ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 5, May 2015

An Efficient Protocol for Clock Synchronization in WSNs

Alina John, P Darsana

***Abstract*— Application of Wireless Sensor Networks often requires a scalable time synchronization service enabling data consistency and coordination. The proposed algorithm recursively synchronizes all the nodes in a network to a global clock using two-hop architecture in an efficient way. It achieves better performance due to the MAC-layer time-stamping, infrequent broadcasts by a dynamically elected reference node, compensation of the offset and adjustment of the time-stamps at each hop, estimation of the relative skew and offset using least square linear regression on two data points. Simulation results show that the computation time and setup time of the proposed algorithm is much lesser in the long run and performs better even in a clustered network.**

***Index Terms*— Clock, efficiency, estimation, protocol, synchronization.**

I. INTRODUCTION

Time synchronization is a crucial aspect of Wireless Sensor Networks for many reasons, such as precise time-stamping of messages, in-network signal processing, time-based localization, TDMA-based medium access, cooperative communication, co-ordinated actuation and energy efficient duty cycling of sensor node. There are several ways to achieve synchronization, which include ordering of events, local synchronization, global synchronization etc. However, many applications need global time synchronization which requires all nodes to be synchronized with one global clock.

The advances in micro electro-mechanical systems (MEMS) technology in wireless communication are leading to smaller, cheaper, low-power sensing and computing devices. Cell phones and hand-held computers already enjoy the increased popularity of the public. These trends point towards radically new systems of thousands or even millions of tiny computing devices interacting with the environment and communicating with each other. Research teams are working on incorporating sensing, processing and communication in a volume of less than one cubic millimeter, while devices of the size of a coin are commercially available already.

This paper describes an efficient Time Synchronization Protocol for WSN. Its design is robust to network topology changes. The proposed method compensates for the skew with linear regression using two data points, synchronizes to a global clock using two-hop architecture, and reference node election by considering remaining energy of the node. While these ideas have been utilized before, their unique combination and its effective implementation yield lower communication overhead than existing approaches on the same platform.

***Manuscript received April 30, 2015* .**

***Alina John**, Department of Electronics and Communication Engineering, Amal Jyothi College of Engineering, Kanjirapally, Kottayam, Kerala, India.*

***P Darsana**, Department of Electronics and Communication Engineering,Amal Jyothi College of Engineering, Kanjirapally, Kottayam, Kerala, India.*

II. RELATED WORKS

Several methods have been proposed for global time synchronization, such as GPS-based clock synchronization, Network Time Protocol (NTP) [1], Precision Time Protocol (PTP) [2], Reference Broadcast Synchronization (RBS) [3], Timing-Sync Protocol for Sensor Networks (TPSN) [4], Flooding Time Synchronization Protocol (FTSP) [5], and Recursive Time Synchronization Protocol (RTSP) [6].

III. PROPOSED METHOD

After WSN starts up, a random sensor node broadcast an enquiry message to ask their neighbors about the ID of the reference node, wait for a period T or until a reply is received. If the network is unaware of the ID of the reference node, dynamic election of a single reference node with highest remaining energy takes place and algorithm is used to compensate offset and drift.

Each node maintains a few variables including identification of reference node and identification of the nodes sending synchronization request through it. In the case of non-clustered or flat network, assume that each node is a cluster head in order to run the algorithm. This algorithm explains two functions, 1) election of the reference node 2) compensation of the offset and drift which are explained in the next two subsections.

A. Election of Reference Node.

The algorithm dynamically elects a single reference node as follows:

1. After startup, sensor node waits for a short time and then sends an enquiry message in order to ask the neighboring node about the ID of the reference node. The cluster head broadcast the message, but a non-cluster head sends the message to its own cluster head. If it does not receive a reply it enters into the contest for a new reference node by broadcasting its own identification number. If it receives a reply, it

saves the identification of the reference node.

2. If a new enquiry message is received by a node it looks the reference node ID field and does the following. If it is an enquiry message the receiving node replies with the ID of the reference node. If the received message is a contest or an announcement and the receiving node does not know the ID or it knows the ID of a node which has energy less than energy of receiving node then it updates its reference node ID field and rebroadcast the message. If the energy of the receiver node is higher then it broadcasts its own identification number in the reference node ID field.

B. Offset and Drift Compensation.

In order to compensate for the offset and drift, and to synchronize all nodes on request path to the reference node, the algorithm works as follows:

1. Node checks the type of newly received message.
2. If the message is a request for synchronization, it notes the receive time T2. If the receiving node is a reference node or a synchronized node it replies with the timestamps T1, T2, T3, and Tr along with other essential information. However, unsynchronized intermediate node forwards the request to the reference node directly after saving ID of its client and values of T1 and T2. Hence two hop synchronization is employed.
3. If the received message is a reply message it calculates the value of propagation delay by equation,

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \quad (1)$$

Then, it finds the new value of Tr by adding d to the received value of Tr as given by the equation,

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

Then it records the global and local time stamps where Tr=global and T4=local. If the node is an intermediate node, it receives the value of T1 and T2, calculates the value of Tr by adding it to the elapsed time since T4 using (3) and then forwards the reply to its client node, if any.

$$Tr = Tr + (T3 - T4) \quad (3)$$

The request and reply mechanism is as follows. A non-cluster head sends a synchronization message to its cluster head. The cluster head stores the ID of requested node, values of T1 and T2 and then forwards this request at its local time T1 to reference node. On receiving this request at T2, the reference node processes the request and send back reply message at T3. When cluster head receives the reply it calculates the value of d, adjusts time by adding d to Tr, retrieves the old T1 and T2, calculates new Tr by adding it to the elapsed time since T4 and forwards the reply to requested non-cluster head at its timeT3. The non-cluster head performs first 2 actions and record the global and local time. The reply follows the same path as request.

IV. Experimental Results

A. Simulation setup

Simulation was performed in MATLAB. The main steps of the simulation are as follows:

1. A class of sensor nodes is defined with ID, position, energy level, parent ID, offset and skew.
2. A set of network parameters such as number of nodes and processing delay at each node are defined.
3. Communication links are defined according to the transmission range and distance between nodes.
4. Clusters are defined.
5. Results are plotted.

B. Results

The simulation was carried out to collect data on efficiency in the long run. Efficiency in computation and set-up time is a very important factor in the performance of time synchronization algorithms. However, actual efficiency may be affected by many factors such as type of the hardware, software and antenna. Therefore, efficiency is measured while assuming same type of hardware, software and antenna for all protocols.

The sensor nodes are deployed randomly over a geographical area as in the figure 1. On startup of the network, reference node and cluster heads are elected as shown in figure 2 and 3 respectively. Then these cluster heads are assigned to remaining sensor nodes based on distance between cluster head and sensor nodes. This is shown in figure 4. Reference node broadcast its ID to cluster heads as in figure 5. A non-cluster head forward request messages and gets their replies accordingly as in figure 6 and 7 respectively. From the reply message the node calculates propagation delay and global time and synchronizes it with the reference node.

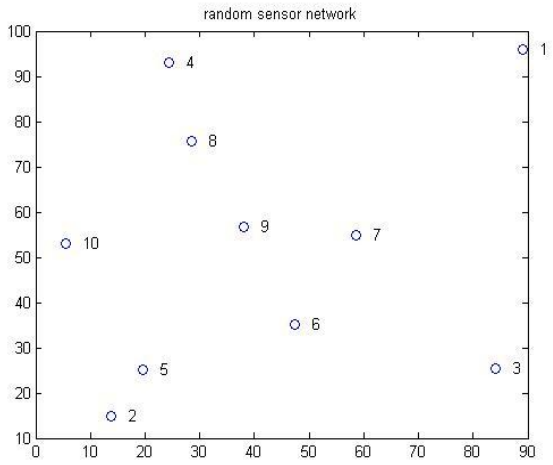

Figure 1. Sensor nodes are deployed randomly.

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 5, May 2015

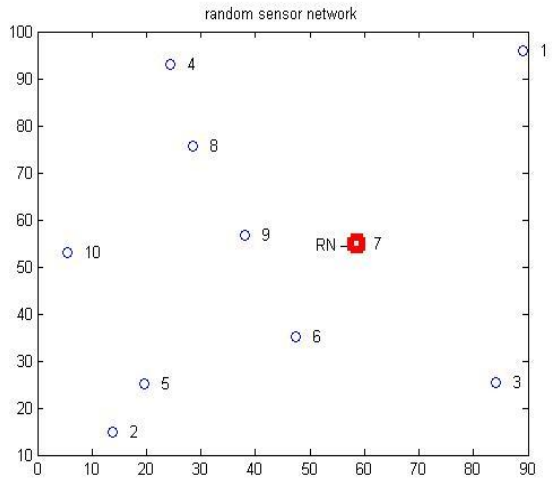

Figure 2. Highest energy node is selected as reference node.

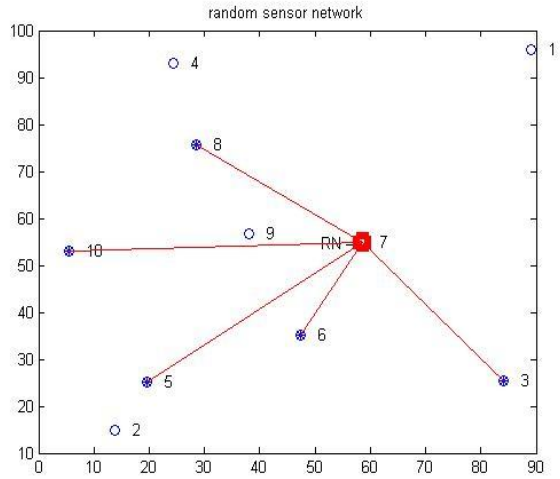

Figure 3. Cluster heads are selected.

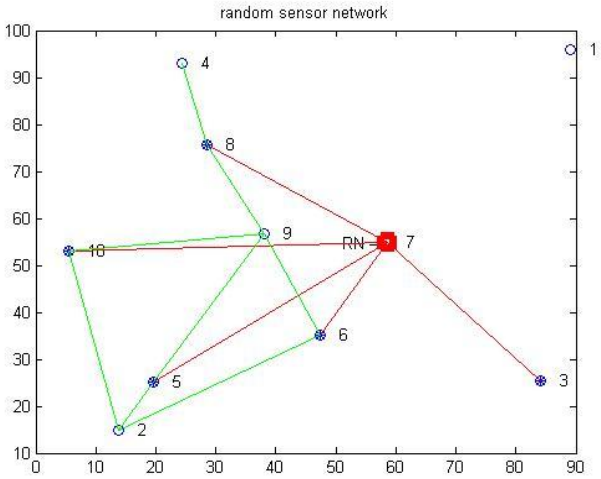

Figure 4. Cluster heads are assigned to sensor nodes.

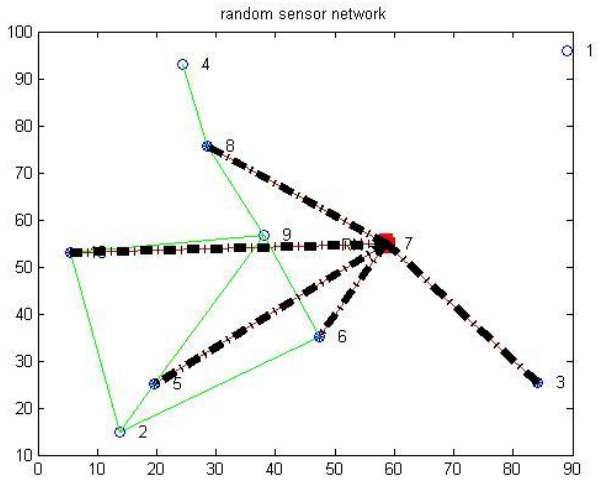

Figure 5. Reference node broadcast its ID to Cluster heads.

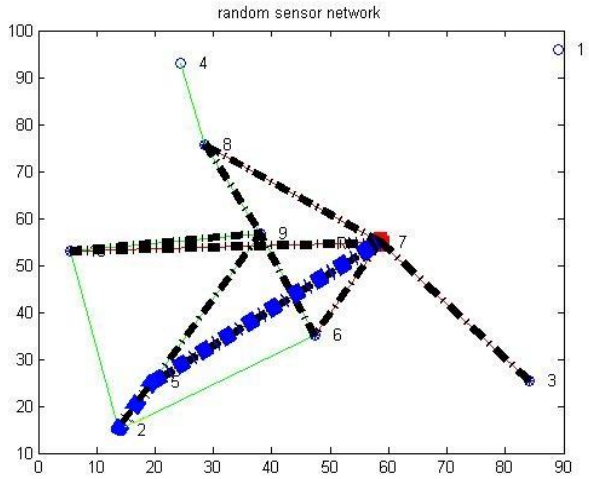

Figure 6. Node sends request to reference node.

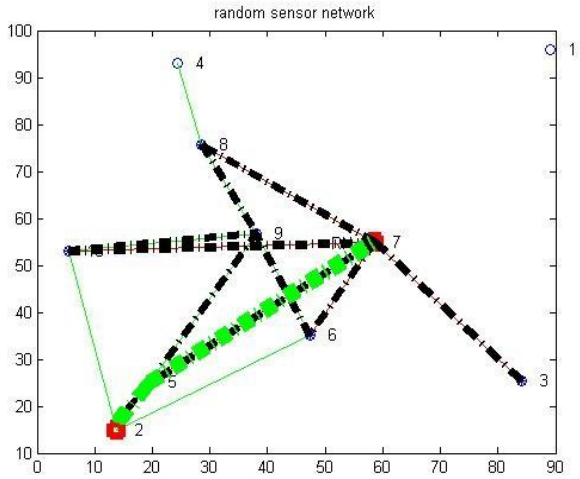

Figure 7. Reference node replies to sensor node.

This algorithm is fast and efficient in terms of setup time or delay and computation time. Figure 8 shows the computation time and setup delay of the algorithm. After the network starts up, the proposed algorithm can synchronize a node as soon as it receives the reply which can be requested by a node as soon as it knows the ID of reference node.

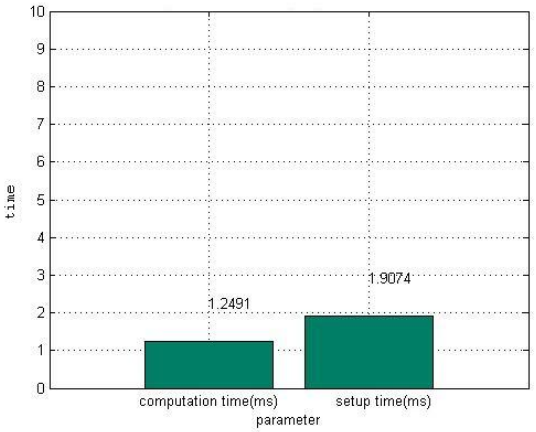

Figure 8. Computation and setup time.

V. Conclusion

The proposed algorithm for global time synchronization in a clustered network, in WSNs uses much lesser computation and setup time. An analysis of the sources of errors show that the two sources of errors are variation in propagation delays and relative drift between local clocks, which are duly compensated by the algorithm. Accuracy is gained by the compensation of propagation delay and adjustment of the timestamps at each hop. The proposed algorithm achieves energy efficiency by skew-estimation using linear regression using two data points.

Acknowledgment

I would like to thank all my friends for their support and valuable suggestions, my teachers who supported me at every stage of my work, my parents for their unending love and support, and God, Almighty for helping me to fight all the challenges that came on my way.

References

[1] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[2] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE Std. 1588-2008, 2008, pp. c1–269, (Revision of IEEE Std. 1588-2002).

[3] 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.

[4] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in Proc. 1st Int. Conf. SenSys, 2003, pp. 138–149.

[5] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The Flooding Time Synchronization Protocol," in Proc. 2nd Int. Conf. SenSys, New York, 2004, pp. 39–49.

[6] Muhammad Akhlaq, and Tarek R. Sheltami, RTSP: "An Accurate and Energy-efficient Protocol for Clock Synchronization in WSNs", *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, march 2013.

First Author Alina John, M.Tech student, Department of Electronics and Communication Engineering, Amal Jyothi College of Engineering, Kanjirapally, Kerala, India.

Second Author P Darsana, Assistant Professor, Department of Electronics and Communication Engineering, Amal Jyothi College of Engineering, Kanjirapally, Kerala, India.