*J. Basic. Appl. Sci. Res.***, 3(3)678-687, 2013**

© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied
Scientific Research
www.textroad.com

A Secure and Global Time Synchronization Algorithm in Wireless Sensor Networks

Mojtaba Joukar, Mehdi Bagherizadeh[*]

Department of computer engineering, Islamic Azad University, Rafsanjan Branch, Rafsanjan, Iran

ABSTRACT

Time synchronization protocols for wireless sensor networks are highly taken into consideration due their abundant applications. Among the various algorithms proposed in this regard, the time synchronization protocols operating distributed and globally are being more considered and it has been inclined to these kinds of protocols more than ever in recent years. In addition, these algorithms are being improved to be more secure and have much more efficiency in real environments. Considering some real cases (Presence of malicious and corrupted nodes which generate unexpected values), this paper presents a secure time synchronization protocol that is globally distributed. Encryption, authentication and hashed methods are not highly efficient due their high overhead and getting massive for the exchanged packets in addition to their computational overhead to sync this type of protocols. In the proposed algorithm, we do not make use of previous common used methods such as authentication and encryption methods which have been presented before, rather the algorithm in itself with enforcing some mechanisms attend to presence of malicious and corrupted nodes and performs the time synchronization as distributed and globally. Finally, we review the results gathered from simulations in terms of time, memory and computational overloads, and the efficiency of the proposed algorithm regarding the current conditions will be found out.

KEYWORDS: wireless sensor networks, malicious nodes, secure global time synchronization protocol.

1. INTRODUCTION

Time synchronization in wireless sensor networks proceeds to unify the time of wireless sensor node distributed in an environment. Different issues are being discussed in synchronization: 1- The number of packets exchanged between nodes for synchronizing and because the packet sending or receiving consumes too much energy compared to data processing, this is an important issue in wireless sensor networks [1, 2]. 2- Acting locally or globally means that whether a group of nodes or our entire sensor network is synchronized. 3- The speed of synchronization. 4- The accuracy of synchronization, etc [9, 10]. In this paper the security is highly considered as an important issue. In previously provided algorithms some security issues haven't been considered and nodes are being assumed as uncorrupted. The problem where some malicious nodes takes synchronization packets and send them to the next node with delay or some nodes have error in time accuracy and send unexpected values to the next node, can be considered as a security issue [22]. In this paper we represent the algorithm that performs synchronization in our entire sensor network that means it globally synchronize the whole network and consider the security issues as well.

As time synchronization has got so many applications in sensor networks and performing many tasks in sensor networks requires time synchronization, the importance of this issue can be found out [7]. For example, speed of calculation, track, syncing nodes for sending and receiving data (Note: Because the nodes are sleeping most of the time due to lower power consumption when they wake up to send and receive data and when sleep, is itself an important category). Now consider the real environment that is risky, for example we understand the importance of this issue when some nodes do not work properly and their times show unexpected values or values with high error rates and or in the other types where some malicious nodes existing in the environment receiving the packets and sending them with a delay. Considering these themes (security issues) that were mentioned, we presented the synchronization algorithm.

This paper presents a secure time synchronization protocol that is globally distributed. Inasmuch as high overhead and making large the exchanged packets in synchronization, Encryption methods, authentication and hashing do not have high efficiency. In addition to that they have computational overhead so that we do not use previous conventional methods such as authentication and encryption, rather the algorithm in itself with enforcing some mechanisms attend to presence of malicious and corrupted nodes and performs the time synchronization as distributed and globally.

[*]**Corresponding Author:** Mehdi Bagherizadeh, Department of computer engineering, Islamic Azad University, Rafsanjan Branch, Rafsanjan, Iran. Email: m.bagherizadeh@srbiau.ac.ir

The rest of this paper is organized as follows: section II examines the works done in the area of securing the time synchronization protocols and also reviews the works done for globally time synchronization. In section III we introduced a secured time synchronization protocol and finally section IV shows the simulation and reviewing the results of it and compares the proposed algorithm to another algorithm. Section V concludes the paper.

II. Previous works

Because time synchronization protocols have many applications in wireless sensor network; different algorithms have been proposed for doing this purpose, while these algorithms do not pay attention to the security issue that we can mention RBS and LTS TPSN and others [6]. In LTS which is based on sender / receiver, the sender sends a packet which contains the current time stamp. The receiver sends a packet containing its time stamp back to the sender upon receiving the packet containing the sender's time stamp. Now sender matches its time with the receiver [2]. The drawback of this method is its low accuracy. TPSN improved the accuracy of LTS by eliminating some of the indeterminacies in time stamping, but the defect of TPSN is in constructing the hierarchical tree that it should construct a tree with least possible height where it wants to synchronize an area in the sensor networks and it should also include all the nodes. Synchronization accuracy doesn't reduce due to more height of some nodes [16, 17]. Constructing this tree has some challenges while the speed of this algorithm was relatively low. In RBS which is based on receiver / receiver; the sends a packet including its time stamp to the receivers and receivers in the same range make their selves synchronized with each other by sending their packets [5]. This algorithm is also locally and has a high overhead if it wants to perform globally. In recent years a special attention has been paid to the security issues and some algorithms are provided with certain assumptions [15, 21]. In some cases it is considered that we have a reliable source node that is far from our main network and it is assumed that it is under no threat. That node sends the synchronization packets and nodes existing in the distant environment are being synchronized by this source node. Packets are being made accessible to the existing nodes in network by means of authentication and encryption methods and no packet type is exchanged in nodes of distant environment. In some other methods [14, 22] directional antennas have been proposed in order to help only those nodes in specially that direction access. Challenges of these methods are: no attention to more real topics (corrupted and malicious nodes) and possibility of reliable source node's absence in distant environment. In addition to that in some papers some threats have been introduced for various algorithms but no solution has been provided for that. Up to now, there has been provided no solution for globally distributed synchronization.

Totally global time synchronization is divided into three general categories [12, 18]: synchronization based on all nodes, synchronization based on clustering, and synchronization based on Asynchronous propagation [8, 16, and 17].

In synchronization based on all nodes, a loop path of all nodes in the network is created and the synchronization is performed based on the time when the packet has been sent from the source node and goes back to it after doing the loop [20]. In this method it is assumed that the arrival time of the packets at each step is equal to other steps. In the clustering based method it is similar to the synchronization based on all nodes. The difference is explained as follows: Consider a cluster which has head node. Now according to the previous method; first of all the head nodes are synchronized to each other similar to the previous method, then the nodes in the clusters synchronize each other with the head nodes and the synchronization performance is done [12, 13]. In section 3 the other method based on asynchronous propagation which is a global algorithm is explained.

III. Proposed security algorithm

The general method of presented time synchronization in this representation is as follows:

For each node with uniform possibility do:

1- The starting node asks all of its neighbor send the amounts of their time stamp.

2- Means from the amounts of neighbors' requested time stamps.

3- New amounts are sent back to the neighbor nodes.

4- The neighbors adjust themselves to the new amount.

Now the threats of the represented modified synchronization algorithm are being reviewed in this section. But before that we attend to review the total methods of malicious or corrupted nodes' performance in order to realize how these nodes challenges a total tome synchronization algorithm and in continue we investigate which strategy is used for the Asynchronous propagation of malicious nodes in each four steps to stop that algorithm and not to overcome the synchronization. Next it will be cleared that through which mechanism the proposed algorithm would overcome malicious nodes and defects of corrupted nodes in the four main steps of its algorithm.

A) Malicious nodes can act in six ways

1-They send unacceptable times with high rates errors (Message manipulation)

2-They send their amounts with delay (Message Delay)

3- They can pretend that are another node and request for an activity (Masquerade attacks)

4- Suppose that there are two states: first the density of malicious nodes is around an applicator node and they send the amounts coordinately with collusion and make mistakes in the synchronization act of that applicator node. Second the possibility of nodes' collusion (malicious nodes) does not exist because of the needed density around the applicator node of synchronization.

5-the malicious nodes send the packet request of synchronization continuously and introduce themselves the applicator node of synchronization.

6-They can pretend to be other nodes and declare their wanted amount. We confront with two amounts for this node.

B) The existing nodes have their own errors that should be differentiated from the previous case (A):

1- The nodes have wrong amounts or have high errors in debut.

2- Some nodes have high rates of delay.

General method of proposed algorithm for solving the problem:

Each node (starting one) sends a request for all of its neighbors and asks them to send the time amounts of themselves for it. Then it propose time amount for itself and its neighbor in order to have equal time amounts with each other (for the first degree the equal time amount). After this because the neighbor nodes have reached a mathematical equation can realize the real amounts from amounts which have been sent from the malicious nodes.

Note: It has been tried to make the proposed amount for synchronization of neighbor nodes more near to the time amount as much as possible. An advantage for this method is increasing in accuracy and more convergence of nodes leading to less time synchronization in specified time duration.

The main step is when the time amount is sent for synchronization and then the main challenge is begun.

A very important assume is that if the percentage of malicious nodes is high, definitely we have a decrease in synchronization accuracy and we can do nothing for this problem in this algorithm.

The proposed algorithm and possible behavior of malicious nodes can be divided into 4 general steps. The effects of malicious nodes and the strategies they follow in each step should be investigated and we may also review whether the proposed algorithm is resistant against threats and attacks.

Notice: Neighbor nodes are that nodes set that are placed in the range of its board.

Step 1: One of the existing nodes in the network (the starting node) sends the requested packet for synchronization of all nodes that it affects on them (these nodes are called neighbor nodes). Next all the nodes those are in range of its board takes this packet. For instance, suppose that node A (in Figure 1) is the starting node the contents the request packet which node A sends, includes the index of node A and the number of request for synchronization (k) (for instance if it is third time it requests it sends 3) that is shown is Figure 1. In this section the attack which is going to begun is pretending the malicious node as a specified one and sending the synchronization request (for example node M sends the Request packet for B and D with the index of A). This problem can be solved by means of a scheduling in the nodes in a way that each node plays down the next request of a neighbor node based on the time it has spent on the synchronization for the neighbor nodes (nodes B and D have a synchronization that respond to a packet request from A again after spending 2 ms). The problem of this thread is making an abundant traffic load in the network.

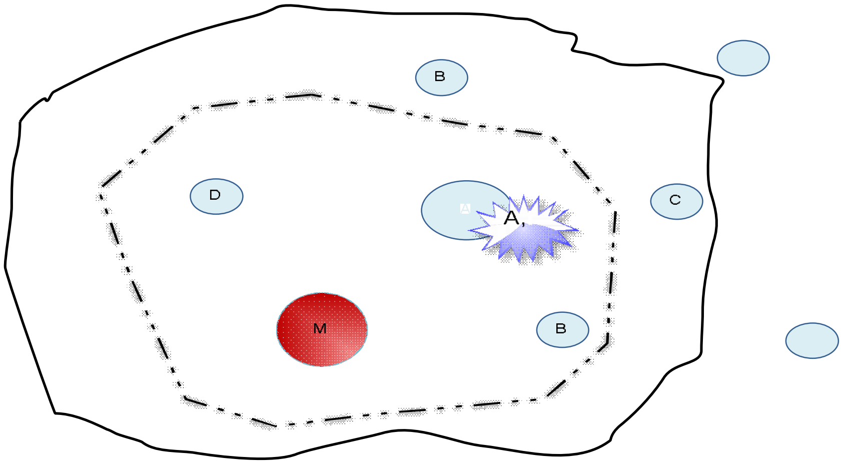


Figure 2- The first stage of fixed-line range of nodes A and line M

Step 2: The neighbor node (shown by j, it may be any neighbor node) takes the request and keeps the time of receipt (node j makes its time stamp as tj,t) and sends the amount of time stamp in the packet (tj,2). In addition it keeps both amounts in its memory. The threat of malicious node is possessed in common which the previous section. It means that if a threat exists is requesting for synchronization from the malicious nodes. The possible solution is represented in the first step. In addition we should identify neighbor nodes for more security which means that each node tries to identify how many nodes it is neighbor with and what is the behavior of its neighbors.

You can clearly see that if the number of colluding nodes is much more than our network nodes. Definitely the existing nodes in the network cannot be synchronized anymore. The more number for colliding nodes causes the more error for the synchronization algorithm. Figure 2 shows the second step of time stamping by the neighbor node (j) upon receiving the packet explained in step 1.

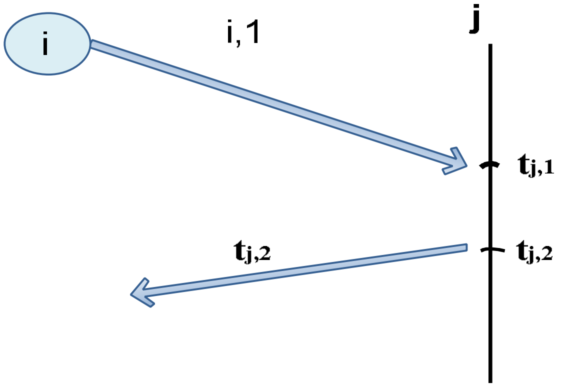


Figure 2 the second step of time stamping by the neighbor node j.

Step 3: Now in the starting node (Generally in this paper the starting nodes are shown as i) the entire time stamp amounts from the neighbor nodes are gathered in a time interval (Max-Delay). A mean is received from these nodes and it is converted to the nearest acceptable time amount (Assume the calculated mean is 3.4 and the acceptable intervals are integers, mean is being considered as 3). The delay time for each of the nodes are recorded in a way that when an amount from the starting node is found it is being considered equal to zero (in Figure 3, node B) and the difference between this amount and the second reached amount is delay for the second node. For instance the delay for node C is the result of $t_{i,rec,2}$-$t_{i,rec,1}$ and it will be recorded for that node in the memory of i). A mean is received from the delay of nodes (or minimum and maximum of them are recorded) and two received means and the index of applicator node are sent in the new packet. Hereby the malicious nodes can act in two different ways: 1- they can send packets with delay (making abundant delay) 2- they can send a wrong time amount (with high or low rates of error)

Meanwhile, the amount of delays from taking the first amount from neighbor should not exceed the Max Delay and if a time stamp comes at the time we do not consider it (1-delays(more or less), 2- wrong time amounts (with high or low rates of error). We have got threshold for high rates of delay that their amounts are not included.

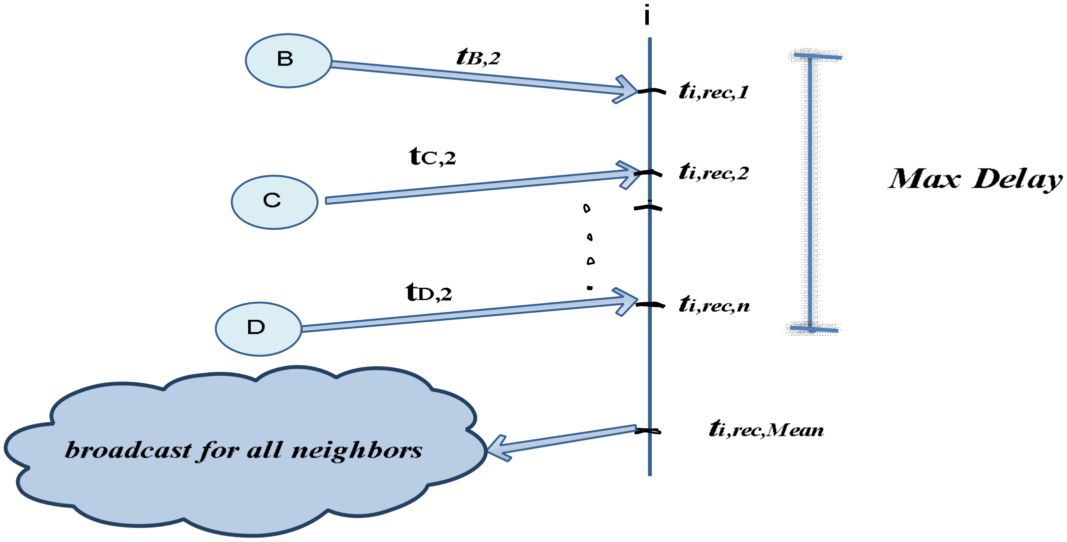


Figure 3. Receiving the values of neighboring nodes by node i is the starting node (In our example, node A).

$$Mean=\frac{\sum_{k=1}^{n} t_{k,2}}{n} \qquad 2\text{-}1$$

Mean=Mean convert to nearest acceptable time; *2-2*

$$T_{C,Delay,k} = t_{i,rec,2} - t_{i,rec,1} \quad \textit{// Delay node C in k's level synchronization} \qquad 2\text{-}3$$

$$T_{C,Delay,Mean} = \frac{\sum_{i=1}^{k} T_{C,Delay,i}}{k} \qquad 2\text{-}4$$

For a high error condition, we also use threshold (meanwhile two amounts that have the maximum difference are eliminated). In addition to what has been mentioned above, the received amounts are being noted and for each node the mathematical relation is written and will be updated.

If two amounts is received for a neighboring node such as C, the amount is selected which its delay of reaching and also amount of calculated (aberration coefficient) is closer to it. The mathematical relation which is in the starting node's memory and it's used there is called mathematical relation of aberration for the number k of synchronization request and it's like as following:

$$\theta_{i,c,k} = \left(t_{c,2,k} - t_{c,2,k-1}\right) / \left(t_{i,rec,c,k} - t_{i,rec,c,k-1}\right); \qquad 2\text{-}5$$

Where the numerator is the amount of sent time stamps of node C and in the denominator time stamps in node I for node C are considered.

Step 4: This step is being taken place in each neighboring node that is shown by j. Packets sent by requesting node is reached to each neighboring node. Consider the case that we have several reached values which one of them is right and the others are wrong. Suppose node j receives 3 values with sequential delays that all 3 claim that those are from the starting node and a mean amount for all of nodes (the amount which they are supposed to adjust with now) is announced. The specifications that should be considered for the received mean amount are that it should be in a specified period of time. It means assume the number are integers results in sending integers from malicious nodes. So we can guess which value is integer more easily because in due to having a relation of that node's behavior, it could opt for the right value from sent values easier. In each two neighboring nodes, two relations are held; one for the mean and the other for neighboring nodes. If the number of collusion nodes gets higher, the percentage for success of the synchronization algorithms decreases.

Neighbor node of the interval (2-7) takes values.

$L_j=\Theta j.H_j(t)+\Phi_j$ 2-6

Max delay+δ(delay between sender and receiver) + $t_{j,2}$ 2-7

After this step it equalizes the amount which is taken from the stating node to Lj. Now we should find an Aberration coefficient (skew) for the taken amount in order to realize its validity by comparing them with previous amounts.

$$\theta_{j,i,k} = \left(T_{i.Mean,k} - T_{i,Mean,k-1}\right) / \left(t_{j,3.k} - t_{j,3,k-1}\right); \qquad 2\text{-}8$$

$$If\ \left(\bar{\theta}_{j,i,Min} < \theta_{j,i,k} < \bar{\theta}_{j,i,Max}\right) \qquad 2\text{-}9$$

$L_j = T_{Mean,i}$ *// the mean from first nodes (in my example from node A)*

Else

$$L_j = \theta_{j,i,Mean}.H_j(t)$$

$$\theta_{j,i,Mean} = \left(\theta_{j,i,k} + \theta_{j,i,k-1} + \ldots + \theta_{j,i,1}\right)/k; \qquad 2\text{-}10$$

If $\overline{\theta}_{\min}$ is named when we consider 10% of minimum values for θ and also we use this action for maximum values (the mean for 10% of maximum values), then we may use above relation (2-9) and define this algorithm secure to use.

Figure 5 shows flowchart of proposed algorithm.

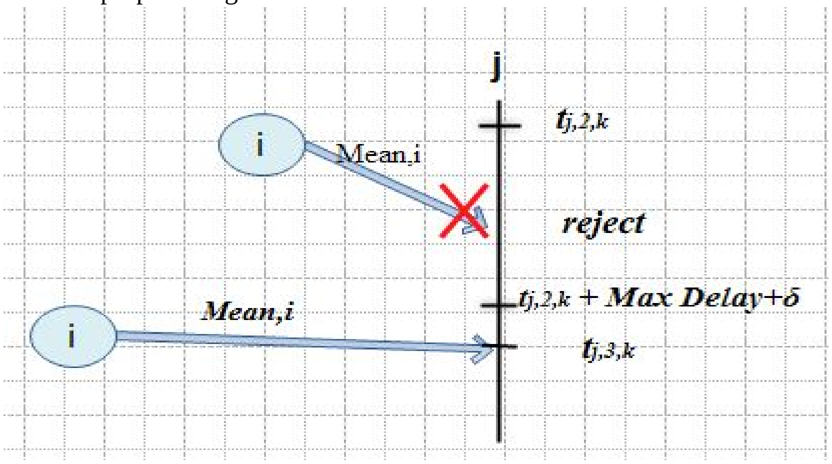
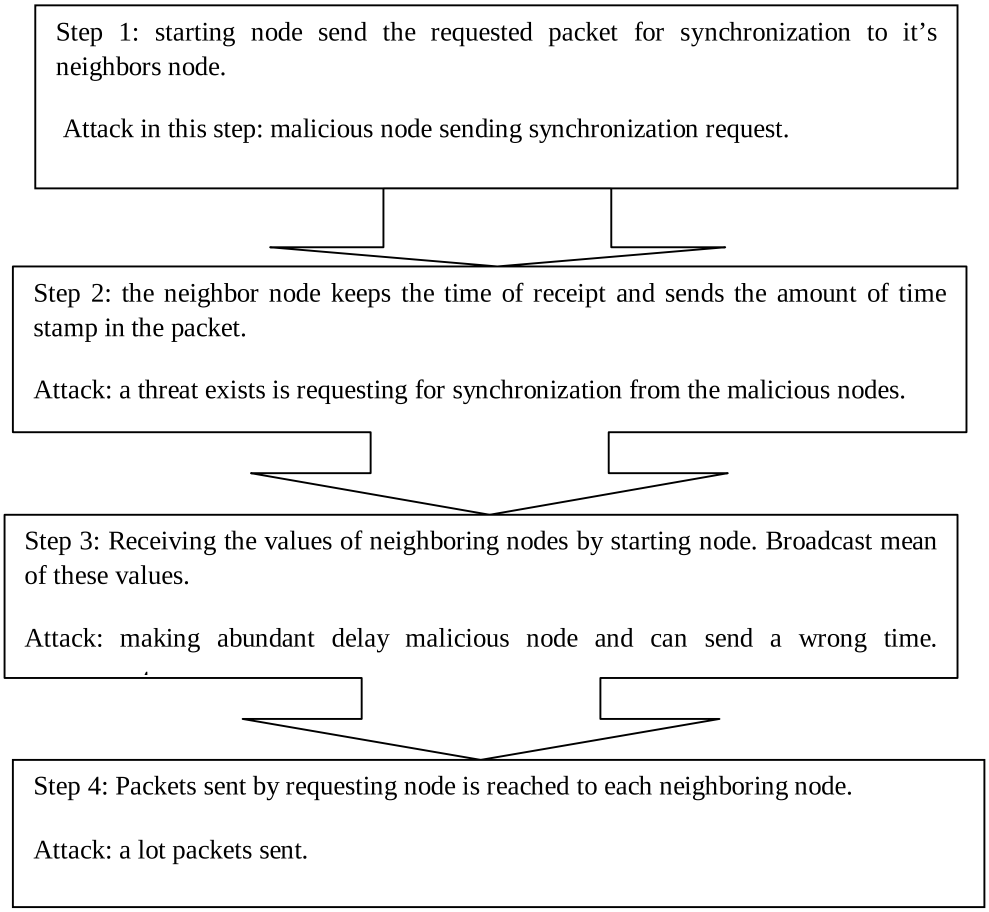


Figure 4. Ignoring packets after a specified time.



Step 1: starting node send the requested packet for synchronization to it's neighbors node.

Attack in this step: malicious node sending synchronization request.

Step 2: the neighbor node keeps the time of receipt and sends the amount of time stamp in the packet.

Attack: a threat exists is requesting for synchronization from the malicious nodes.

Step 3: Receiving the values of neighboring nodes by starting node. Broadcast mean of these values.

Attack: making abundant delay malicious node and can send a wrong time.

Step 4: Packets sent by requesting node is reached to each neighboring node.

Attack: a lot packets sent.

Figure 5. Flowchart of proposed algorithm.

Part IV: Simulation results and comparison:

This part is divided to two sections. In the first section, we simulate a general algorithm of asynchronous propagation and also investigate the influence of different parameters on this algorithm such as such as node density, number of nodes and node ranges. Then we try to analyze the simulation charts taken from the simulation. And finally in the second section we simulate the proposed secured algorithm and compare with asynchronous propagation algorithm.

Simulation is done by means of MATLAB software for various numbers of nodes (100 to 900) and with changing different parameters. The first considered parameter is the nodes' density parameter. If the density of nodes is higher in a range, the synchronization accuracy increases. Then we may say that the accuracy of synchronization has a direct relation with the nodes' density in a range. As shown in Figure 5, one of the disadvantages that we may attribute to this algorithm is that if the density of nodes is low, the accuracy of the algorithm will decrease exponentially.

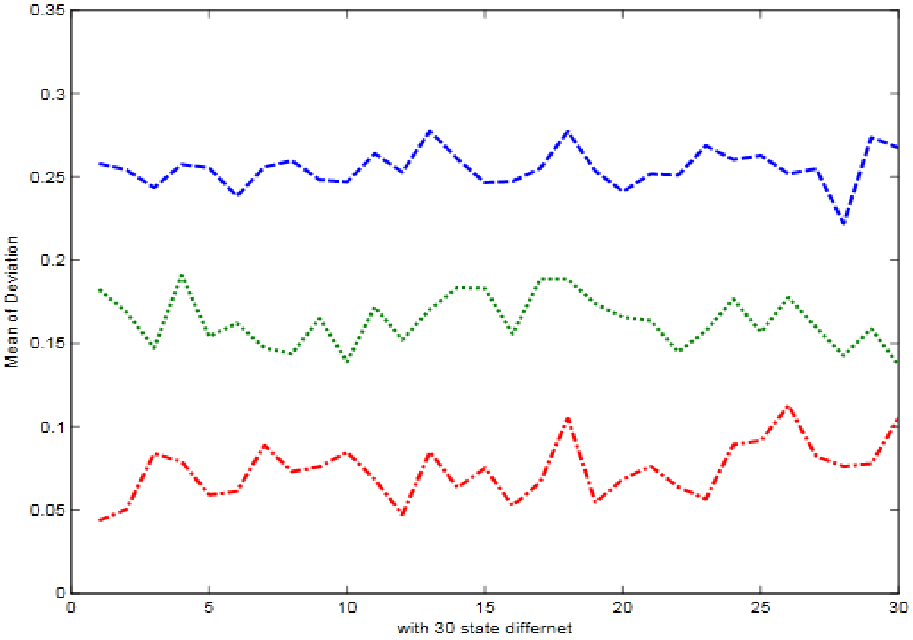


Figure 5 Algorithm have been implemented 30 times with the similar values and the same parameters but different range limits.

In figure 5 the simulation has been done for 30 states with similar parameters and identical initial values but with different ranges. The upper graph has greater mean deviation as a result; nodes in this graph have lower range and if we consider 1 for that amount and 2 for the middle one and 4 for the lower, it is seen that by widening the range of nodes, the convergence speed and accuracy of the algorithm gets double.

About the range limits we can a higher range of nodes causes an increase in the accuracy of the algorithm.

The next thing is about the nodes' density in an environment where the higher density results in the higher amounts of accuracy and convergence speed of the algorithm. In below image you can find 20 sets of data which are tested with different amounts of density in the environment. As observed, in each set, the condition where the density is higher in environment (the higher number of nodes in a similar environment), the accuracy and convergence speed goes up. The densities are simulated with the amounts of 1, 2, 4, 8 and 16. For example, the density in brown set is 16, for the orange one is 8 and finally this amount is equaled to 1 for the blue set of data. Figure 6 shows Comparison of the algorithm with different density values.

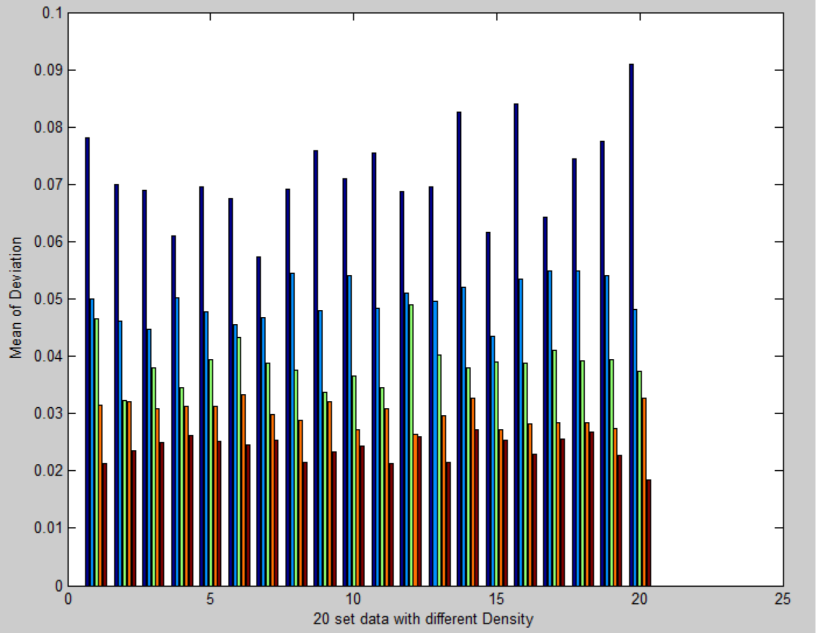


Figure 6. **Comparison of the algorithm with different density values**

When we attend to perform the algorithm without considering the expressed security issues, the mean deviations is much more than when we consider it. Algorithm was run several times and the results indicate the excellent performance of this algorithm. In figure 7 we have used the mentioned algorithm 20 times for a set of data in two states of using securing algorithm and not using it for the other state. The total mean of deviation can be found out in each of sets according to the figure. It has been goes up to 1000s and the number of nodes is 900.

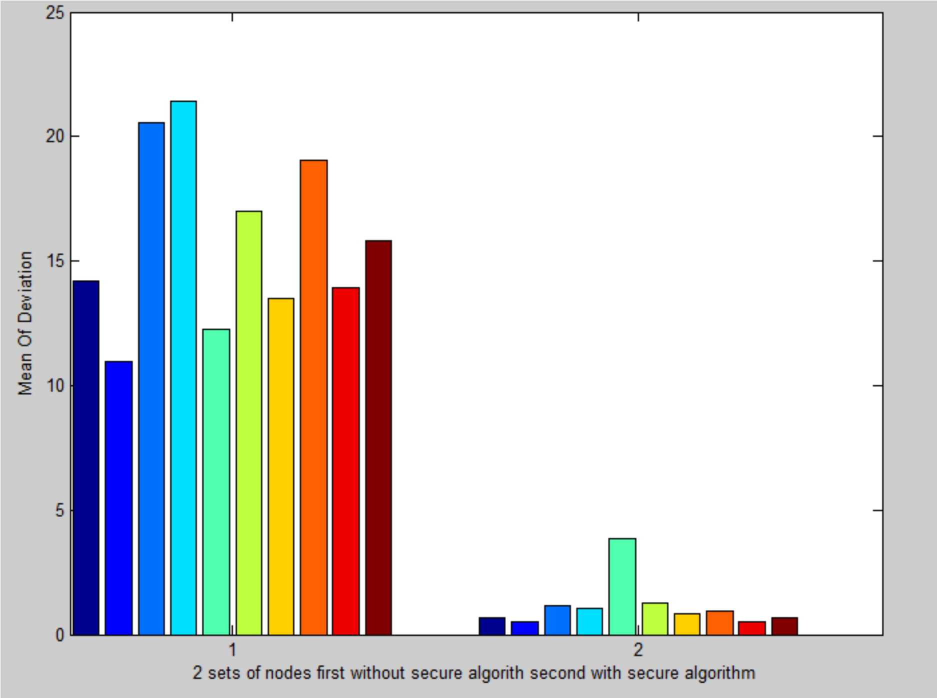


Figure 7. The comparison of the mean deviation between two proposed algorithms that one has been secured and the other has not.

About orders that have been applied if we assume that each node does the calculation of the 2-8 relationship in one clock, it can be considered that the 2-8 relation is done in one CPI. But this calculation does not exist in unsecured conditions. Figure 8 indicates the number of applied calculations based on the number of nodes in the secure mode and shows a computational overhead in the proposed secure algorithm.

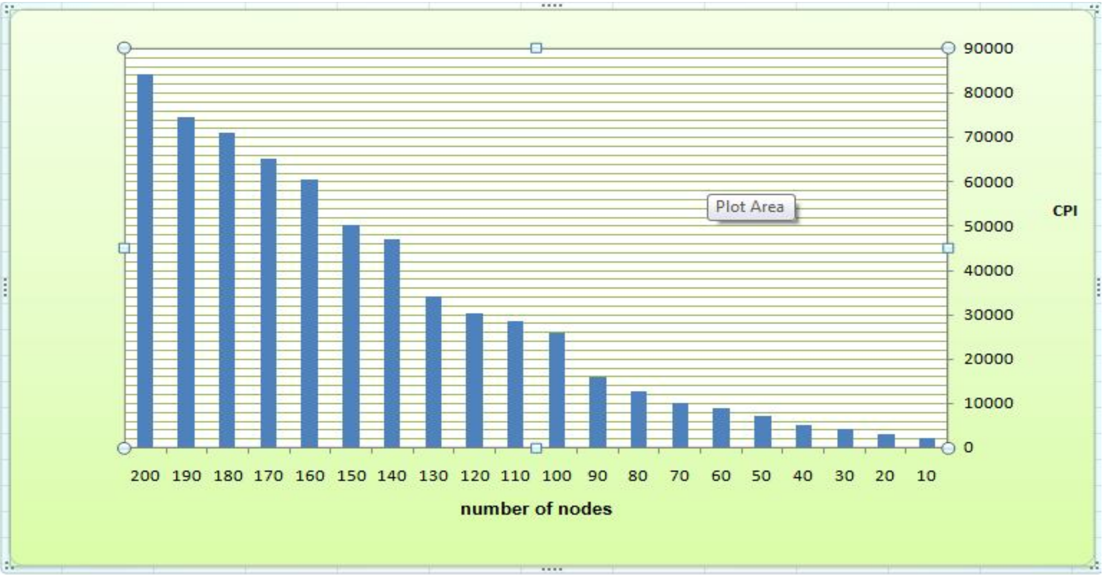


Figure 8. Orders applied in terms of the number of nodes

About the convergence speed of the nodes to a common amount we can use figure 9. This figure shows that he proposed algorithm synchronizes nodes much faster over time and has less standardized mean difference. In this figure the horizontal axis represents time (seconds) and the vertical axis represents the mean standard difference. Simulation has been done considering five malicious nodes and totally 80 nodes in the environment. As shown in this figure proposed secure algorithm has less standard deviation than the asynchronous algorithm.

For overhead of the memory in this algorithm suppose O (n) is the complexity of memory where n is the number of nodes of sensor network . To explain about the amount of this memory space we may say that each node should consider approximately 1KB of memory all of its neighbors. For example, if a node has five neighbors, five-kilo bytes of memory will be required for that node.

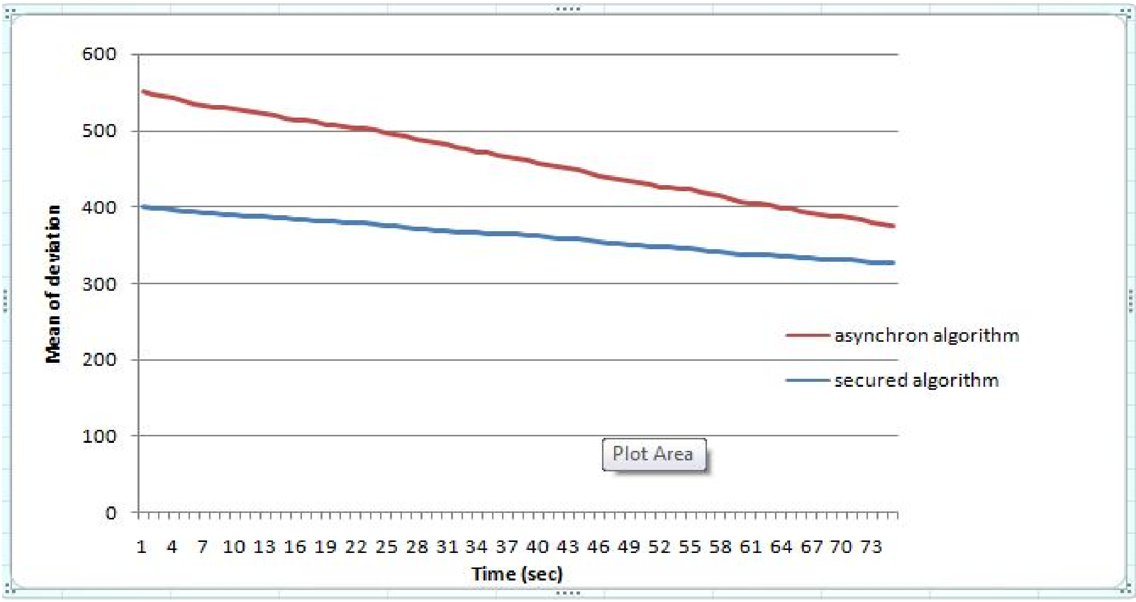


Figure 9. The comparison of the proposed algorithm with Asynchronous secure.

Conclusion

Time synchronization in wireless sensor networks proceeds to unify the time of wireless sensor node distributed in an environment. In previously provided algorithms some security issues haven't been considered and nodes are being assumed as uncorrupted. Inasmuch as high overhead and making large the exchanged packets in synchronization, Encryption methods, authentication and hashing do not have high efficiency. In this paper a secure time synchronization algorithm that is globally distributed is proposed. This algorithm in itself with enforcing some mechanisms attend to presence of malicious and corrupted nodes and performs the time synchronization as distributed and globally.

REFERENCES

1. J.Nieminen,Q.Lijun and J.Riku. 2011. Network-Wide Time Synchronization in Multi Channel Wireless Sensor networks. *Wireless Sensor Network*.39-53.
2. Y.Suyoung, V.Chanchal and L. Sichtiu, 2005. Tight Time Synchronization for Wireless Sensor Networks.
3. F.Kiyani1, A.Aghaee,H.Tahmasebi, 2008. A Distributed Algorithm for Time Synchronization in Wireless Sensor Network.
4. L. Jun,2008.Scalable synchronization of clocks in wireless sensor networks. Ad Hoc Networks,791-804.
5. A.Agarwal, 2010. Analysis and Comparative Study of Clock Synchronization Schemes in Wireless Sensor Networks. IJCSE) International Journal on Computer Science and Engineering. 536-541.
6. R.Keun, L.Jaehan,k.Jangsub, S.Erchin and Ch.Yik, 2009. Clock Synchronization in Wireless Sensor Networks.56-85.
7. G.Wei, H.Yu, M.J.Hong, 2008. Improving the Security of Time Synchronization in WSN.
8. N.Vinod, R.Suresh, 2010. Energy Efficient Global Clock Synchronization for Wireless Sensor Networks.IJWMN.
9. X. *Chaonong, Z.Lei, X.Yongjun, L.Xiaowei*. Broadcast Time Synchronization Algorithm for Wireless Sensor Networks.
10. Y.Suyoung and Mihail L. Sichitiu. Analysis and Performance Evaluation of a Time Synchronization Protocol for Wireless Sensor Networks.27695-7911.
11. S.Luca, F.Federico, 2010. A consensus-based protocol for time synchronization in wireless sensor networks.
12. Jari.Nieminen, Riku.Jäntti, 2008. Time Synchronization in Multi-Channel Wireless Sensor Networks.
13 D. Raskovic, O. Lewis, 2009. Time synchronization for wireless Giessel sensor networks operating in extreme temperature conditions. . 41st Southeastern Symposium on System Theory, 24.
14. K. Sun, P. Ning, C. Wang, 2006. Secure and resilient clock synchronization in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*. 395-408.
15. A-Y.Saravanos. Hybrid energy-aware synchronization algorithm in wireless sensor networks. The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07).
16. R.Prakash, N.Kendall, 2010. Time synchronization in wirelesssensor networks. International of jounrnal of ubicomp.
17. H-H-Bakker1, K-Mercer, H.Wyatt, A Review of Synchronization Methods in Wireless Sensor Networks,Timothy.
18. Johannes Barnickel and Ulrike Meyer, 2009. Secure Time Synchronization Scheme in Wireless Sensor Networks.
19 A.Boukerche and D.Turgut, 2007. Secure time synchronization protocols for wireless sensor networks. IEEE Wireless Communications.
20 Vinod.Namboodiri,Suresh Ramamoorthy, 2010. Global Clock Synchronization for Wireless Sensor Networks,Journal of Wireless Mobile Networks (IJWMN)