Computer Engineering and Applications Vol. 4, No. 2, June 2015

Proposing a novel method for clock synchronization by Reducing the Number of Synchronization Messages and Eliminating Non-Deterministic Errors in Wireless Sensor Networks

Seyed Kazem, Kazeminezhad[1], Shahram, babaie[2], Amir, Shiri[3]

[1, 2, 3] *Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran*
kkazeminezhad@yahoo.com, hw.tab.au@gmail.com, amir.shiri.tab.au@gmail.com

ABSTRACT

Wireless sensor networks (WSNs) of spatially distributed autonomous sensors are used to monitor physical or environmental conditions such as temperature, sound, pressure, etc. Since WSNs have particular constraints and limitations, synchronizing the physical times for these networks is considered to be a complex task. Although many algorithms have been proposed for synchronizing time in the network, there are two main error factors in all the proposed algorithms. The first factor is the clock drift which might be caused by the influence of different environmental factors such as temperature, ambient temperature, humidity, it might be generated on crystal oscillator which is inevitable The second error factor is indeterminacy which is attributed to the existence of non-deterministic delays in sending and receiving messages between sensor nodes. These two factors together reduce the precision of synchronization algorithms. In this paper, the researchers proposed a new approach for dealing with the above-mentioned two problems and achieving better synchronization. The results of simulations conducted in the study indicated that the proposed approach is significantly more efficient than the FTSP and RBS methods in terms of parameters such as accurate synchronization, amount of sent packets and power consumption.

Keywords: Wireless Sensor Networks, Time synchronization, Scalable Time Synchronization, Clock Drift, non-deterministic error, Scalable Time synchronization.

1. INTRODUCTION

Synchronization processes are ubiquitous in nature and play a remarkable role in many different contexts such as biology, ecology, climatology, sociology, technology, and even in the arts [1,2]. Communication is regarded as one of the requirements for implementing and maintaining time synchronization [3]. Time synchronization communication models in WSNs include the followings:

In computer networking, unicast transmission refers to sending messages to a single network destination identified by a unique address. However, the term unicast contrasts with the term broadcast which refers to transmitting the same data to all possible destinations. One more multi-destination distribution method is multicasting which sends data only to interested destinations by using special address assignments [5]. Node A will receive messages sent by node B if and only if node B can receive messages sent by node A. In this case, it is mentioned that the link between these two nodes is symmetrical. Otherwise, it is asymmetrical [3]. In explicit synchronization, time synchronization messages only contain information about time synchronization. Nevertheless, in implicit synchronization, these messages can be mounted in a normal package [3]. In Master–slave synchronization, a master–slave protocol assigns one node as the master and the other nodes as the slaves. The slave nodes consider the local clock reading of the master at the reference time and attempt to synchronize their times with the master. In general, the master node requires CPU resources proportional to the number of slaves and nodes with powerful processors or lighter loads are assigned as the master node [7]. In contrast, in peer-to-peer synchronization, any node can communicate directly with every other node in the network. This eliminates the risk of master node failure which would prevent further synchronization. Peer-to-peer configurations offer more flexibility but they are also more difficult to control [4,5,8,9]. In the Internal synchronization approach, a global time base, called real-time, is not available within the system. The internal synchronization approach aims to minimize the maximum differences between readings of the local clocks of the sensors [7]. In contrast, in external synchronization, a standard time source such as Universal Time is provided. Thanks to an atomic clock which calls reference time, there is no need for a global time base. Reference time provides the actual real-world time. In WSNs, most protocols do not use external synchronization unless the application demands it [7]. In fact, clock drift and uncertainty are considered as the two main error-causing factors in all the algorithms proposed for time synchronization in WSNs. The existence of these two factors together will reduce the algorithm synchronization precision. Hence, in this paper, the researchers aim to propose a new approach which can help eliminate these generic error factors. The approach proposed in the present study combines the two efficient approaches of FTSP and RBS.

The rest of this paper is organized as follows. Related works are presented in section 2. Details of the proposed method are discussed in section 3. Section 4 provides an evaluation on the simulation results. Finally, in section 5, we conclude the paper and discuss future research plan.

2. RELATED WORKS

Time synchronization techniques in WSNs can be divided into two types: traditional synchronization techniques such as remote clock reading [12] and delay estimation method proposed by network time protocol [10] and Advanced synchronization techniques such as RBS[5,6,13] and FTSP[11] techniques. In [12] when a process requires the approximate time of another process, it sends a request and waits for a response from the target process. When it receives a response, it calculates the message round-trip based on the moment the request message is created until it receives a response. The reply message includes the approximate time of the remote procedure. When the process receives a reply message, the source process adjusts its time to the received approximate time in addition to the message half round-trip. The offset delay estimation method is used by the network time protocol [10] has many applications in the internet. This method is based on the strategy used in the [12] and is intended to achieve the remote node time for synchronization. The offset delay estimation method is similar to the Christine method [12] in the averaging policy. In the RBS protocol, serious changes were realized in the network time protocol which led to the improvement of these protocols. This algorithm broadcasts time information. Unlike the traditional methods, the set of receivers in the RBS method are synchronized with each other. In this protocol, nodes periodically send messages to their neighbors by broadcasting on the network at the physical layer. In this message, there is no time and receiver nodes use these messages to locally synchronize themselves. The main limitation of RBS protocol is that it requires a network with a broadcasting physical channel [5,6,13]. Flooding time synchronization protocol (FTSP) aims to achieve a vast network in which the participating nodes are synchronized with the local clock. FTSP protocol makes it possible for a sender to synchronize the times of several nodes with a radio message that has a timestamp [11]. Mill’s NTP [13] is regarded as an outstanding method and is significantly better than many other methods. The followings are the advantages of NTP: scalability, robustness against different types of failures, self-configuration, security in the face of intentional sabotage and ubiquitous deployment. NTP allows the construction of a hierarchy of time servers, multiply rooted at canonical sources of external time. Time Transmission Protocol (TTP) [4,14] is used by a node to communicate the time on its clock to a target node. Then, the target node estimates the time in the source node by using the message timestamps and message delay statistics.

3. THE PROPOSED APPROACH

WSNs are used in diverse applications such as the measurement of humidity, temperature, etc, in which time synchronization is a critical requirement. Since each event should be labeled exactly with respect to the reference time, maintenance and repair of a real-time clock is essential for data fusion in WSNs. Based on the conditions of the environment, the reference time may be made local or global. In all the algorithms proposed for network time synchronization, both of the above-mentioned error causes are taken into consideration. As discussed earlier in the paper, the first error-causing factor is clock drift which is inevitable and may be generated by the influence of different environmental factors such as temperature, ambient temperature, humidity and so on. The second factor is non-determinism and uncertainty which is caused by the presence of non-deterministic delays in sending and receiving messages between the sensor nodes. These two factors together reduce the precision of synchronization algorithms.

In this section, the researchers introduce and explain the novel proposed protocol to develop an effective method for clock synchronization. For this purpose, the outline of the proposed algorithms will be given and then how this algorithm can reduce the number of synchronization messages will be explained. Then, the improved accuracy of the clock at network nodes will be discussed which will enhance the energy efficiency and lifetime of the network.

3.1 THE INITIAL PLAN OF THE PROPOSED PROTOCOL

The proposed protocol is based on one-hop FTSP protocol. The drawback of the FTSP protocol is related to synchronization networks which use more hops to transfer data. It is evident that producing and replicating so many synchronization messages is due to the flood of messages. The creation and repetition of a message in the network not only results in the problem of transmission congestion but also spends a lot of bandwidth for transmitting redundant messages; consequently, it leads to the consumption of more power in network nodes for transmitting and receiving messages. With respect to the above-mentioned problem, the proposed algorithm is intended to create a dynamic tree in the network level. Then, using parent, it will deliver messages to its children in the tree. Hence, the proposed algorithm will extend the one-hop approach used in the FTSP protocol. The proposed protocol is based on clustering and hence it is known as the clustered FTSP or CFTPS.

3.2 TREE INFRASTRUCTURE

The proposed method needs to select a specific node to form a tree so that it can transmit the synchronizing messages in the network. As a result, network nodes are divided into four main categories as follows:

a) **Reference Node**: These nodes in the network play the role of reference time and can be equipped with a global positioning system (GPS). Thus, based on the accuracy requirement of each application, this node can provide external synchronization in the network.
b) **Active Node**: These nodes are the nodes selected in the tree creation phase and their main task is to transmit synchronization messages over the network. It should be noted that the transmission range of all the active nodes must cover the entire network. To ensure complete coverage, the researchers defined a control packet called "start building the tree" for finding and selecting active nodes in the network and starting or continuing the tree making phase.
c) **Passive Node**: These nodes are receiving the synchronization messages which include active network nodes; they synchronize their time with the active nodes. Passive or inactive nodes do not send any synchronized message in the network.
d) **Candidate Node**: Before taking the role of active or passive node in the network, network nodes are referred to as candidate nodes. At network start up-time, all the nodes in the network, except for the time reference nodes, are in this situation. In fact, the candidate node is nominated to become active node.

3.2.1 HOW TO CREATE TREE INFRASTRUCTURE

At the beginning, there is only one reference node and there are a number of candidate nodes. By choosing active nodes among candidate nodes, the reference node starts the first step for making synchronization infrastructure. The tree Infrastructure consists of a set of active nodes which are regarded as the backbone of the network. They disseminate information about the reference clock throughout the network which is necessary for synchronizing the clocks of all nodes in the network. Tree infrastructure is a basic structure used in CFTSP for reducing the number of synchronization messages. The reference node is responsible for starting the process of tree infrastructure construction by broadcasting the "start building the tree" message. From the beginning of the tree construction process, the message "Start building the tree" is responsible for informing the candidate nodes and selecting the active node in the network. This message contains the flag sender's ID for the beginning of the tree construction process.

3.2.2 SELECTING ACTIVE NODES

As mentioned earlier, the "Start building the tree" message provides a mechanism for selecting active nodes in the network. Choosing the active node in the network is accomplished during the process of building the tree as follows: each node has an inverse sequence counter. Inverse sequence counter is a tool for selecting the active node among the candidate nodes. As each candidate node receives the "Start building the tree" message, it initiates counting by a random number and activates it. The node whose inverse sequence counter is activated when it received the first "Start building the tree" message and its counting is finished before it gets the counting repetition message from its neighbor is selected as the active node.

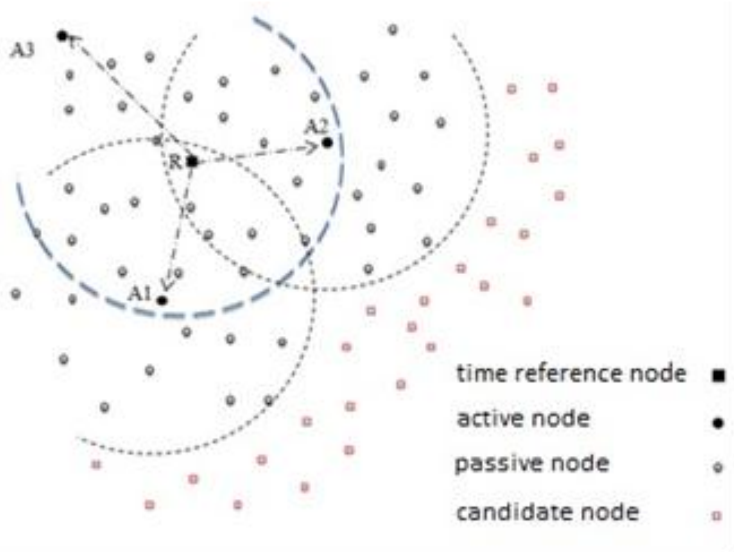

FIGURE 1. An example of Tree Infrastructure

Figure 1 illustrates a tree infrastructure in a hypothetical network. By broadcasting "start building the tree" message in the network, the nodes receiving this message are going to change their status to that of either active node or passive node. Active nodes choose the message sender as their parent node and broadcast the received message to inform other neighbor nodes about their own status as active nodes. Those nodes which receive "Start building the tree" message and its iteration more than once are considered to be passive nodes. Then, these nodes select the sender of the original message as their active parent node and consider others as non-parent active nodes. When inactive nodes receive synchronization messages from the active parent node and active non-parent nodes, they carry out the time synchronization process. Figure 2 demonstrates the pseudo code for selecting active and passive nodes.

```
on tree Setup Message received()
{
   if (me -> state = candidateNode)
  {
      if (rcvdMsg ->creatorID = rcvdMsg ->senderID)
     {
        ADD rcvdMsg to Tree SetupMsgCollection
        ADD an EVENT to selectionTimer(rcvdMsg,
        random(1, 30))
        return
     }
```

FIGURE 2. The pseudo code for selecting active and passive nodes

3.3 THE SYNCHRONIZATION PROCESS

As mentioned before, the proposed method can synchronize clock in the network in two phases. In the first phase, active nodes in the network are synchronized by time reference nodes. Active nodes that are far from the reference node are synchronized by the help of active nodes which are near the reference node. Then, all the passive nodes are synchronized by the active parent nodes and active non-parent nodes. Both of these steps are described in the following review.

3.3.1 SYNCHRONIZING ACTIVE NODES

In the beginning, the reference node starts creating tree infrastructure by sending a “Start building the tree” message. Hence, active nodes in the first level are selected. Next, due to the distributed nature of the proposed approach, reference node does not have to wait for tree infrastructure to be completed; it attempts to send the synchronization message to synchronize the active nodes in the first level. That is, by sending multiple messages which include timestamps, the reference node attempts to fill the synchronization table of the message receiver. Meanwhile, active nodes in another level temporarily don't send synchronization messages after choosing active nodes in the next level of tree infrastructure. Since sending trigger messages based on a local asynchronous clock causes large errors in the network and misleads passive nodes which are close to each other, active nodes wait for the reception of synchronization messages from its active parent nodes or active non-parent nodes. After receiving a good number of synchronization messages, they, then, attempt to calculate their local time and are prepared to send synchronization messages.

3.3.2 PREVENTING THE PROPAGATION OF CUMULATIVE ERROR IN ACTHVE NODE SYNCHRONIZATION

Since the nodes out of the range of reference node are synchronized by using active nodes near the reference node, the aggregation in propagation of cumulative error problem in multi-hop synchronization should be taken into consideration. As we get farther away from a reference node, the number of steps in synchronization increases and average clock error increases. This is explained by the fact that nodes synchronize their clocks by using time information which has more errors. In particular, if the clock slope gets bigger, more synchronization process will be needed to keep the clock synchronized all over the network. However, it should be noted that the frequent repetition of the synchronization approach reduces the energy efficiency of the nodes and the whole network. Hence, with respect to the above-discussed issues, the researchers proposed a technique for reducing the propagation effect of clock slope. In this way, during the synchronization process, each node in the tree infrastructure announces its clock slope to the parent node as a special synchronization message. In as much as clock slope represents the relative ratio of the clock frequency of two nodes, each node can correct the ratio of its clock slope to the reference node through multiplying its parent clock slope by the ratio of its clock slope to that parent. For example, as it is shown in figure 3 the calculated clock slope between node A and node B is denoted by S_{AB} and the slope between node B and node C is indicated by S_{BC}. Thus, if node A is aware of the S_{AB}, it will simply calculate its clock slope to node A through multiplying S_{AB} by S_{BC}.

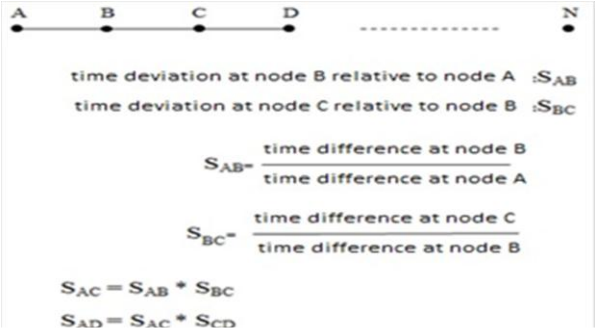

FIGURE 3. Dissemination of clock slope and its formulas

3.3.3 SYNCHRONIZING PASSIVE NODES

In the second phase of the proposed protocol, when passive nodes receive synchronization messages, they synchronize their clocks with the sender node of that message. The protocol used to synchronize the nodes in this phase is for one-hop networks just like the FTSP protocol. As it was discussed earlier, the tree infrastructure ensures that each passive node receives synchronization messages from at least two active nodes where one is the parent node and the other is the active brother node in the same cluster. It should be noted that synchronization messages are received by at least two active nodes. Hence, these messages are likely to be sent from a greater number of active nodes in the network such as the messages sent from the active nodes of the neighboring cluster nodes to the passive nodes in the network.

3.4 MANAGING THE MESSAGES RECIVED FROM MULTIPLE ACTIVE NODES

All the active network nodes which have already been synchronized with the reference node or the parent nodes send their synchronization time to the adjacent nodes by broadcasting "Start building the tree" message. These synchronization messages are received by both the active nodes of tree infrastructure and the passive nodes which are in the range of the sender node. Hence, multiple synchronization messages might be sent from a few active nodes to a passive node in a short time. Thus, using an appropriate way, the recipient must manage the incoming messages to calculate its accurate local time. The received messages can be divided into two groups as follows:

1) **Synchronized messages received from the source node or the parent node:** in this Case, the sender ID inside a message is equal to the active parent node ID and the receiver accepts the incoming message and registers it in the FTSP synchronization message table.
2) **Synchronized messages received from active non-parent node:** In this case, the sender ID inside a message is not equal to the active parent node and the receiver accepts and records the message using a different decision criterion which includes the followings:
   a) **The level value inside the received message is smaller than the level value of the node itself:** It indicates that the received message is sent by an active node that is closer to the reference node than the current node. Due to the proximity of this node to the reference node, it is expected that the clock accuracy in the received message is probably high. Thus, data packets are recorded in the FTSP table of incoming messages.
   b) **The level value inside the received message is equal to the level value of the node itself:** It indicates that the received message is sent by an active node that is at a level the same as that of the node itself. In this case, if the timestamp distance inside message is equal to or less than the farthest timestamp inside FTSP table of receiving messages, the message is accepted and recorded in the table. Otherwise, the message is deleted without being recorded.
   c) **The level value inside the received message is bigger than the level value of the node itself:** It indicates that the received message is sent by an active node which is farther to the reference node than the current node. In this case, the received message will probably have more cumulative deviation. Hence, the information inside the received message can be trusted. Thus, this package is not accepted without checking its timestamp and it is finally deleted. Figure 4 illustrates the pseudo code for the node performance with respect to different messages.

```
on timeSYNCMessage received(){
if (me -> state = activeNode OR me -> state = passiveNode)
  {
     if (rcvdMsg -> creatorID = me -> activeParentID OR rcvdMsg -> creatorID < me -> level)
     {
              ACCEPTMESSAGE_ADDTOTABLE(rcvdMsg)
     }
     else if (rcvdMs -> creatorLevel = me -> level)
          if (rcvdMsg -> timestamp is in  between of lowest and highest timeStamp)
          {
```

FIGURE 4. Pseudo code for node performance with respect to receiving different messages

3.5 CALCULATING LOCAL TIME IN THE NODES

In this paper, averaging method was used to calculate the local time of nodes. Nevertheless, for optimizing this method, unfavorable data are discarded in the time averaging method by combining the CHAUVENET'S CARITERION method and using standard deviation; then, time averaging action is repeated. Doing this process has a remarkable effect on the creation of an appropriate reference time [15,16].

4. SIMULATION AND PERFORMANCE EVALUATION

The proposed algorithm is generally based on FTSP. In this section, the algorithm proposed in this paper is compared with FTSP and RBS methods. Indeed, these methods are regarded as the basic and important techniques in WSN clock synchronization.

4.1. NETWORK MODEL

When the simulation method was applied to CFTSP algorithm, the network system architecture was considered as a distributed event-based system. The network simulation topology was a 300 m*300 m squared region. 50 nodes were randomly distributed in this region. MAC layer of protocol runs IEEE802.11 protocol. The node transmission radius was 30 m. Each packet was 100 bytes. The simulation was repeated in 1,000 cycles and energy consumption was calculated based on table 1. The researchers assumed a simple model for radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier while the receiver dissipates energy to run the radio electronics, as shown in Figure 5. For the experiments described here, both the free space (d^2 power loss) and the multi-path fading (d^4 power loss) channel models were used based on the distance between the transmitter and receiver [17]. Thus, equation 1 was used to calculate energy consumption for transmitting a packet of l bits over distance d. Power control was used to invert the energy loss by appropriately setting the power amplifier. If the distance is less than a threshold d_o, calculated by equation 2, the free space model should be used; otherwise, the multi-path model is used. Energy consumption for receiving a packet of l bits is calculated according to equation 3.

$$E_{Tx}(L,D) = E_{Tx-ELEC}(L) + E_{Tx-AMP}(L,D) = \begin{cases} LE_{ELEC} + L_{EFS}D^2 & D < D_0 \\ LE_{ELEC} + L_{EMP}D^4 & D \geq D_0 \end{cases} \qquad (1)$$

$$d_0 = \sqrt{\frac{\varepsilon fs}{\varepsilon mp}} \qquad (2) \quad , \quad E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \qquad (3)$$

TABLE 1.
Simulation Parameters

Parameters	Values
Transmitter/Receiver Electronics	E_{elec}= 50 nJ/bit
Data Aggregation	E_{DA}=5 nJ/bit/signal
Transmit Amplifier (if dmax to BS < d0)	$\mathcal{E}_{fs}$=10 pJ/bit/m^2
Transmit Amplifier (if dmax to BS ≥ d0)	$\mathcal{E}_{mp}$=0.0013 pJ/bit/m^4
Data Packet Size	100 bytes
d_0	87 m
Initial energy of each sensor	3 Joules
Number of cycles	1000 Cycles
Statues of node mobility	Static
Radio Frequency	2.4GHz – ISM Band

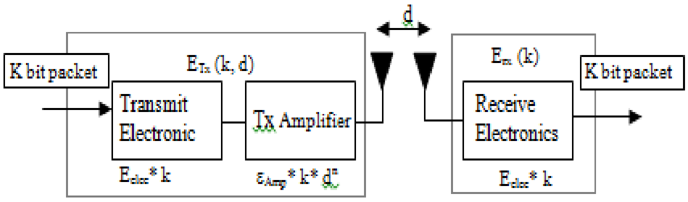

FIGURE 5. Radio energy dissipation model

4.2.THE SIMULATION RESULTS

In this paper, the efficiency of the proposed method was evaluated under the same conditions which applied to the RBS and FTSP algorithms. In other words, the mean difference in time to reference node in a few iterations of the three protocols and network lifetime was used. Figure 6 demonstrates the mean difference in time to the reference node in three protocols which were repeated ten times. It can be clearly observed from this figure that the behavior of the proposed algorithm is close to the FTSP protocol. In another analysis, the researchers attempted to evaluate the performance of the proposed algorithm with the different number of nodes. In Figure 7 mean time difference was used as the parameter for comparing the proposed algorithm with the other two algorithms in different networks with 10, 50, 100 and 1000 nodes. As it can be obviously noted in the figure below, the proposed algorithm has an acceptable behavior and in each phase it has a time difference close to 10ms which is significantly better than that of RBS algorithm. Another parameter which was taken into consideration in the present paper is the lifetime of the network. In the proposed protocol and the other two protocols, this parameter was operationalized as the consumption and finishing of energy over ten hours. The study was conducted in a network composed of hundreds of sensor nodes. Figure 8 depicts the stage in which the respective algorithms consume entire their energy. The loss of active nodes in tree infrastructure in the middle of simulation in CFTSP protocol was noted and it had a better behavior when compared with the other algorithms.

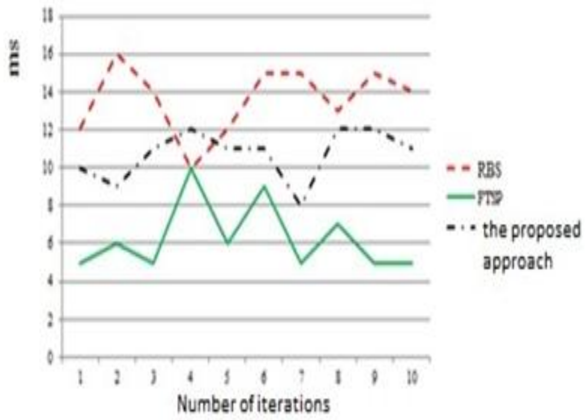

FIGURE 6. Mean Time difference compared to the reference node in three protocols, RBS,FTSP and CFTSP

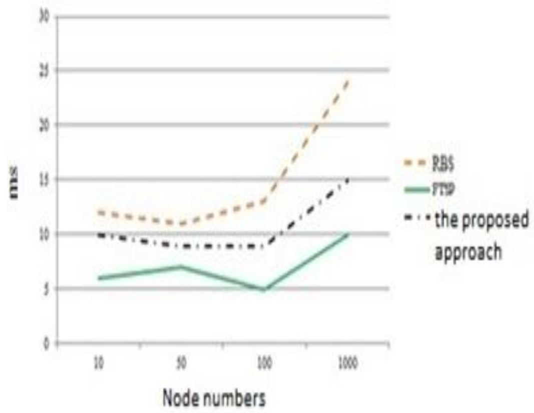

FIGURE 7. Mean Time difference compared to the reference node in four networks

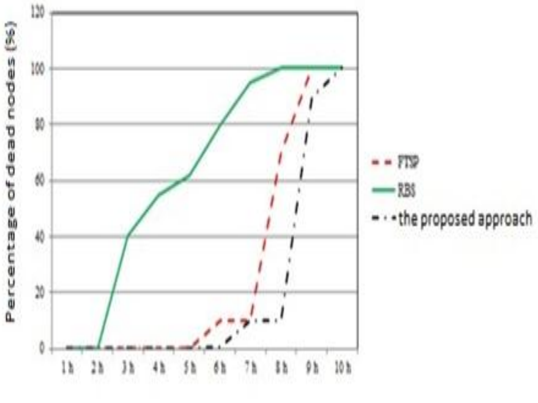

FIGURE 8. Dead nodes due to energy drain

5. CONCLUSION AND FUTURE WORKS

This paper proposed a new approach to achieve more accurate synchronization and eliminate indeterminacy errors in sending messages. It was aimed to improve the accuracy of synchronization and network lifetime. The simulation results of the present study revealed that the proposed approach had a better performance than RBS and FTSP methods in terms of the following parameters: accurate synchronization, the number of packets and power consumption. Therefore, to sum up the discussion, the researchers contend that future endeavors should be made to enhance synchronization accuracy by holding down power consumption in the network. Interested future researchers should take this significant issue into consideration.

REFERENCES

[1] IF. Akyildiz , and W. Su, "Wireless sensor networks: a survey," *jornul of computer network* , 2002, 393-422 [2].

[2] L. Paradis, and Qi. Han, "Fault Management in Wireless Sensor Networks: a survey, " *Journal of networks and systems management*, 2007, 171-190.

[3] I. Stojmenovic, *Handbook of Sensor Networks Algorithm And Architectures*, University of Ottawa, 2005.

[4] B. Sundararaman, and U. Buy, "Clock Synchronization for Wireless Sensor Networks: a survey, " *ELSEVIER, ad hoc network3*, 2005, 281-323.

[5] J. Elson, and L. Girod, "Fine-Grained Network Time Synchronization Using Reference Broadcasts, " Department of Computer Science, University of California, Los Angeles, 2002.

[6] F. Sivrikaya, "Time Synchronization in Sensor Networks, A Survey, " Department of Computer Science Rensselaer Polytechnic Institute, 2003.

[7] M. Mock, and Frings R, "Continuous Clock Synchronization in Wireless real-time applications, " *in proceedings of 19th IEEE symposium on reliable distributed system*, 2000, 125-133.

[8] S. PalChaudhun, and A. Saha, "Probabilistic clock synchronization service in sensor networks, " Department of computer science, Rice University, 2003.

[9] Q. Li, and D. Rus, "Global clock synchronization in sensor networks, " *IEEE*, Vol. 1, 564-574, 2004.

[10] D.L. Mills, "Internet time synchronization: the network time protocol, " *IEEE Transactions of Communications*, 1991, 39(10): 1482–1493.

[11] M. Maroti, and B. Kusy, "The Flooding Time Synchronization Protocol, " *In Proceedings 04, November 3-5,* Baltimore, Maryland, USA, 2004.

[12] F. Cristian, "Probabilistic clock synchronization Distributed computing3, ", 148-153, 1989.

[13] J. Hill, M. Horton, "The platforms enabling wireless sensor network, " *communications of the ACM*, 47(6), 41-46, 2004.

[14] K. Arrind, "Probabilistic clock synchronization in Distributed system, " *IEEE Transactions on parallel and Distributed systems*, 1994, 5(5): 474-487.

[15] P.F. Dunn, "Measurement and Data Analysis for Engineering and Science, " *Chapter 3 Analysis of Experimental Data*, 2005.

[16] H. Song, S. Zhu, and G. Cao, "Attack-Resilient Time Synchronization for Wireless Sensor Networks, " *Ad Hoc Networks*, 112-125, 2007.

[17] W.B. Heinzelman, and A.P. Chandrakasan, "Application specific protocol architecture for wireless micro sensor networks, " *IEEE*, 2002, 660-670.