ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Trends and Development in Time Synchronization Methods in Wireless Adhoc Sensor Networks: A Review

Dr. Arun Saxena

Associate Professor

St. John College of Engineering and Management

Abstract:

Inconsistency in the clocks of nodes of a WSN may be due to several reasons. This may due to variations in their clock skew which depends on various parameters such as temperature, pressure etc. This may lead to difference in observed time or duration for various fundamental operations performed by these nodes. However, many applications and protocols require clock consistency. This implies essentiality of clock synchronization of nodes in the WSN according to some common reference time. The basic purpose of time synchronization is to ensure consistency in the view of time for all the nodes in the WSN. This paper presents an overview of popular time synchronization protocols and their classification criteria.

Keywods: WSN, TAA, Adhoc network, Mobile Network

Introduction

These networks exhibit the basic characteristics of ad-hoc networks including spontaneous formation, no infrastructure requirements and random deployment of sensor nodes. Several physical phenomenon and environmental conditions are studied so that they can be improved or their ill effects can be mitigated. This requires continuous monitoring and reporting of observations. For many environments this is beyond the reach of human capabilities. WSNs are used to perform these activities with the help of sensor nodes. Consequently, a physical activity in the area in vicinity of WSN can be observed and predicted.

Time synchronization is a critical for the operation of any distributed system. However it is more difficult to achieve time synchronization in WSN as compared to traditional distributed systems. The computing devices in the former case are small and may remain unattended for a long time. The communication channel is also wireless, unreliable and has a short range and bandwidth. Time synchronization is required to schedule the fundamental operations performed by sensor nodes and increase the precision of detections made by them. Moreover, time synchronization among node also increases the network lifetime.

This paper presents a review of various approaches and techniques that can be used to address the problems related to time synchronization. This paper aims at facilitating researchers in identifying a suitable solution pertaining to problems in time synchronization of WSN and IoT domains. Subsequent section in this paper identifies the criteria for classification of time synchronization protocols and approaches followed by an explanation of widely used time synchronization protocols along with their advantages and limitations.

2. Time Synchronization Criteria

There are various criteria that can be used to classify time synchronization protocols [1] [2] [3].

Physical time vs Logical Time – Logical time deals with internal consistency between clocks and not with the actual physical time. This implies that the clocks must have same common time. This common time may or may be close to the actual physical time. However physical time is the time in which the clocks of all nodes have the same time and variation of this time from the physical time is bounded by certain limits.

Global vs Local Time Synchronization Algorithms – The algorithms in the former category aim at achieving network wide synchronization whereas local synchronization algorithms synchronize the nodes in a close vicinity only depending on the application requirements.

Absolute vs Relative – Two factors are responsible for the time difference of sensor nodes in a network. First is their sporadic switching time and second is the randomness in the number of times they are switched on. This leads to a difference in their time which may be called as phase difference. Secondly, their frequency of oscillation may vary from their predefined frequencies of oscillation due to several factors. This is known as clock drift or clock skew. Absolute time synchronization schemes correct the time of sensor nodes on the basis of both clock drift and phase while relative time synchronization schemes use only clock drift as the correction factor.

Hardware vs Software-Based Mechanisms – Hardware based mechanisms require special hardware like GPS to synchronize sensor nodes. However, they are not popular in WSNs because they are costly, have heavy weight form factors making them unsuitable to be used in sensor nodes and consume more energy. Software based mechanisms rely mainly on message passing to synchronize nodes in the network. **A-Priori vs A-Posteriori Synchronization –** A-posteriori synchronization is also known as post-facto synchronization. It starts only when an event occurs. However, a-priori synchronization protocols are always running and are continuously synchronizing the nodes in the network.

Deterministic vs Stochastic Precision Bounds – Deterministic algorithms define an upper bound on clock offset. Moreover, this upper bound is defined with some certainty.

Some prominent examples include: Sichitiu and Veerarittiphan's protocol [4], RBS [5], timediffusion protocol [6]. Stochastic or probabilistic methods ensure maximum clock offset. However, this clock offset has some failure probability which is smaller than some fixed bound. Master-Slave vs Peer-To-Peer Synchronization- In master-slave approach sensor nodes are designated as master and slaves. The time of master node is the reference time for all other nodes. All slaves synchronize themselves with the reference time of master node. Although this technique is simple and non-redundant but it requires that the master node should have a powerful processor to cater the synchronization needs of number of slaves assigned to it. Mock et al. [7] and Ping's protocol [9] have adopted this technique for synchronizing nodes in WSN. In peer-to-peer synchronization distinguished designations to the nodes are not given. They communicate with each other to synchronize themselves. It is flexible and robust as the synchronization process does not depend on a single node. However, it is complicated and is difficult to control. Various popular protocols that have adopted this technique include: RBS [5], time synchronization in adhoc networks [10], probabilistic clock synchronization service in sensor networks [8], TDP [6], and ADP [11].

External vs Internal Synchronization – In external synchronization, an external time source is used to synchronize sensor nodes. This time source can be anything such as UTC and is known as reference time. This type of synchronization requires that at least one node in WSN must have access to external time. The accuracy of external synchronization can be increased by using atomic clocks as external sources of time. On the contrary, in internal synchronization nodes in the network do not synchronize themselves according to an external time scale. The nodes simply agree with a common time which is not compliant with external time. NTP can be used in both the methods of time synchronization. NTP is widely used in distributed systems 16 like Internet which makes use of external synchronization. However, energy constraints associated with sensor nodes make external synchronization and NTP unsuitable for WSNs.

Sender-Receiver vs Receiver-Receiver Synchronization – Message flow can also be used to categorize time synchronization algorithms. In sender-receiver approach only one node designated as reference or root initiates and controls the synchronization process [4],[9],[13- 19]. In contrast to this, in receiver- receiver approach a set of receivers synchronize themselves by exchanging message[5-7]. Synchronization protocols can also be classified on the basis of synchronization issues and application-dependent features [20]. The synchronization issue used to classify various protocols include: Master-Slave versus Peer-Peer Synchronization, Clock Correction versus Untethered Clock, External versus Internal Synchronization, Deterministic versus Stochastic Precision Bounds and Sender-Receiver versus Receiver-Receiver Synchronization. Clock Correction vs Untethered Clock – Clock correction synchronization approach changes the clock of each node with reference to a global time scale or an atomic clock whereas in untethered clock synchronization the clocks of nodes are not modified. Latter methods maintain a time-translation table using which a node correlates its local time with other nodes in the network [20]. The clock synchronization protocol for continuous clock synchronization [7] and DMTS [9] are based on clock correction approach whereas RBS [5] and Romer's protocol [10] use untethered clock technique. A taxonomy classification of various popular protocols on the basis of synchronization issues is given in table 1 [20]:

Table 2.1: A Taxonomy Based on Synchronization Issues

Algorithms	Synchronization Issues			
	Master-Slave vs Peer toPeer	Internal vs External	Sender-to receiver vs Receiver-to receiver	Clock correction
RBS 2002	Peer to Peer	Both	Receiver-to Receiver	No
TPSN 2003	Master - Slave	Both	Sender-to Receiver	Yes
FTSP 2004	Master - Slave	Both	Sender-to Receiver	Yes
Miny-Sync 2003	Peer to Peer	Internal	Sender-to Receiver	Yes
LTS 2003	Master - Slave	Both	Sender-to Receiver	Yes
DMTS 2003	Master - Slave	Both	Sender-to Receiver	No
TDP 2005	Peer to Peer	Internal	Receiver-to Receiver	Yes
SLTP 2007	Master - Slave	Both	Sender-to Receiver	Yes
GTSP 2009	Master - Slave	Both	Sender-to Receiver	Yes
TSRT 2011	Master - Slave	Both	Sender-to Receiver	No

On the basis of application dependent features, the classification of these protocols is as follows:

Single-Hop vs Multi-Hop Networks – In former type of topology the nodes in WSN can directly interchange information with each. Single-Hop protocols are clock synchronization protocol for continuous clock synchronization [7], probabilistic clock synchronization service in sensor networks [8], TPSN [21, 22] and DMTS [9]. However, due to large number of nodes in WSN applications, multi-hop networks are used. Multiple domains exist in these type of networks. Nodes in the each domain may interact directly with each other. However, nodes in different domains communicate with each other through an intermediary node. Protocols 18 designed for single hop networks can also be extended to multi-hop networks. Protocols suitable for multi-hop networks include: RBS [5], DMTS [9], probabilistic clock synchronization service in sensor networks [8], and TDP [6].

Stationary Networks vs Mobile Networks – In the former type, sensor nodes do not change their position after their deployment. While in the latter, the sensor nodes can move and connect with the sensors in the new domain after they have changed their position. Time synchronization of nodes in a mobile network is difficult as compared to synchronization of nodes in stationary networks. RBS [5], clock synchronization protocol for continuous clock synchronization [7], probabilistic clock synchronization service in sensor networks [8] and TPSN [21, 22] are designed for stationary networks.

MAC-Layer vs Standard Approach – MAC- layer based protocols enhance energy efficiency by utilizing MAC layer. Several versions of MAC protocols have been proposed to increase energy efficiency [20], [14].

Table 2 presents a summary of protocols based on application dependent features:

Table 2: Taxonomy on the Basis of Application Dependent Features

Algorithms	Application Dependent Features		
	Single-Hop/ Multi-Hop Networks	MAC-Layer Based Approach/ Standard Approach	Stationary Networks/ Mobile Networks
RBS 2002	Both	Standard	No
TPSN 2003	Both	MAC Layer	Yes
FTSP 2004	Both	MAC Layer	Yes
Miny-Sync 2003	Both	MAC Layer	No
LTS 2003	Both	MAC Layer	Yes
DMTS 2003	Both	Standard	No
TDP 2005	Both	Standard	Yes
SLTP 2007	Both	Standard	Yes
GTSP 2009	Both	MAC Layer	No
TSRT 2011	Both	MAC Layer	Yes

3. Time Synchronization Protocols and Approaches

Several time synchronization schemes for WSNs are available in the literature. Each of these addresses various challenges pertaining to sensor nodes. However, any single time synchronization technique cannot meet all the challenges of the time synchronization domain. Each approach has its advantages and limitations. The adaptability of the time synchronization method in a WSN application depends on the requirements of the application itself.

3.1 Network Time Protocol

It used for clock synchronization in computer networks [24]. It was designed to be used in packet switched networks having client server architecture. However, it can also be used in peer to peer networks. In this protocol all the client machines utilize a time server in order to synchronize their time and time servers utilize an external source such GPS to synchronize themselves. NTP has been very popular and widely used in distributed systems especially Internet. However, transmission time in WSN is unpredictable due to MAC layer. This leads to poor estimation of message delay in WSNs. Thus, NTP is not suitable for many WSN applications.

An energy efficient approach for synchronization of nodes in the network is presented in PostFacto synchronization [25]. In this approach the local time of various sensor nodes in WSN remains unsynchronized. The clock synchronization is starts when an event occurs. Whenever an event occurs the nodes record the time of its occurrence as per their local time. This is followed by the broadcasts of a synchronization pulse by a beacon node. This pulse is received by various nodes which are in the radio range of the beacon node receive and they utilize it synchronizing their clocks. The drawback of this approach is that it is not suitable for applications long distance transmission of time stamps is required. However it is suitable for applications where the scope of time synchronization is limited to the range of the beacon node such beam forming applications. The major sources of error in this scheme include: clock skew at receiver side, variable delays at multiple receiver ends and the propagation delay of the synchronization pulse.

3.2 Reference Broadcast Synchronization Protocol

It is one of the popular time synchronization scheme which has adapted receiver-receiver synchronization. It is also called as Reference Broadcast Synchronization (RBS) [5]. In this protocol, a set of receivers synchronize among themselves. In the simplest form of RBS, one transmitter sends a packet to two receivers. These two receivers exchange their information and form a local timescale. This basic mechanism has two limitations. Firstly, WSN application deploy more than two nodes and secondly it does not consider the clock skew in time synchronization. In another variant of this algorithm a transmitter transmits m reference packets to n receivers. These receivers then exchange their observations. Each receiver computes phase offset to other receivers on the basis of received observation. Finally, each receiver computes the average of these phase offsets in order to find the time difference. The accuracy of time difference can be further enhanced by considering the clock skew. In order to find clock skew, phase error is first computed using least-squares linear regression. The advantage of this approach is that it gives a best fit line is obtained using the phase error observations over time. This line can be used to determine the frequency and phase of the local node's clock and remote node's clock.

RBS can be extended for multi-hop networks. This is done by assuming that nodes in the network are grouped as clusters. Linear regression can them be applied to synchronize node that are not immediate neighbours.

RBS removes sender side nondeterministic latency which is due to send time and access time. Moreover, it removes also reduces receive time delays. The primary reason for this is that the clock can be read at the interrupt time. Another reason for this reduction is that RBS considers only the differences in receive time for the two nodes. The energy consumption of RBS is high as large amount of time related information has to be exchanged amongst nodes in its synchronization process. It accuracy is low and convergence is high time in multi-hop networks [15], [26]. Moreover, it requires a physical broadcast channel and is not suitable for point to point networks. Further, the accuracy of RBS decrease for multi-hop networks.

3.3 Lightweight Time Synchronization Protocol

It utilizes pairwise synchronization in order to synchronize the clocks of nodes [15]. It assumes that a single reference node has access to global time reference and the accuracy of this time reference is very high. In pairwise synchronization nodes interchange their timestamps in order to calculate the clock difference and subsequently synchronize themselves. Two different approaches are adapted by LTS namely, centralized and distributed. In centralized approach a spanning tree is created and a reference node sends periodic updates. In this approach the nodes in the network are periodically synchronized after a calculated resynchronization period. The message overhead is fixed to 3 messages per node. The complexity of the tree construction depends on the type for algorithm used for it. In distributed the entire tree is not synchronized after a fixed resynchronization period. In this case, the nodes determines the period after which it has to synchronize itself on the basis the required level of accuracy, their depth in the tree, their clock drift and a record of time elapsed from its last synchronization time. The synchronization process is started by the node itself instead of reference node. This approach increases the energy efficiency of the protocol as all nodes do not always need to resynchronize themselves. LTS consumes less energy but its major drawback is that its accuracy depends on the depth of the tree.

3.4 Tiny/Mini-Sync Protocol Tiny/Mini-Sync

It is another protocol that uses pairwise synchronization using which nodes can make clock corrections [4]. The basis for this protocol is the assumption that both clock drift and clock offset of nodes are linearly related. For a fixed range of these values the clocks are perfectly synchronized otherwise a best fit offset line is estimated using the data points collected from message exchanges between nodes. By applying tight and deterministic bounds on these data points in the best offset line, an accurate estimation clock offset and drift is obtained. The nodes use this information to correct their clocks. This required measurements 22 are repeated several times and several data points are obtained. Thus, estimation of best fit offset line becomes more complex. Consequently a linear constraint is used to reduce this complexity by eliminating redundant packets.

- Both the protocols require very less resources and have following common features: Both algorithms calculate clock offset and drift. They also apply tight bounds on the precision of their estimations.

- Both use deterministic bounds unlike probabilistic bounds used by other algorithms. The complexity of both the algorithms is less in terms of computation and storage.

- Both are immune to errors in communication.

Tiny/Mini-Sync are simple, energy efficient and tolerant to message losses. However, they have several disadvantages. The scalability of the protocol is not discussed. They have high convergence time and are suitable for WSNs where mobile nodes are not mobile.

3.5 Timing-Sync Protocol for Sensor Networks

It is a popular protocol based on sender-receiver approach [13]. TPSN follows "always-on" model in which time of all the nodes is synchronized with the time of a reference node in the network. However, TPSN can also follow other synchronization models. TPSNs system model assigns a unique sensor id to each sensor in the network, uses a 16bit register as a clock and builds a hierarchical topology for the network. In this model hierarchical topology only one node called root node. This node is located at level 0 and rest of the nodes are at higher levels. Moreover, nodes at a particular level can communicate with nodes located at a level one higher then them.

Initially the hierarchical topology is constructed. This phase is also known as level discovery phase. In the subsequent step (called synchronization phase), synchronization of the nodes is done.

First phase starts immediately after deployment of WSN. The construction of hierarchical structure is initiated by root node with the broadcast of a level_discovery packet. The neighbouring nodes assign themselves a level upon receiving this packet as the packet contains the identity and level of the sending node. The level assigned by each receiving node is one greater than the sender's level. After each receiving node has assigned itself a level, they broadcast the same level_discovery packet so that a level can be assigned to other nodes. This process is repeated by all root nodes at various levels of the structure. If a level is already assigned to a node then it discards these level_discovery packets.

In phase two TPSN uses pairwise synchronization to find the clock drift between nodes in the network. It involves bidirectional message exchanges between a pair of nodes.

The advantages of TPSN are its simplicity, scalability along with its consistent accuracy irrespective of the increase in network size. It removes all possibilities of error except reception time and receive time uncertainties. However, its energy efficiency is low as it involves physical clock correction on clocks of sensor nodes. It is not suitable for applications where the nodes are mobile. In TPSN the entire synchronization process depends on the root node(s). If the root node gets disconnected due to less power or due to harsh conditions in which the WSN is deployed, all the nodes in the network remain unsynchronized.

3.6 Delay Measurement Time Synchronization Protocol

Delay Measurement Time Synchronization Protocol (DMTS) follows a master slave approach for time synchronization [9]. It chooses a leader as master node for time synchronization process. Leader broadcasts its local time. All the receiving nodes calculate the time difference on the basis of broadcasted time and their local clock and adjust their time by adding this time difference to the time broadcasted by the leader node.

Time delay calculation of DMTS is based on two timestamps. First timestamp is the time when a receiver synchronizes itself with the leader and second timestamp is the time when it has processed the message. Thus, it measures data transfer time and receiver side processing delay. DMTS can be applied to a network having multiple hops by using the same process for the next level of nodes. This implies that all the nodes are arranged in a hierarchy and a leader has to be chosen for all levels in the hierarchy.

DMTS is energy efficient as it does not use pairwise synchronization. It is flexible, lightweight and in the absence of transmit time estimation error, receivers can synchronize amongst themselves in a better way as compared to their synchronization with the sender. However, it can applied to clocks having lower resolution and frequency. It is less accurate and susceptible to synchronization failures due to failure of master or leader node.

3.7 Flooding Time Synchronization Protocol

Flooding Time Synchronization Protocol (FTSP) can synchronize multiple receivers with the time of a sender using a single message. This message is time stamped at both sender and receiver end. It works on the assumption that the local clocks of sensor nodes in WSN has the timing errors due to crystals. It also assumes that nodes can communicate such errors over an unreliable wireless medium. It improves the accuracy of time synchronization by reducing the 24 delay due to interrupt handling, encoding/decoding time and byte alignment of the radio chips at the sender and receiver ends. The effect of these delays in calculating the clock offset were not considered by earlier protocols. However, in the experimental setup of TPSN delays due to byte alignment were identified but not reported in the sources of errors. Moreover, FTSP uses linear regression in order to improve clock drift calculations.

In the synchronization approach adapted by FTSP, the sender node broadcasts a message having its timestamp. This timestamp is known as global timestamp. The receiving nodes record the local time upon receiving this message. The difference between global and local timestamp is used to calculate clock offset. The packet format for the data packets transmitted and received by sender include: preamble byte, synchronization byte, data bits and the CRC bits. The preamble bytes are used by the receiver to synchronize its frequency with carrier frequency while the synchronization byte is used by it to calculate the bit offset. Bit offset facilitates the receiver in byte alignment using which the message can be reassembled correctly. The key idea using which FTSP reduces delays due to interrupt handling, encoding/decoding time and byte alignment of the radio chips is recording the timestamps at byte boundaries as they are transmitted or received. These byte boundaries start after the end of synchronization byte. Each of these timestamps is first normalized and then their average is taken to find the accurate timestamp. This reduces the errors that are caused due to interrupt handling and encoding/decoding time. This error is further corrected at receiver end by calculating byte alignment time. Byte alignment time can be calculated using transmission speed and bit offset. Although multiple timestamps are taken in FTSP, only the corrected time stamp is included in the message.

FTSP protocol can also be used in multi-hop networks. In these networks a reference points are used for synchronizing nodes in the network. These reference points are generated by a designated node called synchronized-root. This reference is elected by several nodes in the same radio range. It is also called synchronized-root and all other nodes in the network are synchronized according to this node. Synchronized-root is either elected or reelected by a group of nodes in the same radio range. Nodes which are not in the radio range of this synchronizedroot may gather reference points from other synchronized nodes. When a node collects several reference points it can correct its clock by calculating clock offset and skew.

FTSP is more accurate and consumes less resources as compared to RBS and TPSN. Moreover, it supports node mobility and its synchronization process is not affected even if the synchronized-root node fails. However, it cannot be applied to larger networks and is not 25 energy efficient as lot of energy is consumed in sending and receiving messages due to the complexities of creating timestamps.

3.8 Time Diffusion Protocol

Time Diffusion Protocol (TDP) follows two schedules for synchronization namely, active schedule and inactive schedule [6]. Moreover, it uses several algorithms for synchronizing nodes in the network. By using these two schedules and algorithms it synchronizes the local time of nodes in WSN so that it reaches an equilibrium time.

An active schedule consists of multiple cycles. The time period of an active cycle is τ seconds. Each cycle starts with the broadcast of reference time by sink nodes which act as precise time servers to the master nodes in the network. Equilibrium time is the time broadcasted by the sink and master nodes are the elected nodes which synchronize their neighbours. The master nodes broadcast this reference time to all their neighbours. When a neighbour receives this message referred as initialize pulse, they determine whether they will act as master node or not. This is done using Election/Reelection of Master/Diffused Leader Node Procedure (ERP). ERP in turn uses False Ticker Isolation Algorithm (FIA) and Load Distribution Algorithm (LDA) to elect/reelect master node. Once master nodes are elected, each master node uses Peer Evaluation Process (PEP) to remove false tickers from becoming a master node or diffused node. During PEP elected mater nodes send SCAN message to the neighbour nodes, neighbour nodes calculate Allan variance using these SCAN messages and send back the calculated Allan variance with the REPLY message, master nodes calculate outlier from REPLY messages and Allan variance. Finally, master nodes calculate the outlier ratio and average Allan variance and reply back to the neighbours using REPLY message. Each node receiving this information uses ERP and synchronize their clocks using another set of algorithms, namely Time Adjustment Algorithm (TAA) and Clock Discipline Algorithm (CDA). This process is repeated for n hops and starts from the master nodes. Even after the local clock time of node reaches equilibrium, it may vary due to clock drift. Time Adjustment Algorithm (TAA) is used eliminate such drifts.

TDP uses a tolerance level to maintain time of nodes. The level of tolerance can be varied according to the requirements of the application. It does not use fixed or hierarchical network structure. This provides flexibility and support for node mobility. Load of synchronization is distributed on multiple master nodes instead of a single one. The time of all the nodes in the network is compatible with an equilibrium time. TDP is autonomous in nature so it does not require an external time source such as UPTP to synchronize the nodes.

This is useful WSNs where sink node cannot be synchronized with an external source due to line-of-sight related issues or other connectivity problems due to the nature of area where they are deployed. There are multiple cycles and multiple rounds are involved in each cycle. Consequently the synchronization process is complex and the energy efficiency of TDP is reduced. The convergence time for TDP is more in the absence of external time sources.

3.9 Asynchronous Diffusion Protocol

In this protocol nodes flood their neighbours with packets containing their local clock values. When the nodes receives information about all its neighbours they adjust their clocks on the basis of a mutually agreed consensus value [11]. Authors have suggested that the consensus value can be clock reading such as highest or lowest clock reading in the network or some clock value based on statistical calculations. The protocol achieves network wide synchronization without using an external time server. However, it utilizes the approach of RBS to interchange information which makes it complex and less energy efficient. Moreover, clock skew is not utilized in synchronization of nodes.

3.10 Scalable Lightweight Time Synchronization Protocol

Scalable Lightweight Time Synchronization Protocol (SLTP) works in two phases, namely: configuration and synchronization [18]. Configuration phase aims at selecting the cluster heads. Cluster head selection process can be static and dynamic. In static cluster head selection process the node which is most keen to be designated as cluster head is selected whereas in case of dynamic mode cluster head is selected during synchronization phase.

Synchronization process is also carried out in static and dynamic mode. In static mode each cluster head broadcasts packet containing its local timestamp at specific intervals. The member nodes record the receiving time of each of this packet. Then they calculate their clock offset and skew using linear regression.

In dynamic mode firstly the configuration phase is executed in order to identify cluster head and gateway nodes. Subsequently synchronization phase is executed to synchronize various nodes in WSN. The process of synchronization is similar for both static and dynamic nodes. SLTP can be used in both static and dynamic networks. However, its energy consumption is high and does not provide any provisions for message losses due to which the synchronization of nodes in the entire network can be affected.

3.11 Gradient Time Synchronization Protocol

Gradient Time Synchronization Protocol (GTSP) aims at improving time synchronization of neighbouring nodes. Earlier prominent protocols like RBS, TPSN, FTSP etc. focus on global clock skew minimization [17]. However many applications and situations require local clock skew minimization. Global clock skew minimization reduce the clock skew between nodes which are far-away, regardless of their distance with each other whereas local clock skew minimization focuses on precision of time synchronization between neighbouring nodes and assumes that larger pair wise error between far-away nodes is tolerable.

Symmetric error synchronization can be achieved by using a simple moving average filter if various sources of errors are removed [27]. This is the key idea behind GTSP. GTSP characterizes every hop from the reference node with some random error δ. This random error follows a square-root function on each hop, hence the resultant error expected for a chain of k nodes from the reference to the destination node is of the order of $\delta\sqrt{k}$. Thus, the error between two nodes which are at the same level in the hierarchy but located in different subtrees would experience an error in the order of the square-root of their distance in the tree.

In order to synchronize the time of two nodes their relative clock offset is to be compensated. In GTSP each node I broadcasts a synchronization beacon containing its logical time and relative logical clock rate. Any single receiver receives beacons from multiple senders during a synchronization period. Using this information, receiving node updates its absolute logical clock rate.

$$x_i(t) = h_i(t).l_i(t) \quad (1)$$

$$x_i(t_k+1) = (\sum j \epsilon N_i \; x_j \, (t_k)) + xi(tk) \, / \, | \, N_i \, | + 1 \quad (2)$$

Finally, node updates its relative logical clock rate using equations (1) and (2) and corrects its logical clock.

GTSP is fully distributed and has high accuracy. It supports network dynamics as it does require any topology and is robust against node failures. However, it aims at synchronizing nodes in a close vicinity only and ignores global clock skew. It is very complex and consumes a lot of energy as the number of messages exchanged is too high.

3.12 Gradient Clock Synchronization Using Reference Broadcast Synchronization

This algorithm suggests an approach using which effective network diameter can be reduced in order to achieve better time synchronization [28]. The protocol utilizes an estimate layer and another layer that contains the algorithm used for time synchronization.

The estimate layer is used to collect the logical clock values of various nodes in WSN which are in the radio range of each other. This information is represented with the help of an estimate graph. The vertices represent the nodes and the edges represent a mechanism using which two nodes can compute the logical clock values of each other. Any method such as RBS can be used for collecting information related to logical clock values of nodes. Gradient Clock Synchronization Using Reference Broadcast Synchronization uses two methods for collecting information namely, Direct Estimates and RBS estimates. Direct estimate method assumes that logical clock values would be broadcasted by each node at a fixed time interval whereas RBS estimates are obtained by comparing the logical clock values amongst neighbours on the occurrence of an event.

Once the estimate layer is created, any clock synchronization method can be used for synchronizing the clocks of nodes in the network. The synchronization algorithm should be capable of handling links having non-uniform uncertainty. The proposed protocol increases the logical clocks of each node in a continuous manner in several iterations. Moreover, mode of operation of the algorithm in each iteration for each node can be either fast or slow. At each iteration of the algorithm each node compares its clock values with its neighbouring nodes in the graph. If a node finds that the time lag between its logical clock high as compared to its neighbours then it executes the algorithm in fast mode and vica versa.

The approach has all the advantages of RBS. Moreover, it creates an estimate graph instead of physical graph of the network. Consequently, this approach can utilized any scheme for time synchronization. The energy consumption and complexity of RBS is high because larger number of messages are interchanged in its synchronization process. The proposed protocol also uses reference broadcasts so its energy consumption and complexity would be very high.

3.13 A Reliable Time Synchronization Protocol for Wireless Sensor Network

The protocol that synchronization of nodes remains unaffected even in the case of node failures. This is done by employing several nodes as time source. Time source nodes are responsible for initiating the synchronization process for every synchronization period. AT the beginning of 30 every synchronization period, one of the time source nodes is randomly selected for synchronizing other nodes in the network. The protocol associates a link quality with every potential time source. Link quality has several parameters. If a selected potential time source is successful in synchronizing the nodes in the network, link quality parameters associated with it are increased otherwise they are decreased. If the value of these parameters for a potential node falls below a certain limit then it is removed from the list of potential time sources. If in a synchronization period all the time sources fail to synchronize the network then the resynchronization process is followed. The protocol uses linear regression and an error prediction approach to improve the accuracy of time synchronization.

The protocol has achieved higher precision in time synchronization and supports network dynamics. Moreover, its accuracy is compared only with FTSP. However the energy efficiency and the convergence time of the protocol is not discussed.

3.14 A Reliable and Efficient Time Synchronization Protocol for Heterogeneous Wireless Sensor Network

This protocol works for cluster based WSNs and the time synchronization process involves cluster head and cluster members [30]. However, nodes in different clusters can also synchronize each other. The protocol operation involves a configuration phase followed by synchronization phase [23].

The configuration phase involves the selection of a cluster head. The protocol uses Distributed Clustering Algorithm (DCA) which utilizes weight-based algorithm. This approach selects cluster head using several parameters which include: remaining energy, degree, dynamicity, and average distance to neighbours.

The synchronization process adapted by this protocol involves two way message exchanges. Initially the cluster head broadcasts a message containing its identity number and local timeime at t tcs, each cluster member receives this message at time tmr and responds to this message by sending an acknowledgement at time tms. Finally cluster head receives this message from at time tcs. At the end of each broadcast the cluster head estimates the minimum delay using the method suggested in tiny/mini-synch [4]. After sending m broadcasts packets cluster head derives an equation of the form

$$Y = aX + b \quad (3)$$

where a denotes the clock drift and b denotes the clock offset, calculated using m broadcasts. Moreover, values of a and b are different for different cluster members. After m broadcasts cluster head sends the values of a and b to each cluster member. Any two nodes which have 31 to synchronize with each other can exchange their values corresponding to a and b, calculate clock drift and offset using these values and find out their local time with respect to each other. The protocol is scalable adverse environments and has better precision. However, the energy consumption of the protocol is high due to larger number of message interactions involved in the synchronization phase.

3.15 Wireless Deterministic Clock Synchronization (WIDECS) Protocol

In this protocol the network is organized as clusters where nodes in each cluster follows star topology [35]. Each of these clusters are organized as a stack called piconet. It basically propagates the time of master node to all the slave nodes. It uses TDMA MAC layer so that time is divided into slots and each slot is assigned to the nodes in the same level of the tree. It has high convergence time and good synchronization accuracy but the protocol does not address energy consumption and clock drift correction issues.

3.16 Reference Based, Tree Structured Time Synchronization Approach

The working of this protocol can be divided into two steps [19]. Initially a hierarchical tree structure is constructed. A root node at level 0 starts this process by flooding and root nodes at other levels repeat the same. This is followed by synchronization phase. In order to synchronize the network. The root node sends a synchronization pulse to its child nodes. In response to this message an elected leader node replies back to root. The clock drift and propagation delay is calculated by root node on the basis of the reply it receives from the leader before the broadcast. All the non-root nodes fix their clocks using these computations and the process gets repeated for the various tree levels. TSRT utilizes the concepts of TPSN but aims at improving energy efficiency by reducing the complexity of synchronization process adapted by TPSN.

It has been proposed that the accuracy of time synchronization can be improved by eliminating uncertainties due to channel contention [31]. In the said approach, a topology having multiple hops is constructed in a similar manner as adapted by TPSN. However, extra traffic due to duplicate messages, collisions and channel contention is eliminated as the approach uses a different packet delivery process. This is accomplished since each node maintains a forward list for packet delivery. This list is created and updated on the basis of list received by a node which has already sent a packet to it. Consequently, selective forwarding of packets is done instead of flooding which in turn mitigates the adverse effects of flooding on time synchronization of nodes.

The authors have addressed the limitations of this protocols in their subsequent protocols [36,37].

3.17 Two-Hop Time Synchronization Protocol for Sensor Networks

It aims at finding the synchronization pairs of nodes using which all the nodes can be synchronized [32]. This approach is different from traditional approaches as their synchronization process involves synchronization of neighbouring nodes. These neighbouring nodes synchronize their neighbours and this process continues till all the nodes in the network are synchronized. This requires pairwise message exchange among all nodes. In two-hop synchronization protocol a limited number of synchronization pairs can synchronize all the nodes in the network. This involves lesser number of message interchanges and is more energy efficient as compared to traditional protocols. It also reduces the number of hops which contributes to reduction in multi-hop synchronization error.

Due to less communication overhead and reduction in number of hops, the convergence time also decreases. However, the protocol does not discuss the effects of changes in network topology on its operations.

Another approach that is adapted to enhance synchronization accuracy and network lifetime is by varying the packet rate in only that portion of the network where events are detected [33]. This protocol synchronizes networks with selective convergence rate and is suitable for applications which follow event-driven measurements. The protocol is motivated by the preliminary idea suggested in [34]. In this approach the convergence rate in the location where the event is identified increased as compared to the rest of the network. Thus, the nodes in the entire network are divided into two subsets: Improved Synchronized Subset (ISS) and Default Synchronized Subset (DSS). Latter is the area where the event is detected and is characterized with quick convergence whereas former is in quiet state and has low convergence rate so that energy can be saved in order to increase network lifetime. The accuracy of ISS may reduce due to its communication with DSS. In order to remove this incompatibility the authors have described communication policy and ISS connector algorithm. The protocol is energy efficient and improves the synchronization accuracy. However, with the increased packet rate, the number of collisions may increase, affecting adversely both the energy efficiency and the synchronization accuracy

References

[1] Antoni Grilo Chapter 8: Time Synchronization [Online]. Available:http://comp.ist.utl.pt/ecewsn/doc/slides/sensys-ch8-Time

[2] H. Karl, A. Willig, Protocols and Architectures For Wireless Sensor Networks, John Wiley & Sons Ltd., 2005.

[3] Mohamed Ibnkahla, Adaptive Signal Processing in Wireless Communication, Inc. Boca Raton, FL, USA: CRC Press, 2008

[4] M. L. Sichitiu and C. Veerarittiphan, "Simple, Accurate Time Synchronization for Wireless Sensor Networks," in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003), 2003, pp. 1266–1273.

[5] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," in Proceedings Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), 2002, vol 36, pp. 147–163.

[6] W. Su, I. Akyildiz, "Time-Diffusion Synchronization Protocols for Sensor Networks," IEEE/ACM Transactions on Networking, pp. 1626-1645, 2005.

[7] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous Clock Synchronization in Wireless Real-time Applications," in Proceedings 19th IEEE Symposium on Reliable Distributed Systems (SRDS-00), Oct. 2000, pp. 125–133. [8] S. Pal Chaudhuri, A. Saha, and D. B. Johnson, "Probabilistic Clock Synchronization Service in Sensor Networks," Department of Computer Science, Rice University, Houston, Technical Report TR 03-418, 2003.

[9] S. Ping, "Delay Measurement Time Synchronization for Wireless Sensor Networks," Intel Research, Berkley, IRB-TR-03- 013, June 2003.

[10] K. Romer, "Time Synchronization in Ad Hoc Networks," in Proceedings ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)," Long Beach, CA, U.S.A, October 2001, pp. 173–182.

[11] Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," in Proceedings IEEE Conference Computer Communications (INFOCOM 2004), vol. 1, pp. 564–574, Hong Kong,

[13] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-Sync Protocol for Sensor Networks," in Proceedings First International Conference on Embedded Networked Sensor Systems, Los Angeles, California, U.S.A, November 2003, pp. 138-149.

[14] C. E. Jones, K. M. Sivalingam, P. Agrawal. and J. Cheng, "A Survey of Energy Efficient Protocols for Wireless Networks," Wireless Networks, vol. 7, issue 4, pp. 343–358, 2001.

[15] V. Greunn and J. Rabaey, "Lightweight Time Synchronization for Sensor Networks," in Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA'03), 2003, pp. 11-19.

[16] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The Flooding Time Synchronization Protocol," in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04), Baltimore, Maryland, 2004, pp. 39-49.

[17] P. Sommer and R. Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," in 8th ACM/IEEE International Conference on Information Processing in Sensor Networks, April 13-16, 2009, San Fransico, CA, USA, 2009, pp. 37-48.

[18] S.N. Gelyan, A. N. Eghbali, L. Roustapoor, S. A. Y. F. Abadi Firouz and M. Dehghan, "SLTP: Scalable Lightweight Time Synchronization Protocol for Wireless Sensor Network," in International Conference on Mobile Ad-Hoc and Sensor Networks, Springer-Verlag Berlin Heidelberg, 2007, vol. 4864, pp. 536 -547.

[19] S. Rahmatkar and A. Agarwal, "A Reference Based Tree Structured Time Synchronization Approach and it's Analysis in WSN," in International Journal of ADHOC, Sensor and Ubiquitous Computing (IJASUC), vol 2, no. 1, 2011.

[20] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock Synchronization for Wireless Sensor Networks: A Survey," Ad-Hoc Networks, vol. 3, no. 3, pp. 281-323, March 22, 2005.

[21] S. Ganeriwal, R. Kumar, S. Adlakha, and M. Srivastava, " Network-wide Time Synchronization in Sensor Networks," Electrical Engineering Department, Networked and Embedded Systems Lab, UCLA Technical Report, 2003. 105.

[22] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-Sync Protocol for Sensor Networks," in Proceedings First International Conference on Embedded Networked

Sensor Systems, Los Angeles, California, U.S.A, November 2003, pp. 138-149.

[23] Salim el khediri, Nejah Nasri, Mounir Samet, Anne Wei and Abdennaceur Kachouri, ”Analysis Study Of Time Synchronization Protocols In Wireless Sensor Networks,” International Journal of Distributed and Parallel Systems, vol. 3, no. 3, pp. 155-165, 2012.

[24] D. L. Mills, “Internet Time Synchronization: The Network Time Protocol,” in Global States and Time in Distributed Systems, Z. Yang and T. A. Marsland, Eds., IEEE Computer Society Press, 1994.

[25] J. Elson and D. Estrin, “Time Synchronization for Wireless Sensor Networks,” in Proceedings of the 15th International Parallel & Distributed Processing Symposium, April 23 - 27, 2001, pp. 186. 106.

[26] S. Elkhediri, N. Nejah, A. Wei and K. A. Abdennaceur, “Survey of Synchronization Algorithms: Wireless sensor Networks Case Study,” in The Second International Conference on Communications and Information Technology (ICCIT): Wireless Communications and Signal Processing, Hmmamet, Tunisia, June 2012, pp. 6.

[27] P. Sommer and R. Wattenhofer, “Symmetric Clock Synchronization in Sensor Networks,” in ACM Workshop on Real-World Wireless Sensor Networks (REALWSN), Galsgow, Scotland, April, 2008, pp. 11-15.

[28] F. Kuhn and R. Oshman, “Gradient Clock Synchronization Using Reference Broadcasts,” in International Conference on Principles of Distributed Systems, Springer-Verlag, Berlin, Heidelberg, 2009, vol. 5923, pp. 204–218.

[29] F. Wang, P. Zeng, H. Yu1 and X. Zhao, “A Reliable Time Synchronization Protocol Wireless Sensor Network,” in Proceedings of the Third International Symposium on Computer Science and Computational Technology(ISCSCT ’10), Jiaozuo, P. R. China, 14-15, August 2010, pp. 009-013.

[30] M. Jabbarifar, A. S. Sendi, A. Sadighian, N. E. Jivan and Michel Dagenais, “A Reliable and Efficient Time Synchronization Protocol for Heterogeneous Wireless Sensor Network,” Wireless Sensor Network, vol. 2, no.12, pp. 910-918, 2010.

[31] H. Cho, J. Kim and Y. Baek, Enhanced Precision Time Synchronization for Wireless Sensor networks [Online]. Available:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231711/

[32] J. Wang, S. Zhang, D. Gao and Y. Wang (12 March 2014), “Two-Hop Time Synchronization Protocol for Sensor Networks,” EURASIP Journal on Wireless Communications and Networking [Online], Available : https://jwcneurasipjournals.springeropen.com/articles/10.1186/1687-1499-2014-39. [33] Francesco Lamonaca, Andrea Gasparri, Emaanuele Garone and Domenico Grimaldi, “Clock Synchronization in Wireless Sensor Network With Selective Convergence Rate for Event Driven Measurement Applications,” IEEE Transactions on Instrumentation and Measurement, vol. 63, issue 9, pp. 2279 – 2287, Sept. 2014. [34] F. Lamonaca, E. Garone, D. Grimaldi and A. Nastro, "Localized fine accuracy synchronization in wireless sensor network based on consensus approach", in Proceedings IEEE International Instrumentation Measurement Technology Conference(I2MTC), Graz, Austria, May 2012, pp. 2802-2805

[35] T. Beluch, D. Dragomirescu, F. Perget and R. Plana, “Cross-layered Synchronization Protocol for Wireless Sensor Networks”, in ICN’10 Proceedings of the 2010 9th IEEE International Conference on Networks (ICN), Washington, DC, U.S.A, April, 2010, pp. 167- 172.

[36] S. Rahmatkar and A. Agarwal , “An approach towards Lightweight, Reference Based, Tree Structured Time Synchronization in WSN,” in International Conference on Computer Science and Information Technology, Bangalore, India, Springer- CCIS, 2011, vol. 131 (Part-I),pp. 189-198.

[37] S. Rahamatkar, A. Agarwal, P. Sen and A. Yadav, “Tree Structured, Multi-hop Time Synchronization Approach in Wireless Sensor Networks,” in Information Technology and Mobile Communication, V.V. Das V, G. Thomas , F. G. Lumban , Eds., Springer, Berlin, Heidelberg, 2011, vol. 147, pp. 255-258.

[38] A. Kulaki and K. Erciyes, “Time Synchronization Algorithm Based on Timing-Synch Protocol in Wireless Sensor Networks,” in 23rd International Symposium on Computer and Information Sciences(ISCIS’08), Istanbul, Turkey, 2008, pp. 1-5.

[39] S. Hwang and Y. Baek, “Reliable Time Synchronization Protocol in Sensor Networks Considering Topology Changes,” in Proceedings of 7th International Conference on Distributed Computing(IWDC’05), Kharagpur, India, 2005, vol. 3741, pp. 105–110.