

AEESPAN: Automata Based Energy Efficient Spanning Tree for Data Aggregation in Wireless Sensor Networks

Zahra ESKANDARI, Mohammad Hossien YAGHMAEE

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Email: za_es73@stu-mail.um.ac.ir, hyaghmae@ferdowsi.um.ac.ir

Received May 6, 2009; revised July 5, 2009; accepted July 10, 2009

Abstract

In Wireless Sensor Networks (WSNs), sensor nodes are developed densely. They have limit processing capability and low power resources. Thus, energy is one of most important constraints in these networks. In some applications of sensor networks, sensor nodes sense data from the environment periodically and transmit these data to sink node. In order to decrease energy consumption and so, increase network's lifetime, volume of transmitted data should be decreased. A solution, which is suggested, is aggregation. In aggregation mechanisms, the nodes aggregate received data and send aggregated result instead of raw data to sink, so, the volume of the transmitted data is decreased. Aggregation algorithms should construct aggregation tree and transmit data to sink based on this tree. In this paper, we propose an automaton based algorithm to construct aggregation tree by using energy and distance parameters. Automaton is a decision-making machine that is able-to-learn. Since network's topology is dynamic, algorithm should construct aggregation tree periodically. In order to aware nodes of topology and so, select optimal path, routing packets must be flooded in entire network that led to high energy consumption. By using automaton machine which is in interaction with environment, we solve this problem based on automat learning. By using this strategy, aggregation tree is reconstructed locally, that result in decreasing energy consumption. Simulation results show that the proposed algorithm has better performance in terms of energy efficiency which increase the network lifetime and support better coverage.

Keywords: Automata Learning, Wireless Sensor Networks, Data Aggregation, Energy Efficient, Spanning Tree

1. Introduction

Wireless Sensor Networks (WSNs) are networks that consist of low power nodes with limited processing ability. These nodes have sensors which sense light, temperature, jitter and etc. in the environment. These nodes are deployed in environment densely and randomly. In monitoring application, these sensor nodes sense data from the environment periodically and transmit these data to sink node. Since transmitting the data is the most costly function in the network and power of the nodes is limited and cannot usually be charged; this leads to decrease node's power quickly.

After some rounds, network nodes energy is ran out and this leads to situations which the network can not work anymore. To the points mentioned above in order to increase network's lifetime, number of transmitted data packet should be minimized [1,2].

Network nodes, after event occurrence and sensing data from the environment, forward the sensed data to

the sink. In addition to sensed data, each node must transmit other node's data to the sink. As mentioned above, data transmission consumes node's energy quickly. The solution which is suggested to decrease the number of data transmissions is aggregation mechanism. Aggregation mechanism works as follow: each node senses data from the environment and receives other node's data, then aggregates these data, based on the aggregation function and transmits the aggregation result to the sink. Therefore aggregation decreases the data volume that is transmitted and this leads to less energy consumption. In addition to mentioned improvements, aggregation decreases collision and retransmission delay [3]. In aggregation algorithms, we must construct aggregation spanning tree [4]. The spanning tree is a tree contains all network nodes and doesn't have any loop.

Like routing algorithms [5], aggregation algorithms should also be aware of the network topology and based on these information and queries which are propagated by root, network nodes select aggregation function and

aggregate the data, and then forward aggregated data to sink.

Cluster based algorithms [6] needs only local information to construct aggregation tree, therefore they transmit fewer packets to construct the aggregation tree.

In [7], authors investigate the computational complexity of optimal data aggregation in sensor networks and show that it is generally NP-hard; they present some suboptimal data aggregation tree generation heuristics, Center at Nearest Source (CNS), Shortest Paths Tree (SPT) and Greedy Incremental Tree (GIT) and showed the existence of polynomial special cases. Different aggregation algorithms have been presented in recent years [1,4,6,8–10].

Espan [4] is an energy-aware spanning tree algorithm that constructs the aggregation tree to aggregate the data. In Espan, the source node which has the highest residual energy is chosen as the root and other nodes choose their corresponding parent node among their neighbors based on distance to the root and residual energy. In LPT [8] after selecting the node with most energy as root, each node selects neighbors with most energy as parent and its parent forwards its data to the sink.

In this paper, we present an automata based Energy Efficient Spanning tree (AEEspan) algorithm which is a new energy efficient aggregation algorithm for wireless sensor networks. The current work is a modified version of our already published papers [1,11]. In [1] we propose an energy aware data aggregation spanning tree algorithm. In [11], we present an automata based algorithm to construct spanning tree. Automata is an able-to-learn decision-making structure that selects the best action among a number of actions then wait for environment's response to this selection. By using the environment feedbacks, automata update the probability of the selection of each action among the set of actions and select best action for the next step. The main idea of proposed protocol is as follow: each node has an automaton to select the best nodes among its neighbors as its parent.

Since the status of the network is dynamic, the aggregation algorithm should construct the routing tree periodically. When a timer is expired our some nodes are failed in the network, the new aggregation tree must be constructed [4,8].

At the beginning of each time interval, routing packets are flooded into the network. In each routing packet has some routing information likes: number of hops to the root, remaining energy, number of child node. Each node selects optimal path to the sink, based on algorithm parameters. Since the node's energy is limited, transmitting and receiving this volume of routing information is not a good solution to construct aggregation tree. This overhead causes a lot of energy consumption. So, some nodes run out of energy quickly and fail. This causes network to be disconnected.

To solve this problem we use an automaton based approach; if a node in the aggregation tree fails, and a part of tree is disconnected, only this part of tree starts to reconstruct. So it is not necessary to flood routing packets into entire network. To do this, each node uses the environment feedbacks, and updates its automata.

The remainder of this paper is as follow: in section 2, we review some existing aggregation algorithms. The system model is given in section 3. In section 4, we present the proposed algorithm and the performance evaluation of proposed algorithm is presented in section 5. Finally, section 6 concludes the paper.

2. Related Work

Different aggregation algorithms have been presented in recent years. In this section we review them briefly. As presented in [9], DCTC algorithm dynamically constructs the aggregation tree for mobile target tracking. In the presented algorithm depending on the target location, a subset of nodes participates in tree construction.

In [12], the sink saves the entire network state and then by considering link cost, in centralized form, constructs the tree by minimum cost. In cluster algorithm [6], after partitioning the network into clusters, cluster's members construct aggregation tree and transmit data to cluster head. After aggregation, cluster heads transmit aggregated data to the sink in one hop or multihop manner [13].

Espan [4] is an energy-aware spanning tree algorithm that constructs the aggregation tree to aggregate the data. In Espan, the source node which has the highest residual energy is chosen as the root and other nodes choose their corresponding parent node among their neighbors based on distance to the root and residual energy. One of the most important problems of Espan is that the nodes with least distance to root may be selected as parent by many nodes. So these nodes consume their energy quickly and then they will be failed sooner than other network nodes.

In LPT [8] after selecting the node with most energy as root, each node selects neighbors with most energy as parent forwards its data to the sink. In the mentioned algorithm, when a node in the tree fails, the tree will be reconstructed.

In [1] an energy efficient algorithm, which constructs the aggregation tree, is presented. To prevent failing the nodes and to increase the network lifetime, the algorithm considers both the remaining energy and the distance parameters. Each node selects a node which has the most energy within neighbors as its parent. Furthermore, the distance from this parent to the root must be reasonable. To balance the energy and distance parameters, the algorithm uses path's energy and length parameters.

In [14], the proposed algorithm uses machine learning to transmit the sensed data to the sink. Learning algo-

rithm is executed in the sink and its result is propagated throughout the network. In [15] Q-learner is used to construct aggregation tree, to maximize aggregation ratio.

In [16] an algorithm to construct aggregation tree, based on automata, is proposed. In this algorithm, in which each node is equipped with an automaton, the automaton selects a path for transmitting data via the path which the aggregation ratio is maximized. In [17], the algorithm considers an automaton for each node, which selects a path to transmit data to the sink in accordance with network conditions.

3. System Model

We consider a network of N sensor nodes uniformly distributed over a region and one sink. These nodes are non mobile. The sensor nodes have radio communications; two nodes can receive and transmit data if they are in communication range of each other. There are three types of data collection in sensor networks [18]. Event-based data, such as intrusion detection or object tracking, is collected when an event within the deployment region occurs. The event is confirmed by sensors and reported to the sink. State-based data is collected in response to a query sent to selected sensors requesting relevant data. Global state-based data, such as temperature or humidity, is collected by sensors all over the deployment area and is transmitted toward the sink. Our interest here is in global state-based data. All sensor nodes are sources, sense environment and transmit sensed data to the sink periodically. In the following subsections we describe the energy model and data transmission model used in this work.

3.1. Energy Model

We use the same energy consumption model described in [19]. In this model a sensor node consumes E_{elec} (J/bit) in transmitter or receiver circuitry and E_{amp} (J/bit/m²) in transmitter amplifier to achieve an acceptable signal noise ratio. A sensor node expends energy $ET_{ij}(k)$ or $ER_i(k)$ in transmitting or receiving a k -bit packet to or from distance $dist_{ij}$, given by the following equations:

$$ET_{ij}(k) = E_{elec} * k + E_{amp} * k * dist_{ij}^{\lambda} \quad (1)$$

$$ER_i(k) = E_{elec} * k \quad (2)$$

The exponent λ heavily depends on the communication medium [20]. In the current work, we assume that the transmission power is directly related to the squared distance, means $\lambda=2$, which hold for free space. When the distance is small, the free space propagation model is adopted for energy loss, and when the distance is large, the two-ray ground model is adopted for energy loss, which means $\lambda=4$.

In the above functions, k represents the length of transmitted and received data packet. Sensor nodes transmit or receive two types of packet; routing packet and data packet. Routing packets flood in the network to construct or reconfigure the aggregation tree. Data packets include data which sense by nodes from environment and are transmitted to sink. We assume the aggregation function is simple, for example, max or average function, so the input data length is equal to the output data length. Based on this assumption, all data packets in the network have the same length. According above descriptions, we should try to minimize not only distance but also the volume of transmitted and received packets.

As described in [6] if aggregation function is simple, the energy consumption for data aggregation will be negligible.

3.2. Data Transmission Model

After determining children of a node, a node creates a TDMA schedule and notifies its children about it. In the data transmission phase, children send their data to their parent according to the specified TDMA schedule. By using TDMA scheduling mechanism, we can solve the collision problem of data transmission, too. In addition, after sending data each node goes to sleep mode until next round, which cause power saving.

As described in [20], round is defined as the collection of one data unit from every node in the network and delivering the resulting aggregated data to the sink node. In every round, each parent in the tree will wait till it receives data from all its children. A node after participating in a round, wait until next round. Based on [20], lifetime of a tree is defined as the number of rounds that can be performed before the failure of certain percentage of total nodes. Therefore, in this paper, lifetime is defined as the failure of 10% of the total nodes of the tree.

3.3. Automata

Learning automata is an abstract model which has a finite set of actions as its input. Each member of the input set has a selection probability parameter. Automata select an input with highest selection probability as their output. Then the environment evaluates the selected action and responses to the automata. Automata use the response for learning process.

Learning process is as follow: if the environment response is unfavorable based on network parameter, the automata penalize the selected input by decreasing its selection probability and increasing selection probability of the other members of the input set members. But if the environment response is favorable, the automata reward the selected input by increasing its selection probability and decreasing selection probability of the other members of

the input set. The rewarding process increases selection probability of the awarded input for the next step.

As seen in Figure 1, an automaton is learned based on the feedback of the environment.

As described above, an automaton is defined by the quadruple $\{\alpha, \beta, P, T\}$ in which $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\}$ represent the output set, $\beta = \{\beta_1, \beta_2, \beta_3, \dots, \beta_m\}$ represent the input set, $P = \{p_1, p_2, p_3, \dots, p_4\}$ represent probability set and finally $p(i+1) = T[\alpha, \beta, P]$ represent the learning process.

4. Proposed Algorithm

As mentioned in section 1, data aggregation tree construction algorithms construct tree periodically. To construct an aggregation tree, at the beginning each period, routing packets are flooded into the entire network to inform all nodes. After this step, each node selects the best path toward the sink node and transmits data via selected path until the next period. Transmitting these routing packets periodically consumes a lot of energy and has unfavorable overhead for the network.

In automata based algorithms [17,16], at the beginning, routing packets are flooded into the entire network. Each node considers each neighbor as entry in its routing table and then calculates the selection probability of each entry based on the algorithm's parameters, energy or distance and etc., and then each node selects the neighbor with highest selection probability as its parent and sends its data via this parent to root.

In [16] after receiving data, root sends acknowledgment to the sender node; this acknowledgment has some information for automata. Based on acknowledgment information, automata penalize or reward the path's nodes, on the way that if the selected path was optimal based on network parameters, selection probability is increased for the next step, but if selected path was not optimal, selection probability is decreased for next step. This process is called automata learning.

In the next steps, each node selects a new parent based on the updated selection probability of the nodes in the network and this process is repeated by the end of the network's lifetime. By using of this property of automata-learning-the algorithm prevents flooding the routing packets periodically, at the same time, by using ack information, nodes are aware from changes in network topology and paths are updated.

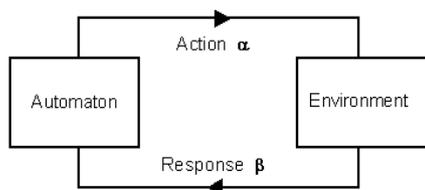


Figure 1. Learning automata.

Proposed algorithm works as follow: at the beginning, routing packets are flooded into the network by the way that each node sends its energy and the distance to root to all its neighbors. Each neighbor, after receiving this message, considers the sender as a new entry in its routing table, and puts sender ID, sender energy and sender distance to root as entry fields.

This sends/receives is performed in entire network, so each node maintains neighbors information in its routing table. Then the routing table entries are considered as input set of automata and the automata calculate the selection probability of each entry as follow:

$$Sel - prob = C_i * \frac{energy_j}{distance_j} \quad (3)$$

In Equation (3), C_i is a constant which is calculated by node and is depended on the sum of energy and distance to root of entries in routing tables of node i . This means that node i adds energy and as well distance to root of all the input set members. Then automaton considers the result of dividing the entry energy by its distance to root multiply a fixed number (C_i) as the selection probability of the entry.

Each node selects neighbor with highest selection probability as its parent, nodes in the network sense data and aggregate them with collected data from their child, then send the result of aggregation to their parents. Their parents forward data to the sink by repeating this process.

In fact, trees show paths that each algorithm selects for transmitting data to sink. These paths have an important effect on energy consumption, if the algorithm selects shortest paths mostly, the nodes in these paths fail quickly, and in continue nodes must send their data via other paths that may be longer, which led to high energy consumption, decrease lifetime and disturb coverage of the network. And as well, if the algorithm selects paths by considering energy parameter as main parameter and does not regard to path's length, nodes send data via longer paths which causes to high energy consumption. Thus, the algorithms should consider parameters, not one parameter, and balance between these parameters.

In these work, we try to select an optimal path by considering both parameters-energy and distance-as main parameters and construct the tree based on both of them.

In order to update automata, each node must collect some information from the network. By using this information, an automaton becomes aware of network changing. In [17] to be aware of the network state, each node after receiving data sends feedback or acknowledgment message to the sender of the data and as mentioned before, this message has some information. By using these feedbacks, automata penalize or reward the selected parent, but sending these acknowledgments have a lot of overhead. In [16] to decrease this overhead, acknowledgment is sent after some data transmissions.

In the proposed algorithm, to be aware of network's changes, one solution that was presented in [11] worked as follow: each node broadcasts its energy and distance to root, in data packet, and does not send in a separate packet, so, node's neighbors, after receiving these packets, update their routing tables. This procedure, perform automata learning. Since node's information of network is updated, optimal path to root is continuously selected.

But, transmitting these addition data led to waste energy because parent's energy becomes less than other nodes in neighborhood after some rounds, mostly. Thus, transmitting additional data in each data packet, based on energy model that mentioned in section 3. A wastes energy.

So, we can improve algorithm performance by working as follow: If a node fails in aggregation tree or the node's energy is lower than a pre determined threshold, then the node's children select a new parent from the nodes in their neighborhoods. Then, it is not necessary to reconstruct aggregation tree globally and periodically. By using this strategy the tree is reconstructed when it is needed, and reconstruction packet broadcast locally. This leads to reduction in data transmission in the network and power saving.

Thus, the reconstruction section of proposed algorithm works as follow: by failing a node in tree, node's children select a new parent base on their automata inputs; each node's children broadcasts an update packet. The neighbors, after receiving this packet, send their information to packet's sender. Then, each sender node updates its automata and then, an input with highest selection probability is selected as a new parent. By using this property, we prevent flooding the routing packets, and also, nodes are aware from changes in the network topology and paths are updated. This process is repeated by the end of the network's lifetime.

In this work, the input and output sets- α , β - of each node's automaton are the entries of its routing table, and the probability set-P-is the set of selection probability of entries. To increase efficiency in proposed algorithm, learning process does not perform each round, but when energy consumption reaches a threshold, automaton, based on responses from environment, performs learning process.

Reconstruction property is an important section in tree construction algorithm that is noted rarely. In this work, we try to achieve two main goals:

- Construct an energy efficient tree by considering both energy and distance parameters.
- Add the reconstruction property, to prevent from flooding packets globally.

The pseudo code of the proposed algorithm which helps us to understand the details of the proposed algorithm is given in Figure 2. In this pseudo code, m represents the message which is sent by each node and con-

tains three fields: node's ID, node's energy and node's distance to root. Input of AEEspan algorithm is the network nodes vector.

5. Performance Evaluation

In this section, using computer simulation, we evaluate the performance of the proposed algorithm and compare it with other algorithms [1,4,8] algorithm. We call the algorithm presented in [1], EEspan in below figures.

We consider a sensor networks with N sensor nodes randomly arranged in a 600m×600m region. The number of nodes (N) varies from 300 to 700. The initial energy of each node varies from 8J to 20J. The communication range of all nodes is set to 60 meter. The size of sensor data packet is 320 bits and a routing packet is 30 bits length. In the following curves the average values over 20 simulation experiments are depicted. Also, we assume all nodes in the network sense the area periodically and send their data to the sink node.

```

Define:
  Message M(sender_Id, sender_Energy, sender_Distance)

AEEspan(nodes)
{
  For each node N in nodes
  {
    Message M1(N.Id, N.Energy, N.Distance)
    N.Broadcast(M1)

    While(N.Receive(M2))
      N.Add_routing_table(M2.Id, M2.Energy, M2.Distance)
  } //flooding routing packets

  For each node N in nodes
  {
    For each entry E in N routing table
      N.Caculate(E.Selection_probability)

    N.Parent=Highest_selection_probability
      (N.Routing_table_entries)
  } //constructing aggregation tree

  While(true)
  {
    N.Transmit(Data, N.Parent)
    N.Receive(Data, N.Parent)
    If(N.Parent.Energy < Threshold)
    {
      N.Broadcast(Update_packet)
      For each entry E in N routing table
        N.Caculate(E.Selection_probability)

      N.Parent=Highest_selection_probability
        (N.Routing_table_entries)
    } //reconstruction
  }
}

```

Figure 2. The pseudo code of the proposed algorithm.

To do simulation, we use centralized version of LPT, however Espan, EEspan and AEEspan work in distributed manner. In below figure, to compare performance of the trees which are constructed by algorithms, we do not consider transmission and receiving energy for routing packet that flooded for tree reconstruction. As mentioned before, nodes send their data via the tree which is constructed by the algorithm, so it is important to compare paths that each algorithm selects to send data to sink.

At the first simulation trial, to evaluate the energy efficiency of proposed algorithm, AEEspan, each node is assigned with an initial energy that is randomly chosen between 8J and 20J. After some simulation rounds, we measure remained energy of network nodes. In Figure 3, sum of the remained energy of all nodes in network is plotted versus number of nodes for four algorithms.

Since LPT algorithm selects paths by considering only energy parameter, nodes transmit their data via longer paths which make higher energy consumption. In Espan algorithm, nodes transmit data via shortest paths, but by failing low power nodes in these paths, data must be transmitted via other paths which may be longer. While in EEspan [1] and AEEspan, nodes consume less energy, because in these algorithms, tree is constructed by applying a reasonable relation between energy and distance parameters. Unlike the algorithms given in [4,8] use only one of these parameters as the main parameter and the other parameter is used as lower priority parameter.

To verify above claim about path length, we study suggested paths in these algorithms. In the aggregation tree construction algorithms, average path length parameter represents average depth of tree that is the number of the hops between the nodes and the root, so, the tree with deeper branch means that nodes transmit data via longer path, and this causes more delay and also more energy consumption.

In Figure 4, the average path length is plotted versus the number of nodes. As in AEEspan, automata select their parents with the highest selection probability, and this value has converse relation to distance parameter, so the node with less distance has higher priority to be selected as parent that causes the parent with higher energy and less distance is selected.

As shown above, LPT tree has longer branches, because of not regarding distance parameter at all, while in Espan which regard distance as main parameter, tree has shorter branches. While in this work, branches are between these two bounds.

By considering distributed algorithms, and apply tree construction cost, consumed energy to transmit and receive routing packet, we evaluate the performance of proposed algorithm by considering reconstruction section.

As describe earlier, the algorithm with automata learning property consumes less energy as a result of prevent flooding routing packet. By considering learning property, transmission volume is decreased that leads to more

power saving. To show this, the remaining energy of network nodes is measured. In Figure 5, sum of the remained energy of all nodes in network is plotted versus number of nodes.

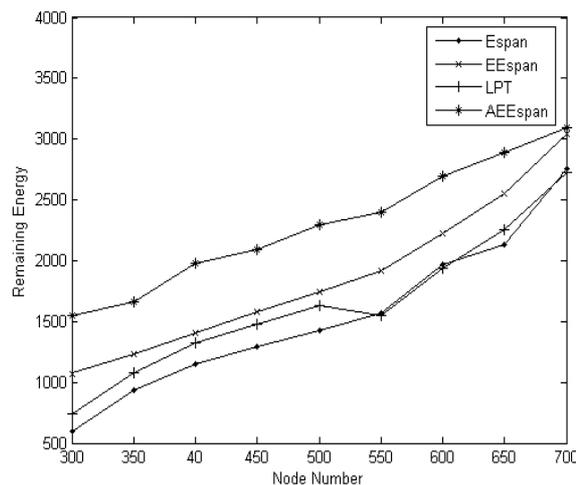


Figure 3. The remaining energy of algorithms without considering tree reconstruction cost.

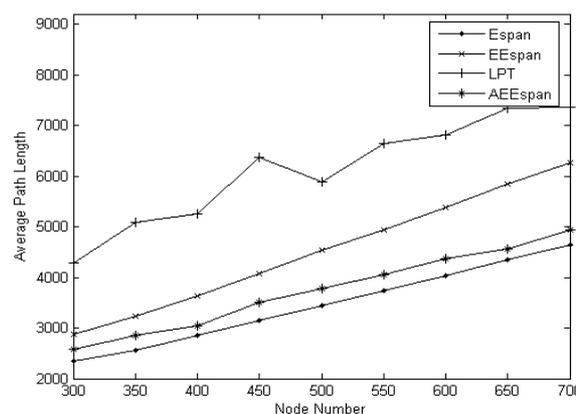


Figure 4. Average hop count to root.

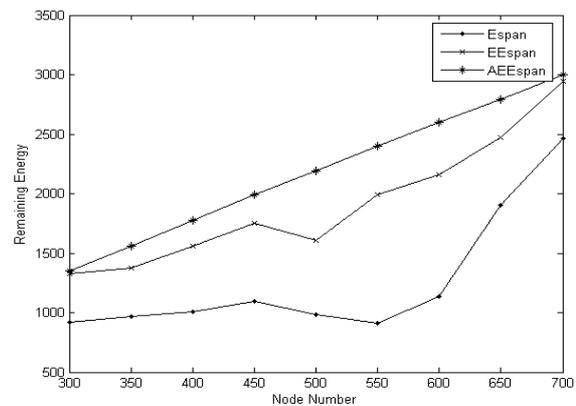


Figure 5. The remaining energy of distributed algorithms with considering tree reconstruction cost.

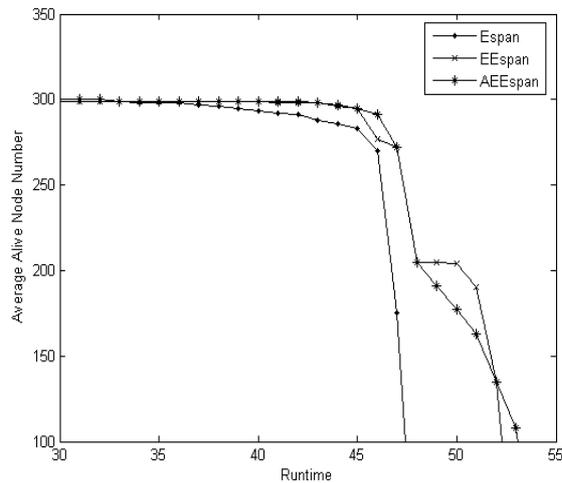


Figure 6. Number of alive nodes at N=300.

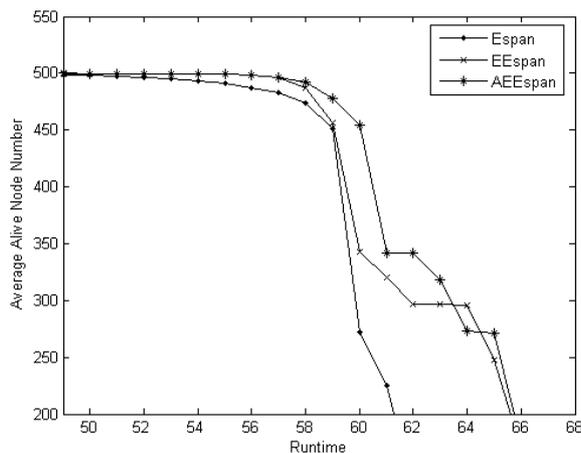


Figure 7. Number of alive nodes at N=500.

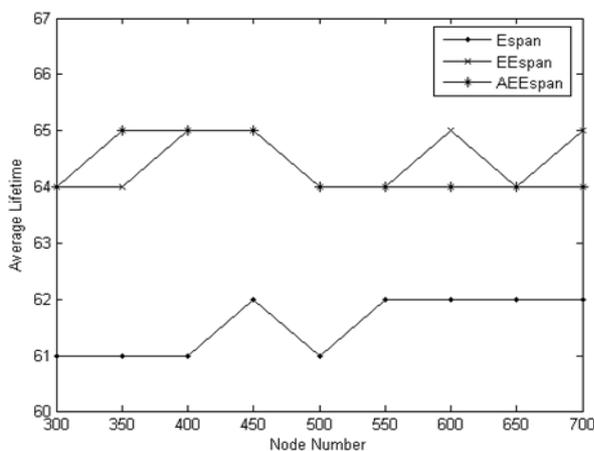


Figure 8. Average lifetime comparison.

We measure the number of alive nodes after each simulation round in Figure 6, 7 when $N = 300$, and 500

nodes, respectively. As in AEEspan, automata select a parent with the highest selection probability which has direct relation to energy parameter, so the nodes with low energy remain a longer time in the network rather than the other algorithms.

For example, in Figure 7, three algorithms work similarly by the round 52, but after that, nodes in Espan tree start to fail sharply, while in AEEspan, nodes failing start later and in slighter manner.

As mentioned before, energy efficiency is a main goal of algorithms in wireless sensor networks. By decreasing energy consumption that led to prevent from failing network nodes, network's coverage whether spatial or temporal is supported better and network's lifetime increases. AEEspan algorithm by decreasing transmission volume, can meet this goal.

In Figure 8, for these algorithms, the average lifetime is plotted versus the number of nodes. The results are obtained after 20 different simulation trials. As seen in Figure 8, the proposed algorithm has higher lifetime than other algorithms. Based on lifetime definition, lifetime has direct relation to alive node numbers.

6. Conclusion

One of the most important limitations of the wireless sensor networks is the network's energy. Aggregation algorithms have a considerable role in decreasing the energy consumption due to the reduction of the transmitted data volume. Aggregation algorithms construct the Aggregation tree based on the algorithm parameters, and determine the transmitting path of the root for each group. In this paper, the tree construction is presented based on automata. An automaton is an able-to-learn structure which tries to choose the best path to send the data to the root by getting feedback from the environment. Also, by prevent from flooding routing packet in entire network, proposed algorithm consume less energy. As the simulation results show, the proposed algorithm has more lifetime and lower energy consumption.

7. References

- [1] Z. Eskandari, M. H. Yaghmaee, and A. H. Mohajezade, "Energy efficient spanning tree for data aggregation in wireless sensor networks," ICCCN, 2008.
- [2] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey, computer networks," Computer Networks Journal, 2002.
- [3] Y. Hu, N. Yu, X. H. Jia, "Energy efficient real time data aggregation in wireless sensor network," IWCMC, 2006.
- [4] M. Lee and V. W. S. Wong, "An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks," IEEE, 2005.
- [5] J. N. Al-Karaki, and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey, supported by the ICUBE initiative of Iowa State University, Ames.

- [6] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE, 2004.
- [7] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," International Workshop on Distributed Event-Based Systems, 2002.
- [8] M. Lee and V. W. S. Wong, "LPT for data aggregation in wireless sensor networks," IEEE GLOBECOM, 2005.
- [9] W. Zhang and G. Cao, "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," IEEE, 2004.
- [10] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A low latency and energy-efficient algorithm for convergecast," IEEE GLOBECOM, 2003.
- [11] Z. eskandari, M. H. Yaghmaee, A. H. Mohajerzade, "Automata based energy efficient spanning tree for data aggregation in wireless sensor networks," IEEE ICCS, 2008.
- [12] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A lowlatency and energy-efficient algorithm for convergecast," IEEE GLOBECOM, 2003.
- [13] Y. P. Chen, A. L. Liestman and J. Liu, "A hierarchical energy-efficient framework for data aggregation in wireless sensor networks," IEEE, 2006.
- [14] P. Radivojac, U. Korad, K. M. Sivalingam and Z. Obradovic, "Learning from class-imbalanced data in wireless sensor networks," IEEE VTC, Fall 2003.
- [15] P. Beyens, M. Peeters, K. Steenhaut, and A. Nowe, "Routing with compression in wireless sensor networks: A Q-learning approach," AAMAS, 2005.
- [16] M. Esnaashari, M. R. Meybodi, "A learning automata based data aggregation method for sensor networks," CSICC, 2007.
- [17] M. Ankit, M. Arpit, T. J. Deepak, R. Venkateswarlu and D. Janakiram, "TinyLAP: A scalable learning automata-based energy aware routing protocol for sensor networks," IEEE, 2006.
- [18] Y. P. Chen, A. L. Liestman, J. Liu, "Energy-efficient data aggregation hierarchy for wireless sensor networks," Proceedings of the 2nd Int'l Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks, 2005.
- [19] J. Kamimura, N. Wakamiya, and M. Murata, "Energy-efficient clustering method for data gathering in sensor networks," BROADNETS, 2004.
- [20] S. Upadhyayula and S. K. S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks," Elsevier, 2006.