Time Synchronization and Energy Efficiency in Wireless Sensor Networks

Mr. K. Rajarajan[1] Mr. J. Rajesh Kumar[2] Mr. M. Srinivasan[3]
[1]M. E. (Final year) [2]M.E., Assistant Professor [3]B.Tech. , M. E., (Ph. D). Assistant Professor. , PEC
[1, 2] RatnaVel Subramaniam College of Engineering & Technology.

Abstract---Wireless sensor networks need accurate time synchronization for data consistency and coordination. Although the existing algorithms for time synchronization offer very good accuracy, their energy consumption is high, and distant nodes are poorly synchronized. We propose a Recursive Time Synchronization Protocol (shortly called as RTSP) which accurately synchronizes all the nodes in a network to a global clock using multi-hop architecture in an energy-efficient way. It achieves better performance due to the MAC-layer time-stamping based on Start of Frame Delimiter byte, infrequent broadcasts by a dynamically elected reference node, compensation of the propagation delay and adjustment of the timestamps at each hop, estimation of the relative skew and offset using least square linear regression on two data points adaptive re-synchronization interval, aggregation of the synchronization requests, and energy awareness. A detailed analysis of the sources of errors is also provided. Simulation results show that the RTSP can achieve an average accuracy of 0.3 microseconds in a large multi-hop flat network while using five-times lesser energy than that of FTSP in the long run and performs even better in a clustered network where it can achieve an average accuracy of 0.23 microseconds while using seven-times lesser energy.

Keywords: Accuracy, algorithm, clock, energy efficiency, estimation, protocol, synchronization.

I. INTRODUCTION

Energy consumption is a very important factor in the performance of time synchronization algorithms. However, actual energy consumption may be affected by many factors, such as type of the hardware, software, and antenna. Therefore, we measure energy consumption in terms of total number of time synchronization messages exchanged among the nodes while assuming same type of hardware, software, and antenna. For all protocols the re-synchronization period is T seconds (based on FTSP) and N is the total number of nodes, then each node sends N messages in RBS protocol, two messages in TPSN protocol, one message in FTSP protocol. Although the existing algorithms for time synchronization offer very good accuracy, their energy consumption is high and distant nodes are poorly synchronized. The computation and memory requirements of FTSP are delay in time synchronization. FTSP is lesser time to execute its steps and lower memory to store the data points.

II. PROBLEM DEFINITION

The main objective of this paper is to provide data consistency and coordination through time synchronization with efficient energy consumption. To achieve this I have proposed the Recursive Time Synchronization Protocol which helps in accurately synchronizing all the nodes in a network to a global clock using multi hop architecture in an efficient way. It may achieve better performance by using MAC layer time stamping based on start of Frame Delimiter Byte. RTSP can achieve an average accuracy of 0.23 microseconds while consuming less energy.

A. Input Scheme

Clock synchronization is important for many reasons. When an event occurs in a WSN it is often necessary to know where and when it occurred. Therefore, the lifetime of WSNs can be significantly prolonged by using an energy-efficient protocol for clock synchronization. However, the total energy consumed by time-synchronization protocol depends on the resynchronization interval T and the ratio of time-synchronization messages to the total number of messages in the network that is determined by the application. Clocks are also used for many system and application tasks. For example, sleep/wake-up schedules, some localization algorithms, and sensor fusion are some of the services that often depend on clocks being synchronized. Application tasks such as tracking and computing velocity are also dependent on synchronized clocks.

The issue of clock synchronization has been investigated extensively, and several methods or protocols have been proposed for global time synchronization, such as GPS-based clock synchronization, Network Time Protocol (shortly NTP), Precision Time Protocol (shortly PTP) Reference Broadcast Synchronization (shortly RBS) Timing-sync Protocol for Sensor Networks (shortly TPSN) and Flooding Time Synchronization Protocol (shortly FTSP). The GPS-based clock-synchronization can provide an accuracy of 1 Microsecond or better, but it is costly, energy-inefficient, and infeasible in obstructed environments. The NTP is commonly used in traditional computer networks including the Internet, but it is not suitable for WSNs because of its very low accuracy high complexity, and energy inefficiency. We can achieve clock accuracy in the sub-microsecond range for networked measurement and control systems, but it is suitable only for hierarchical master-slave architecture. The RBS uses receiver-receiver synchronization to produce an average accuracy of 29.1 Microsecond for a single hop network, but this accuracy is not sufficient for WSNs which require an accuracy of 1 Microsecond or better. The TPSN uses sender-receiver synchronization and MAC-layer time-stamping of messages at the sender side to provide an average accuracy of 16.9 Microsecond for a single hop network and less than 20 Microsecond for multi-hop network, but it still not sufficient for WSNs. The FTSP is the most commonly used protocol for clock synchronization in WSNs. It broadcasts messages with timing information from a single sender to several receivers without any exchange among them, strives to tackle the flaws of TPSN.

It dynamically elects a reference node which regularly floods its current timestamp into the network creating an ad-hoc tree structure of the network instead of a fixed spanning tree. The MAC layer time stamping at both sender and receiver sides eliminates all kind of random delays except propagation delay. The timestamps are made at the end of each byte after Start of Frame Delimiter (shortly called as SFD) or SYNC byte, normalized, averaged error corrected, and then final timestamp is embedded into the message. A node waits for sufficient data points that are pairs of global and local timestamps and then estimates the offset and skew using least square linear regression. Any node that is fully synchronized with the reference node begins flooding its own estimation of the global clock. In this way, the FTSP provides an accuracy of 1.48 Microsecond for single hop case and about 0.5 Microsecond per hop in a multi-hop network. There have been many efforts to improve the FTSP in terms of accuracy, efficiency, energy consumption, etc. For example, have improved the accuracy and power consumption in a single hop network using a different method of time-stamping that is based on the SFD byte. In the SFD-based time-stamping, messages are time-stamped using the time at which radio chip generates an interrupt for the microcontroller after the SFD byte has been sent or received. We have obtained an accuracy of 0.4 Microsecond in a single hop network by using SFD-based time-stamping along with Kalman filter for skew estimation. Although the accuracy of FTSP and its improved versions is sufficiently good, the energy consumption is very high and the distant nodes are poorly synchronized. We have proposed a Recursive Time Synchronization Protocol (RTSP). The RTSP provides an average accuracy of 0.3 Microsecond per-hop in a large multi-hop flat network while using only 1by5th of the energy consumed by FTSP in the long run. However, as WSNs are usually clustered and hierarchical, we have modified and extended the RTSP algorithm to make it work with clustered networks as well. This paper extends the RTSP algorithm for clustered networks. In
Case of non-clustered or flat network, each node is assumed to be a cluster head in order to run RTSP algorithm correctly. Simulation results show that the extended RTSP algorithm further improves the accuracy and energy consumption, it can provide an average accuracy of 0.23 Microsecond in a large multi hop clustered network while using only 1by7th of the energy consumed by FTSP in the long run. The rest of this paper is organized as follows.

III. STRUCTURE MODEL FOR RTSP

First, we briefly describe the system model, assumptions, time-stamping, message structure, and recursion in RTSP.

A. *Clock Model*

Each sensor node has a hardware clock consisting of timer circuitry, which is usually based on quartz crystal oscillator and hence unstable and error prone. The hardware clock can be translated into logical clock which is used for time keeping. The hardware clock with instantaneous oscillator frequency

- $fi(t)$ at an ideal time t,
- where $f0$ is the ideal frequency,
- Δf is the frequency offset
- df is the drift in frequency,
- $rf(t)$ is the random error process
- $fi(t) = f0 + \Delta f + df\, t + rf(t)$.

Assuming $t = 0$ as the initial reference time, the related logical

- Clock reads time $Ci(t)$ at ideal time t,
- Defined as $Ci(t) = Ci(0)+1f0t- 0\ fi(t)dt$.
- We can obtain expression for the time $Ci(t)$ of clock i at a given ideal time *to* by combining
- $Ci(t) = Ci(0) + 1f0t-0(f0+DF+dft+RF(t))\ DT$.

To derive a simple linear model for non-ideal clock, we can assume that there is no frequency drift (*DF)* and that random clock error (*RF* (*t*)) is a zero.

B. *Assumptions*

Following assumptions, which are realistic and commonly found in literature, are made in the proposed algorithm:

- Sensor nodes are uniquely identified by their numeric IDs from0 to n–1.
- Time-stamping of messages at MAC-layer is possible for each node.
- Neighboring nodes can communicate over an unreliable and error-corrected wireless channel.
- Broadcasting of messages is possible in the network.
- Skew and connectivity do not change during the short interval between synchronization request and reply.
- Propagation delay in one direction is exactly equal to the other.
- A simple linear relationship exists between the clocks of two sensor nodes in a short duration.

IV. SYSTEM ARCHITECTURE

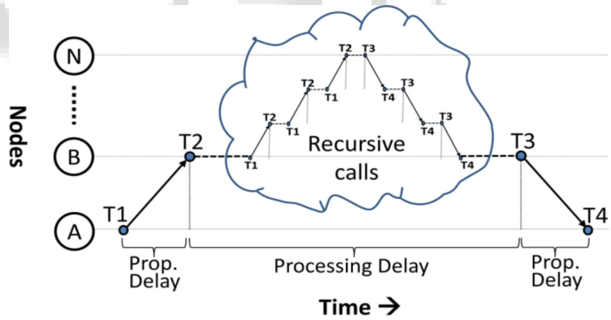

Fig. 1: System Architecture Diagram

A. *Projected Model*

Recursive Time Synchronization Protocol (shortly RTSP) which accurately synchronizes all the nodes in a network to a global clock using multi-hop architecture in an energy-efficient way. A detailed analysis of the sources of errors is also provided. Simulation results show that the RTSP can achieve an average accuracy of 0.3 microseconds. In a large multi hop flat network while using five times lesser energy than that of FTSP in the long run and performs even better in a clustered network where it can achieve an average accuracy of 0.23 microseconds while using seven-times lesser energy. WSNs are commonly deployed for monitoring the remote and hazardous environments, it is not feasible to recharge or replace the battery of sensor nodes.

Advantages of Projected System:

- Sensor nodes should use their energy very efficiently even when a power harvesting technique is used with the battery of limited capacity.
- The issue of clock synchronization has been investigated extensively and several methods or protocols have been proposed for global time synchronization,
- They are GPS-based clock synchronization Network Time Protocol (shortly NTP)
- Precision Time Protocol (shortly PTP)
- Reference Broadcast Synchronization (shortly called as RBS)
- Timing-sync Protocol for Sensor Networks (shortly called as TPSN) and Flooding Time Synchronization Protocol (shortly called as FTSP).

V. RECURSIVE TIME SYNCHRONIZATION PROTOCOL (RTSP)

After a WSN boots up, the sensor nodes broadcast an RTSPERN enquiry message to ask their neighbors about the identification of reference node, wait for the period T or until a

Reply is received, run repeatedly the RTSP algorithm that takes care of the dynamic election of a single reference node with the smallest ID and compensation of the offset and drift. This process is outlined in the Algorithm 1: RTSP and flowcharted in Figs. 3 each node maintains a few variables including my Ref for ID of the reference node, and my Client for IDs of the nodes sending time synchronization request through it. Note that in case of non-clustered or flat network, each node is assumed to be a cluster head in order to run RTSP algorithm. The RTSP algorithm is responsible for two major 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 the Reference Node

The RTSP algorithm dynamically elects a single reference node as follows:

- After start up, a sensor node waits for short time and then sends an RTSP-ERN message containing “−1” value for the ref Node ID field in order to ask the neighboring nodes for ID of the reference node. It is important to note that a cluster head broadcasts the enquiry message, but a non-cluster head sends the enquiry message to its own cluster head. The enquirer node waits for the duration T or until a reply is received, and then acts as follows:
- If it does not receive any reply, it enters into the contest for new reference node by broadcasting an RTSPERN message by putting its own identification in the ref Node ID field. Note that a node that is not a cluster head or is low in energy cannot take part in the contest for reference node.
- However, if it receives a reply, it saves the identification of reference node in a local variable called my Ref, and then broadcasts the RTSP-ERN message.
- If a new RTSP-ERN message is received and authenticated it checks the value
- of ref Node ID field to do the following:
- If the ref Node ID field contains any negative value, the message is treated as an enquiry. If a node knows the identification of reference node, it replies directly to the enquirer.
- However, if the ref Node ID field contains any non-negative value, the message is treated as an announcement or contest, and is flooded as follows:
- If the receiving node does not know the identification of reference node or it knows some identification that is greater than ref Node ID, it learns the identification of new reference node by updating my Ref variable and then rebroadcasts the message.
- If the ID of receiving node is smaller than the current reference node and the receiving node is a cluster head with sufficient energy, it contests for reference node by broadcasting an RTSP-ERN message.

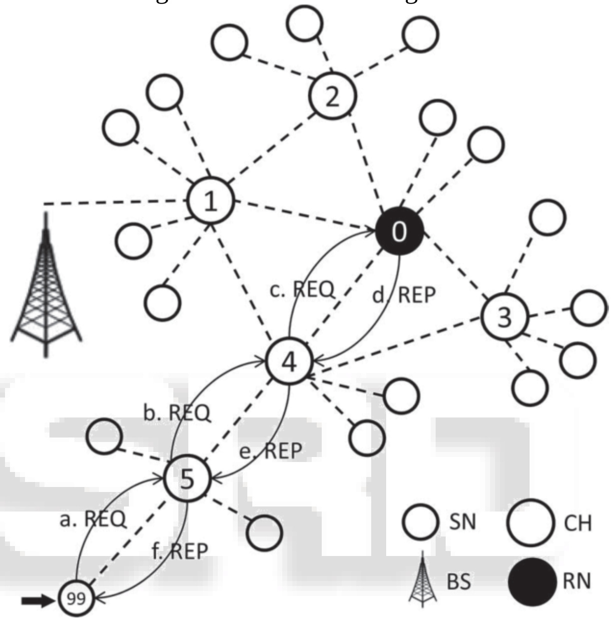

Fig. 2: Example of request-and-reply mechanism in RTSP algorithm.

The request is recused only when intermediate nodes are not synchronized.

VI. CONCLUSION

The RTSP algorithm for global time synchronization in WSNs, which gives an average accuracy of 0.23 microsecond per hop in a large multi-hop clustered network using seven-times lesser energy than that of FTSP in the long run. An analysis of the sources of errors shows that the two sources of errors are variation in propagation delays and relative drift between local clocks, which are duly compensated by the algorithm. The accuracy of RTSP is improved by using the MAC-layer time-stamping based on SFD byte, which is simpler and more accurate. Further improvement in accuracy is gained by the compensation of propagation delay and adjustment of the timestamps at each hop. The RTSP algorithm achieves energy efficiency by using several techniques, which include the infrequent broadcasts by the reference node, skew-estimation using 2LR instead of 8LR, and reducing the number of time synchronization requests through the adaptive re-synchronization interval and aggregation of synchronization requests.

VII. FUTURE WORK

Further improvement in accuracy is gained by the compensation of propagation delay and adjustment of the time stamps at each hop. The RTSP algorithm achieves energy efficiency by using several techniques, which include the infrequent broadcasts by the reference node, skew-estimation using 2LR instead of 8LR, and reducing the number of time synchronization requests through the adaptive re-synchronization interval and aggregation of synchronization requests.

REFERENCES

[1] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in Proc. 3rd ACM Workshop Security Ad Hoc Sens.Netw., 2005, pp.
[2] R. Albu, Y. Labit, T. Gayraud, and P. Berthou, "An energy-efficient clock synchronization protocol for wireless sensor networks," in Proc. WD, IFIP, 2010, pp. 1–5.
[3] R. Albu, Y. Labit, T. Gayraud, and P. Berthou, "An energy-efficient clock synchronization protocol for wireless sensor networks," in Proc. WD, IFIP, 2010, pp. 1–5.
[4] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in Proc. 2nd Int. Conf. SenSys, New York, 2004, pp. 39–49.
[5] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in Proc. 3rd ACM Workshop Security Ad Hoc Sens. Netw., 2005, pp.
[6] M. Akhlaq and T. R. Sheltami, "The recursive time synchronization protocol for wireless sensor networks," in Proc. IEEE SAS, Brescia, Italy, 2012, pp. 1–6.