An Energy Efficient and Trusted Data Fusion by using Cellular Automata in Wireless Sensor Networks

Shaghayegh Jaberi
Department of Computer Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran
sh.jaberi@gmail.com

Amir Masoud Rahmani
Department of Computer Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran
rahmani@sriau.ac.ir

***Abstract*—Wireless sensor networks are becoming more and more common. One of the limitations of wireless sensor nodes is their inherent limited energy resource. Besides maximizing the lifetime of the sensor node, it is preferable to increase the trust value of data fusion results. In this paper, a new protocol is introduced, named EETDFCA (an Energy Efficient and Trusted Data Fusion by using Cellular Automata) in Wireless sensor Networks. EETDFCA uses cellular automata rule to find the most suitable cluster head, perform data fusion, find the most trusted neighbors for sending the fusion result to base station, and transforms from current state to a new state. The network is intended for the long-term monitoring of packets produced by jammer nodes. The data flow of the network is mainly toward a cluster head node, which is responsible for collecting data generated by sensor nodes. When the network is first deployed, an initialization algorithm is performed and preliminary clusters, cluster heads and sensors alive are determined. Simulations and results show that the algorithm can extend the lifetime of the wireless sensor network and boost trusted data fusion frequency.**

Keywords: Wireless Sensor Network; Energy Efficient; Cellular Automata; Trust value; clustering.

I. INTRODUCTION

Wireless sensor networks (WSNs) have been used increasingly in every type of environment due to their ease of deployment. WSNs provide their users which fast and easy access to their data and services anytime and anywhere, especially in remote area such as battlefield, forest and volcano. WSNs have limitations such as: limited energy resources, battery life, computation, and communication capacities, and high cost of transmission. All of these characteristics of wireless sensor networks are complete opposites of their wired network counterparts, in which energy consumption is not an issue, transmission cost is relatively cheap, and the network nodes have plenty of processing capabilities.

In addition, as in many other kinds of network or communications system, the data and services provided require protection. However, WSNs are more vulnerable to security attacks than other traditional networks and Ad Hoc networks due to their unattended nature. For example, an adversary can physically capture some nodes and use them to inject faulty or false data into the network system disturbing the normal cooperation among nodes. Cryptographic and authentication mechanisms, such as TinySec [1] and TinyPK [2], alone cannot be used to solve this problem as internal adversarial nodes will have access to valid cryptographic keys. Moreover, in WSNs, the function of data fusion is mostly shown on saving energy, improving data collection efficiency, enhancing data accuracy, and getting synthesis information [3]. The typical data fusion algorithms, such as Data Funneling, AIDA [4], TAG [5], and TINA [6], do not pay attention to correctness of received data. In addition, clustering is another way to reduce the energy consumption in WSNs. By associating each sensor to one cluster, sensor sends its sensed data to the cluster head of that cluster. Therefore, instead of each sensor sending its message to base station, it sends it to the cluster head; so clustering leads to reducing the message conveyed on the network.

This paper uses cluster heads as fusion nodes and combines data fusion with trust calculation by using a secure data fusion algorithm based on its neighbors' behavioral trust. In addition, selecting cluster head for each round and seeking next-hop after calculating the fusion results are based on cellular automata's rule. Moreover, each cluster head punishes or motivates its neighbors based on their activity and sending accurate data. Moreover ,it lead to sensors transform from its current state to a new state based upon its current state and the states of its neighbors, according to cellular automata's rule to achieve the energy efficiency which is the main purpose of this paper . In Section II, briefly describes the related work about energy efficient data fusion protocols for WSNs and existing network simulators. Section III presents brief description of the network model. Section IV presents the network algorithms. Suggested protocol is discussed in this section. Sections V and VI present results of simulation and conclusion, respectively.

II. RELATED WORKS

Wireless sensor networks have attracted a plethora of research efforts due to their vast potential applications [7]. In particular, extensive research work has been devoted to providing energy efficient routing algorithms for data gathering [8-12]. While some of these approaches assume statistically independent information and have developed shortest path tree based routing strategies, others have considered the more realistic case of correlated data gathering [9-11]. By exploring data correlation and employing in-network processing, redundancy among sensed data can be curtailed and hence the network load can be reduced [8]. The objective of sensor routing algorithms is then to jointly explore the data structure and network topology to provide the optimal strategy for data gathering with as minimum energy as possible

Then it is a critical consideration to collect and fuse sensed information in an energy efficient manner for obtaining a long lifetime of the sensor network. Based on researchers' findings that the conventional methods of direct transmission, shortest path routing, and Dempster-Shafer tool may not be optimal for data fusion of sensor networks, in [13] Low-Energy Event Centric Fusion (LEECF) is proposed, a event-centric-based protocol that utilizes the centric sensor node to aggregate the event data among the triggered sensors in a short delay. LEECF incorporates a fast information fusion into the routing protocol to reduce the amount of information that must be transmitted to the sink and the time complexity of fusion computation of fusion center. Simulations show that LEECF can decrease the energy and fusion time significantly compared with conventional routing protocols.

Many routing, power management, and data dissemination protocols have been specially designed for WSNs where energy awareness is an essential design issue. The focus, however, has been given to the routing protocols which might differ depending on the application and network architecture In general; routing in WSNs can be divided into flat-based routing, hierarchical-based routing, and location-based routing depending on the network structure. In flat-based routing, all nodes are typically assigned equal roles or functionality. In hierarchical-based routing, however, nodes will play different roles in the network. In location-based routing, sensor nodes' positions are exploited to route data in the network. A routing protocol is considered adaptive if certain system parameters can be controlled in order to adapt to the current network conditions and available energy levels. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, QoS-based, or coherent-based routing techniques depending on the protocol operation. In addition to the above, routing protocols can be classified into three categories, namely, proactive, reactive, and hybrid protocols depending on how the source finds a route to the destination. In proactive protocols, all routes are computed before they are really needed, while in reactive protocols, routes are computed on demand. Hybrid protocols use a combination of these two ideas. When sensor nodes are static, it is preferable to have table driven routing protocols rather than using reactive protocols. A significant amount of energy is used in route discovery and setup of reactive protocols. Another class of routing protocols is called the cooperative routing protocols. In cooperative routing, nodes send data to a central node where data can be aggregated and may be subject to further processing, hence reducing route cost in terms of energy use. Many other protocols rely on timing and position information. We also shed some light on these types of protocols in this paper. In order to streamline this survey, we use a classification according to the network structure and protocol operation (routing criteria).

Currently, Agent-based Modeling and Simulation is the only paradigm which allows the simulation of even complex behavior in the environments of Wireless sensors [14]. Simulators like QualNet [15], OPNET Modeler [16], NetSim [17] and NS2 [18] can be used to simulate Wireless Sensor Networks. Other simulators, like IDEA1 – based on SystemC – have hardware-level libraries that permits system-level simulations by taking low-level constraints into account. NS (from network simulator) is a name for series of discrete event network simulators, specifically NS2 and NS3. Both simulators are used in the simulation of routing protocols, among others, and are heavily used in ad-hoc networking research, and support popular network protocols, offering simulation results for wired and wireless networks alike then it is suitable to write CA rules, define different types of nodes that are used in this simulation, and simulate new routing protocol.

III. NETWORK MODEL

The network consists of four types of nodes:

- Sensor nodes: These fixed nodes collect data using their sensors. The collected data are then passed to the cluster head nodes through the network. There may be as many sensor nodes as needed depending on the area to be covered.
- Cluster Head nodes: Fixed cluster head, which has the same design as sensor node, just do some extra activities. These nodes are responsible for collecting data from sensor nodes, performing data fusion algorithm on them, and producing fusion result on their own cluster; run the Find Routing algorithm to find its most trusted neighbor for sending fusion result; and in the end perform Find Cluster Head algorithm to determine new cluster head for later round.
- Jammer nodes: These mobile nodes produce disturbed packet throughout the network and are known as intruders.
- Base station: Data from each cluster are gathered and transferred to a central base station.

The sensor nodes, which receive disturbed packet that is produced by jammer nodes, send a data packet to their cluster head. By using the received data packets, cluster heads make decision about the existence of jammer node and send the final result to base station. The cluster heads are connected to the base station in a hierarchical manner.

Each cluster head send the final result about existence of jammer to its trusted neighbor to send it to the nearest cluster head to base station.

IV. ENERGY EFFICIENT AND TRUSTED DATA FUSION BY USING CELLULAR AUTOMATA (EETDFCA) PROTOCOL

In this section, Trusted Data Fusion by using Cellular Automata (TDFCA) [19] is introduced in terms to increase the trusted data fusion frequency; as can be seen in [19] by using TDFCA WSN has more coverage, longer life time, and greater trusted data fusion in comparison to BCDCP [20]; by using TDFCA,WSN's trusted data fusion frequency will be greater than using Secure Data Fusion Algorithm Based on Behavior Trust [3] scenario; and finally network's lifetime, End-to-end delay, and Packet delivery ratio noticeably decreases, using AODV instead of the TDECA. Therefore, although TDFCA increases the trusted data fusion frequency, it has the higher priority on decrease network's lifetime. In this paper, a new protocol is introduced, named EETDFCA (Energy Efficient and Trusted Data Fusion by using Cellular Automata) in Wireless sensor Network which its main purpose is to extend the network's lifetime and address the main problem of TDFCA protocol (pseudo-codes of the first three algorithms are available in [19]).

A. *Initialization*

Since the network in consideration is an ad hoc network, an initialization algorithm is needed to establish preliminary connections autonomously. The algorithm is based on polling and as such it guarantees connectivity to all the nodes that are acoustically reachable by at least one of their nearest neighbors. During initialization, the nodes create *neighbor tables*. These tables contain a list of each node's neighbors and the premiere clusters and cluster head. The initialization steps can be listed as follows:

All sensor nodes are alive and broadcast a find_premiere_cluster head packet to all the sensors in its transmitted radius.

Each sensor has a variable named *#neighbors*. By receiving find_premiere_cluster head packet adds one to this variable and adds a new row to its *neighbor* table. This table has three columns, the first one is the id of its neighbors filling by the ID field of receiving packet, which contain sender node's x-axis and y-axis, second column is a place to save the data sent by neighbors, and the last one is for saving $normalized_{trustvalue}$. In the beginning of the simulation, the data are set to -1 and the $normalized_{trustvalue}$ set to 1. If the sensor does not receive any packet for ΔT second, then the next step is performed.

Each sensor must calculate the density and distance to the base station by using the following equations:

$$density = (\#neighbors)/2r^2 \qquad (1)$$

$$distance = \sqrt{(pos_x - b.s_x)^2 + (pos_y - b.s_y)^2} \qquad (2)$$

Where r is transmitted radius, pos_x is the x_axis and pos_y is the y_axis of current sensor, $b.s_x$ is the x_axis and $b.s_y$ is the y_axis of base station. All sensors know base station's x_axis and y_axis. Then, calculate apt, which shows its convenience to be cluster head.

$$apt = \alpha_1\, density + 1/\beta_1\ distance + \mu_1\, trst\text{-}val \qquad (3)$$

Where α_1, β_1 and μ_1 are empirical coefficients that define the simulation. The *trst_val* is trust value of sensor node, which is evaluated by using (4) that is explained in following section.

Then, the result of (3) sends by a packet, whose name is find_max_apt to all neighbors of a sensor. By receiving this packet, each sensor performs the Find Cluster Head algorithm to determine whether or not it is a candidate of being this round cluster head. This algorithm is based on cellular automata (CA) rule and makes a decision locally by determining a flag, namely ch_flag. Therefore, it runs for about ΔT second after receiving the first packet.

By receiving the ch_announce packet, each sensor saves the new cluster head's ID.

B. *Data Fusion*

When a sensor node detects the intruder noise, it will compare its $normalized_{trustvalue}$ with normal_trustvalue of its neighbors; if its $normalized_{trustvalue}$ is greater than majority of $normalized_{trustvalue}$, then it generates data packet and sends it to the cluster head (CA rule).

When the cluster head gets this packet, first it updates the $normalized_{trustvalue}$ of the sender neighbor and then performs Data Fusion algorithm to make its cluster decision, which determines whether its cluster detects the intruder noise or not and sends the final decision to base station.

Data Fusion algorithm has two phases, first get the $normalized_{trustvalue}$ of sender node, which is evaluated by using (4) and (5) in individual sensor nodes as follows:

$$trst\text{-}val = \alpha_2\, r\text{-}power + \beta_2\, ch_t + \delta \qquad (4)$$

$$normalized_{trustvalue} = (trst\text{-}val)/(\delta + \alpha_2 r\text{-}power) \qquad (5)$$

where α_2 and β_2 are empirical coefficients that define the simulation; δ is motivation or punishment coefficient of the cluster head after assessing the final_result and compares it with data sender data, then motivates them if they send the same data as final_result and otherwise punishes them; *r_power* is amount of remaining power to maximum amount of power, (6) divided to maximum amount of power of sensor node; and finally, ch_t is the number of times that this sensor is selected as a cluster head.

$$E_t(b,d_2) = E_{tx} b + b\epsilon_1 d_2^2 + processing\ energy \qquad (6)$$

That b is number of bit sent or received by a sensor, d_2 is distance between the sensor and cluster head, E_{tx} is constant determined by simulation, and ϵ_1 is transmit amplifier.

Cluster head multiplies the received data by $normalized_{trustvalue}$. Then, for evaluating that, the fusion result is trusted or not, compares the result of first step to threshold, TL [21].

Therefore, cluster head makes decision and then must create final_result packet and send it to base station using multi hop strategy and make pun_mot packet and send it to sender sensor to punish or motivate them.

At the end of each round of data fusion, it must compare the amount of remaining power with the lower amount of power that requires to do cluster head duties. If the remaining power, assessed by (7), is less than the amount required to cluster head's duties or is equal to zero, the Find Cluster Head algorithm must be performed.

$$E_t(b,d_1)=bE_{tx}x+b\ E_{da}\ x+b\ \epsilon_2\ d_1^4+processing\ energy \quad (7)$$

That E_{da} is data aggregation energy, x is number of received data, d_1 is distance between cluster head, and next hop will be detected in next part.

C. Candidate trusted neighbor to receive data fusion results

The other crucial issue is finding the most trustable sensor for conveying the final_result from cluster head to base station. Each sensor, which is nearer to base station in comparison with its cluster head finds the maximum $normalized_{trustvalue}$ among its neighbors, compares it with its own value if its $normalized_{trustvalue}$ is bigger than maximum value, then candidates itself as next hop to receive fusion result (CA rule).

Therefore, the most trustable receiver node is available for each sensor.

D. Update Trust value

By receiving a pun_mot packet, changing the remaining power or the number of times that this sensor is selected as a cluster head, or the parameters that play a pivotal role in trust value, each sensor must update its $\text{normalized}_{\text{trustvalue}}$ using formulas (4) and (5). Whenever the value of $normalized_{trustvalue}$ would be changed, this new value must be broadcasted urgently. Then, its neighbors replace old value of $normalized_{trustvalue}$ with received $normalized_{trustvalue}$ in their neighbor lists. Therefore, the trust values in their list are always updated.

E. Make decision about transforming state

As mentioned before to extend the network's lifetime and increase the number of active nodes in wireless sensor networks, this new protocol is introduced. The assumption is each sensor has two states one is alive, which sensor sends or receive packets and performs the entire algorithms (clustering, data fusion and routing) and the other is standby , which sensor does not send any packet just received them and discards all the packet except for packets contain updated $normalized_{trustvalue}$ -after receiving this packets sensor updates its neighbor tables, therefore this table always is up to date -, ch_announce packets- by receiving these packets sensor changes cluster head axis, then the sensor always knows cluster head and cluster that belongs to -, and standby packets, which indicate that one of its neighbors transforms from alive state to a standby state.

At the beginning of simulation all the sensors are alive, after the first change is happened in its $normalized_{trustvalue}$; they execute change state algorithm, which is indicated in Table I. The sensor will compare its $normalized_{trustvalue}$ by $normalized_{trustvalue}$ of its neighbors; if its $normalized_{trustvalue}$ value is less than majority of $normalized_{trustvalue}$, then it transforms from alive to a standby state (CA rule). It will remain in this state and updates the $normalized_{trustvalue}$ of its neighbors and cluster head, which is belong to until it receives the standby packet, which indicates that one of its neighbors transforms from alive state to a standby state then it must perform change state algorithm again.

Each sensor, the state of which is set to alive must compare its $normalized_{trustvalue}$ by its neighbors whenever changes is happened to its $normalized_{trustvalue}$ or in neighbor tables ; if its $normalized_{trustvalue}$ value is less than majority of $normalized_{trustvalue}$, then it transforms from alive to a standby state (CA rule) and send standby packet to its neighbors.

TABLE I. CHANGING STATE

```
Data : normalized_trustvalue.i and neighbor table_i
Result : changing sensor state
if change is happened in normalized_trustvaluei or neighbor table_i then
    if normalized_trustvalue.i < = (∑ normalized_trustvalue.neighbors/2) then
        sensor_state = standby;
    else
        sensor_state = alive;
    end
end
if standby packet is received then
    if sensor_state == alive then
        do nothing;
    else
        if normalized_trustvalue.i < = (∑ normalized_trustvalue.neighbors /2) then
            do noting;
        else
            sensor_state = alive;
        end
    end
end
```

By using this algorithm, the amounts of consumption energy is decreased and network's lifetime is extended.

V. SIMULATION RESULTS

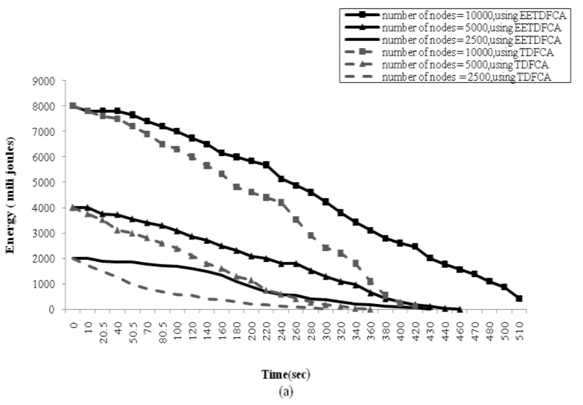
In what follows, results of the new protocol will be described by using NS2 simulator. First, it is necessary to use some assumptions: the network has N fixed sensors that propagate in the area that is L meter square (L×L m^2). Sensors used here also have the same structures and attributes, and then the network is homogeneous and they are trustable and do not show any intruding behaviors. Two different scenarios are used for evaluating this proposed protocol. Then, the new protocol will be assessed in tow region whose scales are 100×100 m^2 (having 10,000, 5000, and 2500 sensors) and 250×250 m^2 (having 62,500, 31,250, and 15,625 sensors). The sensors have a power, which charges up 0.8j, have a timer that is set to 200 ms, and the maximum transmission radius is about 6m (R_{max}=6m). This radio transmission updates by using (8) whenever the remaining power evaluates; in this formula α is coefficient.

$$r\_power = \alpha R_{max} \quad (8)$$

EETDFCA with TDFCA will be compared in terms of three parameters: total energy, coverage, and trusted data fusion frequency.

A. Total Energy

As mentioned above, one of the crucial problem in TDFCA is network's lifetime, which is noticeably decreased because of massive amount of packets sending and receiving by sensor node; Using EETDFCA instead of TDFCA owing to fewer energy consumption by changing the state of sensor node to standby in which the sensor just receives packets and do not participate in sending them. Figure 1 shows the total energy in network. As can be seen in Fig. 1, in all different scenarios EETDFCA has more energy in comparison with TDFCA this fact end in longer networks' lifetime by using changing state algorithm. Figure 1 shows that the effectiveness in teams of energy consumption in EETDFCA protocol is 1.2 times greater. Therefore, it seems that EETDFCA almost achieves to extend network's lifetime.



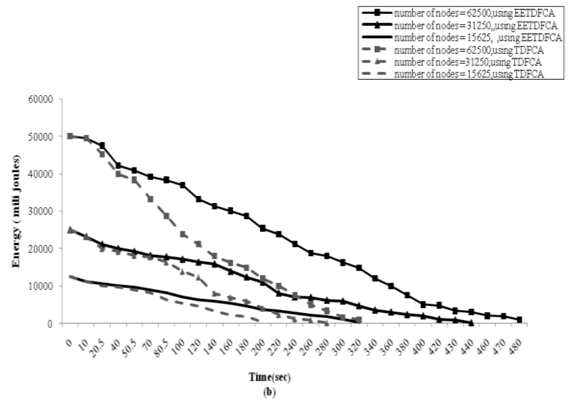
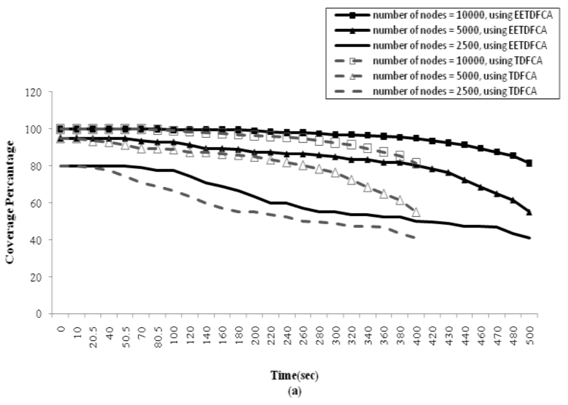


Figure 1. Remaining energy in the network (a) in first scenario (100*100) (b) in second scenario (250*250) EETDFCA in comparison with TDFCA

B. Coverage

As mentioned above in the proposed protocol, changing remaining power of a sensor lead to calculation of transmission radius. Therefore, by using EETDFCA changes are happening in the remaining power of a sensor seem to be happening at less slow a pace; then it has more coverage and can keep it longer in comparison with TDFCA that does not pay attention to sensor's state, then to some extend the coverage decreased. As can be seen, figures Fig. 2(a) and Fig. 2(b) are the same. This similarity leads to the fact that the scale of region does not have direct effect on coverage.



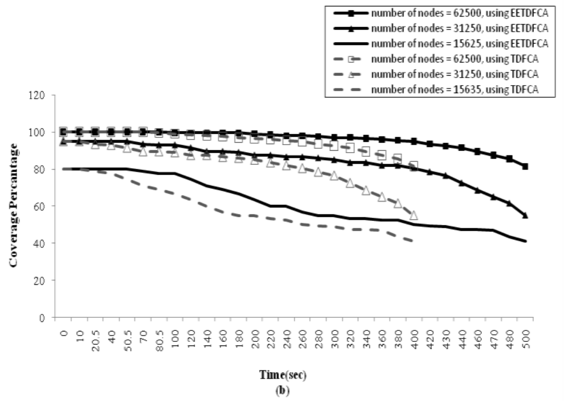


Figure 2. Relationship between coverage and time (a) in first scenario (100*100) (b) in second scenario (250*250) EETDFCA in comparison with TDFCA

C. Trusted data fusion frequency

Whatever data sent to cluster head have higher trust value, the fusion result will be more trusted. By using the both protocols, only when the sensors having higher trust value will send their data to the cluster head; then data fusion is more trusted. Both protocols use this method. Then, the difference between these two protocols is related to standby sate, which has no knock on effect on the trusted fusion results. Similar to coverage, trusted data fusion frequency is same for all regions with different scales, hence just one scenario is shown, Fig. 3. As can be seen in [4] trusted data fusion frequency dramatically increase in comparison with

BCDCP and Secure Data Fusion Algorithm Based on Behavior Trust.

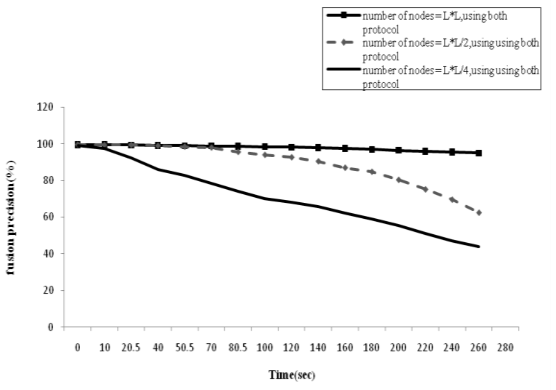


Figure 3. Trusted Data Fusion frequency in all scenarios

VI. CONCLUSION ND FUTURE WORKS

The suggested protocol, named EETDFCA, based on Cellular Automata for a wireless sensor network was tested by NS2. The simulation results show that EETDFCA protocol expands the network's lifetime, and coverage dramatically, and has no effect on trust to fusion result in comparison with TDFCA, in wireless sensor networks simultaneously.

The importance of decreasing energy consumptions and increasing trust value, especially in data fusion, in WSNs leads to various works on it. Then, the following changes are suggested for this protocol to assess its efficiency:

1. Implement this protocol in immobile WSNs.
2. Implement this protocol in WSNs whose their sensors have intruding behavior.
3. Using other types of cellular automata instead of totalistic ones, which are used in this paper, to evaluate this protocol.

REFERENCES

[1] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," Proc. second ACM Conference on Embedded Networked Sensor Systems(SensSys 2004), Nov. 2004, pp. 162-175, doi:10.1145/1031495.1031515.

[2] R. Watro, D. Kong, S.F. Cuti, C. Gardiner, C. Lynn, and P. Kurus. "TinyPK: secure sensor networks with public key technology," , Proc. 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Oct. 2004, pp. 59-64, doi: 10.1145/1029102.1029113.

[3] Z. Cheng, Z. Ming-zheng, X. Jin-sheng, and Y. Qing, "A secure data fusion algorithm based on behavior trust in wireless sensor networks security," 5th International Conference on Wireless Communications, Networking and Mobile Computing(Wicom 08), Oct. 2008, pp. 1-4, doi: 10.1109/WiCom.2008.1106.

[4] T. He, B. M.Blum, J. A.Stankovic, and T. Abdelzaher, "AIDA: Adaptive Application Independent Data Aggregation in Wireless Sensor Networks," ACM Transaction on Embedded Computing System (TECS), vol. 3, May 2004, pp. 426-457, doi: 10.1145/993396.993406.

[5] S. Madden, M. J . Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks," ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, vol. 36 , Dec. 2002, doi : 10.1145/844128.844142.

[6] M.A. Sharaf, J. Beaver, A. Labrinidis, and P. K.Chrysanthis, "TINA: A Scheme for Temporal Coherency Aware in Network Aggregation," Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access(MobiDE03),Sep 2003, pp. 69-76, doi : 10.1145/940923.940937.

[7] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," Proceedings of the IEEE, Aug. 2003,vol. 91, pp. 1247-1256, doi: 10.1109/JPROC.2003.814918.

[8] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of data aggregation in wireless sensor networks," the 22nd International Conference on Distributed Computing System (ICDCSW), July 2002, pp. 575-578.

[9] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," Information Processing in Sensor Networks 2004 (IPSN'04), Apr. 2004, pp. 28-35, doi: 10.1109/IPSN.2004.1307320

[10] W. Zhang and G. Cao, "Dctc: Dynamic convoy tree-based collaboration for target tracking in sensor networks," IEEE Transactions on Wireless Communication, Sept. 2004, vol. 3, pp. 1685–1701, doi: 10.1109/TWC.2004.833443.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Transactions on Networking, Feb. 2003, vol. 11, pp. 2-16, doi: 10.1109/TNET.2002.808417.

[12] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," INFOCOM 2004, Mar. 2004, pp. 2571-2582, doi: 10.1109/INFCOM.2004.1354677.

[13] B. Zeng, J. Wei, and T. Hu, "An energy-efficient fusion protocol for wireless sensor network ," 10th International Conference on Information Fusion , july 2007, pp. 1-7, doi: 10.1109/ICIF4408009.

[14] M.A. Niazi and A. Hussain, "A Novel Agent-Based Simulation Framework for Sensing in Complex Adaptive Environments," Sensor Journal IEEE, vol. 11, Nov. 2010, pp. 404-412, doi: 10.1109/JSEN.2010.2068044.

[15] QualNet manual web page available at http://www.scalable-netwoks.com/products/qualnet/, Last access: june 2011 .

[16] Opnet manual webpage available at http://www.opnet.com/, Last acces: june 2011.

[17] NetSim manual webpage available at http://haemgen.haem.cam.ac.uk/netsim/ , Last access: june 2011.

[18] NS manual webpage available at http://www.isi.edu/nsnam/ns/, Last access: june 2011.

[19] Sh. Jaberi, A.M. Rahmani, and A. Khadem Zadeh, "Trusted data fusion by using cellular automata in wireless sensor networks," unpublished.

[20] L. LI, S. Dong, and X. Wen," An Energy Efficient Clustering Routing Algorithm for Wireless Sensor Networks," The Journal of China Universities of Posts and Telecommunications, vol. 13, Sep. 2006. pp. 71-75 , doi : 10.1016/S1005-8885(07)60015-6.

[21] S. Sahnis and X. Chun Xu, "Algorithm for wireless sensor networks,"International Journal of Distributed Sensor Netwoks,vol. 1, Sep. 2004, pp. 35-56,doi= 10.1080/15501320490886323.

[22] A. Stauffer and M. Sipper ,"Biomorphs implemented as a data and signals cellular automaton," European Conference on Artificial Life, vol. 2801/2003, 2003, pp. 724-732, doi = 10.1007/978-3-540-39432-7_78 .