ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

A Propagation Delay Compensation Protocol for Accuracy Improvement in Clustered Network

Manju S[1], N Angayarkanni[2], Dr.G.K.D Prasanna Venkatesan[3]

PG Student, Department of ECE, PGP College of Engineering And Technology, Namakkal , India[1]

Assistant Professor, Department of ECE, PGP College of Engineering And Technology, Namakkal , India[2]

Professor and HOD, Department of ECE, PGP College of Engineering And Technology, Namakkal , India[3]

Abstract: A wireless sensor network consist of a set of sensor devices which spread over a geographical area. A synchronized network time is essential for energy efficient scheduling, data fusion, localization in wireless sensor network applications. Data collection is one of the most important functions provided by wireless sensor networks. In this paper, we discussing theoretical limitations of data collection and data aggregation in terms of delay and capacity where sensors are randomly deployed. We propose Recursive Time Synchronization Protocol (RTSP) which accurately synchronizes all nodes in a network to a global clock using multihop architecture in an energy efficient way. Simulation results show that RTSP can achieve an average accuracy of 0.3 microseconds in a large multihop flat network. More accuracy improvement is achieved by compensating the propagation delay and adjustment of timestamps.

Keywords: time synchronization, offset, skew, propagation delay, sensor networks

I. INTRODUCTION

Wireless sensor network (WSN) is an emerging technology, which consists of very large number of tiny sensing devices, called motes, distributed over physical space. Each device is capable of limited computing, radio communication and sensing. In the past couple of years, wireless sensor networks have found a wide range of applications such as environmental monitoring, robotic and object localization and tracking.

Time synchronization is very much important in any distributed systems, in particular in wireless sensor networks. Sensor networks are used to monitor real world phenomenon. For such monitoring, physical time often plays a crucial role. Physical time plays a important role for many sensor network applications. While a number of traditional applications of time also apply to sensor networks, here we will focus on areas specific to sensor networks. In sensor networks, many sensor nodes may observe a single physical phenomenon. Since many instances of physical phenomenon can occur within a short time, one of the tasks of a sensor network is the separation of sensor samples, that is the partitioning of sensor samples into groups that each represent a single physical phenomenon. Temporal relationships (e.g., distance) among sensor samples are a key element for separation. Temporal coordination among sensor nodes may also be necessary to ensure correctness and consistency of distributed measurements.

The issue of clock synchronization has been investigated extensively, and several methods of protocols have been proposed for global time synchronization, such as GPS based clock synchronization, Network Time Protocol (NTP), Precision Time Protocol (PTP), Reference Broadcast Synchronization (RBS), Time-sync Protocol for Sensor Network (TPSN) and Flooding Time Synchronization Protocol (FTSP). The comprehensive survey on the clock synchronization algorithms, which should be considered for the measurements of the network delay. We survey the recent progress on these end to end algorithms and special concerns are on the estimation of true the one way delay without the effect of clock skew. End to end one way delay experienced by a packet is the time taken to travel from source to destination by packet can be measured from the difference between the arrival time, according to the time stamp added by the source and the destination clock, which were conveyed by the packet. There are two kinds of clock synchronization approaches according to how they synchronize clocks: external server based methods and end to end measurement methods. In the external server based methods the basic idea is to locate global server providing time information to every host in the network. Every host has to recognize the server and operator under the time synchronization through the server. Network Time Protocol (NTP), Global Positioning System (GPS) and IEEE 1588 could be included within this category. In delay measurement the dynamic part named queuing delay, gets more attention than the static part composed of propagation delay and transmission delay. Many contributions are devoted to determine the clock skew in the measurement and by removing the effect of the skew we can transform the delay measurements so that they are consistent with a single clock.

The NTP topology is typically a set of NTP servers with large group of clients all on the local subnet. Broadcast and multicast modes allow the administrator to use the same configuration file for many clients, since clients would be configured with the same broadcast server. Configuration on the server side allows the client to set up an exchange with the server to improve estimates of the message propagation delay.

The client and server continue to exchange a series of messages to authenticate the source, set clock, and obtain round trip delay estimate until the client time is set. And then after that the exchange terminates; however if the

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

server does not respond, the client will use a default propagation delay to correct its clock.

In Precision Time Protocol messages in the protocol include master synchronization message, master delay response message, and the clock delay request message of the slave. In addition to that all, the Best Master Clock (BMC) algorithm allows multiple masters to negotiate the best clock for the network. The precise time of the delay request message when received at the master. Assuming constant network propagation delays and gradually changing operating conditions such as temperature. The slave clock make use of the offset to adjust the time to agree with the master clock. Typically the slave clock will use a clock tuning algorithm that can account for network propagation delays affecting the offset and the slave clock crystal temperature and aging effect on its stability.

The synchronization pulse is an absolute time reference at the instant of its arrival, that is, that it arrives t every node at exactly the same time. Bt practically, this is not the case due to the finite propagation speed of Radio signals. And it is not possible to achieve synchronization with a precision better than the difference in the propagation delay between the various receivers and the synchronization beacon.

In Reference Broadcast Synchronization (RBS), a reference message is broadcasted over the network. The receivers record the local time when receiving the reference broadcast and then they exchange their recorded time. In this way, they have the knowledge of their time offset with each other. When each mote takes the average of its time offsets to all other nodes that have observed the same reference, a relative network time is achieved among all the receivers.

The TPSN uses sender receiver synchronization in MAC layer time stamping of messages at the sender side to provide an average accuracy of 16.9μs for a single hop network and less than 20μs for multi hop network. TPSN loose their designed microsecond level accuracy under the spatial uncertainty of orders of magnitude large propagation delays in WSN, uncertainty in propagation time (P^{UC}). The TPSN approach eliminates the access time, byte alignment time and propagation time by making use of implicit acknowledgements to transmit information back to the sender.

FTSP broadcasts timing information from single sender to several receivers without any exchange among themselves. It provides an average accuracy of 1.48μs for single hop case and about 0.5μs per hop in a multihop network. There have been many efforts to improve the FTSP in terms of accuracy, efficiency, energy consumption etc. Since the FTSP collects the timing information from the parent node the time stamping approach is complex and it consumes more time. The propagation time is very deterministic in WSN and it depends only on the distance between two nodes. This time is less than one microsecond. The FTSP time stamping protocol effectively reduces all sources of time stamping errors except for the propagation time. Since the FTSP time stamping employs a single radio message, it does not compensate for the propagation delay. Since the synchronization messages are broadcasted periodically and the individual nodes are transmitting asynchronously. The speed of message propagation is limited.

Recursive Time Synchronization Protocol (RTSP) which accurately synchronizes all the nodes in a network to a global clock using multi-hop architecture in an energy efficient way. By using MAC-layer time stamping on Start of Frame Delimiter byte, infrequent broadcasts of dynamically selected reference node, nullifying the propagation delay and adjustment of the timestamps at each hop, estimation of the offset and relative skew using least square linear regression on two data points adaptive re-synchronization interval, accumulation of the synchronization requests, and energy awareness on the RTSP achieves better performance. The MAC-layer time stamping at both sender and receiver sides eliminates all kind of random delays except propagation delay. In RTSP ignoring the propagation delay, the interrupt occurs almost simultaneously both at sender and receiver chips with a difference of 3 microseconds. Therefore, the timestamp at sender side is incremented by 3microseconds. It is important to note that this approach is also unable to compensate the propagation delay. Therefore, we introduce strategy for finding and compensating the propagation delay in order to achieve higher accuracy of time synchronization in the RTSP algorithm.

II. PROPOSED ALGORITHEM

A. *System model*

In the following sections we analyze various synchronization approaches. We will now specify the system model for time synchronization that we use as the foundation of our analysis. we describe how we model clocks. We then specify the characteristics of the communication between nodes in a sensor network.

All our modeling is done in terms of discrete time and events. An event can represent communication between nodes, sensor measurement, the injection of time information at a node, etc. And these all followed by the assumptions time stamping, message structure, recursion in RTSP and the delay management.

B. *Clock models*

Every sensor nodes having hardware clock which consisting of timer circuitry. Digital clocks measure time intervals. They typically consist of a counter refer to as local clock that counts time steps of an ideally fixed length; we denote the reading of the counter at real time. The counter is incremented by an oscillator with a rate of frequency. An ideal clock would have rate 1 at all times, but the rate of real clock fluctuates over time due to changes in supply voltage, temperature etc. the clocks present in the sensor nodes 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 instantaneous oscillator frequency $f_i(t)$ of the hardware clock at an ideal time t, is defined as,

$$f_i(t) = f_o + \Delta f + d_f t + r_f(t) \quad (1)$$

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

where f_o is the ideal frequency, Δf is the frequency offset, d_f is the drift in frequency, and $r_f(t)$ is the random error process.

Assuming $t = 0$ as the initial reference time, the related logical clock reads time $C_i(t)$ at ideal time t, defined as,

$$Ci(t) = Ci(0) + \frac{1}{f0}\int_0^t f_i.(t)\,dt \quad (2)$$

We can obtain expression for the time $C_i(t)$ of clock i at a given ideal time t by combining (1) and (2), s follows:

$$Ci(t) = Ci(0) + \int_0^t (f_0 + \Delta f + d_f + r_f(t))\,dt \quad (3)$$

To derive a simple linear model for non-ideal clock, we can assume that there is no frequency drift (d_f) and that random clock error $(r_f(t))$ is a zero men. Therefore, after elimination of the last two terms in *(3)* and further simplification, we get $C_i(t) = C_i(0)+(1+(\Delta f / f_0))\,t$, which can be written in very simple form as,

$$C_i(t) = \alpha_i + \beta_i t \quad (4)$$

where α_i is the clock offset (deviation from ideal time) at the reference time $t = 0$ and β_i is the clock skew (the frequency of a clock) Clocks are usually specified with maximum drift rate ρ (the rate at which change of frequency with respect to the ideal clock), such that $1 - \rho \le \beta \le 1 + \rho$. That is, the clocks of any two perfectly synchronized nodes may drift away from each other at a rate at most 2ρ. Therefore, in order to keep their relative offset within δ seconds all the time, they need to be resynchronized within $\delta/2\rho$ seconds, i.e., re-synchronization interval $T \le \delta/2\rho$. However, frequent re-synchronization are not feasible due to inefficiency, cost, etc. therefore, it is essential to estimate the drift or skew in order to keep the sensor clocks synchronized without any need for frequent resynchronizations.

C. Message time-stamping

This approach is for time-stamping messages and events of a synchronously ordered computation, that is, when process communicate using synchronous messages. This depends on decomposing edges in the communication topology into mutually disjoint edge groups such that each edge group either forms a star or triangle. To accurately capture the order relationship between synchronous messages, it is enough to use one component per edge group in the vector instead of one component in each process. Timestamps done for events are only slightly bigger than timestamps for messages.

For sensor nodes, the classical approach of doing a handshake between a pair of nodes is a better approach than synchronizing a set of receivers. This observation comes as a result of time stamping of the packets at the moment when they are sent i.e., at MAC layer, which is indeed feasible for this network. The offset between any pair of nodes is computed by exchanging the local timestamps. Thus this scheme successfully removes all possible sources of errors except the variability in processing delay at the receiver.

MAC-layer time-stamping of messages at sender and receiver sides, is very useful for time synchronization in WSNs. The standard IEEE 802.15.4 compliant frame consists of 4 bytes of preamble, 1 byte of SFD, 1 byte of frame length indicator, data payload of 125 bytes maximum, and 2 bytes of frame check sequence as shown in Fig. 1.

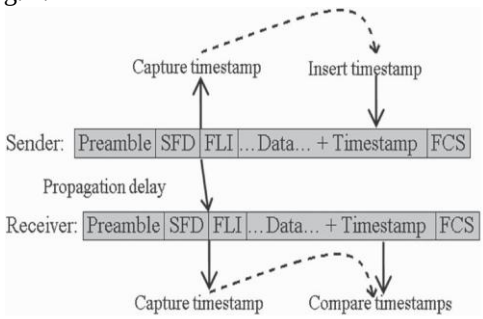

Fig. 1. IEEE 802.15.4 MAC frame and time-stamping structure

The RTSP algorithm timestamps messages when radio chip generates an interrupt for the microcontroller after the SYNC/SFD byte has been sent or received. This interrupt occurs almost simultaneously both at the sender and receiver chips with a difference of 3μs if we considered by ignoring the propagation delay. Thus, the timestamp at sender side is incremented by 3μs.

The method of time-stamping is simpler, faster, and accurate in which timestamps are made at the end of each byte after SFD byte. The final timestamp is embedded into the message. This approach is also unable to compensate the propagation delay. Thus, we introduce a strategy for finding and compensating the propagation delays in order to achieve higher accuracy of time synchronization in the RTSP algorithm.

D. Structure of the RTSP message

The time synchronization is achieved in RTSP algorithm by exchanging messages among nodes. Mainly there are three types of messages: ERN (enquiry/election of the reference node), REQ (request for time synchronization), and REP (reply for time synchronization). The structure of an RTSP message is shown in Table I.

Field	Description
Msgtype	Type of the message: ERN(enquiry/election of the reference node), REQ(request for time sysnchronization), and REP(reply for time synchronization)
msgID	ID of the message
orginID	ID of the node that originated the message
imSrcID	ID of the immediate source node that forwarded the message
imDestID	ID of the immediate destination for message forwarding
refNodeID	ID of the reference node known(-1 for ERN enquiry)
T1	Local time when the message was send or forwarded
T2	Local time when the message was received
T3	Local time when the reply was send or forwarded
Tr	Reference time when the reply was send or forwarded
• We use 0 for null value and * for broadcast address • RTSP-REQ msg takes null values for T2, T3, and Tr	

Table. 1. Structure of the RTSP messages

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

The RTSP-ERN message with a negative value for the refNodeID field is treated as an enquiry, while a non-negative value is treated as an announcement or contest. A node requests timing information by sending an RTSP-REQ message to the reference node, while the reference node or any synchronized node on the request-path replies with an RTSP-REP message. The fields T1, T2, T3, and Tr are used for timestamps after SFD byte has been sent, received, or forwarded.

E. Recursion and Multi-hop synchronization

The problem of time synchronization in WSNs has attracted a great deal of attention; it represents a challenge, having in mind multihop communications, unpredictable packet losses and high probability of node failures .The RTSP algorithm works with any network topology, even if the network is flat and clustered. A time synchronization request can be initiated by any node in the network, and then it is recursively forwarded to the reference node. The reference node, or any synchronized node on the request-path, replies with timing information that is sent back to the requesting node through the same path. It is important to note that the propagation delay is also compensated at each hop before forwarding the reply.

The request-and-reply strategy in RTSP is shown in the Fig 2. The node A in a multi-hop network sends request at T1 to the reference node through another node B that receives it at time T2. The node B, if not synchronized, forwards this request to another node in a recursive manner until it reaches the reference node or a fully synchronized node N which replies with the timing information that is forwarded to node B through the same path. The node B at time T3 forwards this reply to node A which receives it at time T4. It has already proved that the performance of TPSN is insensitive to any variations in processing delay, i.e., the interval between T2 and T3 does not affect the accuracy of time synchronization. This gives us an opportunity to make recursive calls during the interval between T2 and T3 in the RTSP algorithm without losing performance.

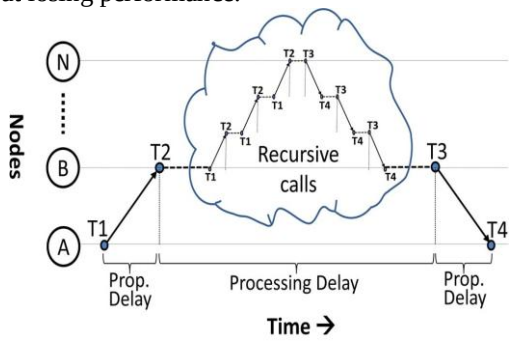

Fig. 2. Request-and-reply mechanism in RTSP

Before the distant nodes, the nodes closer to the reference node are expected to do the following actions:

1. On knowing ID of the reference node the time synchronization request is initiated
2. The initiated requests become synchronized with the reference node.

As the closer nodes are expected to become synchronized before distant nodes and any synchronized node on the request-path can reply, very few time synchronization requests are forwarded all the way to the reference node.

III. RTSP

When network boots up, the node wait for sometime and then sensor nodes broadcast an RTSP- ERN enquiry message to enquire their neighboring node about the identification of reference node. The node wait for the period T or until a reply is received, and then run repeatedly the algorithm that helps in the dynamic election of a single reference node with the smallest ID and nullify of the offset and drift.

The RTSP protocol while running in flat networks assumes each and every node as cluster head. Each node maintains a few variables including myRef for ID of the reference node, and myClient for IDs of the nodes sending time synchronization request through it. Mainly there are two processes taking place in running the RTSP as (1) election of the reference node, (2) Offset and drift compensation, they were explained in the two subsections following.

A. Election of the Reference Node

The process of electing reference node after startup is as follows:

1. After waiting for a short time the sensor nodes sends an RTSP-ERN message with a ”-1” value to for refNodeID field to get the ID of reference node from the neighboring nodes.
   a. If it does not receive any reply, it enters into the contest for new reference node by broadcasting an RTSP-ERN message by putting its own identification in the refNodeID field
   b. If reply is received, it saves the identification in a local variable called myRef, and then broadcasts the RTSP-ERN message
2. For a new RTSP-ERN message is received and authenticated it checks the value of refNodeID field and then carry out the following tasks
   a. If the refNodeID 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
   b. Any non- negative value in the refnodeID field, the message is treated as an announcement or contest and it is flooded as either if the receiving node does not know the identification of reference node or it knows some identification that is greater than refNodeID, it learns the identification of new reference node by updating myRef variable and then rebroadcasts the message the another possibility is 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. The selected reference node takes responsibility of broadcasting the timing information periodically. RTSP algorithm uses this broadcasting mainly for announcing the

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

identification of reference node. Any node that receives a new RTSP-ERN message from the reference node accepts it and then rebroadcasts to its neighbours.

It is possible that more than one node declare themselves as reference nodes simultaneously. To avoid any conflict, a reference node retreats to an ordinary node on receiving an RTSP- ERN message from another reference node with a smaller identification.

B. Compensation of the offset and drift

To compensate for the offset and drift, and to synchronize all the nodes on request-path to the reference node, the RTSP algorithm works as follows

1. The node checks the type of newly received RTSP message.
2. If an RTSP-REQ message is received, it notes the receive time T2. If it is a reference node or a synchronized cluster head or gateway node, it replies with an RTSP-REP message containing timestamps T1, T2, T3, and Tr in addition to the other information.
3. If an RTSP-REP message is received, it computes the value of propagation delay by

$$d = \frac{(T2-T1)+(T4-T3)}{2} \quad (5)$$

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

$$Tr = Tr + d \quad (6)$$

Then it records the global and local timestamps where global = Tr and local = T4. If the node is an intermediate node, it retrives the values of T1 and T2, computes the value of Tr by adding to it the elapsed time since T4 using , and then forwards the reply to its client node, if any

$$Tr = Tr + (T3 - T4) \quad (7)$$

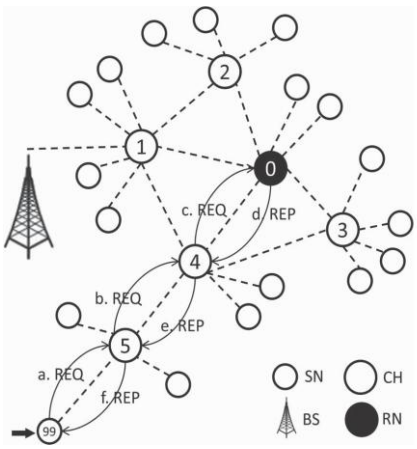

Fig. 5. Example of request and reply strategy in RTSP. The recursion of request happens only when intermediate nodes are not synchronized

Fig. 5 explains the request-and-reply strategy of RTSP algorithm with the help of an example clustered network.

In Fig. 5, node 99 initiates an RTSP-REQ at its local time T1 which is received by an intermediate node 5 at its local time T2. The node 5 stores the identification of node 99 in myClient, values of T1 and T2 in T1old and T2old, respectively, and then forwards this request at its local time T1 to node 4 which recursively forwards it to the reference node 0. On receiving the request at its local time T2, the reference node 0 processes the request and sends back an RTSP-REP message at its local time T3. When node 4 receives the RTSP-REP reply at time T4, it performs the following six actions: 1) computes the value of d, 2) adjusts the timestamps by adding d to Tr, 3) records the global and local timestamps, 4) retrieves the values of T1 and T2, 5) computes the value of new Tr by adding to it the elapsed time since T4, and 6) forwards the reply to node 5 at its local time T3. Node 5 also performs the six actions and then recursively forward the reply to the node 99. Similarly, node 99 performs the first three actions only since it is not an intermediate node.

A reply follows exactly the same path as request, but a new request/reply message may follow a different path from the previous message.

C. Security

WSNs face many security challenges due to their environment and constraints the use of insecure and unreliable wireless channel, and availability of the limited resources such as computing, memory and energy. Moreover, the traditional protocols for time synchronization in WSNs do not consider security and hence prone to several kind of security attacks.

Although security is not the prime objective of RTSP, it provides some basic level of protection against the attacks on time synchronization in which a single compromised node may propagate incorrect timing information through the network. Simple techniques such as redundancy, authentication, and refusal to forward corrupt synchronization information are used to tackle the attacks on time synchronization as follows.

1. *Redundancy:* RTSP algorithm provides some level of redundancy to avoid dependence on the neighbor. A node may receive broadcasts by the reference node by means of a different neighbor at a different time. This provides redundancy for basic level of security.

2. *Authentication:* A node that wants to join the network is authenticated first by using any light-weight authentication protocol. The RTSP algorithm is run only when message is received from an authenticated node.

3. *Discarding the corrupt information:* Each RTSP message carries the current or real time of the sender in Tr field, which helps identify any timing information that is very much deviating from the recent data. A node can discard any corrupt message in order to stop the propagation of incorrect timing information.

IV. RESULTS AND DISCUSSION

A. Errors in RTSP

Mainly there having two kinds of errors in the RTSP. They were:

Variation in the propagation delays

The RTSP algorithm assumes that the propagation delay in one direction is exactly equal to the other. When this is true, one source of synchronization error is eliminated and

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

accuracy of the time synchronization is improved further. However, this assumption may not be true in wireless communication. Consequently, each hop might add possibly an inaccurate value of d to the reference time Tr in order to find the new value of Tr. the speed of radio signal affected by changes in the air temperature and humidity is negligible for a distance of hundreds of miles; rather totally negligible for a distance of few hundred meters in WSNs. Consequently, the effect of this source of error is also minimized in the new system design.

The relative drift between local clocks

The hardware clock in a low-cost sensor node is usually based on quartz crystal oscillator that is vulnerable to huge drift the ideal clock. However, shows that the RTSP is sensitive only to the "relative drift" between the two clocks instead of the drift from ideal clock. It computes the relative drift (β) and offset (α) as,

$$\beta = \frac{g1-g2}{y1-y2} \quad (8)$$

The skew or drift between two clocks can be compensated using (4), (8) and and thus the effect of this source of error is also minimized by the RTSP algorithm[1]

B. Propagation error in multi-hop

Any error at one hop may increase when it propagates to the next hops. However, in case of time synchronization, the error may not always increase due to different sign and magnitude of the drift rate of sensors.

In a network of k hops, every hop may experience a random error of δ. As the sum of randomly distributed errors is the square root of the sum of the squares of errors on each hop, the total error at kth hop may be computes as

$$Error_{k\text{-}hop} = \sqrt{(\delta 1^2 + \delta 2^2 + \cdots \delta k^2)} \quad (9)$$

We can written it in more simpler form as,

$$Error_{k-hop} = \delta\sqrt{k} \quad (10)$$

Thus the total error at the kth hop may be in the order of $\delta\sqrt{k}$ which means that the accuracy of the time synchronization increases with increase in no of hops. There may be extreme situations in which the total error of the system may be equal to zero as well. This occours when all the errors are either negative or positive. Accordingly $-\delta\sqrt{k}$ is the total error for the negative values and $\delta\sqrt{k}$ for positive values. Thus the bounds on total error in multihop is,

$$-\delta\sqrt{k} \leq \text{Error}_{\text{k-hop}} \leq +\delta\sqrt{k} \quad (11)$$

C. Performance evaluation

In order to evaluate the performance of proposed algorithm, we performed a simulation in MATLAB and collected data on accuracy or synchronization error and delays due to propagation. Fig 6 shows that, in the beginning, the no of messages exchanged of the RTSP is higher than the existing protocols.

It becomes equal to TPSN after four or five periods, and then equal to FTSP after seven or eight periods and then remains lower than the two protocols in the long run.

D. Simulation Setup

A simulation was performed in MATLAB. The use of random topology, random walk mobility model, and pause time of zero, which makes the network highly dynamic. A time synchronization protocol that performs better in such a network is expected to perform even better in a static environment.

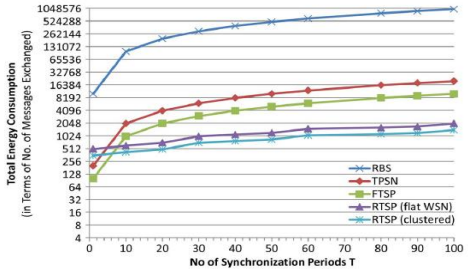

Fig. 6. Total energy consumption for 100 nodes

1. A class for sensor node is defined with different attributes, such as ID, position, energy level, level in hierarchy, parent ID, current time, offset, and skew.
2. For total number of nodes, instances of the sensor node class are created, and random values are generated for the position of each node
3. Communication links are defined according to the trans- mission range and distance between sensor nodes.
4. Protocols are run on their own instance of the network for several times, and results are stored in a workbook.

Actual propagation may be affected by many factors, such as type of the hardware, software, temperature, environmental factors and antenna. Therefore, we measure propagation delay 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. On startup of the network, each node forwards one message, which is broadcasted by the reference node.

Then, a node sends two RTSP-REQ messages and gets their replies accordingly. This makes five messages per node. However, once the nodes are synchronized, they need to send very less number of requests due to the adaptive behavior of RTSP algorithm.

Property	TPSN	FTSP	RTSP
Scheme Used	Sender-to-Receiver	Sender-to-Receiver	Sender-to-Receiver
Time-Stamping	MAC	MAC	MAC
Sensitive to delays	Encoding, Decoding and Interrupt Handling	Propagation	Propagation
Flooding	No	Frequent	Infrequent
Compensation of Propagation Delay	Yes	No	Yes
Request-Reply	Iterative	No	Recursive
Adaptive	No	No	Yes
No of messages per node	2	1	0.14
Energy usage	High	Low	Very low
Accuracy (per hop)	29.1μs (single hop)	0.5μs	0.23μs

TABLE II Comparison Of RTSP With Other Algorithms

While comparing with the existing protocols and RTSP and up on analyzing it is clearly understood that the efficiency in energy utilization and time synchronization all are widely related with the delay in its propagation in the sensor network. Using the Curve fitting toolbox in MATLAB, the equation for RTSP revealed that each node sends only 0.2 messages for flat network and 0.14 messages for clustered network, which means it consumes less energy. Table II provides a quick comparison of the RTSP algorithm with other global time synchronization algorithms

V. CONCLUSION

We have described the RTSP algorithm for compensating propagation delay in the wireless sensor networks. An analysis of the RTSP protocol it is found that there having two sources of errors and they where the 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. Improvement in accuracy is gained by the nullification 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 etc. The RTSP algorithm consumes less energy compared to other protocols and it saves energy through infrequent reference broadcasts and by reducing the no of request through request aggregation. And more over modeling new system at the gateway the propagation delay of the whole system is reduced.

REFERENCES

[1] Muhammad Akhlaq, and Tarek R Sheltami, "RTSP: An Accurate and Energy-Efficient Protocol for Clock Sychronization", IEEE Transation on instrumentation and measurement, Vol 62, pp. 578–589, March 2013.
[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," Ad Hoc Netw., vol. 7, no. 3, pp. 537–568, May 2009.
[3] D. L. Mills, "Internet time synchronization: The network time protocol,"IEEE Trans. Commun., vol. 39, no. 10, pp. 1482–1493, Oct. 1991.
[4] IEEE Standard for a Precision Clock Synchronization Protocol for Net- worked Measurement and Control Systems, IEEE Std. 1588-2008, 2008, pp. c1–269, (Revision of IEEE Std. 1588-2002).
[5] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in Proc. 1st Int. Conf. SenSys, 2003, pp. 138–149.
[6] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time syn- chronization protocol," in Proc. 2nd Int. Conf. SenSys, New York, 2004, pp. 39–49.
[7] P. Ferrari, A. Flammini, D. Marioli, and A. Taroni, "IEEE 1588-based synchronization system for a displacement sensor network," IEEE Trans.Instrum. Meas., vol. 57, no. 2, pp. 254–260, Feb. 2008.
[8] R. L. Scheiterer, C. Na, D. Obradovic, and G. Steindl, "Synchroniza- tion performance of the precision time protocol in industrial automation networks," IEEE Trans. Instrum. Meas., vol. 58, no. 6, pp. 1849–1857, Jun. 2009.
[9] D. Cox, E. Jovanov, and A. Milenkovic, "Time synchronization for Zig- Bee networks," in *Proc. 37th SSST*, 2005, pp. 135–138.
[10] A. Hu and S. D. Servetto, "Asymptotically optimal time synchronization in dense sensor networks," in *Proc. 2nd ACM Int. Conf. Wireless Sens. Netw. Appl.*, 2003, pp. 1–10.
[11] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Trans. Comput.*, vol. C36, no. 8, pp. 933–940, Aug. 1987.