Energy Consumption Analysis of Consensus Time Synchronization Algorithms for Wireless Sensor Networks

Niranjan Panigrahi 1,*, Pabitra Mohan Khilar 2

1 Department of CSE, PMEC, Berhampur, Odisha, India
2 Department of CSE, NIT, Rourkela, Odisha, India

Abstract — Wireless Sensor Networks (WSNs) have received considerable attention in recent years because of its broad area of applications. In the same breadth, it also faces many challenges. Time synchronization is one of those fundamental challenges faced by WSN being a distributed system. Several approaches have been proposed in the last decade for time synchronization in WSNs. Recently, Consensus Time Synchronization (CTS) approaches are gaining popularity due its computational lightness, robustness and distributed nature. Though a rich set of CTS algorithms are proposed, their energy consumption has so far not been studied. Apart from synchronization precision, energy consumption should also be considered meticulously for time synchronization algorithms in energy-constraint WSNs. In this paper, a thorough energy consumption analysis is presented for some recent state-of-the-art CTS algorithms for WSN and tested by simulation. The simulation results will help in selecting an appropriate CTS algorithm that meets the requirements of synchronization accuracy and energy consumption for a specific WSN application.

Keywords — Wireless Sensor Network; Consensus Time Synchronization; Energy Analysis.

I. INTRODUCTION

Time synchronization is a fundamental challenge to any distributed system because of absence of centralized clock. Being a distributed system, WSNs also face the same challenge [1]. A rich set of time synchronization algorithms have been proposed in the literature for traditional wired networks as well as wireless sensor networks in the near past. Being distributed systems, both wired and wireless networks carry some common characteristics. But, time synchronization issues and mechanism in WSNs are quite different from that in wired networks due to certain fundamental differences between these two types of distributed systems [2]. Hence, time synchronization algorithms designed for wired networks cannot be directly used in WSNs.

For the last decade, number of synchronization algorithms has been proposed for WSNs. Some of the well-known works [3, 4, 5, 6] are based on synchronizing to a reference node’s time by considering a hierarchical backbone for the network. Recently, to develop fully distributed and internal time synchronization mechanism, Consensus Time Synchronization (CTS) method has gained much attention [7, 8, 9, 10, 11, 12, 13, 14]. CTS method is essentially based on distributed average consensus principle which states that all the nodes in a network can converge to a consensus or synchronized state after a finite number of iterations, by communicating and performing averaging only with the neighbors. Because of its simplicity, computational lightness, robustness to node or link failure and purely distributed nature, CTS is more suitable for WSNs. But, due to iterative in nature, a series of message exchanges are carried out to achieve the desired synchronization accuracy. This series of message exchange is the main source of energy consumption in CTS algorithms. Though most of CTS algorithms concentrate on convergence speed and synchronization precision, the energy consumption to achieve synchronization is not yet analyzed. The main contributions of this paper are the followings.

* Corresponding author can be contacted via the journal website.

• A comprehensive energy consumption analysis is carried out for some recent state-of-the-art CTS algorithms for WSNs.
• Simulations are performed to validate the analysis.
The rest of the paper is organized as follows. Section II gives a brief related work about CTS algorithms for WSNs. Section III presents the basic system models used for the analysis. Section IV gives the detail energy consumption analysis of three recent state-of-the-art CTS algorithms. Section V shows the simulation results, followed by conclusion in Section VI.

II. BACKGROUND

In recent years, consensus approach from control theory has been widely implemented in many problems of computer science, e.g., peer-to-peer network [15], load balancing in distributed system and sensor network [16]. Consensus problem is the problem of making the scalar states of a set of agents converge to the same value using local communication [17, 18]. Among different types of consensus (max value consensus, min value consensus, and average consensus), the average consensus has gained more popularity because of its feasibility in many applications [19]. The average consensus principle is the mostly adapted principle in recent time synchronization algorithms for wireless sensor network. Some of the recent and state-of-the-art consensus based time synchronization algorithms are presented below.

In most of CTS algorithms, every node in the network participates in the consensus seeking process by communicating with the neighboring nodes. Different authors have proposed different averaging schemes for faster consensus convergence. Based on the averaging schemes, the approaches can be further divided into two types, viz., (i) weighted averaging scheme, and (ii) pair-wise averaging scheme.

In [8], the authors have proposed three weighted averaging consensus methods for clock synchronization, namely, Cumulative Moving Average (CMA), Forwards Weighted Average (FWA), and Confidence Weighted Average (CWA). Using a network of 100 nodes and random linear drift, the authors claimed that CWA proved most reliable. Also, FWA performance is same as CWA and has advantage of reduced computational complexity. The synchronization algorithm in [9] is also based on weighted averaging method. It uses cascading of two consensus algorithm, one for skew compensation and another for offset compensation. The communication protocol used is pseudo-periodic broadcast. The algorithm is claimed to be fully distributed, asynchronous, and computationally light.

In [20], four protocols are proposed, namely, all-node-based method, cluster-based method, diffusion based method and fault tolerant diffusion-based method. The first two methods require a node to initiate the synchronization process. So, these are not fault tolerant and localized. The last two are based on local communication and can achieve the aver-age consensus. These protocols are also analyzed in the presence of byzantine fault and claimed to be fault tolerant. The authors in [21] presented a pure average consensus based synchronization algorithm. It is claimed that the proposed algorithm is fully distributed, asynchronous, includes skew compensation and computationally light. It is also robust to dynamic topology. Similarly, in [22], an average synchronization algorithm is proposed with non-linear dynamical network and with random time delays.

The work in [14] proposes a selective averaging scheme for faster consensus convergence and synchronization accuracy. In [13], the authors have proposed a cluster consensus based time synchronization method. The basic objective of embedding clustering into time synchronization is to minimize energy consumption and to achieve faster convergence. To the best of our knowledge, the work in [13] is the first work whose objective is to design energy-efficient CTS algorithm. Though the authors claim their method as energy-efficient, the energy consumption analysis is not highlighted.

The above discussion on recent literature survey on CTS algorithms for WSNs reveals that the energy consumption metric has not been considered in CTS algorithms. But, WSNs have energy constraint which must be taken into consideration for designing any algorithm or protocol for WSNs. This motivates us to adopt an energy consumption framework and analyze it for recent CTS algorithms.

III. SYSTEM MODELS

In this section, the models adopted in the paper are discussed.

A. Consensus Time Synchronization Model

In each iteration of the CTS algorithm, every node initiates the synchronization process by sending an initiation message. After receiving the time-stamped reply messages from its neighbors, it estimates the arrival time of its neighbor’s messages. Each node then updates its local clock time using pair-wise averaging method [7] or weighted averaging methods [8, 9] until all nodes converge to the aver-age of the initial clock differences between the nodes with some tolerable synchronization error. In the presence of both random and deterministic delays during message exchanges, the clock update rule at each node ‘i’ is given as [23]:

\[
C_i(t_k+1) = C_i(t_k) + \sum_{j \in N_i} C'_j(t_k) - C_j(t_k) \tag{1}
\]

where \(C_i(t_k)\) is the local time at node ‘i’ during iteration ‘k’ and ‘\(\varepsilon\)’ is the constant step size for each iteration. \(C'_j(t_k) = C_j(t_k) + T_{delay}\). The total delay \(T_{delay}\), ignoring system level delay factors, is given as:
\[ T_{\text{delay}} = T_{\text{MAC \ delay}} + T_{\text{PHY \ delay}} \]  

(2)

Where \( T_{\text{MAC \ delay}} \) is the MAC layer delay and \( T_{\text{PHY \ delay}} \) is the physical layer delay.

**B. Basic Energy Model**

To estimate energy consumption in a wireless communication, two mostly used radio models are: free space (fs) model and multi-path (mp) model [24]. Since, CTS algorithms follow one hop communication; the free space model is more suitable. Further, the major energy consumption to achieve synchronization is due to synchronization message transmission and reception. Therefore, we have considered these two factors for energy consumption estimation. Using free space model, the energy consumption \( P_{\text{tx}} \) for a message transmission and the energy consumption \( P_{\text{rx}} \) for a message reception is given as follows.

\[ P_{\text{tx}} = M(\beta_1 + \beta_2(l(i,j))^\zeta), \quad P_{\text{rx}} = M\gamma \]  

(3)

where ‘\( \zeta \)’ is the path loss exponent, typically set to 2 for free space model. The constants \( \beta_1, \beta_2 \) and \( \gamma \) are the energy dissipated by the transmitter module; transmit amplifier, and the receiver module respectively. The estimated distance between nodes ‘\( i \)’ and ‘\( j \)’ is denoted as \( l(i,j) \) and the length of message as ‘\( M \)’.

**C. Consensus Energy Model**

To compute the energy consumption for the CTS algorithms, the power model-1, proposed in [25], is closely followed which in turn follows the basic energy model discussed above. For the sake of simplicity, the model only considers energy consumption in message transmission \( P_{\text{tx}} \) and reception \( P_{\text{rx}} \), defined as above.

Using CTS framework, a node transmits and receives to and from each of its neighbors at every iteration. Assuming local broadcasting, the energy consumed by a node ‘\( i \)’ after ‘\( t \)’ iteration is given by equation (4)

\[ P(i) = M(\beta_1 + \beta_2 \max_{j \in N_i} l(i,j)^\zeta + \gamma | N_i |) \]  

(4)

Thus, the average nodal energy consumption for a network of ‘\( n \)’ nodes is given by equation (5):

\[ P_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} P(i) \]  

(5)

**IV. ENERGY CONSUMPTION ANALYSIS**

Using the above discussed models, the following section gives a detail energy consumption analysis of three recent CTS algorithms, namely, SATS [14], ATSP [7], and CCS [8]. Though, the authors in their papers have used different name for type of synchronization messages, for the sake of simplicity and uniformity, generic names for type of messages are used for all the algorithms in this paper.

(i) Test Algorithm-1: SATS

To achieve synchronization up to an acceptable synchronization error bound, the SATS algorithm executes at each node in the network by a certain number of iterations. In a particular iteration, the energy consumption at node ‘\( i \)’ is the sum of energy consumption for SYN_INIT message transmission, energy consumption for receiving SYN_ACK messages from neighbors and energy consumption for sending a SYN_AVG message to a selected node. Following the energy model given in section III, the following lemma gives an estimation of average energy consumption to achieve desired level of synchronization.

**LEMMA-1.** The average energy consumption to achieve synchronization in a network of ‘\( n \)’ nodes using SATS algorithm is:

\[ P_{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} I(i) * P(i) \]

where

\[ P(i) = M(\beta_1 + \beta_2 \max_{j \in N_i} l(i,j)^\zeta + l(i,k)^\zeta) \]

is the total energy consumption per iteration at node ‘\( i \)’ and \( I(i) \) is the number of iterations required at node ‘\( i \)’ to reach the acceptable synchronization error bound.

**Proof:** Using the energy model given in section III, the following equations can be derived.

For broadcasting a SYN_INIT message at node ‘\( i \)’, the energy consumption is given by:

\[ P_{\text{tx}} \text{ SYN\_INIT}(i) = M(\beta_1 + \beta_2 \max_{j \in N_i} l(i,j)^\zeta) \]  

(6)

For receiving SYN_ACK messages from neighbors at node ‘\( i \)’, the energy consumption is given by:

\[ P_{\text{rx}} \text{ SYN\_ACK}(i) = M\gamma | N_i | \]  

(7)

Similarly, for sending a SYN_AVG message by node ‘\( i \)’ to a selected node ‘\( k \)’, the energy consumption is given by:

\[ P_{\text{tx}} \text{ SYN\_AVG}(i) = M(\beta_1 + \beta_2 \max_{k \in N_i} l(i,k)^\zeta) \]  

(8)

Summing up the above equations, the total energy consumption per iteration at node ‘\( i \)’ is given by:
\[ P(i) = P_{\text{syn-init}}(i) + P_{\text{syn-ack}}(i) + P_{\text{syn-avg}}(i) \]
\[ = M[2\beta_1 + \beta_2 \max\{|l(i,j)|^c + l(i,k)|^c, j, k \in N_i| + \gamma | N_i|] \]
\[ \text{(9)} \]

Thus, the average energy consumption to achieve network wide synchronization for a network with ‘n’ nodes is given by:
\[ P_{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} P(i) * I_t(i) \]
\[ \text{(10)} \]

where \( I_t(i) \) is the number of iterations required at node ‘i’ to reach the acceptable synchronization error bound. This proves lemma 1.

(ii) Test Algorithm-2: ATSP

This algorithm follows pair-wise message passing paradigm to perform averaging. At each iteration, a node randomly selects a neighbor node and unicasts a SYN_INIT message. After receiving the SYN_ACK message from the selected node, it estimates the skew and offset and send to the selected node a SYN_AVG message. This process iterates until acceptable synchronization accuracy is achieved. Following the same principle as in energy analysis of SATS, the following lemma gives the energy consumption for ATSP.

**LEMMA-2.** The average energy consumption to achieve synchronization in a network of ‘n’ nodes is
\[ P_{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} P(i) * I_t(i) \]
\[ \text{(10)} \]

where \( I_t(i) \) is the number of iterations required at node ‘i’ to reach the acceptable synchronization error bound.

**Proof:** Using the energy model given in section III, the following equations can be derived for ATSP.

For unicasting a SYN_INIT message at node ‘i’ to a selected neighbor ‘k’, the energy consumption is given by:
\[ P_{\text{syn-init}}(i) = M(\beta_1 + \beta_2 \max\{|l(i,j)|^c, j \in N_i| + \gamma | N_i|) \]
\[ \text{(11)} \]

Similarly, for sending a SYN_AVG message by node ‘i’ to a selected node ‘k’, the energy consumption is given by:
\[ P_{\text{syn-avg}}(i) = M(\beta_1 + \beta_2 \max\{|l(i,k)|^c, k \in N_i| + \gamma | N_i|) \]
\[ \text{(13)} \]

Summing up the above equations, the total energy consumption per iteration at node ‘i’ is given by:
\[ P(i) = P_{\text{syn-init}}(i) + P_{\text{syn-ack}}(i) + P_{\text{syn-avg}}(i) \]
\[ = M[2\beta_1 + 2\beta_2 \max\{|l(i,j)|^c, j \in N_i| + \gamma | N_i|] \]
\[ \text{(14)} \]

Thus, the average energy consumption to achieve network wide synchronization for a network with ‘n’ nodes is given by:
\[ P_{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} P(i) * I_t(i) \]
\[ \text{(15)} \]

where \( I_t(i) \) is the number of iterations required at node ‘i’ to reach the acceptable synchronization error bound. This proves lemma 2.

(iii) Test Algorithm-3: CCS

Unlike SATS and ATSP algorithms which follow two-way message passing paradigm, CCS algorithm follows one-way message passing paradigm to perform cumulative weighted averaging. At each iteration, a node broadcasts a SYN_INIT message to its one-hop neighbors. Upon receiving this message, the neighboring node performs the weighted averaging. Following the same principle used for ATSP and SATS, the following lemma gives the energy consumption of CCS algorithm.

**LEMMA-3.** The average energy consumption to achieve synchronization in a network of ‘n’ nodes using CCS algorithm is
\[ P_{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} P(i) * I_t(i) \]
\[ \text{(15)} \]

For receiving SYN_ACK message from the selected neighbor ‘k’ at node ‘i’, the energy consumption is given by:
\[ P_{\text{syn-ack}}(i) = M\gamma \]
\[ \text{(12)} \]

Proof: Using the energy model given in section III, the following equations can be derived.
For broadcasting a SYN INIT message, the energy consumption at node ‘i’ is given by:

\[ P_{tx}^{\text{SYN \_INIT}}(i) = M[\beta_1 + \beta_2 \max \{l(i, j) \mid j \in N_i\}] \]

\[ (16) \]

For receiving SYN INIT messages from neighbors at node ‘i’, the energy consumption is given by:

\[ P_{rx}^{\text{SYN \_INIT}}(i) = M\gamma |N_i| \]

\[ (17) \]

Summing up the above equations, the total energy consumption per iteration at node ‘i’ is given by:

\[ P(i) = P_{tx}^{\text{SYN \_INIT}}(i) + P_{rx}^{\text{SYN \_INIT}}(i) = M[\beta_1 + \beta_2 \max \{l(i, j), j \in N_i\}] + \gamma |N_i| \]

\[ (18) \]

Thus, the average energy consumption to achieve network wide synchronization for a network with ‘n’ nodes is given by:

\[ P^{\text{avg}}(i) = \frac{1}{n} \sum_{i=1}^{n} P(i) \times I(i) \]

\[ (19) \]

where \( I(i) \) is the number of iterations required at node ‘i’ to reach the acceptable synchronization error bound. This proves lemma 3.

V. SIMULATION RESULTS & DISCUSSIONS

From the above analysis, it is observed that the average energy consumption of CTS algorithms depends on average number of iterations required for consensus convergence and average number of messages exchanged to achieve desired synchronization accuracy. To validate this, the simulations are performed using PROWLER simulator [26] on a random topology of 50 to 500 nodes in a square deployed area of side (L) equals to 10 units. To ensure a connected topology for consensus propagation, the connectivity radius (r) is calculated using the following formula [27].

\[ r = L \cdot \sqrt{2\log n / n} \]

\[ (20) \]

As per TelosB data sheet specification mentioned in [9], the typical skew range is between -5 PPM to 5 PPM. So, to have a close resemblance with the realistic environment, the skew is generated in the specified range using random uniform distribution. To have a fair comparison between SATS [14], ATSP [7] and CCS [8], the clock offsets are generated using random uniform distribution between 0 and 1 which is same as specified in the above algorithms [7, 8, 14]. The interval for one iteration, which denotes an upper bound for maximum oscillation period, is set to 10 seconds. The clock offsets are observed at a pause time of 10 sec, which is the interval of one iteration. The default MAC protocol provided in PROWLER simulator is CSMA/CA.

(i) Average Iterations for Convergence

The offset convergence is tested and observed for an acceptable synchronization error of 0.0001 sec. as shown in Figure 1 (a)-(c). The initial average of random offset distribution is recorded as 0.49. It is observed that SATS algorithm has faster convergence than ATSP and CCS algorithm. Hence, average iterations required for consensus convergence is less in SATS and it converges 16% faster than CCS and 50% faster than ATSP.

(ii) Average Number of Messages

To compare the average number of messages exchanged to achieve synchronization with the given error bound, the number of messages exchanged at each node is recorded and the average is computed for the whole network with different network size varying from 100-500 nodes. It is observed from Figure 2 that SATS algorithm has almost exchanged 50% less messages than ATSP and 10% less messages than CCS algorithm.

Mathematical analysis in [14] shows that for a network of ‘n’ nodes, SATS algorithm exchanges \( 4n - 2\) number of messages per iteration which is comparatively higher than ATSP algorithm which exchanges \( 3n \) number of messages per iteration and CCS algorithm which exchanges \( 3n - 2\) messages. But, due to faster convergence of SATS algorithm, the total number of messages exchanged is minimized.
(iii) Average Energy Consumption

The average energy consumption is estimated using the mathematical derivations obtained in section IV. The number of nodes are varied from 100-500. It is observed from Figure 3 that in an average, SATS algorithm has consumed 60% less energy than ATSP and 20% less energy than CCS algorithm. This is also, due to faster convergence and hence, minimization of total iterations in SATS algorithm which has major impact on energy consumption of consensus based synchronization algorithms.

VI. CONCLUSION

Recently, Consensus Time Synchronization (CTS) algorithms have gained much popularity among WSN research community due its computational lightness, robustness and distributed nature. Motivated by its inherent advantages and suitability for WSNs, a rich set of CTS algorithms are proposed in recent past. But, their energy consumption has so far not been explored which is a primary requirement for energy-constraint WSNs. In this paper, a thorough energy consumption analysis is carried out for recent, state-of-the-art CTS algorithms for WSNs. From the analysis, it is observed that the energy consumption is solely dependent on two metrics, average iterations for consensus convergence and average number of messages exchanged. Simulation results reveal that SATS algorithm is more energy-efficient than ATSP and CCS. This is due to its faster convergence than CCS and ATSP. Further, the energy consumption of CCS is also comparable with SATS because the average number of message exchanged is quite same as SATS due to its one-way message passing paradigm.

REFERENCES


