A Comparative study of Time Synchronization Protocols in Wireless Sensor Network

Amit Nayyer
Shoolini University
Solan, H.P, India

Meenakshi Nayyer
Baddi University
Solan, H.P, India

Lalit Kr. Awasthi
NIT Hamirpur
H.P, India

ABSTRACT
Wireless sensors network is a type of the ad-hoc network. It is comprised of many sensors which are interlinked with each other for performing the same function collectively such as monitoring the weather conditions, temperature, different kind of vibrations and sound etc. Any distributed system requires time synchronization. In particular, time Synchronization is extremely important for Wireless sensor network applications e.g. for data fusion, TDMA schedules, synchronizes sleep periods, etc.

In this paper, we study different time synchronization protocols available for sensor networks, like Reference Broadcast Synchronization (RBS), Flooding Time Synchronization Protocol (FTSP) and Time Synchronization Protocol for Sensor networks (TPSN). Network Time Protocol (NTP) which is very famous in the computer network is also considered. The simulations of these different protocols are performed on the sensor network with the help of a simulator. The effects of these protocols on different parameters are studied and results obtained are compared.

Keywords
Time synchronization, clock synchronization, wireless sensor network.

1. INTRODUCTION
New advancement towards the minimization, reducing the cost and power requirements have motivated the researchers towards wireless sensor network [1]. The aim of many researchers is to create an environment that is rich of sensors. The deployment of sensor can be very helpful in detecting the different conditions of the environment like sound, temperature and movement of objects [6]. There are a wide range of applications envisioned for such sensor networks, including micro-climate studies, groundwater contaminant monitoring, precision agriculture, condition-based maintenance of machinery in complex environments, urban disaster prevention and response, and military interests etc. These applications are not fulfilled by the traditional and existing architecture. In many scenarios, deploying wired sensors in large areas is impractical due to requirement of infrastructure. Putting manual observation in fields such as environmental monitoring is not only time consuming but also it require a lot of man power to cover a large area. Moreover the events in such type of environment are low enough that sometime even a few occurs in a day. Deploying a large number of sensors in such area makes sure that all the area is covered well. Generally, the sensors are thrown rather than manually deployed to cover the area.

In sensor networks, different factors demands flexible and robust time synchronization, while simultaneously is making it more difficult to achieve as compared to computer networks [8]. Some sensors are so battery constrained that they only wake up occasionally, take a reading, transmit it and return to sleep [10]. To notify the time of events that occur in the environment is the basic requirement for nodes. For example, accurate time is needed to measure the time-of-flight of sound; distribute an acoustic beam forming array; form a low-power TDMA radio schedule [17]; integrate a time series of proximity detections into a velocity estimate; or suppress redundant messages by recognizing duplicate detections of the same event by different sensors. In this research work, we compare the existing time synchronization protocols and show the results based on different parameters selected. We try to show that the particular protocol is better in a particular situation and need of the application. We consider the various uses of time synchronization in detail, and describe the axes along which these applications can be characterized. Based on our experience exploring this problem space, we propose several general guidelines for the use of time synchronization protocols in sensor networks. No single synchronization scheme can be optimal on all axes (e.g., precision, lifetime, scope, energy, availability), but many applications do not require best performance on all the above mention axes. A range of schemes should be available to system designers, such that the synchronization that is provided matches what is needed. An ideal synchronization system will minimize its energy use by providing service that is exactly necessary and sufficient for the needs of the application. Tunable parameters can allow synchronization modes to be matched more closely to the requirements of the application. Most existing time synchronization schemes make a common assumption: that their goal is to keep the clock synchronized all the time. Applications assume that they can query the clock at any time, and it will be synchronized. Another approach is to let clocks at sensors to run at their natural rate, and when any event of interest occur the node time stamp the event with the clock of the cluster head [18]. This has many advantages; for example, it enables post-facto synchronization, peer-to-peer synchronization, and participation in multiple timescales.

The outline of the paper is as follows: In section II, we briefly review literature survey in clock synchronization. It also describes various challenges that should be considered during the design of time synchronization protocol for sensor networks. Section III reviews related work in clock synchronization. This chapter discusses the results of related research which compare time synchronization protocols. We explore a number of metrics that have found relevant for evaluating time synchronization in the sensor network domain in this related research. In Section IV, we show the comparison results of our simulation of the time synchronization protocols. Finally, in Section V, we present our conclusions and describe directions for future research in this area.

2. BACKGROUND

In this section we summarize various existing synchronization protocols. There are different methods available for synchronization in distributed systems [9], but we are considering only few of them here. These all methods work in different way as compared to just ordering of events [7]. We specifically consider the following protocols.

2.1 Network Time Protocol

The time synchronization is also required in computer networks for various different purposes. The network time protocol (NTP) [2] is the protocol that is used in traditional computer network to keep the clock synchronized. It provides coordinated universal time (UTC). NTP uses level of clock servers for the purpose of synchronization. Each level called stratum is assigned a different level starting with 0. The next level server has another level as 1 and the hierarchy is maintained for the levels. Normally stratum 0 is GPS clock or atomic clocks, where as stratum 1 are those servers that get time from stratum 0 level. Servers at stratum 1 act as time synchronization source for stratum 2 servers. In NTP client sends multiple requests to the server and stores the pair of offset and delay for later calculations. The main disadvantage of NTP is that it needs to send multiple messages to the server for synchronization.

2.2 Reference Broadcast Synchronization

The Reference Broadcast Synchronization (RBS) [4] protocol utilizes the concept of broadcast nature of wireless communication. According to this property, two receivers located within listening distance of the same sender will receive the same message at approximately the same time. It utilizes the concept that when we send the message using the physical layer than it will arrive at different receivers at the same time. The propagation delay in sensor network is very minimal and the range of sensor network is very low. Therefore the message send from physical layer will arrive at same time to different nodes. When the nodes receive the message they will record the time of arrival of that message and compare their clock with each other. This process will allow them to synchronize at high degree of precision. This protocol uses a sequence of synchronization messages from a given sender in order to estimate both offset and skew of the local clocks relative to each other. The protocol exploits the concept of time-critical path, that is, the path of a message that contributes to non-deterministic errors in a protocol. Fig. 1 and Fig. 2 compare the time-critical path of traditional protocols, which are based on sender-to-receiver synchronization, with receiver-to-receiver synchronization in RBS. The delays that occur at the sender side are eliminated by using the physical layer broadcast in sensor networks. In general, the time involved in sending a message from a sender to a receiver is the result of the following four factors, all of which can vary non-deterministically.

Send time: The time spent by the sender for message construction and the time spent to transmit the message from the sender's host to the network interface.
Access time: The time spent waiting to access the transmit channel.
Propagation time: The time taken for the message to reach the receiver, once it has left the sender.
Receive time: The time spent by the receiver to process the message.

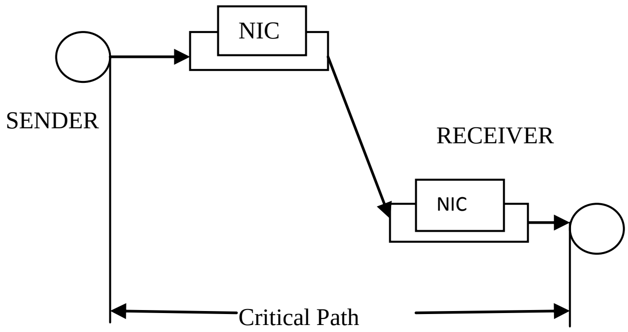

Figure 1: Critical path analysis for a sender receiver protocol

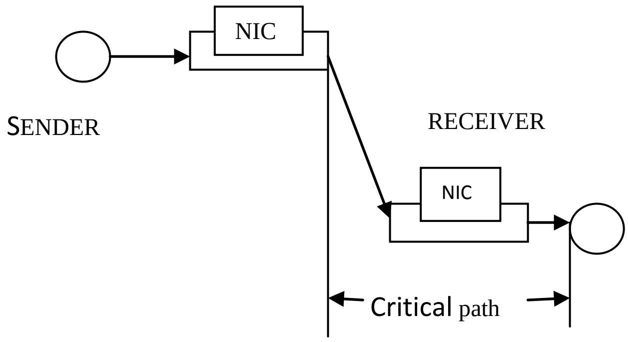

Figure 2: Critical path analysis for RBS protocol

The RBS protocol removes two of the sources of non-determinism involved in message transmission i.e. the send time and the access time, as it consider only the times at which a message reaches different receivers. Thus, this protocol can provide a high degree of synchronization accuracy in sensor networks. Algorithms are used in RBS to estimate the phase offset between the clocks of two receivers. RBS protocol can produce highly accurate if each receiver can record its local clock reading as soon as the message is received. This is often the case for single-hop communication in a wireless network. In real scenario it is possible that messages sent over a wireless sensor network can be corrupted. The reason may be that node was busy with other computations when the message arrived. To solve this problem RBS protocol uses a sequence of reference messages from the same sender, rather than a single message. Any Receiver say 'x' will compute its offset relative to any other receiver 'y' as the average of clock differences for each packet received by nodes 'x' and 'y'. The main advantage of RBS is that the largest sources of error (send time and access time) are removed from the critical path by using physical broadcast medium of the sender. The disadvantage of using RBS is that the protocol is not applicable to point-to-point networks, a broadcasting medium is required. As the network size grows the requirement of message exchange also grows.

2.3 Flooding Time Synchronization Protocol

FTSP protocol achieves time synchronization with very low error rate. It is also scalable up to hundreds of nodes due to its flooding property. This protocol is robust to network links and nodes failure due to any reasons. The FTSP maintain accuracy with MAC layer time-stamping algorithm and skew compensation with linear regression method [3]. It assume that each node has a local clock that can have timing errors due to nature of crystal clock and can communicate over an unreliable wireless link to its neighbors. In FTSP the synchronization between a sender and multiple receivers is obtained using a single message that is broadcasted by the sender to multiple receivers. The message is time stamped by both the sender and receivers at their MAC layers. The use of this type of time stamping i.e. at MAC layer will eliminate many errors. Linear regression is used in FTSP to compensate for clock drift. Typical WSN operate in areas larger than the broadcast range of a single node. Therefore the WSN generally are multi hop in nature. The root of the network is a single node that is dynamically elected. This root node maintains the global time and synchronizes other nodes to the clock of root node. The FTSP use the ad-hoc type of structure as compared to the spanning tree type approach used in TPSN. The approach of FTSP saves the initial phase of maintaining the tree. FTSP approach is more robust to different failures reasons in the network due to its flooding nature. In RBS protocol the time stamp is not incorporated in the message that is broadcast where as in FTSP the time stamp of the sender is embedded in the currently transmitted message. Therefore, the time-stamping on the sender side must be performed before the message is transmitted to the receiver. In the real scenario if we have the clocks that works on same frequency always, setting the clock offset once will be sufficient. But this is not the case and we need to send synchronization message again and again. However, the frequency differences of the crystals used in Mica2 motes introduce drifts up to 40µs per second [3]. In FTSP, it is mandatory to resynchronize the network with a period that is less than one second. If we do the resynchronization, less than a second than only we can achieve the accuracy within microseconds. Otherwise the clock looses their accuracy due to skew in them. The main disadvantage of FTSP is that sending resynchronization message very shortly is a big overhead in terms of energy as well as it also utilize a lot of bandwidth in the network

2.4 Timing-Sync Protocol for Sensor Network

Timing-sync Protocol for Sensor Networks (TPSN) [5] is a sender-receiver approach whereas RBS works on receiver-receiver approach therefore the time stamp is not required in the RBS network. TPSN require time stamping with in the broadcast message to make it work. TPSN claims that, for sensor networks the handshaking type of approach is much effective as compare to receiver-receiver synchronization. This result came when we time stamp the packet at Mac layer. Time stamping at Mac layer is feasible in sensor networks. An effective and simple solution of time synchronization is provided by the TPSN for sensor network.

TPSN has two different algorithms to provide synchronization. First algorithm is called "level discovery phase" in which levels are assigned to the node. The root node is assigned a level 0 and other nodes are assigned level according to their distance in hops from the root node. The node at level x must communicate only to node at level x-1. Even the author claim that this "level discovery phase" is not only required for time synchronization only. Much other application like localization, target tracking and aggregation used this phase and therefore this phase should not be considered as an overhead to TPSN. After the implementation of first algorithm, the second algorithm starts its work. This algorithm is known as "synchronization phase" which actually synchronize the node at level x with node at x-1. This synchronization process repeated and finally the whole network get synchronized. Each node in the network is synchronized to a single node called root node and in this way network wide time synchronization is achieved. Root node is one that acts as interface between the external world and the sensor network. The root node can also be equipped with GPS receiver to synchronize the sensor network to the real world. In "synchronization phase" pair wise synchronization is performed. The synchronization is performed along the edges of the hierarchical structure that is maintained by the level discovery phase. The figure below analyzes the message exchange between two nodes.

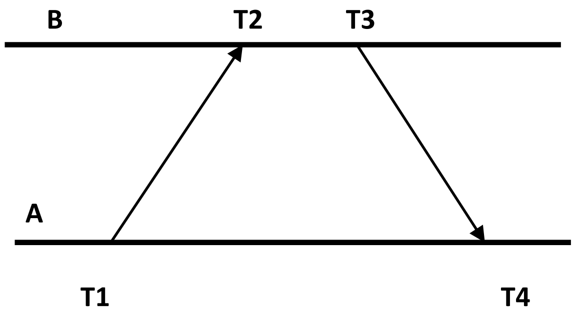

Figure 3 Two way message exchange between nodes

Figure shows this message-exchange between nodes 'A' and 'B'. Here, T1, T4 represent the time measured by local clock of 'A'. Similarly T2, T3 represent the time measured by local clock of 'B'. At time T1, 'A' sends synchronization packet to 'B'. The synchronization packet contains the level number of 'A' and the value of T1. Node B receives this packet at T2, where T2 is equal to T1 + D + d. Here D and d represents the clock drift between the two nodes and propagation delay respectively. At time T3, 'B' sends back an acknowledgement packet to 'A'. The acknowledgement packet contains the level number of 'B' and the values of T1, T2 and T3. Node A receives the packet at T4. Assuming that the clock drift and the propagation delay do not change in this small span of time, 'A' can calculate the clock drift and propagation delay as:

$$\Delta = \frac{(T2 - T1) - (T4 - T3)}{2}$$

$$d = \frac{(T2 - T1) + (T4 - T3)}{2}$$

When the drift is calculated, node A can correct its clock; so that it synchronizes to node B. TPSN is a sender initiated approach, where the sender synchronizes its clock to that of the receiver. The root node first sends the packet to all. This packet is called time synchronization packet that intimates that root is ready to distribute message. When a node receives the packet it waits for some time and then request the time from the root node. The concept of random wait is incorporated so that more than one node on receiving the time synchronization message should request the time at a same moment. Root node replies the request for time stamp with a message that contain time stamp. Sensor node on receiving the message synchronizes itself to the root node. The process is repeated for each level and network wide synchronization is achieved. Another property of TPSN is that it is flexible enough that one can tune it to meet the desired level of accuracy in the network.

3. RELATED WORK

In this section, we are again going to review the literature. This time we are going to provide the results that are already provided in different research papers. Many of the research articles provide a new method for synchronization and compare their protocol with existing ones. The results from such articles with reference to our simulation results are covered in this section.

In Flooding Time Synchronization Protocol, every node must broadcast its time stamp message to the other node in a hop area upon receiving beacons, so that they can estimate the relative clock offsets among each other. The performance of FTSP is very good as compared to other protocols, but experimental results shown in paper [11] proves that as the hops increases in multi hop network the accuracy of FTSP decreases. The efficiency of FTSP is checked with the help of simulation and it provides the decrease in accuracy where the number of hops is more [12]. Therefore FTSP can be used in the networks where the number of hops is few. In our simulation scenario the decrease in accuracy is not much visible due to limited number of hops. Our model assumes only 2 hop network and hence can not show the decrease in accuracy of FTSP. Implementation of FTSP on mica and mica2 for experimental results show that FTSP removes various jitters as compared to RBS with the help of time stamping at the MAC layer [3]. Our simulation model also proves the same concept, where we show that the accuracy of FTSP is much better than that of RBS protocol.

NTP is a famous synchronization method for setting the clocks of computer. In our research work we simulate the NTP on sensor networks. NTP is developed for computer network, assumes that the sending and receiving of the messages in the network is free. In paper named post-facto synchronization [16], the accuracy of NTP is found to be very accurate at the level of 1 microsecond. Our simulation results also prove that the accuracy provide by NTP is greater than all the other protocols. As sensor network are very energy constraints therefore NTP is not fit for sensor network despite the fact that it provide high accuracy. As discussed in article [13], wireless sensor nodes have onboard memory, for example Crossbow's Mica2 and Mica2Dot sensors. A large the sensor network has large requirement of memory. These time synchronization messages will require a lot of space and leave little space to hold other data that is monitored by the sensor. This will restrict the use of NTP for WSN.

TPSN proposed by S. Ganeriwal uses two way communications that increase the message load on the network. The first phase is the root discovery phase and other is the synchronization phase. The output result shows that the accuracy of TPSN is more than RBS which is also concluded in other papers [14]. The accuracy of TPSN is lower than FTSP in our results, which is due to the fact that TPSN does not loose the accuracy in multi hop network, where as FTSP looses the accuracy with increase in hops in the network. Root discovery phase in TPSN generates extra messages. The flooding used in FTSP eliminates the need of root discovery phase. Due to this root discovery phase the TPSN is in need of more messages as compared to FTSP and hence more energy consumption results. TPSN is a time synchronization method that provides a balance between the energy consumption and accuracy. It consumes more energy as compared to FTSP but it provides accuracy much better than FTSP [5] in multi hop networks. The performance analysis of RBS and TPSN is done in paper [5]. The author claim that the accuracy of RBS is not achieved with one synchronization message. Our simulation results show the same output that the RBS accuracy is increased with increase in resynchronization time.

RBS does not perform any time synchronization based on UTC or other external source. This eliminates the error occurrence due to the sending and accessing time. In the case of multi hop network it is again not as suitable as TPSN is because the common node that acts as interface between the two hops if dies, need to be chosen again. The TPSN due to its level discovery phase overcome these issues. The chances to die of common node are more as it is more involved in time synchronization. More a node is involved more is the chance to exhaust its energy and sooner it dies. Moreover the working of RBS in multi hop network is not suitable for the network as discussed in paper [14]. The life time of sensor is shorten with multiple messages are being exchanged between the nodes. Similar kind of results is available with our simulation. Another factor that affects the credibility of RBS is that it requires more resynchronization to achieve its best accuracy. Our simulation result prove that the accuracy of RBS keep on increasing with resynchronization time. It is not possible to achieve its accuracy with the single synchronization message and its accuracy keep on increasing with the increase of resynchronization messages. In research article [15], the work is based on the assumption that by reducing transmission rate the energy used can be improved and hence increase the network life time. The goal was not to make high accurate time synchronization but to make it energy efficient. The comparative energy efficiency analysis is performed over Mica motes. Motes use TinyOS operating system. For comparison purpose protocols are set with each parent node with 3 or less children. They conclude that sender-receiver synchronization is good for large networks and receiver-receiver synchronization work better in smaller network. In our simulation criteria a small network is modeled and RBS which is based on receiver- receiver synchronization also perform well in our model.

In multi hop ad-hoc networks, a depleted sensor drop information that came from other sensors node through it. This factor affects the area monitored by the network. Considering the various situations the performance of TPSN and RBS is compared on mat lab [5] and a new hybrid type of protocol is proposed. The author compares their results on different parameters and according to their result a new hybrid type of algorithm is devised. We refer to the paper for the purpose of comparison results that they found between TPSN and RBS. TPSN and RBS both achieve accurate clock synchronization but they both worked for small network. Although these algorithms will work for large network but as the nodes start losing power they become inefficient. TPSN does not perform well in sparse network where as RBS deteriorates in denser network. Our simulation results show that the TPSN accuracy is more than that of RBS network in initial phases. Our model is not a dense network and it comes under the category of sparse network where RBS outperform TPSN in later resynchronization phases.

4. SYSTEM MODEL & RESULTS

The network consists of a root node and many sensor nodes. The nodes that are in reach of root node are called common nodes as they are the nodes that act as interface between the end sensors and the root node. The time request from the root node is routed to rest of the network with the help of these common nodes. Delays are calculated according to the protocols. We also assume that the messages generated for the synchronization are of same length. The common nodes are in the range of root node and the end nodes are in the range of common nodes. Root cannot send a message directly to the end nodes as they are not in the reach of the root node. We named root node as sink node according to the name proposed in the simulating protocol. Each protocol needs to send the synchronization message again and again. The value of resynchronization time can be set by the user. Before starting the simulation the resynchronization time is entered by the user for that protocol. After that specified time the network again forces the root node to send the synchronization message.

- The simulation model chosen for protocols is described as:
- The nodes were randomly deployed over simulation area.
- The Area in which nodes were deployed was 500 X 800 unit area. (Simulator specific)
- The resynchronization time message was initially 30s. It is a tunable parameter which can be set accordingly ("s" represent simulation time here)
- The range for delays was chosen with distribution over the range of 10 to 600.
- The number of nodes chosen was 21 for simulation purpose.
- The transmission range of the sensor nodes chosen was 100 units. (Simulator specific)

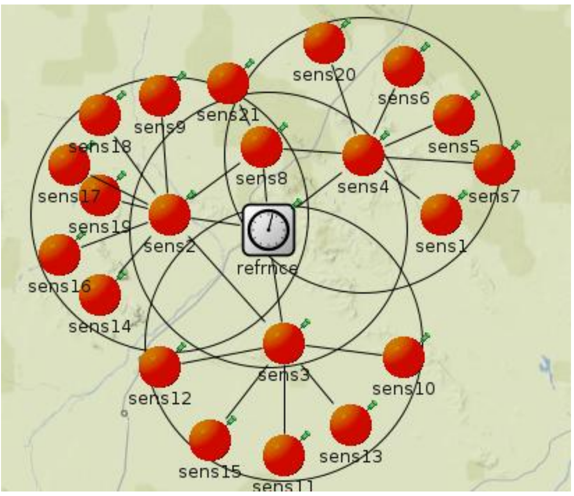

Figure 4: System model for WSN

In the Figure 4 which is actually the snapshot of the network we model on our system for the purpose of simulation The wireless sensor network is one in which the sensors are not connected to each other but they send the message to each sensors which are in the range of a given sensor. In our model we use round circles to show the transmission range of a specific sensor. The particular sensor can only send the message to those sensors which come under the area of this circle. The lines connected to sensors are only to show that they can communicate, as in real scenario they are wireless and no such line connecting sensor exists.

The following parameters are assumed for the purpose of simulating various protocols. The parameters remain same for all the protocols. The simulation is tested for the resynchronization time of 30s, but it is a user specific parameter can be changed to other value also.

Table 1 Parameters for simulation

Parameter	Value
Max. simulation time	1000
Non deterministic delay range (micro sec.)	10-600
Transmission range (units)	150
Initial energy of the node (joule)	1000
Transmission Energy consumed (mille joule)	10
Receiving Energy consumed (mille joule)	5
Resynchronization time	30
Initial message at each node	0

In this section, we describe the results that we got after simulating the protocols on simulator. The results are collected and shown using line chart for clarity. The line chart we show contain four different lines of different colors. Each line represents the specific protocol and its performance on the parameter used for drawing that graph. The different results are shown as explained below.

4.1 Number of Messages vs. Resynchronization Time

Figure 5 shows the comparison of FTSP, NTP, TPSN and RBS Protocols. The graph is drawn between the resynchronization times vs. the number of messages. The vertical axis shows the number of messages that keep on increasing with resynchronization time. Horizontal axis represents the resynchronization time. We choose the time interval of 30 seconds for simulation and we observe the data up to 150 seconds for each protocol. The total of the messages that are created for a particular protocol at a particular moment when the synchronization ends is collected and shown in the graph. The simulation is run for different value of resynchronization time of nodes. The NTP line is far above than all other nodes, this is due to the reason that NTP generate more messages as compared to other protocols. The FTSP generate the lowest number of messages according to our results. The TPSN provide more messages in our simulation results, this is due to the fact that TPSN provide level discovery phase and that phase also need to send messages which are further added to the messages that are sent for the purpose of time synchronization. RBS produce more messages as compared to FTSP but lower than TPSN, as RBS need to exchange messages between two neighbors for comparing their clock timing for the purpose of time synchronization.

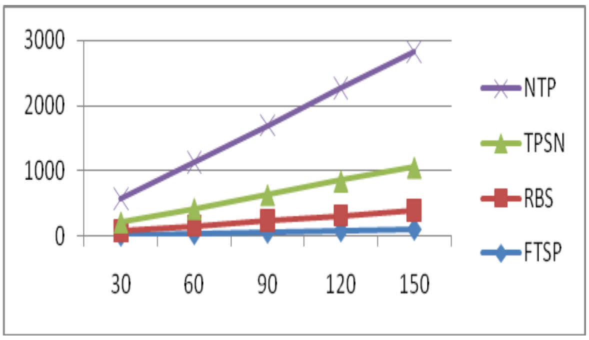

Figure 5 Number of messages (vertical axis) vs. Resynchronization time (horizontal axis)

4.2 Synchronization error time vs. Number of nodes

This line graph is used to show the performance of the time synchronization protocols for the parameter of accuracy. We draw the graph in which we show that that error decrease. As the error decreases between the two clocks, more accurate they become. The graph shows that the error in seconds for different protocols. The line of all the protocols go in a same direction, but the RBS line is coming downwards. This represents that the accuracy of the RBS increases with the increase of resynchronization period. The FTSP here maintain their clock at lowest error therefore, the FTSP shows better results as compare to other protocols. The network design in our project is just 2 hop network. Due to small number of hops in our simulation the accuracy of FTSP is not affected much. The accuracy of FTSP decreases with the increase in hops in the multi hop sensor network [15]. FTSP provides a good accuracy in our model. The accuracy of NTP is better than both TPSN and RBS, but as the resynchronization time increase the RBS outperforms NTP. This is due to the fact that more the resynchronization time occurs more is the time exchanges happen between two nodes. More the time exchanges are there more accurate the clock of sensors will be there.

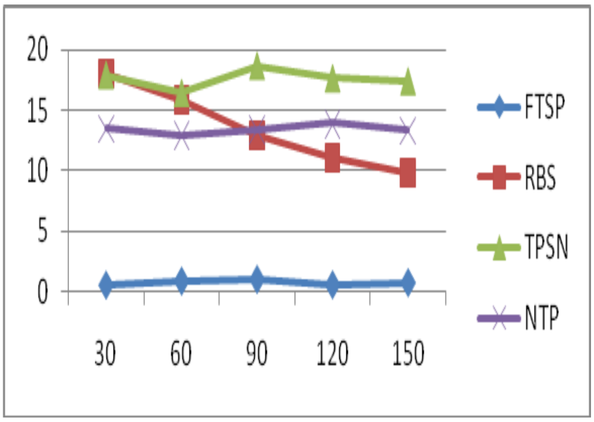

Figure 6 Synchronization Error in time (vertical axis) vs. Resynchronization time (horizontal axis)

4.3 Average energy consumption vs. Resynchronization time

This line graph is used to show the energy consumption of different protocols. We assume an energy decrement of 10 mille joule for a message sending and 5 mille joules for message receiving. As a message is received on a node we decrease the energy, again as soon as it sends the message we again decrease the energy of a node. The graph plotted here shows the average energy consumption of the network. This means that we are taking about the total energy of the network rather than for a single node. As the number of messages increases in the network with the increase in resynchronization time, similarly the consumption of energy also increases in the network. The average energy consumed of node at with different numbers is shown in figure 7.

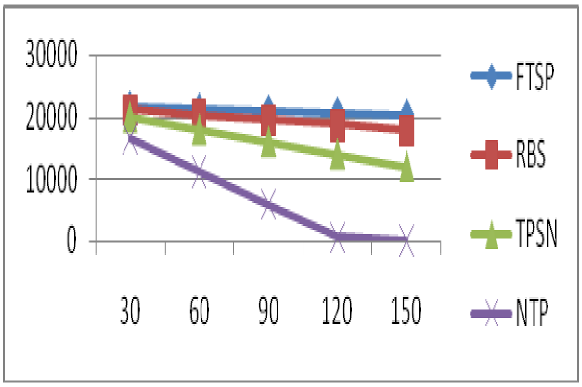

Figure 7 Average energy consumption (vertical axis) vs. Resynchronization time (horizontal axis)

The Figure shows that NTP make the system exhaust very fast as compare to other protocols. FTSP is the least energy consuming protocol according to our simulation results. As only one message is flooded to each node and no other step is followed to make it more effective, which reduce the number of messages and hence also decrease the energy consumptions of the protocol. RBS due to its time exchange consume more energy as compared to FTSP and TPSN due to its synchronization level needs to send more messages and therefore consumes more energy as compared to FTSP and RBS.

5. CONCLUSION AND FUTURE SCOPE

A wireless sensor network is a multi-hop ad hoc network of hundreds or thousands of sensor devices. The sensor nodes collect useful information such as sound, temperature, and light. Moreover, they play a role as the router by communicating through wireless channels under battery-constraints. Time synchronization is required in many application of distributed system. The comparison of various protocols and their results bring out some facts that can be used before deploying any network. NTP which is famous protocol for network also proves its performance in sensor network in term of accuracy but it lack in term of energy consumption. It consumes the more energy as compared to other protocols and therefore is not suitable for sensor networks. RBS is another protocol that lacks in accuracy first but its accuracy increases as the resynchronization increases. Therefore initial accuracy is not provided by this protocol and hence avoided in circumstances where initial accuracy is desired. FTSP perform very well in terms of energy and accuracy, but as discussed in previous section the accuracy decreases with increase in multiple hops. Therefore it is not fit for multi hop networks but provide a good accuracy with limited hop of network.

On the basis of the result a new Hybrid type of method that can take the features of these protocols can be designed. The ideas presented here could also be fully or partially applied to improve the performance of existing protocols. Experimental performance evaluation and comparisons with other existing protocols represent an open research problem.

6. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey, Computer Networks”, The International Journal of Computer and Telecommunications Networking, Vol.38, No.4, pp.393-422, 15 March 2002.

[2] D.L. Mills, “Internet time synchronization: The Network Time Protocol” In Z. Yang and T.A. Marsland, editors, Global States and Time in distributed Systems. IEEE Computer Society Press, 1994.

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

[4] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” SIGOPS Oper. Syst. Rev., vol. 36, pp. 147–163, 2002.

[5] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in Proceedings of the 1st Int’l Conf. on Embedded Networked Sensor Systems. ACM Press, 2003, pp. 138-149.\

[6] Alberto Cerpa, Jremy Elson, Deborah Estrin, Lewis Girod, MichaelHamilton, and Jerry Zhao. “Habitat Monitoring: Application Driver for Wireless Communitation Technology”. In proceeding of the 2001 ACM SIGCOMM workshop on Data Communications in LatinAmerica and the Caribbean, April 2001

[7] Leslie Lamport, “ Time, Clocks and the Ordering of Events in a Distributed System”, Communications of the ACM, Vol. 31, No. 7, pp. 558-565, July 1978

[8] B. Sundararaman, Ugo Buy, Ajay D Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey”, Ad Hoc Networks, Vol. 3, No. 3, pp. 281-323, 2005.

[9] Kopetz, H., and Ochsenreiter .W,” Clock Synchronization in Distributed Real-Time Systems.” IEEE Transactions on Computers, C-36(8), pp. 933–939, August 1987.

[10] Y. W. Fahmy, S. Shro, N.B. “Optimal Sleep/Wake Scheduling for Time-Synchronized Sensor Networks with QoS Guarantees”. In Proc. of the 14th IEEE International Workshop on Quality of Service, 2006.

[11] Tanmay Ratnaparkhe, Sagar Natekar, Sridhar Chandan , Vaishali P.Sadaphal. “Selection of Time Synchronizing Nodes in Wireless Sensor Network”

[12] Khurram Shahzad, Arshad Ali, N.D Gohar. “ETSP: An Energy-efficient Time Synchronization Protocol for Wireless Sensor Networks”, International Conference on Advanced Information Networking and Applicatoins.

[13] Xiaoya Hu, Bingwen Wang. “A Novel Energy-Balanced Time Synchronization protocol in Wireless Sensor Networks for Bridge Structure Health Monitoring

[14] Youngtae jo, Chongmyung park, Joahyoung Lee, Inbum Jung. “Energy Effective Time Synchronization inWireless Sensor Network”. Internationa conference on Computational Science and Applications.

[15] Hyojung Lee, Wonpil Yu and Youngmi Kwon. “Efficient RBS in sensor network”. Third international conference on Information Technology: New Generations (ITNG’06).

[16] J. Elson, D. Estrin, “Time Synchronization for Wireless Sensor Networks,” Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing, San Francisco, California, USA, April 2001.

[17] V.Classo, H. Lonn, N. Suri, “Efficient TDMA Synchronization for Distributed Embedded Systems” 20th symposium on Reliable distributed Systems (SRDS), pp 198-201, October, 2001.

[18] Nayyer Amit, Sharma Rajesh, Nayyer Meenakshi, Awasthi L.K. “Wait and Watch Technique:Removing Time Synchronization overhead in Wireless Sensor network”. International conference on Advance computing (ICAC) held at Thapar University, India, March 2009.