JACET Journal of Advances in Computer Engineering and Technology, 2(1) 2016

EIDA: An Energy-Intrusion aware Data Aggregation Technique for Wireless Sensor Network

Nafiseh Daneshgar Moghaddam[1], M. Habibi Najafi[2], Mohsen Jahanshahi[3], Ehsan Ahvar[4]

Received (2015-04-18)
Accepted (2015-10-17)

***Abstract*— Energy consumption is considered as a critical issue in wireless sensor networks (WSNs). Batteries of sensor nodes have limited power supply which in turn limits services and applications that can be supported by them. An efficient solution to improve energy consumption and even traffic in WSNs is Data Aggregation (DA) that can reduce the number of transmissions. Two main challenges for DA are: (i) most DA techniques need network clustering. Clustering itself is a time and energy consuming procedure. (ii) DA techniques often do not have ability to detect intrusions. Studying to design a new DA technique without using clustering and with ability of finding intrusion is valuable. This paper proposes an energy-intrusion aware DA Technique (named EIDA) that does not need clustering. EIDA is designed to support on demand requests of mobile sinks in WSNs. It uses learning automata for aggregating data and a simple and effective algorithm for intrusion detection. Finally, we simulate and evaluate our proposed EIDA by GloMosim simulator.**

***Keywords*- data aggregation, learning automata, energy-Intrusion aware**

1- Islamic Azad University, Qazvin Branch, Qazvin, Iran (n.daneshgar@qiau.ac.ir)
2- Islamic Azad University, Qazvin Branch, Qazvin, Iran (m.habibi@qiau.ac.ir)
3- Dept. of Computer Engineering Central Tehran Branch, Islamic Azad University, Tehran, Iran (mjahanshahi@iauctb.ac.ir)
4- Institut Mines-Telecom, Telecom SudParis, Evry, France (ehsan.ahvar@telecom-sudparis.eu)

I. INTRODUCTION

A Wireless Sensor Network (WSN) includes a large number of sensor nodes, which are typically self-organized in a multihop fashion. By working together, sensor nodes cooperate to finish a task [1]. These nodes have the ability to communicate either among each other or even directly to a sink. The sink can be a fixed node or a mobile node capable of connecting the WSN to an existing communications infrastructure or to the Internet where a user can have access to the reported data. Energy consumption in sensor networks is an integral factor. Because batteries carried by each mobile node have limited power supply, processing power is limited, which in turn limits services and applications that can be supported by each node. This becomes a bigger issue in sensor networks, as each node is acting as both an end system and a router at the same time, additional energy is required to forward packets from other nodes [2].

Energy consumption in WSN is a critical factor. Because batteries of sensor nodes have limited power supply which in turn limits services and applications that can be supported by each sensor node. In addition, as each node is acting as both an end system and a router at the same time, additional energy is required to forward and receive packets from other nodes.

DA is an effective approach to save energy because it can reduce the number of transmissions. In the literature, DA protocols could be classified into two classes: structured and structure-free. Structured solutions use a tree-based or a cluster-based structure constructed at the network initialization phase to achieve efficient data gathering. They usually rely on a structured architecture to accomplish data gathering.

Such structure-based methods suffer from high maintenance overhead in a dynamic environment where sensors can move or fail unexpectedly. On the contrary, structure-free approaches do not spend energy building any structure [3]. In this paper, we propose a structure-free DA technique that we focus on the designing without clustering.

Intrusion is an unauthorized (unwanted) activity in a network that is achieved either passively (e.g., information gathering, eavesdropping) or actively (e.g., harmful packet forwarding, packet dropping, hole attacks). In a security system, if the first line of defense, "Intrusion Prevention," does not prevent intrusions, then the second line of defense, "Intrusion Detection," comes into play. It is the detection of any suspicious behavior in a network performed by the network members [4]. Due to restricted operating conditions (constrained computational and energy resources along with an ad hoc communication environment) of WSNs, most of the security techniques (including intrusion detection techniques) devised for traditional wired/wireless networks are not directly applicable to a WSN environment [4] [5]. Therefore, designing a simple and efficient intrusion detection technique that is applicable to WSNs is a very big challenge.

This paper proposes an energy-intrusion aware data aggregation technique (named EIDA) that does not use clustering to support WSNs with mobile sinks. It uses a Learning Automata (LA)-based algorithm for DA. A simple and effective intrusion detection algorithm also attached to our proposed DA technique for supporting intrusion detection. Our EIDA is simulated and evaluated by GloMosim simulator.

Section II introduces related work. Section III describes different types of attack models. Section IV presents our proposed DA technique. Section V evaluates the proposed technique and Section VI concludes the paper.

II. RELATED WORK

Krishna et al. [6] explained the energy-efficient DA protocols in WSNs. The attributes of DA like latency, energy, node scheduling, and cluster size were explained in detail. They also classified the energy-efficient DA protocols into two models: structure-free and structure-based models.

LEACH [7] is a first clustering algorithm in WSNs and a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network. M. Jahanshahi et al. [8] proposed an efficient cluster head selection algorithm to select the cluster head for WSNs. Selection of the clusters is according to the residual energy, number of the neighbors, and the centrality of each node. The proposed algorithm uses a Fuzzy System to select the cluster head. The algorithm not only balances the energy load of all nodes, but also provides a reliable selection of a new cluster head and optimality routing for the whole networks. In a nutshell, most of the DA protocols rely on clustering approaches, though, such methods suffer from high maintenance overhead in a dynamic environment.

Structure-free DA model uses random and probabilistic methods to aggregate the data, so they do not maintain any structures. This method is very useful in event based on application where event region changes frequently and if we use structure- based approach, we have to maintain the structure repeatedly [9].

On the other hand, security in WSNs is an important issue, especially if they have mission-critical tasks [10]. In [4], a comprehensive survey of the state-of-the-art in Intrusion Detection Systems (IDSs) that are proposed for WSNs is presented. Secure DA protocols are threatened by two types of adversaries: passive and active. Differences between the two types are explained in the next section.

Clustering is an energy and time consuming technique in both initialization and maintenance phases. Therefore, finding a simple and quick DA method for DA without clustering can be essential. In our previous study [11], we have proposed a cluster-free DA technique, EDQD, to reduce and balance energy consumption in WSNs. Although EDQD could improve energy consumption of the network totally, we will add some extra features to it to improve its performance. In EDQD, each event witnessed node used LA to select a node from its Neighbor List (including its neighbors and itself) as its aggregator. In the situation where a Neighbor List has two or more candidates with same probability (same energy level and hop count), each sender node in EDQD selects one of them randomly as an aggregator. This type of random selection increases the number of paths from an event region to the sink.

In this paper, we reform this problem of EDQD to improve its energy consumption. We

also consider intrusion detection mechanism to design a cluster-free energy aware DA method with the ability to detect intrusion.

III. ATTACKER MODEL

Generally, an attacker can eavesdrop, replay, change or even destroy the ongoing communication. We can divide attackers into passive and active types. A passive attacker eavesdrops data sent in the network and tries to obtain sensitive information from the data (e.g., sensed data, shared cryptographic keys). This type of attacker is relatively easy to prevent by encryption mechanisms. Another type of attacker is active attacker, which, in addition to the passive attacker, is able to alter, replay or destroy the data. Active attacks are divided into external and internal ones. External attacks are carried out by nodes that do not belong to the network. In order to prevent such attacks a node and data authentication can be used. Internal attacks are carried out by the captured nodes. An attacker can access an area of deployment and capture a fraction of sensor nodes when the area is not physically protected. Moreover, sensor nodes are not tamper-resistant and hence the attacker can extract cryptographic keys that are used for authentication and encryption purposes. Being a legitimate participant of a network, the attacker can launch a variety of internal attacks [13]. This paper considers an internal attacker who captures a limited number of sensor nodes and performs selective forwarding and packet alternation attacks. In the selective forwarding attack a malicious node may refuse to forward certain packets and simply drop them. In the packet alternation attack a malicious node modifies packets that it forwards for neighbors [13][14].

IV. ENERGY-INTRUSION AWARE DATA AGGREGATION TECHNIQUE (EIDA)

This section is divided into three main parts. The first part introduces some useful definitions and terms. Second part describes the EIDA logical architecture and the third part shows a general overview of the EIDA.

A. Definitions

1) Query packet: A query packet is a request packet for receiving information on a particular event that is sent by Query source or sink node. Information such as packet sender Id, its energy level and hop count is attached to the packet.

2) Data packet: A data packet is created and sent by an event witnessed node. It includes information to be sent to sink. Energy level of the data sender node is attached to the packet.

3) Neighbor List: Each node has a Neighbor List. The neighbors information is recorded in a Neighbor List. The components of the Neighbor List for each neighbor are as follows: Neighbor ID (NID) holds the Id of a neighbor, Energy Level (EnLevel) holds the energy level of the sender of neighbor, Hop Count (HopCnt) stores the number of hops from a neighbor to the sink, Event Witness (EvtWintness) shows a neighbor detects an event or not. EventWitness fields whether of each neighbors are updated based on EventWitness field of the a query packet that is received from that neighbor. Finally the Aggregator Selection Probability (AggSelProb) holds the probability associated with a neighbor as computed by the learning automata. In addition to saving information of the node neighbors, the Neighbor List also records information of the node.

B. EIDA architecture

Our proposed EIDA technique basically includes the following components (Fig.1):

- *A Neighbor Update Unit (NUU),*
- *A Probability Computation Unit (PCU),*
- *A Probability Update Unit (PUU),*
- *An Intrusion Detection Unit (IDU).*

Figure 1. EIDA architecture

In brief, the NUU gets information from query or data packets and inserts them into the Neighbor List. Aggregation Module (AM) is in charge of selecting an aggregator for each event witnessed node. It works based on LA and includes PCU and PUU components. The probability of each neighbor is computed by the PCU and then updated by the PUU. And IDU monitors behavior of the neighbors and tries to find the ones that perform the attack. We describe

each unit in detail in the following sub-sections.

1) Neighbor Update Unit (NUU):

As mentioned earlier, there are two types of packets: query and data. The fields of each type of packet have been described in previous sections. After receiving a query packet, the NUU will insert the neighbor Id, energy level, hop count into the Neighbor List. For data packet, it just updates information of the neighbors.

2) Probability Computation Unit (PCU):

When a node (i.e., node i) receives a query packet from a neighbor for the first time (i.e., from neighbor K), it creates a new entry in its Neighbor List. The Neighbor List is composed of fields, and each part of the data has to be stored in its related field. The PCU can then compute the probability of neighbor K from the information contained in the Neighbor List received from neighbor K. The probability PK(t) associated with neighbor K is computed according to the equation (1).

$$Pk(t) = \frac{1}{2}\left(\frac{\varepsilon_k(t)}{\sum_{i=1}^{m}\varepsilon_i(t)} + \frac{\frac{1}{H_k(t)}}{\sum_{i=1}^{m}\frac{1}{H_i(t)}}\right) \qquad (1)$$

Where *PK(t)* is probability of selecting neighbor k as an aggregator. $\varepsilon_i(t)$ is the energy level advertised by neighbor *i*, *m* is the size of Neighbor List of node *k* (including now node k), $H_i(t)$ is the number of hops advertised by neighbor *i* to the sink *S*. $\varepsilon_k(t)$ is the energy level advertised by node *k* and $H_k(t)$ is the number of hops advertised by node *k* to the sink *S*.

As we already mentioned, node i creates an entry in its neighbor for itself to record its information. Node i also computes its probability based on equation (1).

The rationale of using equation (1) is that it produces a good balance between energy and distance, though at the cost of the potential re-computation of the probabilities immediately after each query packet is received, since the sum of the probabilities for all neighbors must be equal to one.

3) Probability Update Unit (PUU):

Updating probabilities is considered as a main part of LA. We describe updating procedure with one example, Fig.2. The updating procedure is based on piggybacking and overhearing. We Assume node i has already selected node j as its next hop and sent its data packet to it (selected action of LA in node i). Then, node j attaches its energy level to the original data packet, piggybacking. When data packet is forwarded to neighbor with highest probability (i.e. node k), it receives data packet and updates energy level of the sender node (node j). Also, all other neighbor nodes can receive and update energy level of sender node by overhearing.

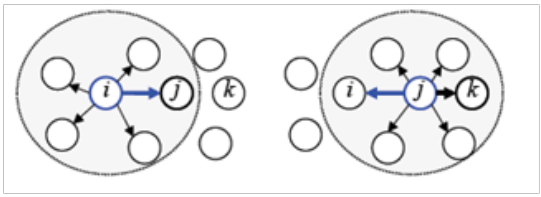


Figure 2. Updating procedure

In this situation, the previous data sender node (i.e. node i) receives the data packet again by overhearing. It updates the energy level of sender node (i.e. node j) and then updates probabilities in its Neighbor List based on one of four following behavioral scenarios:

- *En_j < (0.5*AvgEn) then the aggregator penalizes with 3β. β is computed based on Equation(4),*
- *(0.5*AvgEn) <= En_j < (0.8*AvgEn) then the aggregator penalizes with β′.β′ is computed based on Equation(5),*
- *(0.8*AvgEn) <= En_j < (AvgEn) then the aggregator rewards with α/2. α is computed based on Equation(3),*
- *En_j > AvgEn then the aggregator rewards with α. α is computed based on Equation (3).*

That En_j is energy level of the sender node (node j) that is attached to the data packet. *AvgEn* is the average energy of neighbor nodes and is computed based on equation (2).

$$Avgenergy = \left(\sum_{i=1}^{m} EnergyLevel_i\right)/m \qquad (2)$$

In the equation (2), $EnLevel_i$ is the energy level of neighbor *i* (sender node *i*) and *m* is the Neighbor List size.

$$\alpha = \lambda + \delta_1(\frac{En + Maxhop - HopCount}{EnLevel + Maxhop}) \quad (3)$$

$$\beta = \lambda + \delta_2(\frac{AvgEn - En + HopCount}{AvgEn + Maxhop}) \quad (4)$$

In Equations (3) and (4), α and β are reward and penalty parameters, respectively, *HopCount* is hop numbers between the sender node and the sink node, *EnLevel* is the initial energy of each node, *Maxhop* is maximum number of received hop counts and λ is minimum value for reward and penalty parameter. δ_1 and δ_2 are selected which cause dose not to exceed a threshold for α and β parameters. And β' is calculate as follow:

$$\beta' = (1 + \frac{\frac{En}{AvgEn} - 50}{30})\beta \quad (5)$$

Where, 30 is equal to 80 - 50.

4) Aggregation Module (AM):

The AM is the core routing-aggregation module with the responsibility for choosing the aggregator and next-hop during the data packet forwarding process. It can work on three different modes: Normal, aggregator and warning. In warning mode, when the IDU warns the AM about selective forwarding attack, the AM will select another node. In condition where IDU warns about a packet alternation attacks, the AM will send a warning message to its neighbors.

If a node (i.e. node i) witnesses an event, the AM will operate in its aggregation mode. It looks at its Neighbor List to select a node as its aggregator. It checks probability of all neighbors of node i that their EvtWintness field of which is True (and also its probability). If there is only a node with the highest probability, it select the node as its aggregator. If the selected aggregator is the node itself, it waits to receive packet and after that will forward the aggregated information to an intermediate node (described in normal mode). But if AM selects another node as aggregator of node i, it will send the packet to aggregator. In condition where there are more than one event witnessed node with the highest probability in the Neighbor List, AM selects the one with lowest Id. If the selected node is the node i, AM does not do anything and after a short delay (to aggregate information) it will go to its normal mode. Otherwise, the AM will forward the packet to the selected aggregator.

In the normal mode, the AM only acts as a learning automata based routing protocol to forward the packet to the best next hop. It selects the neighbor with highest probability as its next hop. If there are more than one neighbor with the highest (and same) probability, AM selects one of them randomly. In normal mode, AM does node consider the node itself in making its routing decision and only considers the neighbors.

The processing of the AM module is summarized in Algorithm 1.

5) Intrusion Detection Unit (IDU): Our IDU is designed to detect selective forwarding and packet alternation attacks. Selective forwarding attack (SFA)- lets us consider the scenario in Fig.2, where a sensor node (i.e. node i) sends a packet and the packet is dropped by a malicious neighbor node (i.e. Node j). In our proposed intrusion detection method, the data packet sender node (node i) should check its data packet by overhearing receiver node (node j). If receiver node (node j) does not forward the data packet, the sender node detects the receiver as a malicious node. It changes situation of the receiver node from “normal” to “attacked” and after that selects another node to send the data packet. Packet alternation attacks (PAA)- when node j, in Fig.2, forwards the data packet, node i receives its data packet by overhearing. Then node i (IDU) compares received data packet with the data packet that has already sent to node j. If both packets are same, node i does not do any action. But if there is a difference between original data packet and its overheard version, node i (AM) will consider node j as a malicious node and will send a warning message to its neighbors.

C. EIDA overview

EIDA is a new energy aware on demand DA technique that supports sensor network with a sink mobility. It has also ability to detect intrusion detection. We use learning automata to find the best node in terms of energy level and hop count both. If a node witnesses an event, it changes its *EventWitness* field to True in its Neighbor

List. EIDA can be divided into two phases: (i) query broadcasting, and (ii) data aggregation and forwarding.

Algorithm 1: The AM Algorithm

```
1  //warning mode;
2  foreach warning alarm from IDU do
3      if Received SFA alarm then
4          re-send the packet to another neighbor;
5      if Received PAA alarm then
6          broadcast warning massage to neighbors;
7  // agrregation mode;
8  foreach witnessed even do
9      // I am an event witnessed node;
10     if there is only one node with highest probability then
11         select it as an aggregator;
12     if I am selected aggregator then
13         wait, aggregate and go to normal mode;
14     else
15         send the packet to the selected aggregator;
16     if there are some nodes with highest (and same)probability then
17         select one with lowest Id as an aggregator;
18     if I am selected aggregator then
19         wait, aggregate and go to normal mode;
20     else
21         send the packet to the selected aggregator;
22 // normal mode;
23 foreach packet to be forwarded do
24     // I am an intermediate node;
25     if there is only a neighbor with highest probability then
26         forward the data packet to it;
27     else
28         send the packet to the neighbor with highest probability that
           has lowest Id;
```

Query broadcasting- the mobile sink node generates a query packet and inserts 1 into its *HopCount* field, False into *EventWitness* field, its id into *NodeId* field and its energy level into Energy Level field. Then the sink node broadcasts the Query packet. Each neighbor node that receives the Query packet computes the aggregator selection probability of the Query Source based on energy level and hop count of Query packet, and inserts the id, energy level, event witness and the aggregator selection probability of the Query Source into its Neighbor List. Then it increments the hop count by 1, replaces its hop count, Id, event witness situation and energy level and forwards the Query packet. Therefore, each receiver node simply can get information of its neighbor. Then it increases the hop count by 1, inserts its information into the query packet and forwards it to its neighbors. In this phase, each node broadcasts its id and energy level only once. Therefore, each node can maintain a list of its neighbors and their energy and their aggregator selection probability.

Data aggregation and forwarding- When a node decides to select its aggregator, it observes the *EventWitness* fields in its Neighbor List. If its *AggregatorProbability* field is the highest, it selects itself as an *aggregator*. Otherwise, the data is forwarded to a neighbor with highest probability field *EventWitness* field of which is *True.*

V. PERFORMANCE EVALUATION

In this section, we evaluate the EIDA by comparing it to EDQD [11] and Minimum Hop Routing (MHR) [16]. EDQD is a structure-free energy aware DA -routing technique and can be a good candidate to show effectiveness of the proposed algorithm. MHR is a simple routing algorithm that does not use any aggregation technique. By comparing the proposed idea to MHR we can see effect of aggregation technique on energy consumption. We used the GloMoSim simulator [15] developed by UCLA. The simulation model used and the results we obtained with it are described below.

A. SIMULATION MODEL

The performance of the different schemes is evaluated using a surface of 1000m^2. The radio range for each sensor is set to 110m, with an available bandwidth of 2Mbps and a radio transmission (TX) power of 0.0dBm. We considered 1000 sensors. The placement of the sensors in the terrain and their initial energy levels were selected randomly. It is worth highlighting that, even though the placement and initial energy of the nodes were set randomly, once set those factors remained fixed for rest of the trials to obtain comparable results across experiments.

We have considered four different scenarios (S1, S2, S3 and S4) to evaluate ability of the algorithms in different situations. In all scenarios, the traffic in the network is initiated by a mobile source sink S, which acquires information from a particular region d. The sink S moves at a speed of 30 Meter/Minute. There are 10 sensors in the region d and all of them can detect each other as a one-hop neighbor. Once the query is received at region d, the sensors will immediately should find their aggregator and send back the response to S with the requested information. In S1 all sensors in the event region d have same situation or probability (see Equation (1)). This situation often happened in first hours of starting the network. Scenario2 (S2) assumes 70 percent of

the nodes in the region d have highest (and same) probability. In S3 we consider 50 percent and in S4 only 25 percent of the nodes in the region d have highest (and same) probability.

We evaluate the different routing schemes considering three different tests:

Test 1: Number of paths- This test is one of the indicators of the effectiveness of aggregation schemes in terms of energy management and reducing traffic.

Test 2: Total number of hops- This test computes the total number of hops that data packet(s) traverse to deliver the event information from region d to the sink.

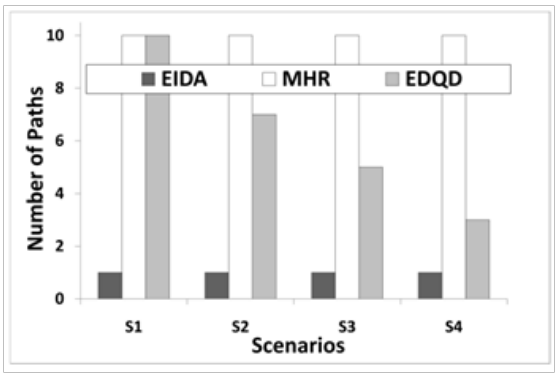
Test 3: Total energy consumption- This test computes total energy consumption for transferring the event information from region d to the sink for each algorithm. It provides another indicator for which routing scheme is more efficient in managing energy.

B. SIMULATION RESULTs

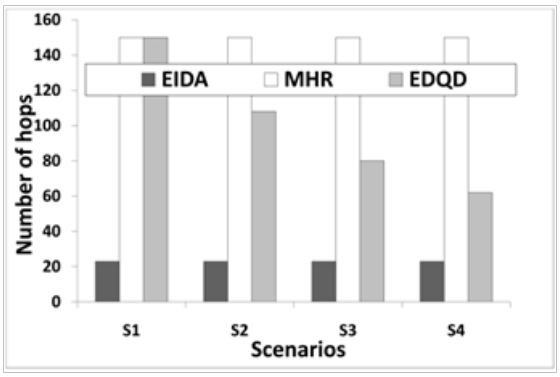
In this section we evaluate simulation results. Results of Tests 1 and 2 have a direct effect on total energy consumption (Test.3). Generally, a scheme with lowest number of paths and the lowest number of hops can offer lowest energy consumption (packet transmission) to transfer the event information from region d to the sink S. Fig.3(c) shows total energy that is needed to receive information from the event occurred in the region d. In all scenarios MHR is the worst. It is because MHR does not use any DA technique and, as Fig.3 (a) shows, it sends an independent response from each event witnessed sensor to the sink S. Sending independent response to the sink increases its total number of hops (Fig.3 (b)).

As we already mentioned, in EDQD if some candidates in the event region d have highest (and the same) probability, each node selects one of the candidates as an aggregator randomly. This method of selecting aggregators can leads to have multiple aggregators and paths. In contrast, EIDA always selects only a candidate as its aggregator for all event witnessed nodes in the same region. If some candidates in the event region d have highest (and the same) probability, in EIDA each node selects the candidates with lowest Id. Therefore, we do not have problem of multiple aggregators in the same region for EIDA.

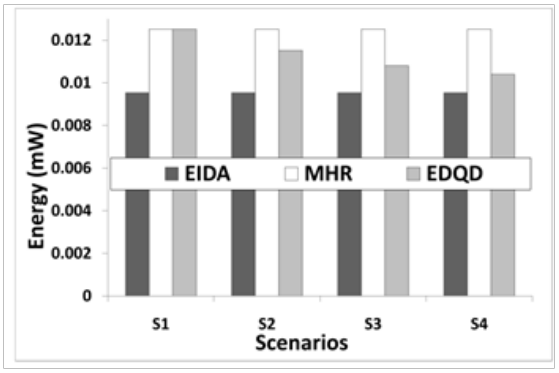
In a nutshell, as parts (a), (b) and (c) of the Fig.3 show, EIDA performance is independent of different scenarios (situations of event witnessed nodes), while changing situations (i.e. energy and hop count of the event witnessed sensors) in the region d can make a direct effect on performance of EDQD.



(a) Test1: Path numbers.



(b) Test2: Total number of hops.



(c) Test3: Total energy consumption.

Figure 3. Tests results for different senarios.

VI. CONLUSION

This paper has addressed energy consumption and intrusion detection as two important factors for current WSNs. Data aggregation is considered as an efficient technique to reduce energy in WSNs. We have proposed a new data aggregation technique to improve energy consumption in

WSNs and equipped the technique with an intrusion detection module. Clustering is a time and energy consuming technique for gathering data. Unlike current data aggregation methods that often use clustering, we have proposed a structure-free data aggregation method, EIDA; that works based on learning automata. The number of paths and total number of hops were considered in this paper as two important metrics for reducing energy consumption. We have defined different scenarios to show effectiveness of the EIDA. The Results showed, by improving these two metrics, EIDA could reduce total energy consumption. The results also showed, unlike other methods, different situation of event witnessed nodes in the event region (different scenarios) can not affect EIDA performance.

REFERENCES

[1] Mo Li, Zhenjiang Li, and A. V. Vasilakos, "A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues'', Proceedings of the IEEE, Vol. 25, Issue 10, Pages 2367-2380, October 2013.

[2] Jamal. N. Al-Kamal, and Ahmed. E. Kamal, "Routing Techniques in Wireless Sensor Networks, A survey", Wireless Communications, IEEE, vol. 11, pp. 6-28, 2004.

[3] Chih-Min Chao, Tzu-Ying Hsiao, "Design of structure-free and energy-balanced data aggregation in wireless sensor networks", Journal of Network and Computer Applications 37 (2014) 229-239.

[4] Ismail Butun, Salvatore D. Morgera, and Ravi Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks", IEEE COMMUNICATIONS SURVEYS and TUTORIALS, VOL. 16, NO. 1, 2014.

[5] I. Butun and R. Sankar, "A Brief Survey of Access Control in Wireless Sensor Networks", in Proc. IEEE Consumer Communications and Networking Conference, Las Vegas, Nevada, January 2011.

[6] M. Bala Krishna and Noble Vashishta,"Energy Efficient Data Aggregation Techniques in Wireless Sensor Networks", 5th International Conference on Computational Intelligence and Communication Networks, 2013.

[7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy effcient communication protocol for wireless microsensor networks," in in Proc. of the 33rd International Conference on System Sciences (HICSS00), January 2000.

[8] M. Jahanshahi, S. Rahmani, S. Ghaderi, "An Efficient Cluster Head Selection Algorithm for Wireless Sensor Networks Using Fuzzy Inference Systems", The International Journal of Smart Electrical Engineering (IJSEE), Vol.2, No. 2, 2013.K.S.Narendra, M.A.L.Thathachar, "learning automata: An introduction", Prentice Hall 1989.

[9] Vaibhav Pandey, Amarjeet Kaur and Narottam Chand, "A review on data aggregation techniques in wireless sensor network", Journal of Electronic and Electrical Engineering, ISSN: 0976-8106 and E-ISSN: 0976-8114, Vol. 1, Issue 2, 2010.

[10] I X. Chen, K. Makki, K. Yen and N. Pissinou, "Sensor network security: A survey", IEEE J. Communications Surveys and Tutorials, vol. 11, num. 2, pp. 52-73, 2009.

[11] Ehsan Ahvar, "EDQD: An Energy-Distance aware Query-based Data aggregation technique for wireless sensor network", IEEE 24th International Conference on Advanced Information Networking and Applications workshop, australia, 2010.

[12] K.S.Narendra, M.A.L.Thathachar, ``learning automata: An introduction'', Prentice Hall 1989.

[13] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," Ad Hoc Networks, vol. 1, no. 2-3, pp. 293-315, 2003.

[14] Andriy Stetsko, Vashek Matyas, "EffectivenessMetrics for Intrusion Detection in Wireless Sensor Networks", European Conference on Computer Network Defense, 2009.

[15] Zeng X., Bagrodia R., and Geria M. "Glomosim: A Library for Parallel Simulation of Large Scale Wireless Networks" Proceedings of the 12th Workshop on Parallel and Distributed Simulations 1998; 154-161.

[16] S. S. Chiang, C. H. Huang, and K. C. Chang, "A Minimum Hop Routing Protocol for Home Security Systems Using Wireless Sensor Networks", IEEE Transactions on Consumer Electronics, Vol. 53, no. 4, pp. 1483-1489, 2007.