Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 8, No. 3, August 2017), Pages: 1-17
www.jacr.iausari.ac.ir

A High Performance and Secure Way to Time Synchronization in Wireless Sensor Network

Rezvan Kazemi and Mehdi Bagherizadeh✉

Department of Computer Engineering, Rafsanjan Branch, Islamic Azad University, Rafsanjan, Iran

kazemi8374@gmail.com; m.bagherizadeh@srbiau.ac.ir

Received: 2016/07/28; Accepted: 2016/08/30

Abstract

Time synchronization protocols, because of having many applications in WSN[1] have been highly regarded. These protocols have been more considered due to operate in a distributed and globally way. In addition, time synchronization protocols are improving toward to becoming secure and having a high applicability. This paper presents a new algorithm with respect malicious nodes to provide a secure distributed and global time synchronization protocol. Methods of encryption, authentication and hashing don't have a high performance because of high overhead for the synchronization. In the proposed algorithm the previously used methods, like authentication and encryption are not used, but instead the algorithm by applying mechanisms within itself takes into account the corrupted and malicious nodes. Finally, the gained results of the simulation in terms of the deviation average of the standard criterion, time computational, and memory overheads are considered. In the simulation also the applicability of the proposed algorithm will be realized.

Keywords: *Wireless Sensor Networks; Time Global Synchronization Protocols; Security; Malicious Nodes.*

1. Introduction

Time synchronization in wireless sensor networks attend to make synchronize the times of the wireless sensor nodes which spread in an environment. In synchronization, various issues are raised: 1- The number of the exchanged packets between nodes to perform the process of synchronizing. Since the sending or receiving packets consumes a lot of energy in comparison to processing data, this is also an important issue in the wireless sensor networks [1], [2] . 2 - To operation locally or globally (means whether a region of nodes synchronize or the entire wireless sensor network synchronize). 3- The speed of synchronization. 4- The accuracy of synchronization and so on [26]. In this paper, the important issue of security has been noted. In the presented algorithms [2], [5], [6], [18], [22] some security issues have not been regarded and it is assumed that the nodes are safe. Some security issues mean that some malicious nodes receive the packets of synchronization and send them to the next nodes with delay [41] or some nodes have errors in terms of time and send the values that are out of our expectation

[1] Wireless sensor networks

[8]. In this paper, an algorithm that has done the operation of synchronization globally and also pays attention to the security issues has been considered [21], [25].

Since the time synchronization has many applications in the sensor networks, the importance of this issue can be realized [7]. Examples of this can be named to calculate the speed, tracking, synchronization of nodes for sending and receiving data (Note: Since the nodes are mostly in sleep due to consuming less energy in sleep [39], what when to wake up for sending or receiving and when go to sleep is an important issue) [38] . Now if we consider in a real environment that may be a risky one, for example some nodes don't work properly and their times show unexpected values or a rate of high errors or there are some malicious nodes in the environment that receive the packets and send them with delay, the importance of the matter can be taken. Regarding the topics (security cases) that were raised, it has been tried now to present the time synchronization algorithm.

Methods of encryption, authentication, and hashing due to the high overhead and bulky becoming of the exchanged packets don't have a high applicability for synchronization, in addition to having computational overhead as well. In this algorithm we don't use the already used and conventional methods [2], [6], [18], [22], such as authentication and encryption algorithms, but instead the algorithm by applying mechanisms within itself pays attention to the damaged and malicious nodes and does the global time synchronization.

In the section2 the works done in the scope of securing the time synchronizing protocols are examined and the works done for global time synchronizing are considered too. In the section 3 the proposed secured time synchronizing protocol is attended. In the section 4 simulation and the study of the results in the scope are examined and ultimately in section 5 shows concludes the work.

2. Related Work

Because of having many applications of the time synchronization protocols in wireless sensor networks[6], [17] there have been various algorithms such as RBS[1] [20], TPSN[2][24], LTS[3][2] and FTSP[4] [26] . In LTS[4] [2] based on the sender / receiver, the transmitter sends the packet that contains the current time stamp, and as soon as receiving the packet containing the time stamp of the transmitter, the receiver re-sends a packet containing his own time stamp to the transmitter, then the sender makes one his own time with the receiver. This algorithm is a focused one which has made a spanning tree and then the nodes along the N-1 edge of the spanning tree synchronize and work in a distributed way which means that each node can decide for the time of its synchronizing. The disadvantage of this method is its low accuracy and tolerance in the face of error.TPSN[3] [24] has two phases .The first phase is the determining level in which a level number is assigned to each node. and the second one is the phase of synchronization in which the node of the level I till the node of the level I-1synchronize.TPSN [24] by eliminating some of the unknown in time stamping, improved the accuracy of LTS [2], but the disadvantage of TPSN[24] was making the

2. Reference Broadcast Synchronization

3.Timing-sync Protocol for Sensor Networks

4.Lightweight Time Synchronization

5.Flooding Time Synchronization protocol hierarchy tree. When it wants to synchronize a location in the sensor networks, making its own tree in terms of that the height of the tree should be the minimum bring some challenges about too, because it must contain all the nodes not to let the accuracy of the synchronization lessen due to the more height of some of the nodes. In addition, the speed of the algorithm was also relatively low. In RBS [20] based on the receiver / receiver, the node of the transmitter sends the packet which includes its own time stamp to the receivers, and the receivers being in one scope synchronize with each other by sending packets too[5].This algorithm is locally and to be applied globally has a lot of overhead. In FTSP [26] nodes are dynamically selected and keep the Global Time and other nodes synchronize their time with the root node. An ad hoc structure is created by the node to transmit the Global Time from the root to the other nodes. of course this protocol has also some advantages which are to be more resistant in front of the failure of links, nodes and topology changes.

The notable point in these protocols is that the security issue is not addressed [6]. In the field of the security of the nodes that has received a special attention in the recent years [8], [22], [25] with certain assumptions they have attended to some algorithms. In some cases, they consider that we have a reliable source node that is away from the main network [8] and there is no threat against it and that node sends the synchronization packets and then the nodes existing in the remote environment synchronize by the source node .The packets by authentication and encryption are provided for the nods existing in the network. In addition, any package cannot be exchanged between the nodes in the remote environment. In some other methods [8] directional antennas have been proposed, so that only the nodes existing in that direction access to the information and the other is that there has not been much attention to the actual topics (malicious or corrupted nodes) and the impossibility of existing a reliable source node in the distance. In addition, some articles [9], [23] have attended to introduce several threats for different algorithms, but have not provided a solution to it. Almost there has not been presented any solution for the distributed and global synchronizing protocols yet. Generally, the global time synchronizing is divided into the three general categories: Synchronizing based on the all nodes [12], sync based on clustering [12], sync based on asynchronous release [12]. In this study it has been tried to show the time synchronization based on the asynchronous release and also to present an algorithm that does the process of synchronizing globally which means it synchronizes the network nationwide and pays attention to the security issues of the nodes too.

In synchronizing based on the all nodes[12], in a way that a ring route of all of the existing nodes in the network is created in which based on the time that the packet was sent from the source node and after passing through the ring returns to the source node, the operation of synchronization is performed. In this method it is assumed that the time of the packets arrival is the same from any step to the other one. The synchronization method based on clustering [12], is the same method as the one based on the all nodes with the difference that in this method there should be considered a cluster, that this cluster contains a head node. Now, according to previous methods, firstly the reference nodes of the cluster synchronize with each other according to the first method and then the existing nodes in clusters synchronize themselves with the reference nodes of their own cluster and the act of synchronization end [12],[13]. In the Third part of the method based on asynchronous release, it will be explained that it is a global algorithm.

3. Providing the Proposed Secured Algorithm

Notice to the asynchronous release algorithm in the following:

```
1: For each node $n_i$ with uniform probability do
2: Ask its neighbors the clock readings (read values from $n_i$ and its neighbors)
3: Average the reading (compute)
4: Send back to the neighbors the new value (write values to $n_i$ and its neighbors)
```

The above figure is an algorithm that in papers[25], [30] refer to it by the name of based on asynchronous release which we have rewritten it in the following way for a closer look:

```
For each node $n_i$ with uniform probability do
1: send a request packet for all neighbors and ask send back clock reading
2: neighbors time stamps from clocks and send it to $n_i$ (requested node)
3: node $n_i$ average from neighbor's clock readings and send the new value to all neighbors
4: neighbors adjust it's with new value
```

The overall style of time synchronization presented in this plan is as follow:

For each node with equal probability the following operations can be performed:

1. The first node asks all of its neighbors to send their timestamp values.
2. It averages the whole of the read time stamp values from the neighbors.
3. The new values will be sent back to neighboring nodes.
4. The neighbors synchronize themselves with the new value.

Now the threats that threaten the presented reformed synchronizing algorithm are going to be considered. But before that, the overall styles of the performance of the malicious or corrupted nodes are considered to determine how these nodes challenge a general time synchronization algorithm[15] ,[21] ,[22] ,[25] . In the following it will be checked that at each stage of the four phases of asynchronous algorithm, by using which strategy the malicious nodes can act and disable the algorithm. Then it will be determined that the proposed algorithm by which mechanism affords the attack of the malicious nodes and the fault of the corrupted nodes in the four main stages of the algorithm [7].

A. Malicious nodes: Can do their work in four ways[36]:

1. Send unacceptable times with a lot of errors (Message manipulation).
2. Send their values with delay [41] (Message Delay).
3. Pose themselves as another node and request for doing a work (masquerade attacks).
4. Suppose we have two modes: The first one, the density of the malicious nodes is around a requesting node and the malicious nodes with collusion send some values consistently with each other and bring about errors in the act of synchronizing of that

requesting node. The second one, the impossibility of collusion of the nodes (malicious nodes) for not having the required density around that synchronizing requesting node.

5. Malicious nodes continuously send synchronization requesting packet and continuously introduce themselves as a synchronizing requesting node.

6. Pose themselves as the other nodes and declare their own desired value. So we are faced with two values for this node.

B. The existing nodes themselves include some drawbacks that must be separated from the previous ones(A):

1. The nodes themselves have wrong values or a high error at the beginning of their work.

2. Some nodes have too many delays [33], [41].

The general approach of the proposed algorithm to solve the problem:

Each node (named the initial) sends a request to all of its neighbors and asks them to send it their own time values. Then the node for having the same time values as its neighboring nodes (in the first stage the same time value) suggests the value of time for itself and its neighbors. (We will talk about it to determine what value for the synchronization). After this process, as the neighboring nodes have gained the time mathematical equation from each other’s behavior, they have the ability to recognize the real values from the values sent by the malicious nodes. The nodes are also required after synchronizing their own amount of time with the proposed value of the initial node, (After the time that we specified in our algorithm) declare that have elapsed that amount of time. (This overhead also helps to detect whether we have realized the previous value of the initial node correctly or not and to reform the existing mathematical equation in memory too).Notice that the values the initial nodes declare after reading the values of their neighbors should consider some reservations. For example, they should offer some values in certain time periods (for example, integer values). Additionally the packets sent by the initial node include the minimum and the maximum delay and the neighboring nodes regarding these values and the data from before and passing the time period synchronize.

Note: Firstly, each initial node reads the whole of the values of its neighbors because the neighboring values are a matter of importance [17]. If we can suggest a value to the neighboring nodes to make one our own times, this offered time value is closer to the time values of the neighboring nodes. As the nodes include less difference than that proposed value, they can continue to their work for more time and with fewer errors, without needing synchronizing and without taking help from the Synchronization process.

The main stage is the stage where the value of time for synchronization is sent, the real challenge begins. It is a very important assumption that if the percentage of the malicious nodes is high the accuracy of synchronization will definitely decrease and nothing special can be done for this problem in this algorithm.

The general rule which is true in the case of asynchronous release algorithm is that as much as more information are sent, the algorithm will be more vulnerable against the malicious nodes. If we divide the algorithm and the probable behavior of the malicious nodes into the 4 stages, the influences of the malicious nodes and the strategies they apply should be evaluated in each stage.

First step: An existing node (the initial node) in the network sends the requesting packet for synchronizing by broadcasting to all of the nodes that are in the range of that node [18]. Then all of the nodes that are in the board of that node receive this packet. For example, if the node A (Fig.1) is the starter, the contents of the requesting packet, includes the characteristics of the node(A) and the number of which synchronization requests (k), (For example, if it is the third time that requests, it sends number three) which is shown in Fig.1. At this stage an attack that may take place is that the malicious node poses itself as some specific nodes and requests for synchronizing nonstop(For example the node M by characteristics of the node A sends constantly requesting packets to the node B and D).This problem could be considered by putting a schedule in the nodes themselves, in this way that each node depending on the amount of time that has done the action of synchronizing for each of the neighboring node, disregard the next request of that neighbor. (Nodes B and D have a schedule so that after passing for example the next two milliseconds, will respond to the requesting packet that comes from the node A).The problem this threat could create, is creating additional traffic load in the network [37].

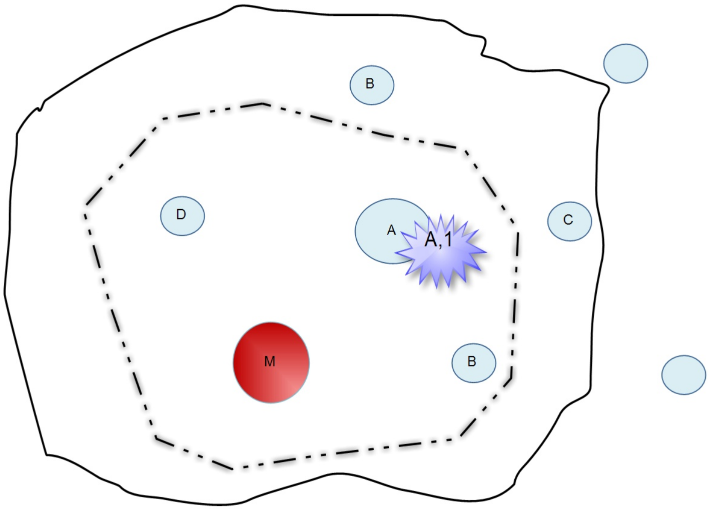

Figure1. Relating to the first stage of the fixed line

The second stage:(Fig.2)The neighboring node (which is shown by j and any node could neighbor) receives the request and keeps the receiving time (Node J has time stamped this value by $t_{j,\ 1}$) and sends the value of its time stamp in the packet to the requesting node (t_j, 2). Additionally it keeps both of the values in its memory for its neighbor. The threat of the malicious node is in common with the first stage, which is the threat of consistently requesting the fake nodes for synchronization .The possible solution was provided in the first stage as well. In addition, to have more security we must recognize our neighboring nodes, and for that it is only enough to know how many nodes we are neighbor with and what their features are. Notice that in the case of collusion, if the number of the nodes which collude is much more than the nodes of our network, the node will certainly go out of the synchronization process. In general, if a large number of nodes are colluding, the synchronization algorithm also continues to its work with a higher error and if there is no collusion and there is only one node, it’s recognizable and therefore our accuracy will be very high.

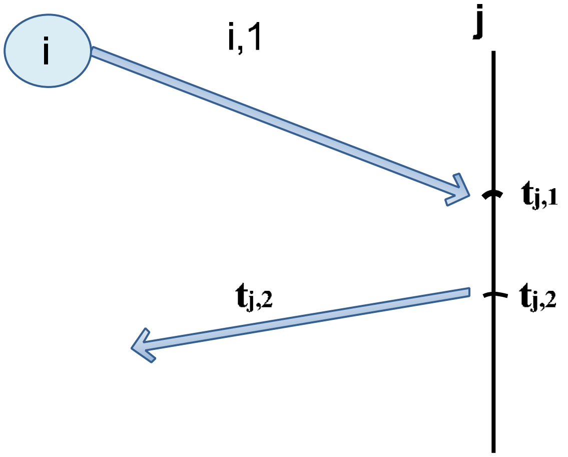

Figure2.time stamping by the neighboring node (j)

The third stage: (Fig.3) in the initial node (generally every initial node is displayed by I), all the values of the timestamps are earned from the neighboring nodes .They are then averaged, and the average become rounded down to the nearest acceptable amount of time. (For example, if the calculated average was equal to 3.4 and the acceptable time periods were integer, we put the average equal to 3) (1),(2). Each node time delay is also recorded [33], [41], so that the value which will come from the first node, will take the amount of zero.(fig.3: Node B) And its difference with the second reached amount will be considered a delay for the second node (for example, the value of delay for node C will be ($t_{i,\ rec,\ 2}$ - $t_{i,\ rec,\ 1}$) and this amount will be recorded in the memory of that node)(3).The delay of the nodes, will be also averaged (or the minimum and the maximum are recorded) and within the new packet the two obtained mean value and the feature of the requesting node are sent(4).Here there are two general types of threats from the malicious nodes [41]: (In addition, the amount of delays from taking the first value of the neighbor, shouldn’t be more the Max Delay, and if the timestamp value is gained after this period of time, we don’t take it into account). 1. Delays [41], [33] (few and many), 2.wrong time values (with high or low error). For high delays we have a threshold limit value that their values are not considered. We use the existing mathematical equation or disregard that value [25]. (Notice that in case of the high amount of time with high error we also use threshold limit, in addition two of the values which have the most difference are deleted as well). Note that any amount can be offered for synchronizing, but the average causes that the nodes synchronize for a more amount of time. Furthermore, the established (received) amount will be noted and for each node the mathematical equation will be written and updated [10].

$$mean = \frac{\sum_{k=1}^{n} t_{k,2}}{n} \tag{1}$$

Mean =Mean convert to nearest acceptable time (2)

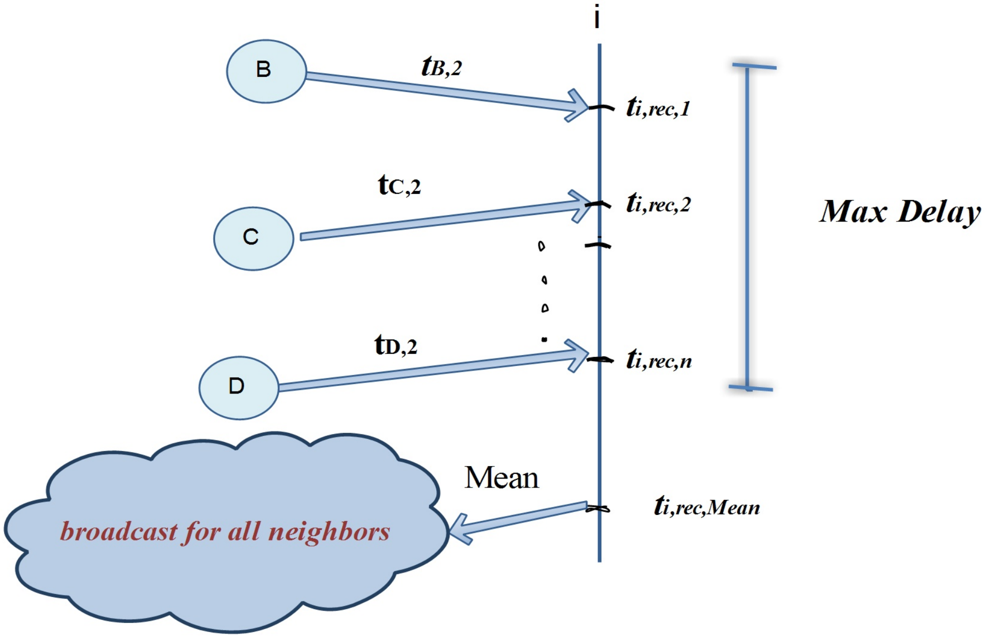

Figure 3. Taking values of the neighboring nodes by node i

$$T_{C,delay,K} = t_{i,rec,2} - t_{i,rec,1} \tag{3}$$

// Delay node C in k's level synchronization

$$T_{C,delay,Mean} = \frac{\sum_{i=1}^{k} T_{c,delay,i}}{k} \tag{4}$$

If two values arrive for a neighboring node (eg C) we choose a value for C that both the delay of its arriving and the value of θ(coefficient of deviation) calculated from it, is a value that is closer to the equation of mathematics we have in our memory (we use the impact of the coefficient of deviation more than delay) and use it [24],[32].The mathematical equation of the coefficient of deviation has been brought for the K-th request of synchronization: (5)

$$\theta_{i,c,k} = \frac{(t_{c,2,k} - t_{c,2,k-1})}{(t_{i,rec,c,k} - t_{i,rec,c,k-1})} \tag{5}$$

That both of the values in the numerator are the time stamps sent from the node C and in the denominator there are the time stamps for node C that have been stamped in node I [25] .

The fourth stage: (Fig.4)The packet sent from the requesting node (the initial node) reach the neighboring nodes of that node in a distributed way. Consider a case that we have multiple values that one of them is right and the others are wrong. Suppose you take three values with successive delays [33] that all three of them claim to be sent from the initial node and declare a value (a value that should now co-ordinate themselves with it) for the average of all of the nodes. The characteristics that should be considered for the average is that it should be in a certain time period, ie, they be supposedly integer, then the malicious nodes are also forced to send the integer values, so that to guess which value is correct will be more comfortable. Because regarding having an equation from the behavior of that node, it will be easier than before to choose the right value from between the sent values. While considering that the next time period that all nodes announce, for example the time of x can recognize the previous right value. Then here the right value in synchronization could be obtained through using two methods and its previous equations be updated constantly. Then when the node co-ordinated its value with the average declared of the initial node, has the duty to be sent after a time you specify(this time can be conventional or put in the packets themselves).Two equations are keptone for the average and the other for the neighboring nodes [25]. If the number of the nodes which collude is high, the percentage of success of the synchronizing algorithm declines (6).

$$L_j = \theta_j . H_j(t) + \Phi_j \tag{6}$$

- Neighboring node takes the values after this time period:(7)

$$Maxdelay + \delta(delaybetweensenderandreceiver) + t_{j,2} \tag{7}$$

After this stage, the value that takes from the initial node equalizes L$_j$ [10], [25]. Now we have to find a coefficient of deviation (skew) for the taken value, so that by putting it in accordance with the previous values of the coefficient of deviation we realize the reliability of that value [28] (8),(9) .

$$\theta_{j,i,k} = \frac{(T_{i,mean,k} - T_{i,mean,k-1})}{(t_{j,3,k} - t_{j,3,k-1})} \tag{8}$$

$$\begin{aligned}
&If\ \overline{\theta_{j,i,\min}} \langle \theta_{j,i,k} \langle \overline{\theta_{j,i,\max}} \\
&L_j = T_{mean,i} \\
&else \\
&L_j = \theta_{j,i,mean}.H_j(t)
\end{aligned} \tag{9}$$

$$\theta_{j,i,mean} = \frac{(\theta_{j,i,k} + \theta_{j,i,k-1} + \ldots + \theta_{j,i,1})}{k} \tag{10}$$

If we call the average of 10% of the minimum values of θ, $\overline{\theta}_{\min}$ and do this action also for the maximum (the average of 10 percent of higher values) we can use the above equation and consider the algorithm to be safe (10).

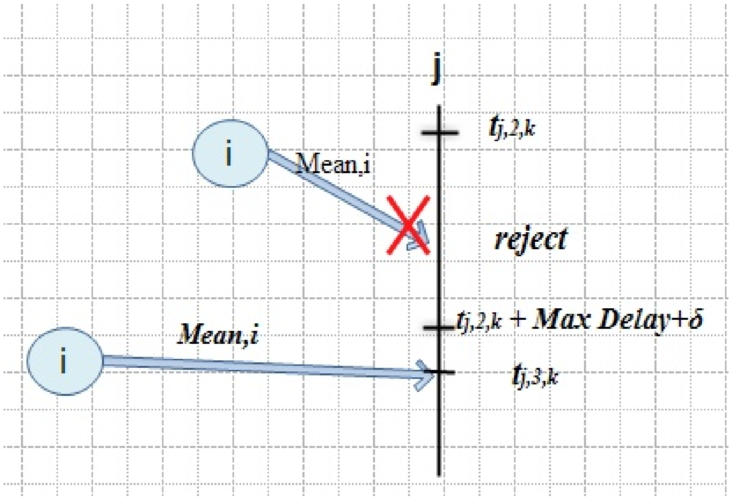

Figure 4.Ignoring the packets after a specified time

Suppose we want to review the fourth stage for the neighboring node B. The last action that this node does for the k-th act of synchronizing is to put the amount of $T_{B,\,2}$ in a packet and to send it. Now is waiting for the value that is sent from the initial node. Here the malicious node can as soon as the Node B finished its work in the second stage send constantly responding packets on behalf of the node A asit is the node A and the node B will face several values which claim to be from the node A. Here to make the node (B) resistant against this threat, we consider a rule for that which is to accept the values of A after the time period that the related packet has obtained in the equation 4. After that the neighboring nodes and the initial node value their time value with the new value (regarding the conditions mentioned above) then all of the neighboring nodes are required to declare after a period of time that have passed this period of time. (For example the node B after synchronizing its time value with the mean value of the node A is required to declare for example after half of a second that has passed this period of time. And with an advertising packet, all of the other neighboring nodes like C,D will do this work as well.) Now, when all of the nodes are obliged to do this work, it causes that the accuracy and the reliability of synchronization also rise. The reason for this is that any of the neighboring nodes takes multiple ads (declaration) from its other neighboring nodes and supposedly based on the first two cases of the neighbors that have arrived earlier make a decision (Fig.5).

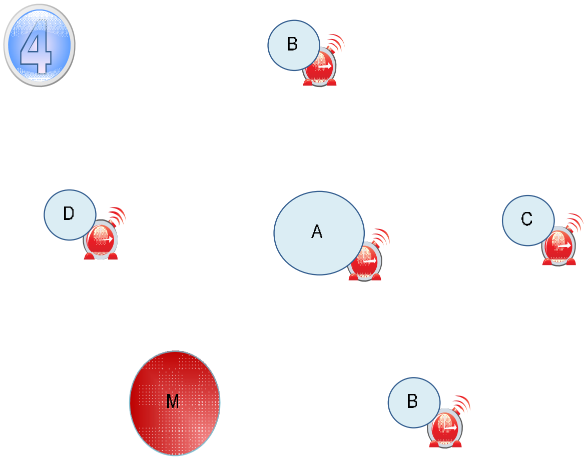

Figure 5. All nodes after the fourth stage

notice: All the nodes that have participated in sync export ads packets at the moment that we specify.

4. Simulations and Results

This part has been divided into two sections. In the first part we will attend to the overall simulation of asynchronous release algorithm and the impact of various parameters such as the density of nodes, the number of nodes and the board of nodes on the is algorithm[27] , and then we will analyze the gained graphs of simulation. In the second part, we will consider the simulation of the proposed safe algorithm and its performance comparison with asynchronous release algorithm.

In first part simulation has been performed by MATLAB software for a various number of nodes (100 to 900) and by changing some different parameters. The first parameter that has been considered is the parameter of density of the nodes. As much as the density of nods is greater in a range [26], the accuracy of synchronization also rises and it can be said that the accuracy of synchronization with the density of the nodes on a range correlates directly. One of the disadvantages that can be considered for this algorithm is that if the node density is low, the accuracy of the algorithm will almost decline exponentially. Fig.6 is expressive of this matter. In this figure we have done the act of simulation for 30 modes with the same parameters and the same first values, but with different board ranges. In the upper graph that has a more average deviation, the nodes include a less board in it. For example, if we consider the value of one for that and the value of two for the middle graph and the value of four for the lower graph, we will understand that by increasing the range of the board of the nodes the accuracy and speed of the convergence of the algorithm will almost increases twice.

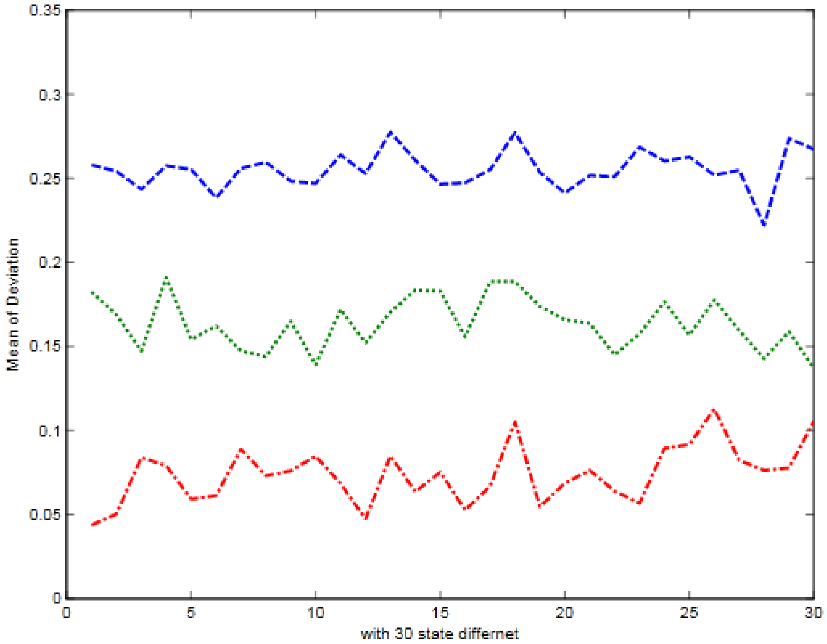

Figure 6.Proposed algorithm with 30 times running

It is important to note that as much as the range of the board of nodes increases, the accuracy of algorithm also rises [26].

The next matter (Fig.7) is the density of nodes in an environment where by increasing the density of nodes, the accuracy and speed of the convergence of the algorithm will also rises. In the following figure, we have tested 20 sets of data with some values of densities in the environment. As it was seen in any set, the state that includes a further density in the environment (more number of nodes in an environment with the same size) the accuracy and speed of the convergence of it rises as well.The density of the values of 1, 2, 4, 8, and 16 have been simulated, for example the brown set shows the value of density of 16, the orange 8 and so on, and finally the dark blue set shows the value of density of 1.

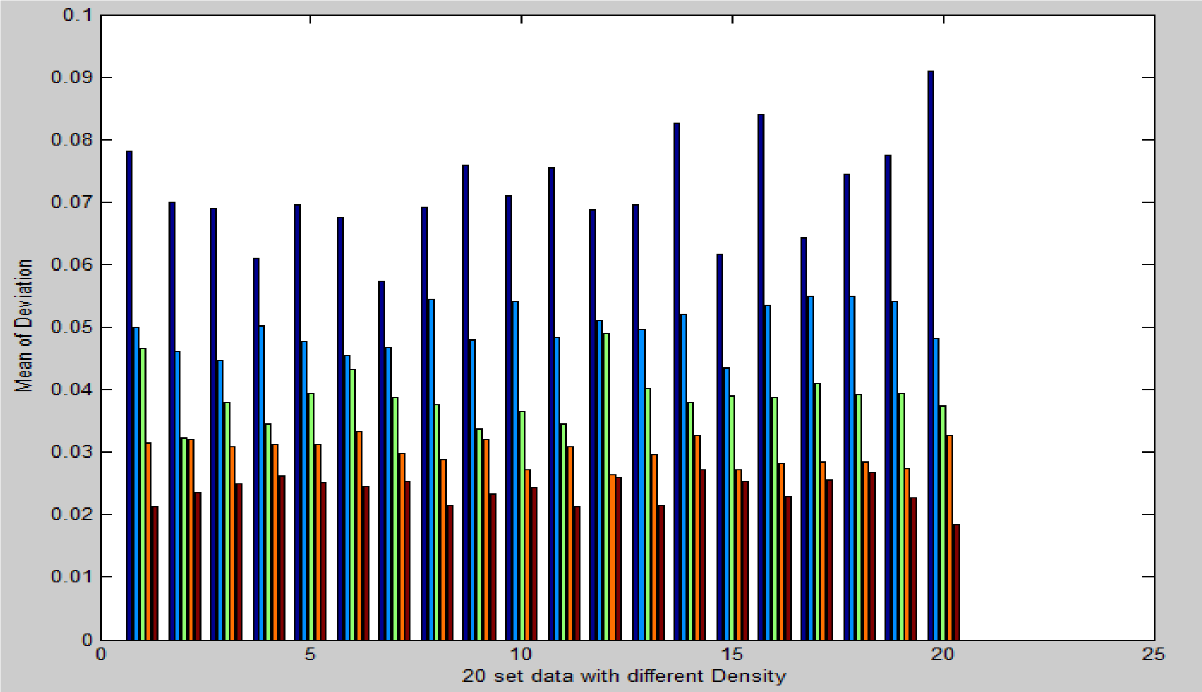

Figure 7.The comparison of the algorithm with different values of densities

When we perform the algorithm regardless of the expressed security issues, the average of deviations becomes much more than the cases that we take into account the security issues. There too much differences. The algorithm has been executed several times, and the results show the good performance of this algorithm.

In Fig.8 the algorithm has been performed 20 times for a set of data in two modes, one without performing the proposed securing algorithm and the other by performing the set of data by utilizing the proposed secured algorithm that by using the figure we can find out the overall average of deviations in each of the sets. This state has gone up to 1000 seconds. The number of nodes is 900.

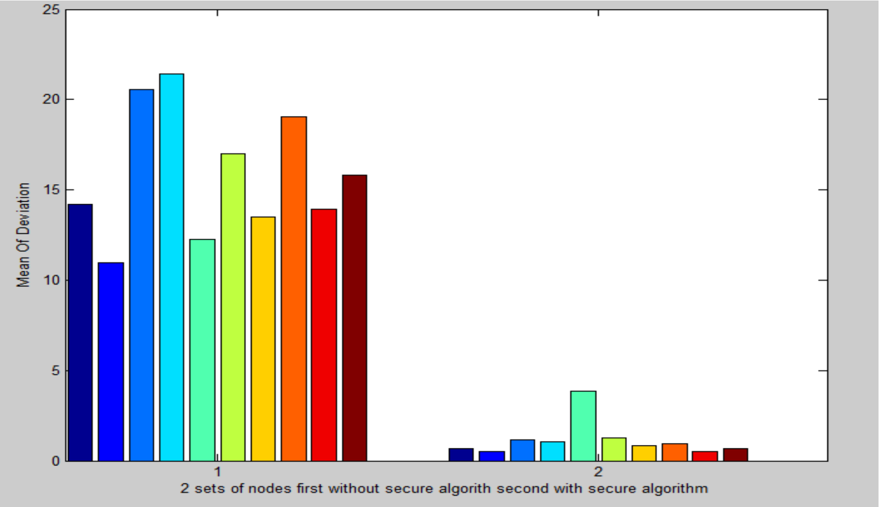

Figure 8.The comparison of the average of deviation between the two modes of the proposed secured algorithm

About the number of the executed instructions, if we assume that each node does calculations in a clock, it can be considered that the equation is done in one cpi that in an insecure mode these calculations don't exist. Fig.9 determines the number of calculations done based on the number of nodes for a secured mode that this graph represents the computational overhead in the proposed secured algorithm.

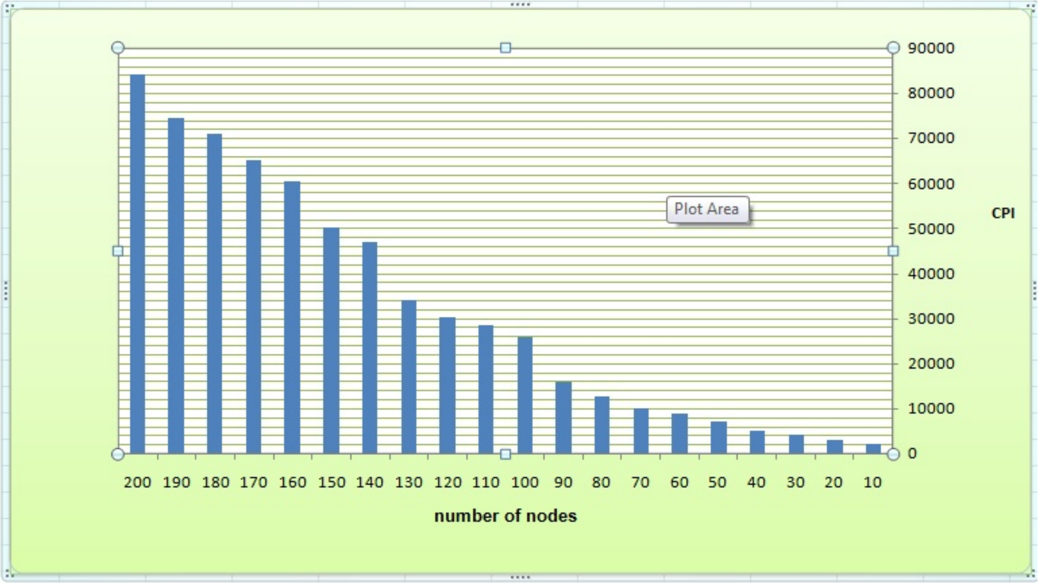

Figure 9. The value of the executed instructions performed.

In case of the speed of the convergence of nodes, to a common value, the Fig.10 can be used. The figure shows that the proposed secured algorithm synchronizes the nodes over time much faster and has a less difference standard average[26].The horizontal axis represents time based on seconds and the vertical axis represents the standard average difference. This simulation has been considered by taking into account 5 malicious nodes and 80 nodes in the environment. The horizontal axis has gone forward until the time of 73 and the vertical graph also shows the value of the deviation of the standard average which is seen that the secured algorithm includes a less standard deviation comparing to asynchronous algorithm.

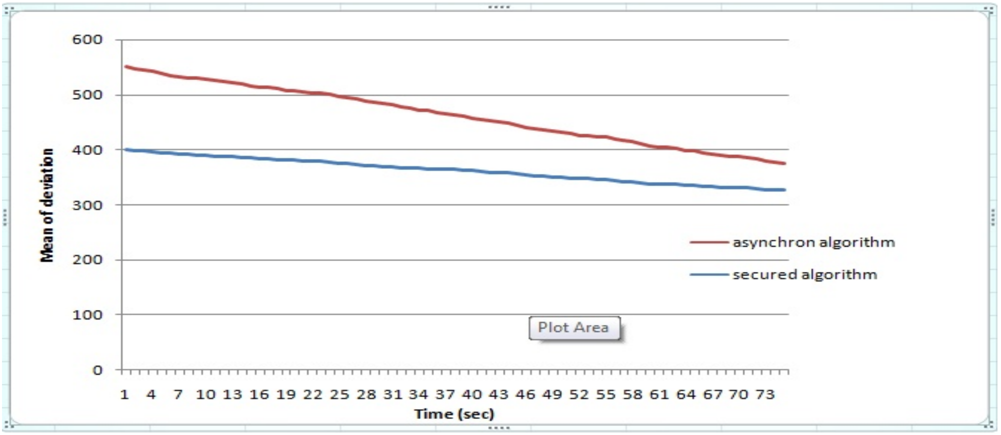

Figure 10.comparing the proposed secured algorithm

In the case of the overhead of memory[26], [30], consider that the complexity of memory is O(n) where in 'n' is the number of the nodes of the sensor network. In explaining this amount of the space of memory it must be said that each node must consider the approximate value of one kilo byte for each of its neighbors and if for example a node has five neighbors, that node will need five kilo byte memory.

5. Conclusions

In this paper, despite malevolent and malicious nodes, a safely distributed nationwide synchronize protocol is presented. This means that algorithm within itself by applying sum mechanisms detects malicious nodes and it does synchronization for global distribution.

Studies show that by increasing the range of nodes, the accuracy of the algorithm increases. Also, whatever the density of nodes is greater, the accuracy and speed of convergence of the algorithm is higher. When we run the proposed algorithm that considers security the deviation is much lower.

References

[1] J. Nieminen, L.Qian, and J.Riku,"Network-wide time synchronization in multi-channel wireless sensor networks", Wireless Sensor Network, 2011.

[2] S. Yoon, C.Veerarittiphan, and M.L.Sichitiu, "Tiny-sync: Tight time synchronization for wireless sensor networks", ACM Transactions on Sensor Networks, 2007.

[3] F.Kiani, andM.Fahim, "Energy Efficiency in Wireless Sensor and Actor Networks by Distributed Time SychnronizationAlgorithm", Computational Intelligence & Communication Technology (CICT),2015.

[4] J. Liu, "Scalable synchronization of clocks in wireless sensor networks", Ad Hoc Networks, 2008.

[5] S. Rahamatkar, A. Ajay, andN. Kumar, "Analysis and comparative study of clock synchronization schemes in wireless sensor networks",2010.

[6] I.K. Rhee, et al, "Clock synchronization in wireless sensor networks: An overview", Sensors, 2009.

[7] W. Guo, Y. Hua, and H.j. Ma, "Improving the security of time synchronization in WSN", Mechtronic and Embedded Systems and Applications, 2008.

[8] P.K. Singh, K. Sahu, and A. Kumar, "Clock Synchronization Protocols: Analysis and Comparisons in WSNs", International Journal of Science, Engineering and Computer Technology, 2013.

[9] C. Xu, et al, "Broadcast time synchronization algorithm for wireless sensor networks",Proceedings of the 1st International Conference on Sensing, Computing and Automation,2006.

[10] S. Yoon, andM.L. Sichitiu, "Analysis and performance evaluation of a time synchronization protocol for wireless sensor networks", The Int'l Conf on Telecommunication Systems, 2005.

[11] Y. Saravanos, "Energy-aware time synchronization in wireless sensor networks", Diss. University of North Texas, 2006.

[12] F. Fiorentin, andL. Schenato, "Average timesync: A consensus-based protocol for time synchronization in wireless sensor networks", IFAC Proceedings, 2009.

[13] Z. Dengchang, Z.An, and Y.Xu, "Time synchronization in wireless sensor networks using max and average consensus protocol", International Journal of Distributed Sensor Networks, 2013.

[14] D. Raskovic, O. Lewis, and D. Giessel, "Time synchronization for wireless sensor networks operating in extreme temperature conditions", Southeastern Symposium on System Theory, 2009.

[15] K. Sun, P. Ning, and C. Wang, "Secure and resilient clock synchronization in wireless sensor networks",Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks, 2007.

[16] R. Akl, andY.Saravanos, "Hybrid energy-aware synchronization algorithm in wireless sensor networks", Indoor and Mobile Radio Communications, 2007.

[17] P. Ranganathan, and K.Nygard, "Time synchronization in wireless sensor networks: a survey", International Journal of UbiComp, 2010.

[18] T. Brooks, H. Bakker, and K. Page,"A Review of Synchronization Methods in Wireless Sensor Networks", 2009.

[19] P. Sommer, and R. Wattenhofer, "Gradient clock synchronization in wireless sensor networks",Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, 2009.

[20] K. Shahzad, A. Arshad, and Gohar. N.D, "Etsp: An energy-efficient time synchronization protocol for wireless sensor networks", Advanced Information Networking and Applications-Workshops, 2008.

[21] J. Barnickel, and U. Meyer, "Secsywise: A secure time synchronization scheme in wireless sensor networks", International Conference on Ultra Modern Telecommunications & Workshops, 2009.

[22] B. Winkler, and T. Damla, "Secure time synchronization protocols for wireless sensor networks", IEEE Wireless Communications, 2007.

[23] V. Namboodiri, and S. Ramamoorthy, "Energy Efficient Global Clock Synchronization for Wireless Sensor Networks", 2004.

[24] F. Sivrikaya, and B.Yener. "Time synchronization in sensor networks: a survey." IEEE network 18.4 (2004): 45-50.

[25] S. Ganeriwal, et al, "Secure time synchronization in sensor networks", ACM Transactions on Information and System Security (TISSEC), 2008.

[26] S.K. Bae,"Power Consumption Analysis of Prominent Time Synchronization Protocols for Wireless Sensor Networks", JIPS, 2014.

[27] L. Zhou, J. Li, and L. Yang, "Improvement of Mechanisms for Network Time Synchronization Algorithm Based on Wireless Sensor Network", International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME), 2015.

[28] W. Liu, and W. Zhi, "On the Joint Synchronization of Clock Offset and Skew in PBS-Protocol", 2015.

[29] I. Sari, et al, "On the joint synchronization of clock offset and skew in RBS-protocol",IEEE Transactions on Communications, 2008.

[30] C. Benzaïd, M. Bagaa, and M. Younis, "An efficient clock synchronization protocol for wireless sensor networks", International Wireless Communications and Mobile Computing Conference (IWCMC), 2014.

[31] Q. Li, and D. Rus."Global clock synchronization in sensor networks", IEEE Transactions on computers, 2006.

[32] A. Raghuvanshi,A. Kumar, G. Krishna Yadav,Analysis of Time Synchronization Protocols for Wireless Sensor Networks: A Survey", International Journal of Computer Science and Mobile ComputingA Monthly Journal of Computer Science and Information Technology,2015.

[33] Y.P. Tian, and S. Zong, "Average TimeSync with bounded random time-delay is almost divergent", Control Conference (CCC), 2014.

[34] D. Djenouri,and M.Bagaa. "Implementation of high precision synchronization protocols in wireless sensor networks",Wireless and Optical Communication Conference (WOCC), 2014.

[35] A.R. Swain, and R.C. Hansdah, "An energy efficient and fault-tolerant clock synchronization protocol for wireless sensor networks", Second International Conference on COMmunication Systems and NETworks (COMSNETS), 2010.

[36] D.S. Mantri, N.R. Prasad, and R. Prasad, "Synchronized Data Aggregation for Wireless Sensor Network", Wireless Computing and Networking (GCWCN), 2014.

[37] S. Prakash, et al, "A Novel Time Synchronization Recursive Algorithm (TSRA) in WSN", Proc. of Int. Conf. on Advances in Communication, Network, and Computing, 2014.

[38] S. Watwe, A. Bhatia, and R. C. Hansdah, "A design for performance improvement of clock synchronization in WSNs using a TDMA-based MAC protocol", Advanced Information Networking and Applications Workshops (WAINA), 2014.

[39] M.K. Maggs, S.G. O'Keefe, and D.V. Thiel, "Consensus clock synchronization for wireless sensor networks", IEEE sensors Journal, 2012.

[40] J. Elson, and D. Estrin, "Time synchronization for wireless sensor networks", IPDPS, 2001.

[41] S. Ping, "Delay measurement time synchronization for wireless sensor networks", Intel Research Berkeley Lab 6, 2003.