t) = dt$$
$$\alpha = 1$$

If node's clock is fast, $\alpha > 1$
If nodes' clock is slow, $\alpha < 1$
Drift: Drift (ρ) is defined as deviation of node's clock rate from global clock rate (perfect clock rate). For global clock rate is 1, because it is perfect clock. Thus drift for node's clock is

$$\rho = 1 - \alpha$$

If we know that in certain time interval, after synchronization node's clock goes ahead or lags behind by Δt, then we can calculate correct time by subtracting or adding Δt in nodes current time h(t).
The offset Δt can be found using drift value. Because drift indicates by which rate node's clock is going ahead or lagging behind as compared to rate of global clock.

$$\Delta t = (1- \alpha)\, t$$

Where t is self-synchronization period and it should be kept minimum.
Three different values of Tr and T_4 are to be noted from three different "REP" message packets received successively with last packet new. T_r values used should be corrected values by adding delay "d" in it.

Table 1

REP packet no.	Global time (T_r)	REP packet receiving time (T_4)
1	$X_1 = T_r$	$Y_1 = T_4$
2	$X_2 = T_r$	$Y_2 = T_4$
3	$X_3 = T_r$	$Y_3 = T_4$

Node clock rate,

$$\alpha 1 = \frac{(Y2-X1)}{(X2-X1)} \qquad (4)$$
$$\alpha 2 = \frac{(Y3-X2)}{(X3-X2)} \qquad (5)$$

in equation (4) to find difference in time in node's clock, X_1 is subtracted from Y_2. Because after synchronization, time in nodes clock is $X_1 = T_r$ and from this value the node's clock runs further up to value $Y_2 = T_4$. This is shown in figure 4.

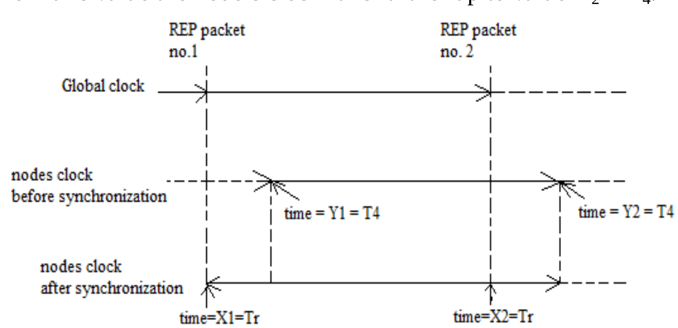

Figure 4

Average rate,

$$\alpha = \frac{(\alpha 1+\alpha 2)}{2}$$

$$\text{Drift} = 1- \alpha$$

Equation for correction of clock time by self-synchronization,

$$[h\ (t)]_{new} = (1-\alpha)\ t + [h\ (t)]_{old}$$

Where, t is self-synchronization period.

6. RESULTS AND DISCUSSION

For clock synchronization in WSNs measurement of accuracy of synchronization and energy consumption are important for performance evaluation. Figure5 (a) shows average absolute error for time synchronization for existing RTSP algorithm. The average absolute error of existing algorithm is 0.23 microsecond. This is less than all previously developed algorithms such as RBS, TPSN and FTSP. As drift calculation is done in better way in proposed system, therefore more accuracy will be achieved. Expected accuracy is assumed to be 0.15 μs and this is shown in figure 5 (b). Also by properly optimizing the re-synchronization interval more accuracy can be achieved.

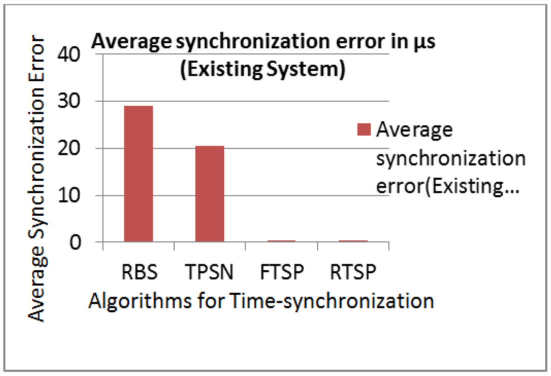

Figure 5 (a) Average absolute error of time synchronization

The actual energy consumption depends on many factors, such as type of hardware, software and antenna. Therefore in existing RTSP algorithm a straight forward method is used for measurement of energy consumption. According to this straight forward method, if there are N nodes in network then, in a re-synchronization period T seconds (based on FTSP), each node sends N messages in RBS protocol, two messages in TPSN protocol, one message in FTSP protocol and only 0.14 messages in RTSP protocol. In existing RTSP protocol, on startup of the network, reference node first broadcast one message and it is forwarded by every node. Then for synchronization purpose every node sends two REQ messages and receives two REP messages. Thus total five messages are needed for synchronization per node, after this node sends very less messages. Figure 6 shows the energy consumption in the long run. Initially energy consumption is more for RTSP but after five periods it is less than all protocols other protocols.

In the proposed algorithm only difference is that instead of initial two REQ messages a node will send there REQ messages and will receive three REP messages. Means instead of five, seven messages will be needed for initial synchronization. Therefore in the proposed algorithm initially energy consumption will be little more. But with this little more consumption of energy accuracy will be more and energy consumption will be again less afterwards because very few resynchronization requests will be sent to reference node. While plotting energy graph it is assumed that for proposed system, increase in messages sent by nodes for synchronization will be half of the messages sent in existing system by nodes for increase in synchronization period T by value 10.

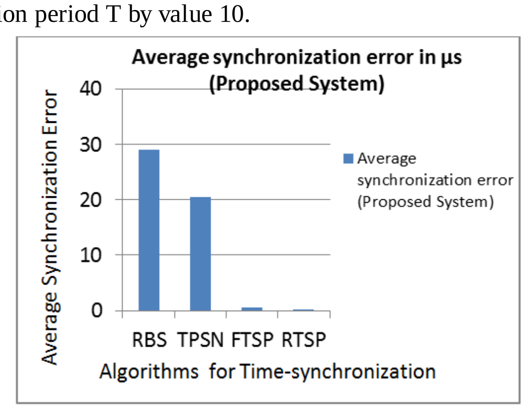

Figure 5 (b) Average absolute error of clock synchronization

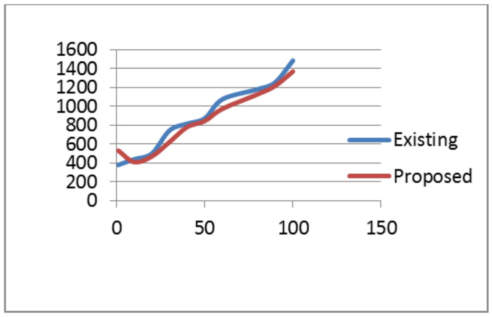

Figure 6 Graph of energy consumption in terms of number of messages sent for synchronization against number of synchronization periods T (based on T for FTSP)

7. CONCLUSION

We have discussed results to be obtained by proposed change in RTSP algorithm. The existing system achieves average accuracy of 0.23 microseconds in clustered accuracy. By proposed change accuracy will be more than this value because of average calculation of drift value and then offset between global and node clocks and by optimizing resynchronization period. The accuracy will be more because of MAC layer time stamping based on SFD byte, by compensation of propagation delay are aggregation of synchronization of request.

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. 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 provide 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, BranislavKusy, 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 PTTI Conf., 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] Vilgot Claesson, HenrikLonn, Neeraj Suri, “An Efficient TDMA Synchronization Approach for 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 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes”, IEEE Trans. Instrum. Meas., vol. 56, no. 5, pp. 1632–1639, Oct. 2007.

[14] Muhammad Akhlaq and Terek R. Sheltami, “TRSP: An Accurate and Energy –Efficient Protocol for Clock Synchronization in WSNs”, IEEE transaction on instrumentation and measurement, vol.62,no.3,March 2013

[15] Kay Romer, “Time Synchronization and Localization in Sensor Networks”, Ph.D. Thesis, Swiss Federal Institute of Technology Zurich (ETH Zurich) 2005.

[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.

AUTHOR BIOGRAPHY

Mr. Devdatta B. Bagul has done the B.E. degree in Computer Science from NMU University, India in 2012, Pursuing M.Tech degree in Computer Engineering from Mukesh Patel School of Technology, Management & Engineering of Narsee Monjee Institute of Management Studies, Deemed to be University, Mumbai Shirpur off Campus. His research interest is Adhoc Network.

Mr. Suresh Kurumbanshi has done B.E. and M.Tech degrees in Electronics Engineering from Nagpur University. He has worked in Finolex Cables ltd,Pune & worked as Head of Electronics Engineering Department of Rajeev Gandhi College of Engineering Nagpur. .He is Pursuing PhD from G.H.R.C.E, Nagpur & worked in the same organization in Electronics & Communications Engineering Department for more than 6 years as a Sr.Lecturer. He is currently working as Assistant Professor & Head [First Yr-ASH] & In charge ISTE Chapter in Mukesh Patel School of Technology, Management & Engineering of Narsee Monjee Institute of Management Studies, Deemed to be University, Mumbai Shirpur off Campus. His research area includes Adhoc Network, mobile communication, Automation Technology ,wireless access network using stratospheric platforms.

Mr. Upendra Verma has done B.E. and M.E. degrees in Computer Science and Engineering from RGTU University. He has worked in PCST and LKCT Engineering college. . He is currently working as Assistant Professor in Mukesh Patel School of Technology, Management & Engineering of Narsee Monjee Institute of Management Studies, Deemed to be University, Mumbai Shirpur off Campus. His research area includes Adhoc Network, Network Security, Software Engineering.