

A Novel Cluster Based Time Synchronization Technique for Wireless Sensor Networks

Gopal Chand Gautam, Narottam Chand

Department of Computer Science & Engineering, National Institute of Technology Hamirpur, Hamirpur, India

Email: gopalcgautam@gmail.com, nar.chand@gmail.com

How to cite this paper: Gautam, G.C. and Chand, N. (2017) A Novel Cluster Based Time Synchronization Technique for Wireless Sensor Networks. *Wireless Sensor Network*, 9, 145-165.

<https://doi.org/10.4236/wsn.2017.95008>

Received: May 1, 2017

Accepted: May 22, 2017

Published: May 25, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Time synchronization is one of the important aspects in wireless sensor networks. Time synchronization assures that all the sensor nodes in wireless sensor network have the same clock time. There are various applications such as seismic study, military applications, pollution monitoring where sensor nodes require synchronized time. Time synchronization is mandatory for many wireless sensor networks protocols such as MAC protocols and also important for TDMA scheduling for proper duty cycle coordination. Time synchronization is a stimulating problem in wireless sensor networks because each node has its own local clock which keeps on varying due to variation in the oscillator frequency. The oscillator frequency is time varying due to ambient conditions which leads to re-synchronization of nodes time and again. This re-synchronization process is energy consuming whereas energy is constraints in WSN. This paper proposes a novel cluster based time synchronization technique for wireless sensor networks in which cluster head rotation is based on minimum clock offset. Simulation results based on energy analysis of the proposed model demonstrate that proposed novel cluster based time synchronization technique reduces the energy consumption and also the synchronization error compared with other existing protocols.

Keywords

Cluster, Offset, Delay, Time Synchronization

1. Introduction

Recent advancements in the electronic and communication technologies have resulted in the emergence of wireless sensor networks (WSNs). WSNs are composed of small size, low power and low cost wireless micro-sensors, known as Sensor Nodes (SNs) [1] [2] [3]. These SNs can be embedded in various objects in order to form intelligent multipurpose distributed systems. These SNs generally

perform three major tasks *i.e.* sensing, data processing and communication [4]. SNs are capable of sensing various environmental conditions such as sound, temperature, humidity, strain, acidity pressure, vibration, motion or pollutants [5] [6], etc.

Wireless Sensor Network (WSN) consists of large number of SNs deployed in unattended environment [7]. These SNs monitor objects in the sensing field and report the activities and events to sink. In order to establish correct logical order of the events, the data must be time stamped by the SNs. In WSN some applications such as military applications require accurate time of the event. Data fusion in WSN also demands time stamp of SNs to suppress the duplicate information of the SNs. Time is also important to implement the TDMA schedule in WSNs. Due to various challenges and constraints in WSNs, the clock synchronization protocols (NTP) [8] [9] designed by researchers for wired networks do not work in wireless sensor networks. In WSNs each node has its own local clock that may vary from the local clock of other nodes in the network which leads to clock offset. Higher clock offset degrade the WSNs performance and the information collected by the nodes may not provide the accurate timing of the event. Whereas time synchronization in WSNs is important for efficient duty cycling, location based monitoring, target/event tracking data fusion and network scheduling and routing.

Keeping in view the importance of time synchronization as well as the constraints of SNs, the objective of the proposed research is to develop time synchronization technique that helps to synchronize the SNs in an energy efficient manner. This paper proposes a novel cluster based time synchronization technique for wireless sensor networks to keep local clocks of all the SNs synchronized with global clock.

In wireless sensor networks, a lot of research [10]-[31] has already been carried out to design time synchronization protocols such as Reference Broadcast Synchronization (RBS) [11], Romer synchronization [13], TPSN (Timing Sync Protocols for Sensors Networks) [15], Two-hop time synchronization protocol for sensor networks (TTS) [18], Fast distributed multi-hop relative time synchronization protocol and estimators for wireless sensor networks [19], Long term and large scale time synchronization in wireless sensor networks (2LTSP) [20], and Average time synchronization in wireless sensor networks by pairwise messages (ATSP) [21], Time synchronization protocol based on spanning tree [28], Lightweight and Energy Efficient Time Synchronization (LEETS) Protocols [29] and Lightweight Fault-tolerant Time synchronization [30]. Some of these protocols synchronize the nodes internally by placing the nodes on common notion of time and some synchronize externally by using large number of messages to adjust the clocks of the sensor nodes with global clock.

The proposed novel cluster based time synchronization technique for wireless sensor networks to keep local clocks of all the SNs synchronized with global clock by using less number of messages. In the proposed technique initially the SNs are arranged in clusters by using energy efficient clustering algorithm

(EECA) and then time synchronization is performed.

The rest of the paper is organized as follows. In Section 2 describes the proposed energy efficient clustering algorithm (EECA) for WSNs. Section 3 describes the proposed time synchronization algorithm and synchronization error estimation. Energy analysis is performed in Section 4. Section 5 contains simulation and analysis and finally we conclude the paper in Section 6.

2. Energy Efficient Clustering Algorithm

In this section, we describe our proposed protocol, called Energy Efficient Clustering Algorithm (EECA) for wireless sensor networks. The proposed technique has been segregated into different phases; creation of clusters to prolong the network lifetime, CH selection and cluster head rotation. EECA forms clusters before selecting cluster-head. Our proposed EECA works in three steps: (2.1) Cluster formation process, (2.2) Cluster head selection process and (2.3) Cluster head rotation process.

2.1. Cluster Formation Process

In EECA clustering process is initiated by the sink. Sink calculates the preliminary mean points by sensing field size, optimum number of clusters and average distance of the nodes from the center of the sensing field. After the deployment of SNs in sensing field, the sink initiates the clustering process by finding the centroid of the sensing field by using the Equation (1) where $x_i = \frac{L_i}{2}$ and

$y_i = \frac{B_i}{2}$ where L_i and B_i is the size of the sensor field and A_i is the area of the sensing field.

$$\text{Centroid} = (\bar{x}, \bar{y}) \quad \text{where } \bar{x} = \frac{\sum x_i A_i}{\sum A_i} \text{ and } \bar{y} = \frac{\sum y_i A_i}{\sum A_i} \quad (1)$$

When the value of centroid is calculated, sink node finds the average distance d between centroid and all the SNs by using the Equation (2). Where X_i and Y_i are the coordinates of the node SN_i .

$$d = \left(\frac{\sum_{i=1}^n |X_i - \bar{x}|}{n}, \frac{\sum_{i=1}^n |Y_i - \bar{y}|}{n} \right) \quad (2)$$

Sink also computes the optimum number of clusters k using the Equation (3) [32]. The total number of nodes deployed in the sensing field is n , M is the side of square sensing field and $d_{to BS}$ is average distance to cluster heads from the base station. ε_{fs} is energy coefficient of power amplifier stage of SN for free space energy dissipation model and ε_{mp} is energy coefficient of power amplifier stage of SN for multipath energy dissipation model.

$$k = \frac{\sqrt{n}}{2\pi} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}} \frac{M}{d_{to BS}}} \quad (3)$$

Once the values of centroid, d and k are calculated, sink calculates the value of

the preliminary mean points $M_j(m_{j_x}, m_{j_y})$ where $j = 1, 2, \dots, k$ using Equation (4) and broadcasts the locations of preliminary mean points to the SNs.

$$\left. \begin{aligned} m_{j_x} &= d \times \cos\left(\frac{2\pi}{k} \times (j-1)\right) + \bar{x} \\ m_{j_y} &= d \times \sin\left(\frac{2\pi}{k} \times (j-1)\right) + \bar{y} \end{aligned} \right\} \quad (4)$$

The objective of finding the preliminary mean points is to significantly reduce the iteration time of cluster formation. **Figure 1** shows the preliminary mean points in $100 \text{ m} \times 100 \text{ m}$ sensing filed area with 50 sensor nodes.

When the SNs receive the sink message of preliminary mean points, each node find its minimum distance from preliminary mean points by using the Equation (5) and retain the minimum distance.

$$\text{Arg.Min} \sum_{j=1}^k \sum_{X_i \in C_j} \|X_i - m_j\|^2 \quad (5)$$

X_i is the coordinate of SN_i, C_j is the cluster j and m_j is the coordinate of preliminary mean points. The minimum distance between preliminary mean points and SNs helps to form uniform distributed clusters. Nodes join the cluster on the minimum distance to the preliminary mean points using the Equation (6). In each round SNs check their distance with all nearby clusters and node nearest to the preliminary mean point m_j in the r^{th} round joins the cluster j .

$$C_j^{(r)} = \left\{ X_i : |X_i - m_j^{(r)}|^2 \leq |X_i - m_{j^*}^{(r)}|^2 \text{ for all } j^* = 1, 2, \dots, k \right\} \quad (6)$$

$C_j^{(r)}$ is the Cluster j in r^{th} round. After tagging all the nodes with different clusters new mean points are created by using the Equation (7).

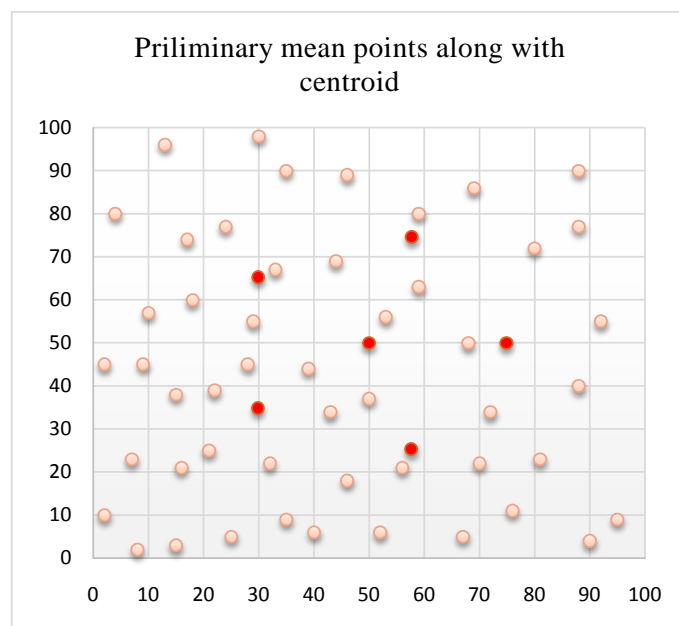


Figure 1. Preliminary mean points along with centroid where $n = 50$ and $k = 5$.

$$m_j^{(r+1)} = \frac{1}{|C_j^{(r)}|} \sum_{X_j \in C_j^{(r)}} X_j \quad (7)$$

$|C_j^{(r)}|$ is the number of SNs in the cluster j . **Figure 2** shows the new mean points.

The process of finding the new mean points is iterative till the new mean points stop moving. As long as the preliminary mean points keep on changing the nodes are re-arranged again and again as per the Equation (6) and (7). The cluster formation process is complete when the new mean points are fixed and nodes join the final mean points with minimum distance. **Figure 3** shows the flowchart of clustering process.

2.2. Cluster Head Selection

The clustering algorithm divides the whole network into different clusters. The next step is to elect CH within each cluster. Each node within the cluster calculates its distance from the cluster by using the Equation (8). Where X_i is the coordinates of the SN_i and C_j is the coordinates of the final mean point in which the SN_i tagged itself.

$$D_i = |X_i - C_j| \quad (8)$$

The cluster head selection is performed using back off timer strategy. SN_i sets its back off timer by using the Equation (9).

$$\text{Counter}_i = \frac{D_i}{D_{\text{average}}} * RV_i \quad (9)$$

RV_i is the random variable between [0.9, 1]. D_i is distance of SN_i from final mean point in a cluster. D_{average} is average distance of all the sensor nodes within a cluster.

The back off timer of each node starts decrementing and the node whose back

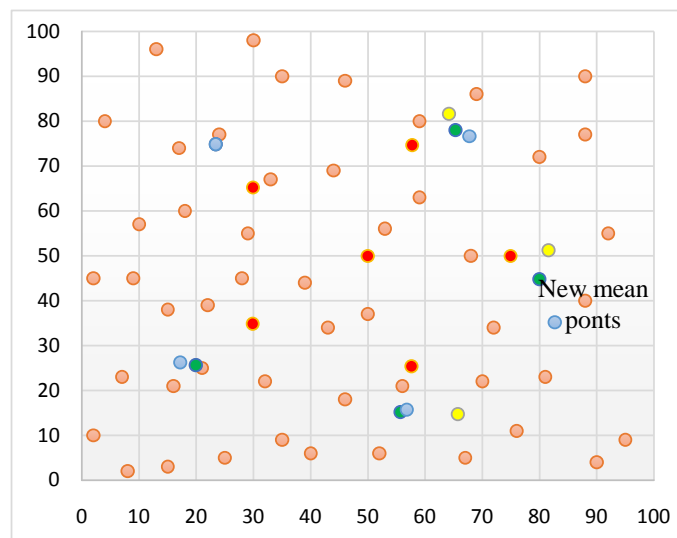


Figure 2. New mean points after three iterations where $n = 50$ and $k = 5$.

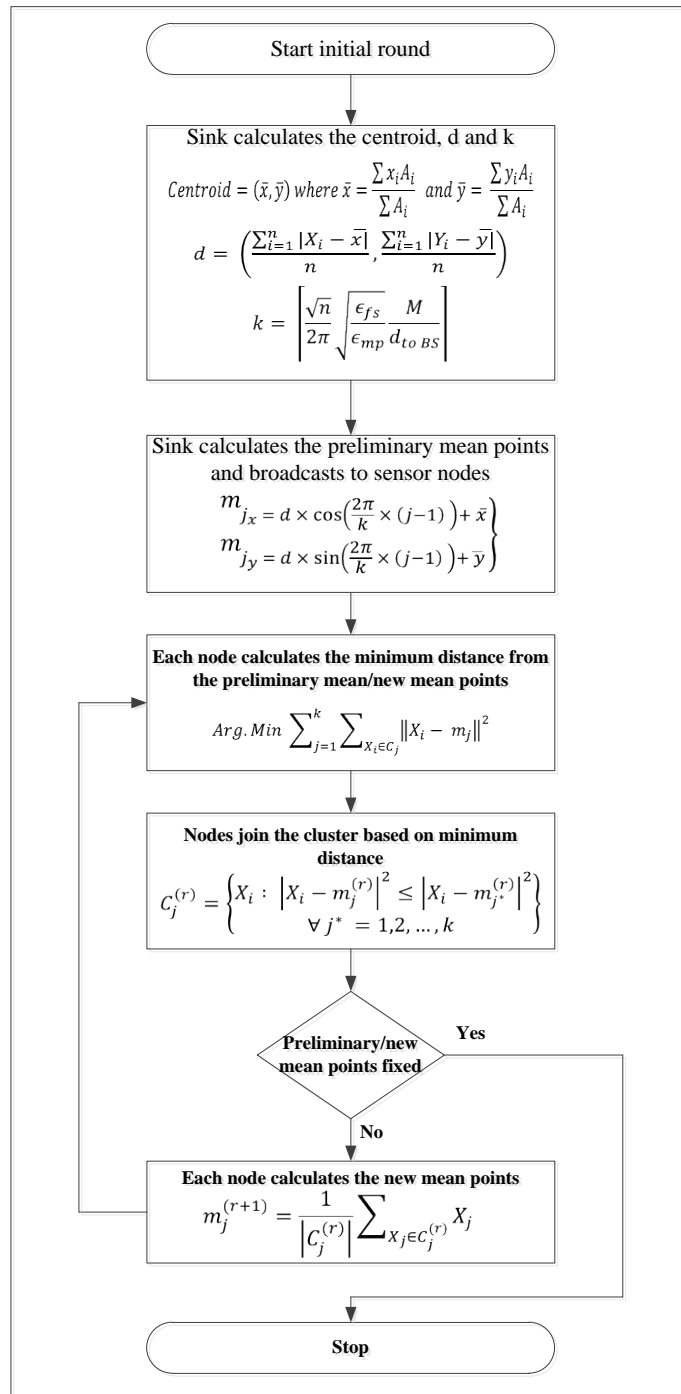


Figure 3. Flowchart of clustering process.

off timer expires first within the cluster, broadcasts a CH advertisement message to other nodes. The nodes receive a CH advertisement message joins the CH by sending the cluster member join message to CH. A node may receive more than one CH advertisement messages. If a node receives more than one message, checks the received signal strength of the messages and send the reply to the CH message received with larger signal strength. Figure 4 shows the clusters with their cluster heads formed after CH selection.

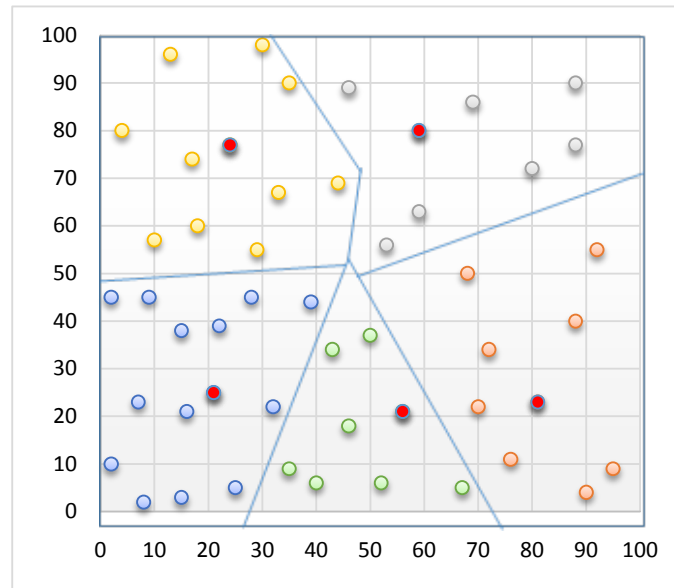


Figure 4. Clustering with CH where $n = 50$, $M = 100$ m and $k = 5$.

2.3. Cluster Head Rotation

The role of CH in a cluster must be rotated regularly among its members to prolong the network lifetime of sensor network by balancing the energy consumption of various sensor nodes. CH is required to perform extra tasks such as data gathering, data aggregation, etc. compared to the other sensor nodes. Energy consumption of CH is more compared to other sensor nodes therefore, some mechanism must be applied for CH rotation among the cluster members. A number of methods for CH rotation have been discussed in literature [33]-[38].

In CH rotation process, the new CH is selected from cluster members. CH rotation/re-election process is initiated when residual energy of a node falls below the threshold value. The new CH is elected with higher value of Candidacy factor (CF). Candidacy factor of SN_i is defined as

$$CF_i = \frac{E_{Resi}^i}{D_i \times \theta_i} \quad (10)$$

where E_{Resi}^i is residual energy of SN_i , D_i is the distance between SN_i and centroid of the cluster. θ_i is clock offset of SN_i . Each node sends its CF_i value to cluster head. CH would choose the node with highest CF_i value and pass information to sink and other members about new cluster head. **Figure 5** shows the flowchart of CH selection/rotation process.

3. Cluster Based Time Synchronization

The proposed cluster based time synchronization (CBTS) algorithm is based on external synchronization. Once the clustering process is complete using EECA protocol proposed in section 3, time synchronization process is initiated by the parent node, which is the sink/base station during the initial phase. Sink/base station is used as a reference node for time. The proposed algorithm is using k

CHs. There are k' CHs at level-1 and k'' CHs at level-2. The proposed model is shown in **Figure 6**.

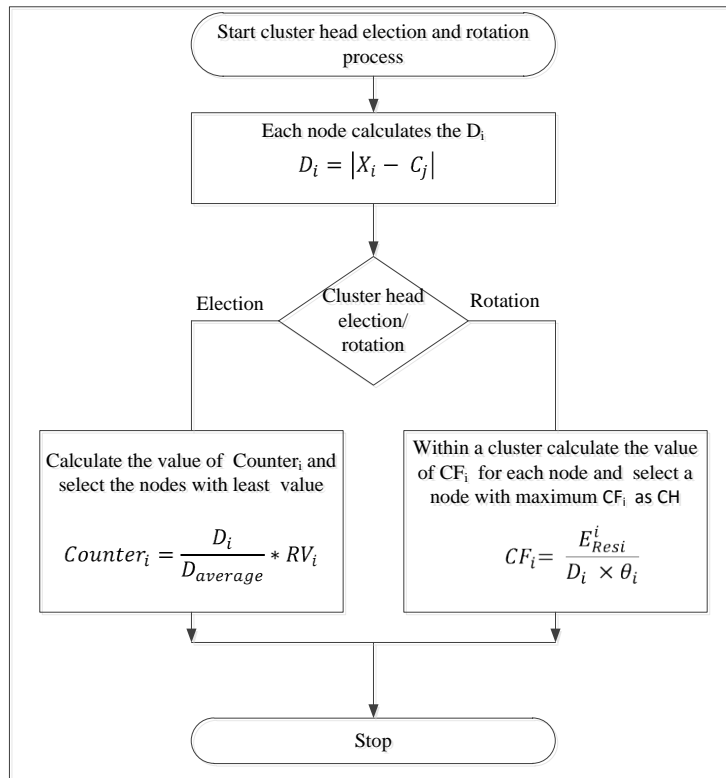


Figure 5. Flowchart for selection and rotation of cluster head.

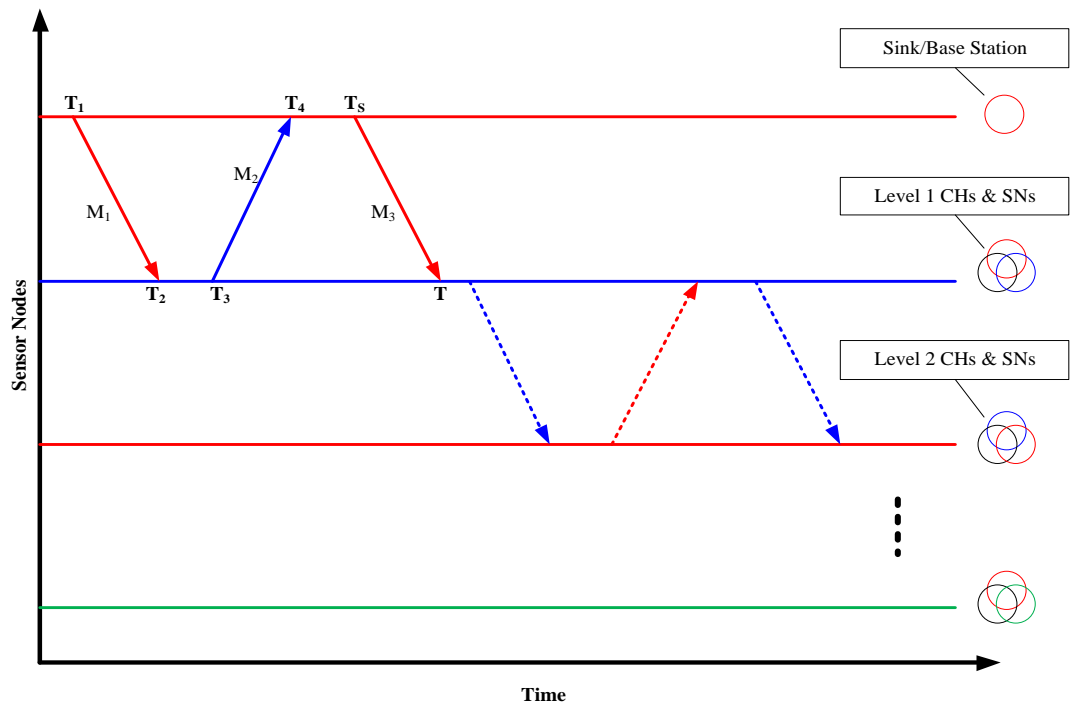


Figure 6. Exchange of timestamps among parent node, CHs and nodes.

The parent node (P_n) initiates the synchronization process by broadcasting the *Syn_start* packet which contains the timestamp T_1 and then waits for the reply. When the CHs of level 1 (CH_{L1}) receive *Syn_start* packet, they send reply to sink/base station by sending the *Syn_ack* packet. The *Syn_ack* packet contains the timestamp T_1, T_2, T_3 where time T_1 is the time of sending *Syn_start* packet, T_2 is the time when the CHs of level 1 receive the packet and T_3 time when CHs of level 1 send the *Syn_ack* packet. The parent node after receiving the *Syn_ack* packets, broadcasts *Syn_pkt* with timestamp T_1, T_2, T_3, T_4, T_s where T_4 is the time when the parent node receives the *Syn_ack* packet and T_s is the time to set the clocks of the CHs of level-1. All CHs of level-1 receive the *Syn_pkt* and calculate the offset (θ) and delay (δ) and set the clocks as:

$$T = T_s + \theta \pm \delta \quad (11)$$

In the initial synchronization phase, sink/base station functions as parent node. After synchronizing the CHs of level 1, it further functions as parent nodes and synchronizes the cluster nodes and CHs of level 2. Cluster based time synchronization algorithm is as below:

Sink Base Station

1. $T_m = t + t_a$
2. **while** nodes are not synchronized **do**
3. set SCOUNT = $k' \times T_m$
4. P_n broadcasts (Syn_start, T_1)
SCOUNT - -
5. **if** P_n wait for reply && SCOUNT = 0 **then** repeat step 4
6. **else**
 P_n receive (Syn_ack, T_1, T_2, T_3)
7. P_n broadcasts (Syn_Pkt, T_1, T_2, T_3, T_4, T_s) to CH_{L1}
8. **endif**

Cluster Heads

9. **if** CH_{L1} receives (Syn_start, T_1) **then** send (Syn_ack, T_1, T_2, T_3) to P_n
10. **else**
 CH_{L1} waits
11. **endif**
12. **if** CH_{L1} receives (Syn_Pkt, T_1, T_2, T_3, T_4, T_s) from P_n **then**
calculate $\delta = [(T_2 - T_1) + (T_4 - T_3)] / 2$
calculate $\theta = [(T_2 - T_1) - (T_4 - T_3)] / 2$
synchronize $T = T_s + \theta \pm \delta$
13. **else**
14. CH_{L1} waits
15. **endif**
16. Repeat step from 1 to 15 in next Synchronization Phase

In the algorithms, T_m is the maximum time required to receive a message from CH and P_n , or vice versa, whereas t is the current time and t_a is the maximum propagation delay. This algorithm sets the logical clock of the CHs and nodes with the global time.

Synchronization Error

All WSN synchronization schemes [10]-[31] have four basic packet delay components: send time, access time, propagation time, and receive time as shown in Figure 7. The send time is that of the sender constructing the time message to transmit on the network. The access time is that of the MAC layer delay in accessing the network. The time for the bits to be physically transmitted on the medium is considered the propagation time. Finally, the receive time is the time spent by the receiver for processing the message. The synchronization error is calculated by using the time stamp message shown in the Figure 6. In synchronization error calculation, access time is combined with send time. The main problem of time synchronization besides having a packet delay is that, it is able to foresee the time spent on each, which can be challenging. Disregarding any of these will significantly surge the performance of the synchronization technique.

In the proposed CBTS synchronization, a message from the sink to SNs transmits through multi-hops that induce the time delay error because of various delays as shown in Figure 7. Exchange of timestamp messages among parent node and cluster heads (shown in the Figure 6) are used to analyse the synchronization error of the proposed CBTS. Simply by using send time, propagation time, receive time and clock drift and then equating the timestamps of cluster head and parent node following equations obtained:

$$T_2^{i+1} = T_1^i + \alpha^i + \delta^{i \rightarrow i+1} + \beta^{i+1} + \gamma^{i \rightarrow i+1} \tag{12}$$

$$T_4^i = T_3^{i+1} + \alpha^{i+1} + \delta^{i+1 \rightarrow i} + \beta^i - \gamma^{i \rightarrow i+1} \tag{13}$$

T_2^{i+1} and T_4^i represent the timestamps of cluster head and parent node respectively, where $i = 1, 2, \dots, n$ is representing the level. α, β and δ represent the send time, receive time and propagation delay, respectively. γ is the clock drift of the node. To calculate one-hop synchronization error $\omega_{i \rightarrow i+1}$ subtract the Equation (12) from Equation (13) as:

$$T_2^{i+1} - T_4^i = T_1^i - T_3^{i+1} + \alpha^i - \alpha^{i+1} + \delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i} + \beta^{i+1} - \beta^i + \gamma^{i \rightarrow i+1} + 2\gamma^{i \rightarrow i+1} \tag{14}$$

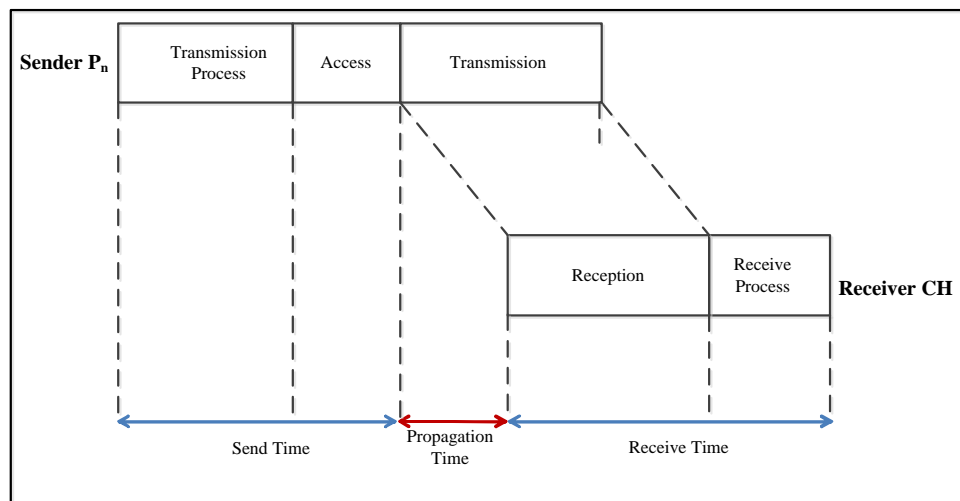


Figure 7. Time delay model.

Dividing both sides by 2

$$\frac{(T_2^{i+1} - T_1^i) - (T_4^i - T_3^{i+1})}{2} - \gamma^{i \rightarrow i+1} = \frac{\alpha^i - \alpha^{i+1}}{2} + \frac{\delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i}}{2} + \frac{\beta^{i+1} - \beta^i}{2} \quad (15)$$

$$\theta^{i \rightarrow i+1} - \gamma^{i \rightarrow i+1} = \frac{\alpha^i - \alpha^{i+1}}{2} + \frac{\delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i}}{2} + \frac{\beta^{i+1} - \beta^i}{2} \quad (16)$$

where Offset $\theta^{i \rightarrow i+1}$ is computed as $\theta^{i \rightarrow i+1} = \left[(T_2^{i+1} - T_1^i) - (T_4^i - T_3^{i+1}) \right] / 2$ and by subtracting the clock drift from the clock offset, synchronization error for one-hop $\omega_{i \rightarrow i+1}$ is obtained as:

$$\omega_{i \rightarrow i+1} = \frac{\alpha^i - \alpha^{i+1}}{2} + \frac{\delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i}}{2} + \frac{\beta^{i+1} - \beta^i}{2} \quad (17)$$

The Equation (17) can be modified to obtain multi-hop synchronization error as:

$$\omega_i = \frac{\alpha^i - \alpha^{i+1}}{2} + \frac{\alpha^{i+1} - \alpha^{i+2}}{2} + \dots + \frac{\delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i}}{2} + \frac{\delta^{i+1 \rightarrow i+2} - \delta^{i+2 \rightarrow i+1}}{2} \quad (18)$$

$$+ \dots + \frac{\beta^{i+1} - \beta^i}{2} + \frac{\beta^{i+2} - \beta^i}{2} + \dots$$

$$\omega_i = \frac{1}{2} \sum_{i=1}^l \left[(\alpha^i - \alpha^{i+1}) + (\delta^{i \rightarrow i+1} - \delta^{i+1 \rightarrow i}) + (\beta^{i+1} - \beta^i) \right] \quad (19)$$

Equation (19) calculates synchronization error for multi-hop communication. The objective of finding the synchronization error ω_i is to ensure that at any real time T , ω_i is between lower and upper bounds $[\min \varepsilon, \max \varepsilon]$. If the value of synchronization error goes below $\min \varepsilon$ or exceeds $\max \varepsilon$, the resynchronization process is initiated by the sink/base station.

4. Energy Analysis of CBTS

Energy analyses of CBTS algorithm is performed for l levels of CHs by using the energy models [32] [39] [40] [41] [42]. Since most of the energy is dissipated by the SNs in communication, therefore the energy analysis is performed for the transmission and reception of m bit message for a distance of d . To transmit m bit message over a distance d , the energy cost of transmission (E_{Tx}) is as:

$$E_{Tx}(m, d) = \begin{cases} m * E_{elec} + m * \varepsilon_{fs} * d^2 & \text{if } d \leq d_0 \\ m * E_{elec} + m * \varepsilon_{mp} * d^4 & \text{if } d > d_0 \end{cases} \quad (20)$$

E_{elec} is the energy dissipation per bit to run transmitter and receiver electronics circuitry. ε_{fs} is the energy coefficient of power amplifier stage of sensor nodes for free space energy dissipation model when transmission distance is less than threshold *i.e.* $d < d_0$. ε_{mp} is the energy coefficient of power amplifier stage of sensor nodes for multipath energy dissipation model when transmission distance is greater than threshold *i.e.* $d \geq d_0$. For $d = d_0$, distance threshold can be calculated as $d_0 = \sqrt{\varepsilon_{fs} / \varepsilon_{mp}}$.

The energy consumed to receive m bit data is expressed as:

$$E_{Rx} = m * E_{elec} \quad (21)$$

Initially two-level network is considered, where

Total number of nodes	=	n
Total number of cluster heads	=	k
Number of level 1 CHs	=	k'
Number of level 2 CHs	=	k''
Number of normal sensor nodes	=	$n - k$

The proposed CBTS algorithm uses $n + 2(k + 1)$ messages to synchronize n number of SNs. The energy dissipation for receiving and transmitting m bit time synchronization message derived below depends on the number of messages received and transmitted by the SNs. As the sink/base station is at level 0 with no constraint of energy, the energy exhausted during transmitting and receiving m bit time synchronization message at level 1 CHs and level 2 CHs is calculated by using the Equations (20) and (21).

Energy consumption at level 1 CHs: Total energy consumed in CBTS protocol while receiving and sending synchronization messages at level 1 CHs is given by Equation (22) and (23) ($E_{Rx,1}, E_{Tx,1}$) respectively and the total energy consumed at level 1 CHs is given by Equation (24) ($E_{Total,1}$).

$$E_{Rx,1} = 3k'mE_{elect} \quad (22)$$

$$E_{Tx,1} = mE_{elect} + m\varepsilon_{mp}d^4toP_n + 2k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] \quad (23)$$

$$E_{Total,1} = 3k'mE_{elect} + mE_{elect} + m\varepsilon_{mp}d^4to P_n + 2k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] \quad (24)$$

Energy consumption at level 2 CHs: Energy consumed by level 2 CHs while receiving and transmitting synchronization messages is given by Equations (25) and (26) ($E_{Rx,2}, E_{Tx,2}$) respectively and the total energy consumed by level 2 CHs is given by Equation (27) ($E_{Total,2}$).

$$E_{Rx,2} = 3k''mE_{elect} \quad (25)$$

$$E_{Tx,2} = k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k'} \right] + 2k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi(N-k)} \right] \quad (26)$$

$$E_{Total,2} = 3k''mE_{elect} + k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k'} \right] + 2k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi(N-k)} \right] \quad (27)$$

Energy consumption at SNs level: Energy consumed by SNs while receiving and sending synchronization messages is given by Equations (28) and (29)

($E_{Rx,sensor}, E_{Tx,sensor}$) respectively and the total energy consumed by SNs for receiving and sending time synchronization messages is given by Equation (30) ($E_{Total,sensor}$).

$$E_{Rx,sensor} = 2(n-k)mE_{elect} \quad (28)$$

$$E_{Tx,sensor} = k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] \quad (29)$$

$$E_{Total,sensor} = 2(n-k)mE_{elect} + k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] \quad (30)$$

Total energy consumption for communication: The energy consumption rate for sensors in a WSN contrasts significantly on the basis of the protocols the sensors employ for communications. However, the total energy expended for communication of synchronization messages in the proposed model is given by Equation (31) which is further simplified in Equations (32)-(39). Therefore, the total energy dissipation for a single round of time synchronization for l level of hierarchical sensor networks is given by Equation (39).

$$E_{Grand\ Total} = 3k'mE_{elect} + mE_{elect} + m\varepsilon_{mp}d^4toP_n + 2k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] + 3k''mE_{elect} + k' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] + 2k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi(N-k)} \right] \quad (31)$$

$$+ 2(n-k)mE_{elect} + k'' \left[mE_{elect} + m\varepsilon_{fs} \frac{M^2}{\pi k''} \right] = 3k'mE_{elect} + mE_{elect} + m\varepsilon_{mp}d^4toP_n + 2k'mE_{elect} + 2k'm\varepsilon_{fs} \frac{M^2}{\pi k''} + 3k''mE_{elect} + k'mE_{elect} + k'm\varepsilon_{fs} \frac{M^2}{\pi k''} + 2k''mE_{elect} + 2k''m\varepsilon_{fs} \frac{M^2}{\pi(n-k)} \quad (32)$$

$$+ 2nmE_{elect} - 2kmE_{elect} + k''mE_{elect} + k''m\varepsilon_{fs} \frac{M^2}{\pi k''} = 3k'mE_{elect} + 2k'mE_{elect} + k'mE_{elect} + 3k''mE_{elect} + 2k''mE_{elect} + k''mE_{elect} + 2nmE_{elect} + mE_{elect} + 2k'm\varepsilon_{fs} \frac{M^2}{\pi k''} + k'm\varepsilon_{fs} \frac{M^2}{\pi k''} + k''m\varepsilon_{fs} \frac{M^2}{\pi k''} + 2k''m\varepsilon_{fs} \frac{M^2}{\pi(n-k)} + m\varepsilon_{mp}d^4toP_n + 2k'mE_{elect} \quad (33)$$

Solving above, following equation is obtained:

$$= (3k' + 2k' + k' + 3k'' + 2k'' + k'')mE_{elect} - 2kmE_{elect} + 2nmE_{elect} + mE_{elect} + \left[\frac{2k'}{k''} \right] m\varepsilon_{fs} \frac{M^2}{\pi} + 2m\varepsilon_{fs} \frac{M^2}{\pi} + 2k''m\varepsilon_{fs} \frac{M^2}{\pi(n-k)} + m\varepsilon_{mp}d^4toP_n \quad (34)$$

$$= 6(k' + k'')mE_{elect} - 2kmE_{elect} + 2nmE_{elect} + mE_{elect} + \left[\frac{2k''}{(n-k)} + \frac{2k'}{k''} + 2 \right] m\varepsilon_{fs} \frac{M^2}{\pi} + m\varepsilon_{mp}d^4toP_n \quad (35)$$

$$= 4kmE_{elect} + 2nmE_{elect} + mE_{elect} + \left[\frac{2k''}{(n-k)} + \frac{2k'}{k''} + 2 \right] m\varepsilon_{fs} \frac{M^2}{\pi} + m\varepsilon_{mp}d^4toP_n \quad (36)$$

$$= (4k + 2n + 1)mE_{elect} + \left[\frac{2k''}{(n-k)} + \frac{2k'}{k''} + 2 \right] m\varepsilon_{fs} \frac{M^2}{\pi} + m\varepsilon_{mp}d^4toP_n \quad (37)$$

$$= (4k + 2n + 1)mE_{elect} + 2 \sum_{i=2}^n \left[\frac{k^i}{(n-k)} + \frac{k^{i-1}}{k^i} + 1 \right] m\varepsilon_{fs} \frac{M^2}{\pi} + m\varepsilon_{mp}d^4toP_n \quad (38)$$

Equation for l Level ($i > 2$) can be standardized as:

$$E_{total,i} = (4k + 2n + 1)mE_{elect} + i \sum_{i=2}^l \left[\frac{k^i}{(n-k)} + \frac{k^{i-1}}{k^i} + 1 \right] m\varepsilon_{fs} \frac{M^2}{\pi} + m\varepsilon_{mp}d^4toP_n \quad (39)$$

where the value of d_{toCH}^2 is taken as $\frac{M^2}{\pi k}$ and the value of $d^4_{toP_n}$ can be obtained as:

$$d^4_{toP_n} = \int_{y=0}^{y=M} \int_{x-x_{P_n}-M}^{x-x_{P_n}} \frac{\left(\sqrt{x^2 + (y - y_{P_n})}\right)^4}{M^2} dx dy. \quad (40)$$

5. Simulation and Results

In this section, the performance of Cluster Based Time Synchronization (CBTS) algorithm has been evaluated through simulation. The simulation has been performed in MATLAB 2013a. The performance of CBTS protocol is compared with TPSN [15] and TTS [18] protocols. The performance metrics include the number of nodes in the WSN, initial energy of the SNs, total number of messages per synchronization round, total energy dissipation per synchronization round, propagated synchronization error and convergence time. The simulation parameters are listed in **Table 1** where for each parameter, simulation has been run many times and the average result of all runs has been taken for evaluation.

The performance evaluation includes message complexity, total energy dissipation per synchronization round, propagated synchronization error with number of hops and convergence time.

The message complexity in terms of the total number of time synchronization messages per synchronization round while the number of nodes varies from 50 to 500 and the sensing field area is kept constant *i.e.* 300 m × 300 m as shown in the **Figure 8**. The simulation results show that the number of time synchronization messages increase with the increase in the number of nodes in a constant field size because increase in number of nodes leads to increase in time synchronization messages. There is a significant reduction in terms of time synchronization messages in the proposed CBTS method than TPSN and TTS methods. TPSN requires $n + n$ messages to synchronize n SNs and TTS requires $n + n/2$ messages. CBTS reduces the number of synchronization messages by 40% as compared to TPSN and 25% as compared to TTS. The proposed CBTS method

Table 1. Simulation parameters.

Parameter	Default Value	Range
Network size (side of square sensor field)	300 m	100 m ~ 500 m
Number of nodes	300	50 ~ 500
Initial energy of node	2 Joule	
Sink location	(0, 0)	
Data packet size (k)	16 KB	
E_{elec}	50 nJ/bit	
ϵ_{fs}	10 pJ/bit/m ²	
ϵ_{mp}	0.00134 pJ/bit/m ⁴	
Data rate	10 Kbps	
Transmission range	75 m	

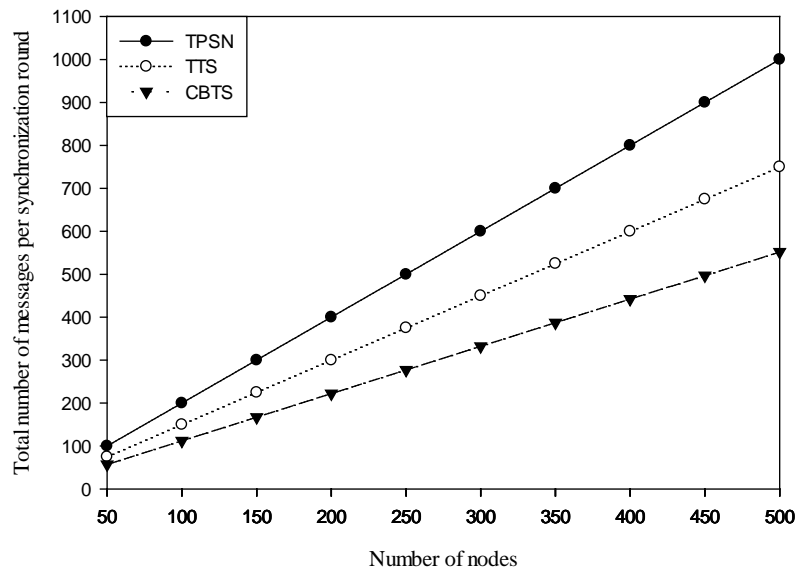


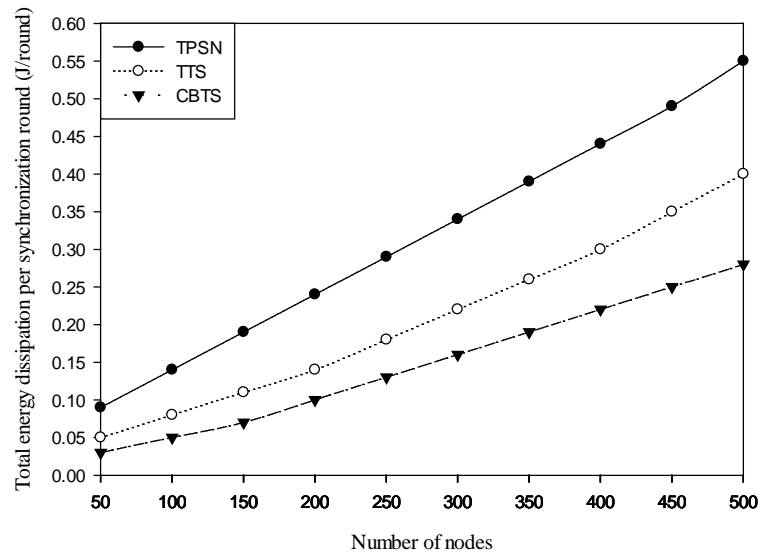
Figure 8. Variation in time synchronization messages with number of nodes.

provides the best results in comparison with other protocols. Fewer time synchronization messages decrease the overhead in terms of communication and thus increase the network lifetime.

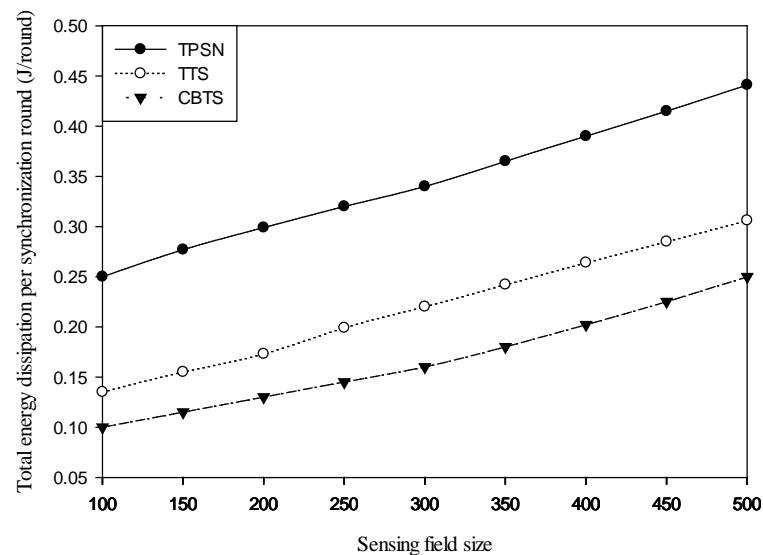
Total energy dissipation is calculated as the amount of energy dissipated within the sensor network to perform time synchronization. **Figure 9** shows the comparison of simulation results of energy variation for different protocols such as TPSN, TTS and CBTS. The variation in the energy dissipation is studied by varying number of nodes from 50 to 500, while keeping the sensing field size constant as 300 m. It has been observed that the energy dissipation per round increases with increase in the number of nodes. Increased number of nodes leads to increase in the number of time synchronization messages which increases the total routing energy and thus total node energy consumed. The simulation results in **Figure 9(a)** show that CBTS protocol provides better performance compared to other two protocols. This is due to the less number of synchronization messages and better clustering technique as compared to TPSN and TTS.

It has been observed that the energy dissipation incessantly increases with the increase in the sensing field size. The distance between the sink and the sensing node increases with the increase in the sensing field size, which increases the transmission energy cost, although the time synchronization messages remain the same. Aforementioned, the performance of cluster based time synchronization (CBTS) is better than other two protocols due to the reduction in the number of synchronization messages and better clustering approach.

Figure 9(b) shows the total energy dissipation per round for variation in the sensing field sized between 100 m to 500 m while the number of nodes remains constant at 300 nodes. It has been observed that the energy dissipation goes on increasing with the increase in the sensing field size. The distance between the sink and the sensing node increases with the increase in the sensing field size, which increases the transmission energy cost. In case of TPSN protocol the value



(a)



(b)

Figure 9. Variation in total energy dissipation per synchronization round with (a) number of nodes in sensing field, and (b) sensing field size.

is the highest because it uses $n + n$ messages to synchronize n nodes as compared to TTS and CBTS.

Synchronization error is the difference between the times of a CH with respect to the times of the sink. The synchronization error is added with the increasing number of hops generally. The variation in the synchronization error is studied by fixed number of nodes *i.e.* 300 nodes, while keeping the sensing field size constant at 300 m. The synchronization error increases with number of hops for CBTS, TTS and TPSN protocols in particular as shown in **Figure 10**. The values shown in the **Figure 10** are the average of 50 simulation results.

The synchronization error of TPSN is high *i.e.* 16 μ s for one hop and 46 μ s for five hops because of the topology and number of messages required for synchronization of the TPSN network. Synchronization error in TPSN is high because it

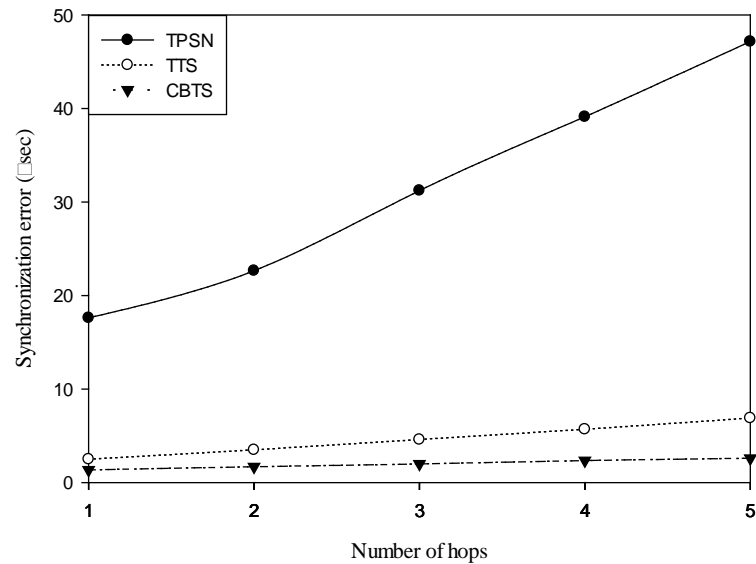


Figure 10. Variation in propagated synchronization error with number of hops.

uses large number of messages, since several messages bump into each other causing a collision and required to be sent again and leads to error. A large amount of message transmission leads to collisions and further to performance degradation. Synchronization error of TTS is also high because of using a pair of nodes to synchronize the SNs which increase number of messages and adds errors at each hop. The performance of cluster based time synchronization (CBTS) is better than other two protocols *i.e.* 2.6 μ s for five hops due to less number of synchronization messages and better clustering approach which reduces the packet collisions and thereby reducing the synchronization error. Simulation results show that our proposed protocol outperforms other protocols in terms of synchronization error. In case of TPSN, the unbalanced clustering technique and the overhead of messages for synchronization are responsible for higher synchronization error. CBTS uses EECA clustering technique which required less number of synchronization messages to synchronize the sensor network.

The total time required to synchronize the network is known as convergence time. The variation in the convergence time is studied by fixed number of nodes *i.e.* 300 nodes, while keeping the sensing field size constant as 300 m. Convergence time is measured for five hops and observes that it increases with number of hops for TPSN, TTS and CBTS protocols as shown in **Figure 11**. Convergence time is directly proportional to message complexity and bandwidth use. TPSN requires $n + n$ messages to synchronize n SNs and TTS requires $n + n/2$ time synchronization messages, therefore the convergence time for TPSN and TTS is very high. In TPSN for five hops it is 4 s and for TTS it is 3 s. In CBTS it is 2 s for five hops. Simulation results shows that CBTS protocol outperforms the other protocols in terms of convergence time with respect to number of hops. In TPSN convergence time is higher because of unbalanced hierarchical technique and more number of messages. CBTS uses less number of messages and better clustering topology.

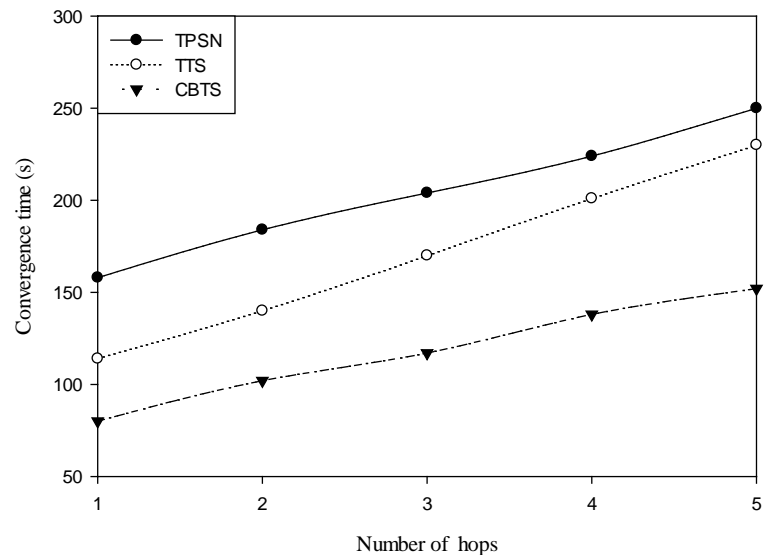


Figure 11. Variation in convergence time with number of hops.

6. Conclusion

The proposed cluster based time synchronization protocol ensures the synchronization of the nodes with global time. The proposed CBTS protocol uses the EECA clustering technique proposed in section 2 for clustering the network. Synchronization is performed using top to bottom approach. CHs synchronize with the sink and SNs synchronize with their respective CHs. The proposed technique uses $n + 2$ messages to synchronize the n SNs. Therefore, the proposed CBTS is suitable for time synchronization in energy efficiency manner. Simulation results demonstrate the effectiveness of the proposed protocol in terms of message complexity, total energy dissipation per synchronization round, propagated synchronization error, convergence time and synchronization precision level. The simulation results are compared with TPSN and TTS protocols. The proposed CBTS protocol provides better results in comparison with these protocols and can be used in WSNs for less energy consumption and prolonging the network lifetime.

References

- [1] Akyldiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) Wireless Sensor Networks: A Survey. *Computer Networks*, **38**, 393-422.
- [2] Mottola, L. and Picco, G.P. (2011) Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art. *ACM Computing Surveys*, **43**, Article No. 19. <https://doi.org/10.1145/1922649.1922656>
- [3] Kuorilehto, M., Hännikäinen, M. and Hämäläinen, T.D. (2005) A Survey of Application Distribution in Wireless Sensor Networks. *EURASIP Journal on Wireless Communication and Networking*, **2005**, Article ID: 859712. <https://doi.org/10.1155/WCN.2005.774>
- [4] Hwang, S. and Baek, Y. (2005) Reliable Time Synchronization Protocol for Wireless Sensor Networks. In: Yang, L.T., Amamiya, M., Liu, Z., Guo, M. and Rammig, F.J., Eds., *Embedded and Ubiquitous Computing—EUC 2005. Lecture Notes in Com-*

puter Science, Vol. 3824, Springer, Berlin, Heidelberg, 663-672.

https://doi.org/10.1007/11596356_66

- [5] Thakur, S., Reddy, M.V., Siddavattam, D. and Paul, A.K. (2012) A Fluorescence Based Assay with Pyranine Labeled Hexa-Histidine Tagged Organophosphorus Hydrolase (OPH) for Determination of Organophosphates. *Sensors and Actuators B: Chemical*, **163**, 153-158.
- [6] Thakur, S., *et al.* (2011) A Flow Injection Immunosensor for the Detection of Atrazine in Water Samples. *Sensors and Transducers Journal*, **31**, 91-100.
- [7] Romer, K., *et al.* (2002) Middleware Challenges for Wireless Sensor Networks. *ACM SINGMOBILE Mobile Computing and Communication Review*, **6**, 59-61. <https://doi.org/10.1145/643550.643556>
- [8] Mills, D.L. (1991) Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions of Communications*, **39**, 1482-1493. <https://doi.org/10.1109/26.103043>
- [9] Mills, D. (1992) Network Time Protocol (Version 3) Specification, Implementation and Analysis. Technical Report, University of Delaware, Delaware.
- [10] Mills, D. (1995) Improved Algorithms for Synchronizing Computer Network Clocks. *IEEE/ACM Transactions on Networking* **3**, 245-254. <https://doi.org/10.1109/90.392384>
- [11] Elson, J., Girod, L. and Estrin, D. (2002) Fine-Grained Network Time Synchronization Using Reference Broadcasts. *5th Symposium on Operating Systems Design and Implementation*, **36**, 147-163. <https://doi.org/10.1145/1060289.1060304>
- [12] Elson, J. and Romer, K. (2003) Wireless Sensor Networks: A New Regime for Time Synchronization. *ACM SIGCOMM Computer Communication Review*, **33**, 149-154.
- [13] Romer, K. (2001) Time Synchronization in Ad Hoc Networks. *ACM Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, CA, 4-5 October 2001, 173-182. <https://doi.org/10.1145/501416.501440>
- [14] Mock, M., *et al.* (2000) Continuous Clock Synchronization in Wireless Real-Time Applications. *19th IEEE Symposium on Reliable Distributed Systems*, Nurnberg, 16-18 October 2000, 125-133. <https://doi.org/10.1109/RELDI.2000.885400>
- [15] Ganeriwal, S., *et al.* (2003) Timing-Sync Protocol for Sensor Networks. *International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, 5-7 November 2003, 138-149. <https://doi.org/10.1145/958491.958508>
- [16] Ganeriwal, S., *et al.* (2003) Network-Wide Time Synchronization in Sensor Networks. Technical Report, Networked and Embedded Systems Lab, Electronic Engineering Department, UCLA.
- [17] Maróti, M., *et al.* (2004) The Flooding Time Synchronization Protocol. *2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, 3-5 November 2004, 39-49.
- [18] Wang, J., *et al.* (2014) Two-Hop Time Synchronization Protocol for Sensor Networks. *EURASIP Journal on Wireless Communications and Networking*, **2014**, 39-48. <https://doi.org/10.1186/1687-1499-2014-39>
- [19] Djenouri, D., *et al.* (2013) Fast Distributed Multi-Hop Relative Time Synchronization Protocol and Estimators for Wireless Sensor Networks. *Ad Hoc Networks*, **11**, 2329-2344.
- [20] Huang, G., *et al.* (2014) Long Term and Large Scale Time Synchronization in Wireless Sensor Networks. *Computer Communications*, **37**, 77-91.

- [21] Wu, J., *et al.* (2012) Average Time Synchronization in Wireless Sensor Networks by Pairwise Messages. *Computer Communications*, **35**, 221-233.
- [22] PalChaudhuri, S., *et al.* (2003) Probabilistic Clock Synchronization Service in Sensor Networks. Technical Report TR 03-418, Department of Computer Science, Rice University, Houston.
- [23] Li, Q. and Rus, D. (2004) Global Clock Synchronization in Sensor Networks. *IEEE Conference on Computer Communications*, **1**, 564-574.
- [24] Ping, S. (2003) Delay Measurement Time Synchronization for Wireless Sensor Networks. Intel Research, IRB-TR-03-013.
- [25] Arvind, K. (1994) Probabilistic Clock Synchronization in Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, **5**, 474-487.
<https://doi.org/10.1109/71.282558>
- [26] Sichitiu, M.L. and Veerarittiphan, C. (2003) Simple, Accurate Time Synchronization for Wireless Sensor Networks. *IEEE Conference on Wireless Communications and Networking*, New Orleans, LA, 16-20 March 2003, 1266-1273.
- [27] Su, W. and Akyildiz, I. (2005) Time-Diffusion Synchronization Protocols for Sensor Networks. *IEEE/ACM Transactions on Networking*, **13**, 384-397.
<https://doi.org/10.1109/TNET.2004.842228>
- [28] He, L.-M. (2008) Time Synchronization Based on Spanning Tree for Wireless Sensor Network. *4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, 12-14 October 2008, 1-4.
- [29] Xu, M., Zhao, M. and Li, S. (2005) Lightweight and Energy Efficient Time Synchronization for Sensor Network. *International Conference on Wireless Communications, Networking and Mobile Computing*, **2**, 947-950.
- [30] Searesavetrat, S., Pornavalai, C. and Varakulsiripunth, R. (2008) A Light-Weight Fault-Tolerant Time Synchronization for Wireless Sensor Networks. *8th International Conference on ITS Telecommunications*, Phuket, 24-24 October 2008, 182-186.
- [31] Park, C., *et al.* (2007) Reference Interpolation Protocol for Time Synchronization in Wireless Sensor Networks. *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing*, Hong Kong, 11-13 July 2007, 684-695.
- [32] Heinzelman, W.B., *et al.* (2002) An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *Transactions on Wireless Communications*, **1**, 660-670. <https://doi.org/10.1109/TWC.2002.804190>
- [33] Chandrasekaran, V. and Shanmugam, A. (2013) An Energy Efficient Multi Hop Hierarchical Routing in Wireless sensor Networks. *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, **5**, 61-65.
<https://doi.org/10.9790/2834-0546165>
- [34] Zhao, F., Xu, Y. and Li, R. (2012) Improved LEACH Routing Communication Protocol for a Wireless Sensor Network. *International Journal of Distributed Sensor Networks*, 1-6.
- [35] Manjeshwar, A. and Agrawal, D. (2001) TEEN: A Routing for Enhanced Efficiency in Wireless Sensor Networks. *15th International Parallel and Distributed Processing Symposium (IPDPS)*, San Francisco, CA, 23-27 April 2000, 2009-2015.
<https://doi.org/10.1109/ipdps.2001.925197>
- [36] Manjeshwar, A. and Agrawal, D. (2002) APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS)*, 15-19 April 2002.

- [37] Nehra, V. and Sharma, A.K. (2013) PEGASIS-E: Power Efficient Gathering in Sensor Information System Extended. *Global Journal of Computer Science and Technology*, **13**, 14-18.
- [38] Demirbas, M., *et al.* (2004) FLOC: A Fast Local Clustering Scheme in Wireless Sensor Networks. *Workshop on Dependability Issues in Wireless Ad Hoc Network and Sensor Networks (DIWANS)*.
- [39] Zhang, H. and Shen, H. (2009) Balancing Energy Consumption to Maximize Network Lifetime in Data-Gathering Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, **20**, 1526-1539. <https://doi.org/10.1109/TPDS.2008.252>
- [40] Ferng, H.W., *et al.* (2012) Energy Efficient Routing Protocol for Wireless Sensor Networks with Static Clustering and Dynamic Structure. *Wireless Personal Communications*, **65**, 347-367. <https://doi.org/10.1007/s11277-011-0260-4>
- [41] Wu, X., Chen, G. and Das, S.K. (2008) Avoiding Energy Holes in Wireless Sensor Networks with Non-Uniform Node Distribution. *IEEE Transactions on Parallel and Distributed Systems*, **19**, 710-720. <https://doi.org/10.1109/TPDS.2007.70770>
- [42] Halgamuge, M., *et al.* (2009) An Estimation of Sensor Energy Consumption. *Progress in Electromagnetics Research B*, **12**, 259-295. <https://doi.org/10.2528/PIERB08122303>



Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact wsn@scirp.org