1

Data Aggregation Tree Construction: Algorithms and Challenges [1]

Zahra Eskandari[2] and Fatemeh Ayughi
Islamic Azad University of Kashmar, Islamic Azad University of Qazvin
Iran

1. Introduction

Wireless Sensor Networks (WSNs) are a set of communication networks which consist of sensor nodes. These nodes sense the events occurred in their own area and transmit the data related to these events to the sink node. The sink node is considered as the gateway. The sink is in relation with the end user and disseminates the queries requested by the user (Akyildiz et al, 2002) in the network. Receiving the query, the nodes of the network should sense the query's required data from the environment and send them to the sink.

The first advantage of wireless technology is easy deployment of sensors, so that outdoor environments like forests, deserts and the wildness in general can be covered. The second advantage is the possibility of networking mobile nodes. The application scenarios are various, ranging from the obvious military applications, such as distributed battlefield sensing or frontier control, to peaceful and civilian uses. Examples are: habitat monitoring (birds, whales), home intelligence (e.g. local climate control and smart appliances), biomedical, patient tracking, disaster relief, surveillance, fire control, agricultural, and industrial control (Cantoni et al, 2006).

WSNs have specific characteristics. In these networks, the nodes are randomly deployed in the environment, i.e. the geographical locations of these nodes are undetermined (Eskandari et al a, 2008) and these nodes are inaccessible. Furthermore, the nodes are deployed in the environment densely. These nodes have generally low capability for processing and storing. So the tasks that the nodes perform should not be computationally complex.

Furthermore, one of the main constraints in these networks is energy resource due to size and cost limitation in their nodes (Lee & Wong c, 2006), so, the tasks should be energy efficient. Up to now, many attempts have been made to minimize energy consumption (Chlamtac & Kutten, 1987; Chlamtac & Weinstein, 1991; Heinzelman et al, 2000; Min & Chandrakasan, 2001; Upadhyayula et al, 2003; Krishnamachari et al, 2002; Intanagonwiwat et al, 2004).

In monitoring application, the sensor nodes sense data from the environment periodically and transmit these data to the sink node. The nodes in the network are densely developed,

[1] This work is supported by Islamic Azad University of Kashmar.

[2] Professor of computer engineering department, Islamic Azad University of Kashmar.

so the information associated with an event is sensed by more than one sensor node. The nodes transmit the redundant information to the sink (Liang & Liu, 2006).

Transmission of these redundant data wastes energy. As energy resources are the most important limitation of WSNs and data transmission is the most costly function in the network. This leads to decrease in the node's power, quickly (Akyildiz et al, 2002). After some rounds, network nodes energy is finished and this leads to cases in which the network can not work anymore. Regarding the above mentioned points, in order to increase the network's lifetime, the number of transmitted data packets should be minimized (Akyildiz et al, 2002; Eskandari et al a, 2008).

As described in (Upadhyayula & Gupta, 2006), a 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. And, also based on this work, the lifetime of a tree is defined as the number of rounds that can be performed before the failure of certain percentage of total nodes.

Based on energy model described in (Kamimura et al, 2004), a sensor node consumes E_{elec} (J/bit) in transmitter or receiver circuitry and Eamp (J/bit/m^2) in transmitter amplifier to achieve an acceptable signal noise ratio. A sensor node expends energy ETij (k) or ERi(k) in transmitting or receiving a k-bit packet to or from distance distij, given by the following equations:

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

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

The exponent λ heavily depends on the communication medium (Upadhyayula & Gupta, 2006). As described in (Younis & Fahmy, 2004) if aggregation function is simple, the energy consumption for data aggregation will be negligible.

1.1 Data Aggregation

A number of mechanisms called aggregation algorithms are suggested in order to omit the redundant data. Aggregation algorithms, after receiving data from several sensors, process data and omit the redundancy and send the result of aggregation to the sink (Liang & Liu, 2006). Due to the reduction in data volume, these algorithms decrease the energy consumption (Lee & Wong a, 2005).

Therefore the networks which perform aggregation have more life time (Eskandari et al a, 2008; Lee & Wong a, 2005) and draw more attention (Eskandari et al a, 2008; Lee & Wong a, 2005; Lee & Wong b, 2005). In addition to mentioned improvements, aggregation decreases collision and retransmission delay (Zhu et al, 2006).

Data aggregation is performed during routing in wireless sensor networks. Finding the route from several nodes to the sink in a way that maximizes the shared path and redundancy removing is one of the main objectives in these protocols (Liang & Liu, 2006).

In aggregation algorithms, we must construct aggregation spanning tree (Lee & Wong a, 2005). The spanning tree is a tree which contains all network nodes and doesn't have any loop.

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.

2. Aggregation Tree Construction

As a result of energy saving of data aggregation, different aggregation algorithms have been presented. In this section, we review them briefly and compare their efficiency, and then we introduce a new algorithm, describe it and evaluate its efficiency. Finally, we consider a new challenge, i.e. tree construction cost.

2.1 Recent Works

In (Krishnamachari et al, 2002), the 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 show the existence of polynomial special cases.

As presented in (Zhang & Cao, 2004), 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 (Upadhyayula et al, 2003), the sink saves the entire network state and then by considering link cost, in centralized form, constructs the tree with minimum cost. In cluster algorithm (Younis & Fahmy, 2004), 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 (Chen et al, 2005).

Espan (Lee & Wong a, 2005) 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. Each node selects the closest neighbors to root as its parent. If there are multiple neighbors with equal distance, the node which has the most remaining energy is selected as parent.

As Espan protocol considers distance as main parameter and remaining energy as second, one of the most important problems of Espan is that the nodes with the least distance to root maybe selected as parent by many nodes. So these nodes consume their energy quickly and then they will fail sooner than other network nodes, so the network cannot cover region completely.

In LPT (Lee & Wong b, 2005) after selecting the node with most energy as root, each node selects neighbors with the most energy as parent and its parent forwards its data to the sink. In the mentioned algorithm, when a node in the tree fails, the tree will be reconstructed. LPT aims to prolong the lifetime of the sources which transmit data reports periodically. But in LPT, the parents may have higher distance to root and this cause more energy consumption. LPT does not consider the distance parameter in parent selection.

We have presented an energy efficient algorithm, which constructs the aggregation tree in (Eskandari et al a, 2008). To prevent failing of 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.

2.2 An Efficient Aggregation Tree Construction Algorithm

In this section, we present an Energy Efficient Spanning tree (EEspan) algorithm which is a new energy efficient algorithm for wireless sensor networks. The current work is a modified version of our former published papers (Eskandari et al a, 2008; Eskandari et al b, 2008). Unlike the algorithms given in (Lee & Wong a, 2005; Lee & Wong b, 2005) which use only one of the distance and energy parameters as the main parameter, to decrease the number of failed nodes and to increase the network lifetime, this algorithm considers both remaining energy and distance parameters.

To control the energy and distance parameters, the algorithm uses path's energy and path's length parameters. Using this strategy, a node with low remaining energy can be alive more than that of Espan protocol. This increases the lifetime of the network and supports better coverage. Also, unlike the LPT algorithm, the presented algorithm prevents selecting a parent with high remaining energy, and far distance to the root.

In fact, the presented algorithm might select a node with higher energy but farther from root as its parent. If the selected neighbor with highest energy is in a distance farther than a threshold, the presented algorithm selects the less energy path. In addition, to provide fairness in energy consumption, the algorithm considers a third parameter which is the maximum number of children. In the presented algorithm, the nodes have a predetermined maximum number of children. Based on (Upadhyayula & Gupta, 2006), if the nodes have the same number of children, we can conclude that the nodes will be prepared to transmit data at the same time and their parent will have to be awake for a shorter duration to collect data from all its children.

An example which helps us to understand the details of the presented algorithm is given in Figure 1.

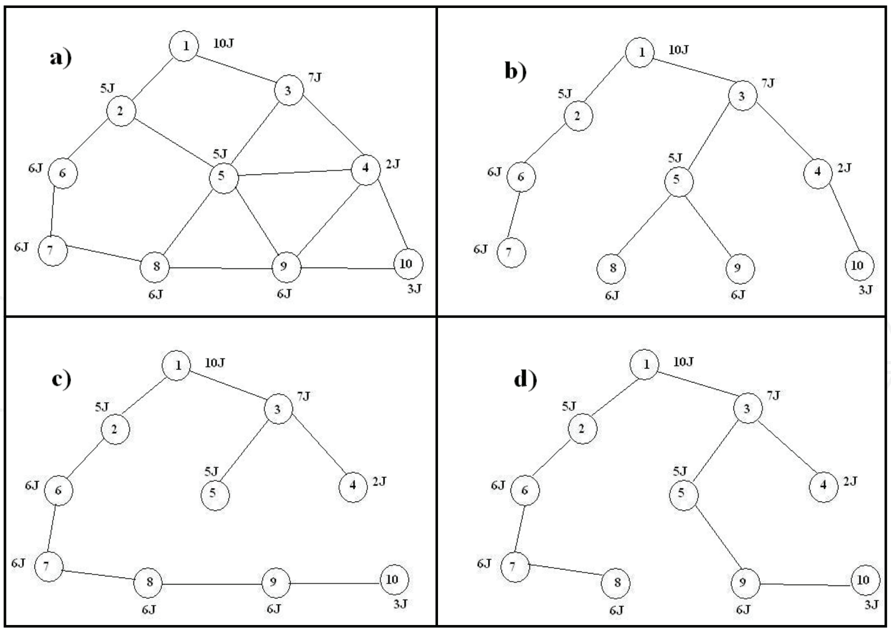


Fig. 1. The spanning trees of different algorithms a) connectivity graph, b) Espan's tree, c) LPT's tree, d) EEspan's tree

In this example, a connectivity graph with 10 different sensor nodes is used. The Espan, LPT and EEspan spanning trees are shown in figure 1. The remaining energy of nodes 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 are equal to 10J, 5J, 7J, 2J, 5J, 6J, 6J, 6J, 6J and 3J, respectively. Suppose that node 10 wants to select its parent.

Using Espan algorithm, node 4 which has the minimum distance to the root will be selected, while in the presented algorithm, node 9 which has more average path's energy is selected as the parent of node 10. The selected parent by Espan algorithm has low energy and fails quickly. As shown in Figure 1(c), LPT's tree has longer path length which causes more energy consumption.

The algorithm is a distributed algorithm which does not need to save global information about the entire network. This makes the presented algorithm more scalable. Furthermore, in the algorithm, routing is done in a multihop manner.

To verify the energy efficiency of the algorithm, here, we evaluate performance of the algorithm. Figure 2 shows the average path length of the three algorithms. At the beginning rounds, the Espan algorithm has lower energy consumption. This is because in this algorithm, nodes transmit data via shortest paths, but by ruining the low power nodes in these paths, data must be transmitted via other paths which may be longer. Since LPT algorithm selects paths by considering only the energy parameter, nodes transmit their data via longer paths which make higher energy consumption.

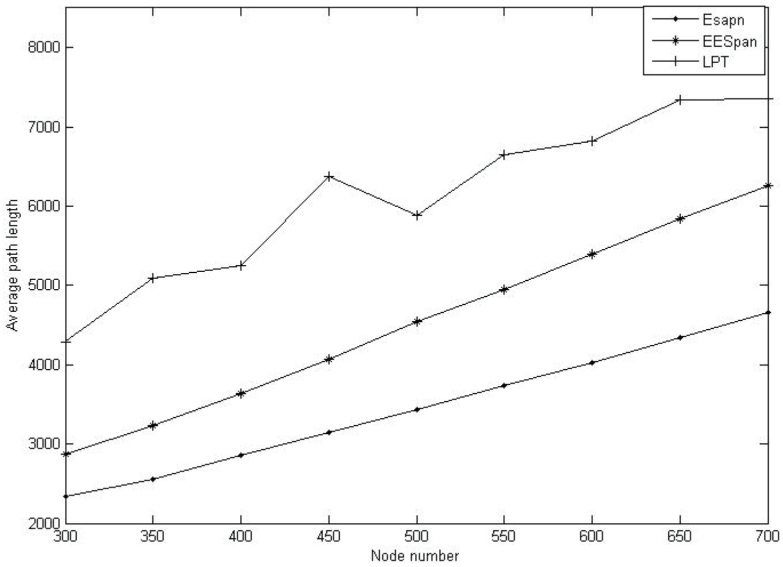


Fig. 2. The average path length of three algorithms

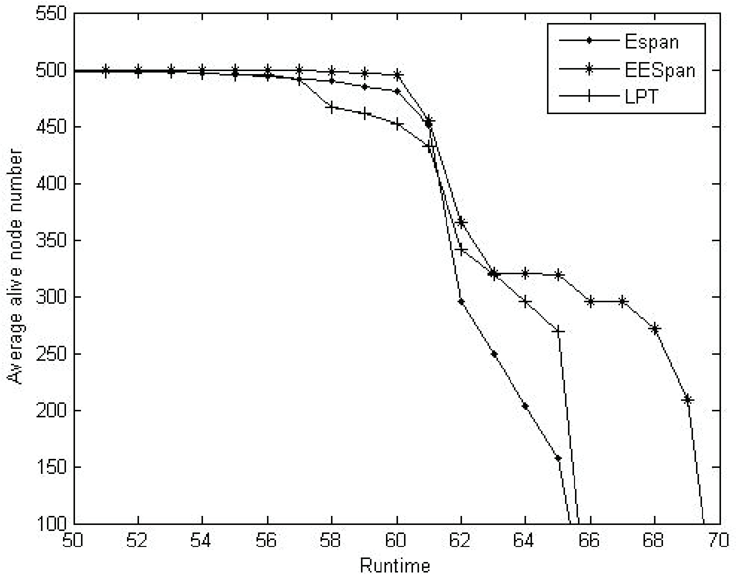


Fig. 3. Number of alive nodes at N=500

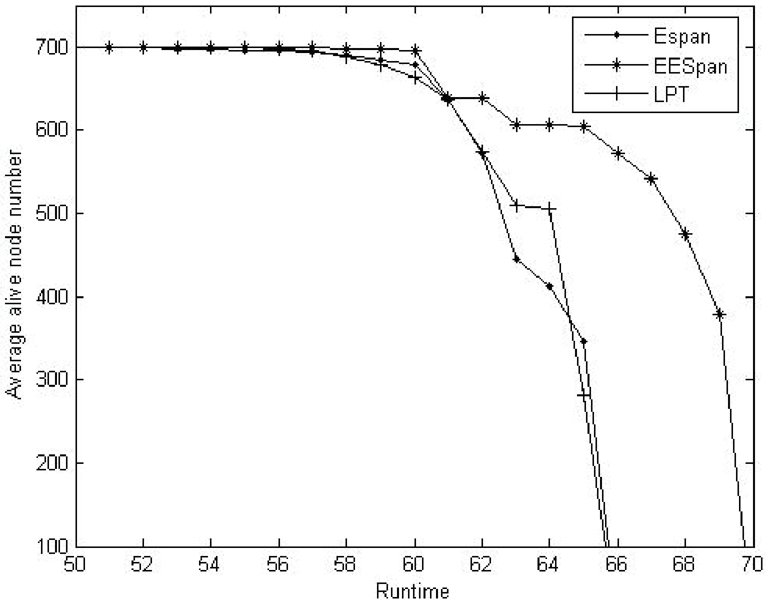


Fig. 4. Number of alive nodes at N=700

In Figures 3, 4, for different values of N, N = 500, and 700 nodes, the average number of alive nodes is plotted versus runtime. As the EEspan selects the nodes with high remaining energy, the nodes with low energy remain longer time in the network. Therefore the number of alive nodes is more than that of Espan algorithm. Furthermore, the LPT algorithm transmits data via the longer paths that leads to consume more energy and the failure of more nodes. More alive nodes can sense environment better, that means the network nodes have better coverage.

In Figure 5, for the three algorithms, the average lifetime is plotted versus the number of nodes. The main objective of all the algorithms is to achieve high energy efficiency. In addition to reducing the energy consumption, balancing energy consumption in nodes is important, too. In Espan algorithm, nodes transmit data via the smallest paths, but this leads the low power nodes in these paths to fail quickly and so the network's lifetime is decreased. To balance energy consumption in the network, the EEspan algorithm operates in an energy aware and transmit data via paths with more energy. Note that EEspan algorithm considers the path length to find the best tree.

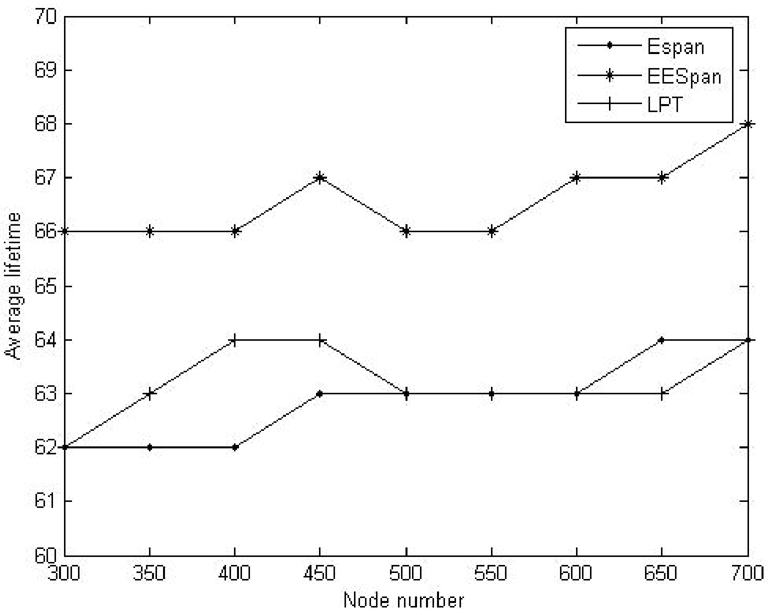


Fig. 5. Average lifetime comparison

2.3 Aggregation Tree Construction Cost

Since the status of the network is dynamic, like routing algorithms, 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 the aggregated data to sink. And, also they should construct the aggregation tree periodically. To construct an aggregation tree, at the beginning of each period, routing packets are flooded into the entire network to inform all nodes. After this step, each node selects the best path towards the sink node and transmits data via the selected path until the next period. When a timer is expired or some nodes fail in the network, the new aggregation tree must be constructed (Lee & Wong a, 2005; Lee & Wong b, 2005). Since the node's energy is limited, transmitting and receiving this volume of routing information is not a good solution to construct an aggregation tree. This overhead causes a lot of energy consumption. So, some nodes run out of energy quickly and fail. This causes the network to be disconnected.

3. Reconfiguration

To solve the mentioned problems, in this section we introduce reconfiguration property; if a node in the aggregation tree fails, and a part of the tree is disconnected, only this part of tree starts to reconstruct locally, so it is not necessary to flood routing packets into the entire network. To do this, each node uses the environment feedbacks, and updates its information on its neighbors. In this section we introduce an automata-based algorithm to reconstruct spanning tree, the current work is published in [Eskandari et al b, 2008; Eskandari et al c, 2009).

3.1 Recent Works

Cluster based algorithms (Younis & Fahmy, 2004) needs only local information to construct the aggregation tree; therefore they transmit fewer packets to construct the aggregation tree. In (Radivojac et al, 2003), the presented algorithm uses machine learning to transmit the sensed data to the sink. Learning algorithm is executed in the sink and its result is propagated throughout the network. In (Beyens et al, 2005) Q-leaner is used to construct aggregation tree to maximize aggregation ratio.

In (Esnaashari & Meybodi, 2007), an algorithm to construct the automata-based aggregation tree, is presented. In this algorithm, in which each node is equipped with an automaton, the automaton selects a path for transmitting data via the path whose aggregation ratio is maximized. In (Ankit et al, 2006), the algorithm considers an automaton for each node, which selects a path to transmit data to the sink in accordance with network conditions.

3.2 An automata Based Aggregation Tree Reconstruction Algorithm

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. The 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 follows: 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. 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 shown in figure 6, an automaton is learned based on the feedback of the environment.

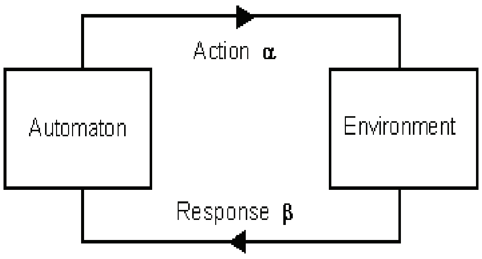


Fig. 6. learning automata

In automata-based algorithms (Ankit et al, 2006; Esnaashari & Meybodi, 2007), 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 the root.

In (Esnaashari & Meybodi, 2007) after receiving data, the 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 the network parameters, the selection probability is increased for the next step, but if the selected path was not optimal, the selection probability is decreased for the 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 till the end of the network's lifetime. By using this learning property of automata, the algorithm prevents flooding the routing packets periodically, at the same time, by using ack information, nodes become aware of changes in network topology and paths are updated.

The presented algorithm in this section works as follows: at the beginning, routing packets are flooded into the network. Each neighbor, after receiving these packets, considers the sender as a new entry in its routing table.

This sending/receiving is performed in the 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:

$$Sel - prob = C_i * \frac{energy_j}{dis\tan ce_j} \tag{3}$$

In equation 3, Ci is a constant which is calculated by node and is dependent on the sum of energy and distances to the root of entries in routing table of node i.

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 order to update the automata, each node must collect some information from the network. By using this information, an automaton becomes aware of the network changing. In (Ankit et al, 2006) 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 (Esnaashari & Meybodi, 2007) to decrease this overhead, acknowledgment is sent after some data transmissions.

But, transmitting these additional data leads to waste of energy because parent's energy becomes less than other nodes in the neighborhood after some rounds. So, we can improve algorithm performance by working as follows: if a node in the aggregation tree fails 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 the aggregation tree globally and periodically.

By using this strategy the tree is reconstructed when it is needed, and reconstruction packet broadcasts locally. This leads to reduction in data transmission in the network and power saving.

Reconstruction property is an important section in the 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.

In this section, to evaluate the performance of the presented algorithm, we compare it with other algorithms (Lee & Wong a, 2005; Lee & Wong b, 2005; Eskandari et al a, 2008).

At the first simulation trial, to evaluate the energy efficiency of the presented algorithm, the automata-based Energy Efficient Spanning tree (AEEspan), we measure remained energy of the network nodes. In figure 7, sum of the remaining energy of all nodes in network is plotted versus the number of nodes for four algorithms.

Since LPT algorithm selects paths by considering only energy parameter, nodes transmit their data via longer paths which cause 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 (Eskandari et al a, 2008) and AEEspan, nodes consume less energy, because in these algorithms, the tree is constructed by applying a reasonable relation between energy and distance parameters.

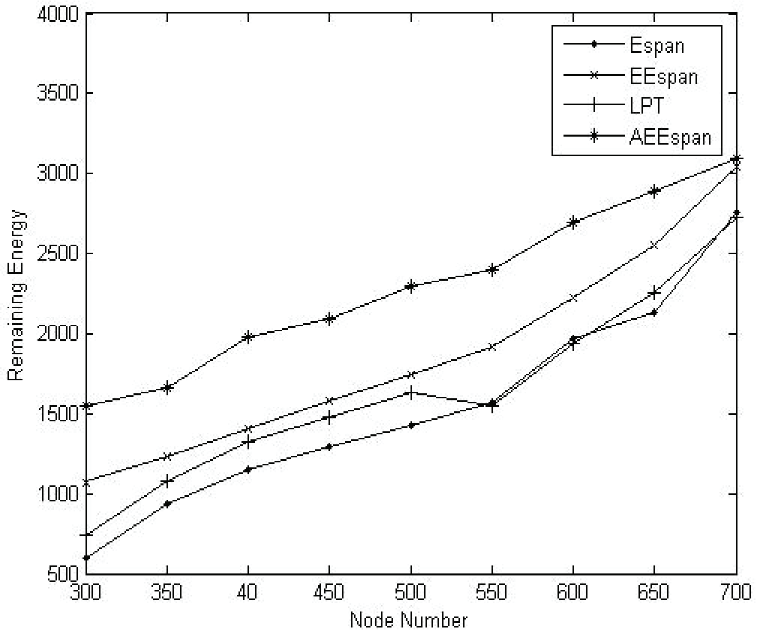


Fig. 7. The remaining energy of algorithms without considering tree reconstruction cost

In figure 8, 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, the tree has shorter branches. While in this work, branches are between these two bounds.

As described earlier, the algorithm with automata learning property consumes less energy as a result of preventing from flooding routing packet. By considering learning property, transmission volume is decreased, that leads to more power saving. To show this, the remaining energy of the network nodes is measured. In figure 9, the sum of the remained energy of all nodes in the network is plotted versus the number of nodes.

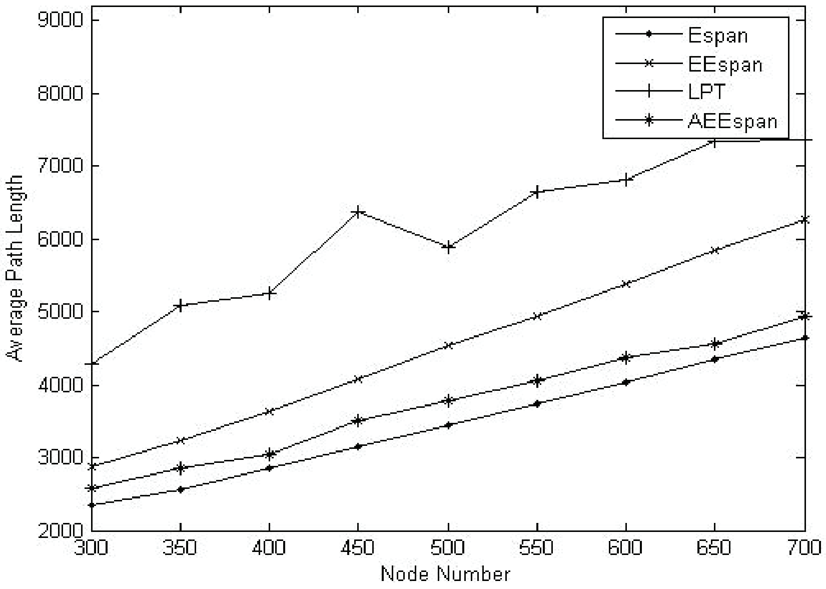


Fig. 8. Average hop count to root

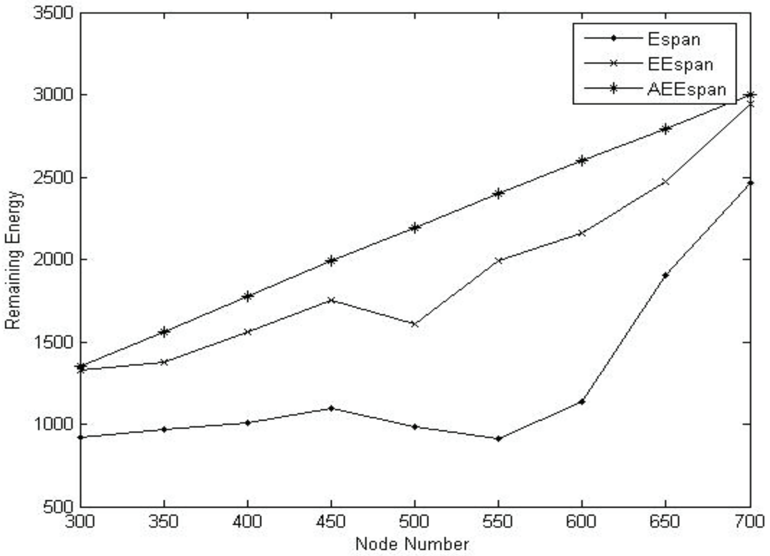


Fig. 9. The remaining energy of distributed algorithms with considering tree reconstruction cost

We measure the number of alive nodes after each simulation round in figures 10 and 11 when N = 300, and 500 nodes, respectively. As in AEEspan, the 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.

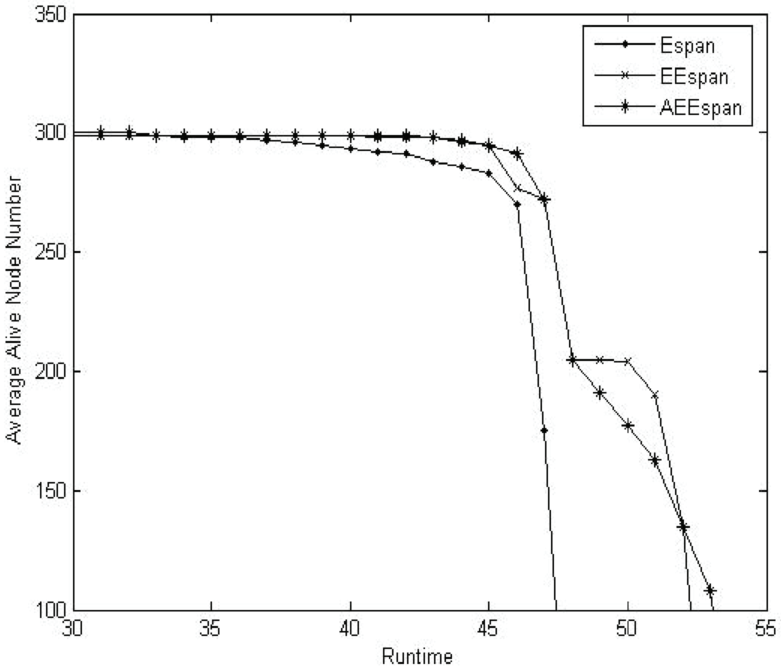


Fig. 10. Number of alive nodes at N=300

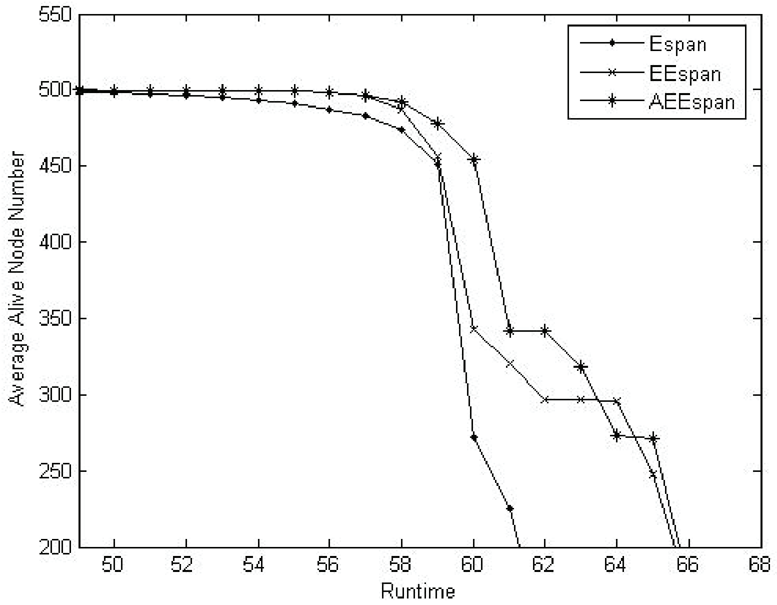


Fig. 11. Number of alive nodes at N=500

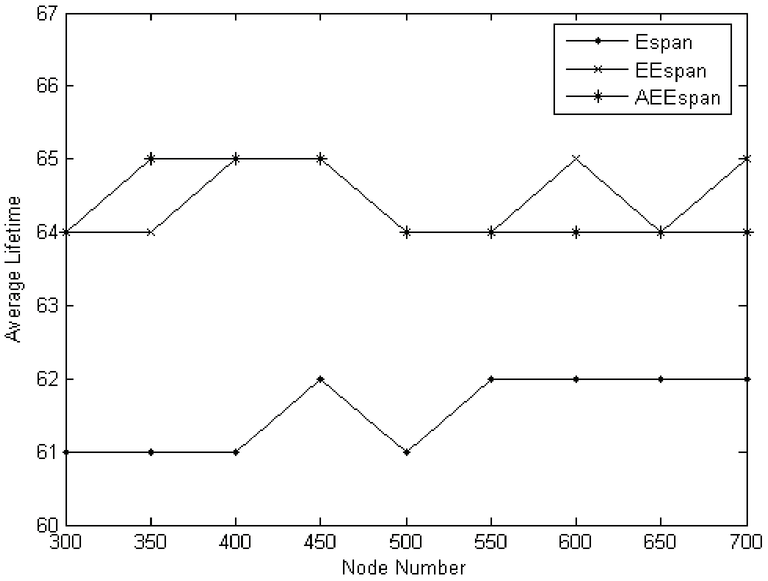


Fig. 12. Average lifetime comparison

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

In figure 12, for these algorithms, the average lifetime is plotted versus the number of nodes. The results are obtained after 20 different simulation trials. As shown in figure 8, the presented algorithm has higher lifetime than the other algorithms. Based on the lifetime definition, lifetime has direct relation to alive node numbers.

4. Conclusion

One of the most important constraints in wireless sensor networks is the energy consumption. Aggregation algorithms have a considerable role in decreasing the energy consumption due to the reduction of the transmitted data volume. Data aggregation has been put forward as an essential paradigm for wireless routing in sensor networks. The idea is to combine the data coming from different sources, eliminating redundancy, minimizing the number of transmissions and thus saving energy. In this work, an energy efficient algorithm to construct the aggregation tree is presented. The algorithm considers both energy and distance to construct the aggregation tree. Simulation results show that the algorithm has better performance than the existing algorithms and also, the algorithm decreases the number of failed nodes and provides higher network lifetime and better coverage. To construct the aggregation tree, routing packets are flooded into the network periodically that leads to waste of energy. To omit this overhead, we introduce automata-based reconfiguration property. 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 preventing from flooding the routing packet into the entire network, the presented algorithm consumes less energy.

5. References

Akyildiz, F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002), Wireless Sensor Networks: A Survey, *Computer Networks Journal*

Ankit, M.; Arpit, M.; Deepak, T.; Venkateswarlu, R. & Janakiram, D. (2006), TinyLAP: A Scalable Learning Automata- Based Energy Aware Routing Protocol for Sensor Networks, IEEE

Beyens, P.; Peeters, M.; Steenhaut, K. & Nowe, A. (2005), Routing with compression in wireless sensor networks: a Q-learning Approah, AAMAS

Cantoni, V.; Lombardi, L. & Lombardi, P. (2006), Challenges for Data Mining in Distributed Sensor Networks, IEEE Computer Society

Chen, Y.; Liestman, A. & Liu, J. (2005), Energy-Efficient Data Aggregation Hierarchy for Wireless Sensor Networks, the 2nd Int'l Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks

Chlamtac, I. & Kutten, S. (1987), Tree-based broadcasting in multi-hop radio networks, *IEEE Transactions on Computers*

Chlamtac, I. & Weinstein, O. (1991), The wave expansion approach to broadcasting in multi-hop radio networks, *IEEE Transactions on Communications*

Eskandari, Z.; Yaghmaee, M.H.; Mohajerzade, A.M. (2008), Energy Efficient Spanning Tree for Data Aggregation in Wireless Sensor Networks, IEEE ICCCN

Eskandari, Z.; Yaghmaee, M.H & Mohajerzade, A.M. (2008), Automata based Energy Efficient Spanning Tree for Data Aggregation in Wireless Sensor Networks, IEEE ICCS

Eskandari, Z.; Yaghmaee, M.H. (2009), AEESPAN: Automata based Energy Efficient Spanning Tree for Data Aggregation in Wireless Sensor Networks, *International Wireless Sensor Network (WSN) journal*

Esnaashari, M. & Meybodi, M. (2007), a learning automata based data aggregation method doe sensor networks, CSICC

Heinzelman, W.R; Chandrakasan, A. & Balakrishnan, H. (2000), Energy-efficient communication protocol for wireless micro sensor networks, International Conference on System Science

Intanagonwiwat, C.; Estrin, D.; Govindan, R. & Heidemann, J. (2002), Impact of network density on data aggregation in wireless sensor networks, International Conference on Distributed Computing Systems

Kamimura, J.; Wakamiya, N. & Murata, M. (2004), Energy-Efficient Clustering Method for Data Gathering in Sensor Networks, BROADNETS

Krishnamachari, B.; Estrin, D. & Wicker, S. (2002), The Impact of Data Aggregation in Wireless Sensor Networks, International Workshop on Distributed Event-Based Systems

Lee, M. & Wong, V. (2005), An Energy-Aware Spanning Tree Algorithm for data aggregation in wireless sensor networks, IEEE

Lee, M. & Wong, V. (2005), LPT for Data Aggregation in Wireless Sensor Networks, IEEE GLOBECOM

Lee, W.M. & Wong, V. (2006), E-Span and LPT for data aggregation in wireless sensor networks, *Elsevier*

Liang, B. & Liu, Q. (2006), A Data Fusion Approach for Power Saving in Wireless Sensor Networks, IEEE

Min, R. & Chandrakasan, A. (2001), Energy-efficient communication for ad-hoc wireless sensor networks, the Thirty-Fifth Asilomar Conference on Signals Systems and Computers

Radivojac, P.; Korad, U.; Sivalingam, K.M. & Obradovic, Z. (2003), learning from class-imbalanced data in wireless sensor networks, IEEE VTC

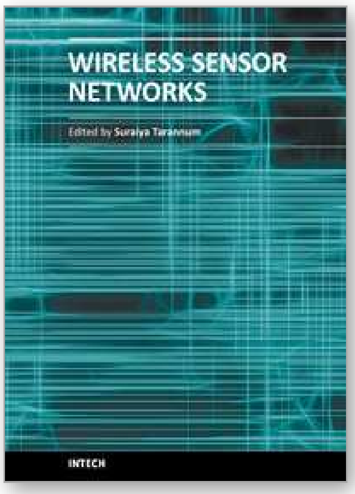
Upadhyayula, S.; Annamalai, V. & Gupta, S. (2003), A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks, IEEE Global Communications Conference

Upadhyayula, S. & Gupta, S.K.S. (2006), Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks, *Elsevier*

Younis, O. & Fahmy, S. (2004), HEED: A hybrid, energy- efficient, distributed clustering approach for ad hoc sensor networks, IEEE

Zhang, W. & Cao, G. (2004), DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks, IEEE

Zhu, J.; Papavassiliou, S.; Kafetzoglou, S. & Yang, J. (2006), An Efficient QoS-Constrained Data Aggregation and Processing Approach in Distributed Wireless Sensor Networks, IEEE


WIRELESS SENSOR NETWORKS
Edited by Suraiya Tarannum
INTECH