Energy Signal Scheduling: A Simple Scheduling Mechanism for Wireless Sensor Networks

Hamid Shokrgue[1] and Arash Nikdel[2]

[1,2]*Sama Technical and Vocational Training College, Islamic Azad University, Shushtar Branch, Shushtar, Iran*
shokrgue.academic@yahoo.com, a.nikdel@khouzestan.srbiau.ac.ir

Abstract

The nodes in Wireless Sensor Networks (WSNs) are usually powered by batteries with finite capacity and one design challenge in sensor networks is to save limited energy resources to prolong the network lifetime. Moreover, the definition of Quality of Service (QoS) and the metrics to evaluate the performance of a wireless sensor network are different from traditional networks. There are different Metrics for quality of service in wireless sensor networks and most of them are application-dependent. One of the most important of them is the number of active nodes. Regarding the sensors redundancy, we can improve the network lifetime by activating enough number of nodes and making the others sleep. Also, proposing a method that uses the nodes energy equally can prolong the network lifetime. In this paper, we propose an energy efficient scheduling method called Energy Signal Scheduling (ESS). Then, we present QoS-aware Energy Signal Scheduling (QoS-ESS). The QoS-ESS method improves ESS by dividing the sensor network to some clusters and applying the ESS for each cluster separately. The effectiveness of the proposed method is evaluated using NS2 simulator. We have simulated our proposed method and compared its functionality to some other methods. Simulation results show high efficiency of the proposed method.

Keywords: *Wireless Sensor Network, Scheduling, Energy consumption, Network lifetime, Quality of Service*

1. Introduction

The Wireless Sensor Networks (WSNs) are designed to conduct military surveillance, disaster prediction, and environment monitor [1, 2, 3]. The nodes in WSNs are usually powered by batteries with finite capacity and it is always impossible to replenish the power [4]. Therefore, the applications are hindered by limited energy supply, and one design challenge in sensor networks is to save limited energy resources to prolong the network lifetime [4, 5, 6]. Thus, a number of studies for decreasing the energy consumption of the sensor network have been performed in recent years [4]. Power saving techniques in wireless sensor networks can generally be classified in two categories [4]:

- Scheduling the sensor nodes to alternate between active and sleep mode
- Adjusting the transmission or sensing radius of the wireless nodes

In this paper, we deal with the problem using first methods [4]. The proposed methods are based on sleep scheduling, where nodes are turned on and off based on certain algorithm to increase the network lifetime.

Due to the importance of wireless sensor networks, there are the large number of studies in the field of network architecture, the design of communicating protocol and the way of decreasing energy consumption. But there is not so very notice to the Quality of Service (QoS) in these networks since the quality of service in various applications in wireless sensor networks is different. In spite of traditional networks, there is not the same metrics for evaluating quality of service in these networks and regarding the application, the essential of quality of service is different. Thus, the definition of quality of service and the metrics to evaluate the performance of a wireless sensor network are different from traditional networks.

The quality of service can be evaluated in two aspects such as application and network. In the field of application, different essentials are considered in the network quality of service. In this field, network coverage [7], event detection [8] and the number of active sensors [9] are considered. From the network aspect, the quality of service depends on data delivering model, i.e. regarding the model in which the network provides data, different essentials are defined for quality of service. There are three main models for delivering data consist of event driven, query driven and continuous delivery models. In event driven model, the network begins to collect and deliver data after happening one event. For example, a network designed to recognize an animal crossing in a part of jungle, uses this kind of data model. In this model, the truth of recognition and delay of its report are the metrics of evaluating quality of service. In the query driven model, the query is prepared from central node and network responses by collecting proper data. In this model, the response time to query and response accuracy are the metrics of quality of service. In continuous deliver model, the nodes send information to the central node continuously in certain time periods. Network coverage is the main metric of quality of service in this model. In this paper, we use the number of active nodes as a metric to evaluate the quality of service. We try to decrease the energy consumption by changing the node mode to active and passive.

The remaining of this paper is organized as follow: related works is explained in Section 2. Proposed method is explained in Section 3. Simulation results are shown in Section 4. Section 5 is the conclusion.

2. Related Works

In *Gur Game*, some of players select an option between “1” and “0” in each round without knowing other player’s decision. At the end of each period, the number of “1” (k) is counted and the amount of reward is calculated based on one reward function ($r = f(k)$). Based on this amount of reward, it is cleared that how possible it is to give the players reward in this period. The purpose of this game is to have optimum number of selection “1”. This optimum number is shown with parameter k^*. The value of reward function is between “0” and “1”. The more k value nears k^* the nearer to “1”. Each player decides by a state machine with $2 \times n$ states. It is shown in Figure 1. This state machine has n conditions with positive label and n conditions with negative label as shown in Figure 1. If the players are in positive condition, they will select “1”; otherwise they’ll select “0”. After getting r value, the nodes change to the condition with larger label with possibility r and change to the condition with lower label with possibility $1-r$. As shown in paper [10], playing this game and after enough period, the number of players who select “1” converge to k^*.

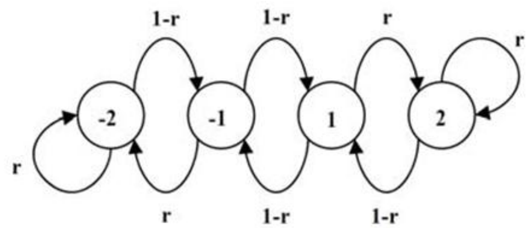

Figure 1. State Machine with *2×n* States

Paper [9] uses *Gur Game* in order to control the number of sensor nodes. In this method, each node has a state machine and based on it the node becomes active or passive. The central node sends the control signal to nodes by a direct channel and controls the quality of service by changing the condition of the nodes state machine. In this method, the receiver of all nodes should wait for winning the reward amount until the end of period. This will decrease the network lifetime.

Paper [11] uses *Ack Automata*. The nodes send based on the possibility determined by their state machine condition. Figure 2 shows the example of *Ack Automata*. As *Gur Game* method, in *Ack Automata* the central node controls the number of sender nodes by sending control signal to the sensors and changes their condition. In this method, the central node sends the proper control signal immediately after receiving each packet and evaluating the number of received packet. So, the sensor receives the control signal from central node immediately after sending the package. As a result, sensor's sender-receiver is passive in most of the time and this leads to decrease energy consumption in comparison with *Gur Game* method. But in this method, as in *Gur Game* method, regarding the nodes decision to send alternately, those which send earlier than others, will win a reward and resend in next periods, so other nodes will not win the reward even they sends packet and their states change to the states with lower possibility. This mechanism causes the k^* first node resend packet in the next periods. This process continues until the energy of these nodes finish while other nodes have still consumed a little energy.

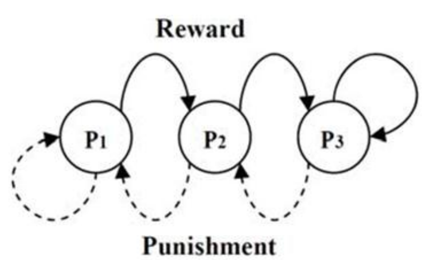

Figure 2. Example of *Ack Automata*

This unbalance energy consumption and early death of sender nodes cause the network not to be able to prepare the required quality of service. Making the nodes energy consumption equal can leads to increase network lifetime. For example suppose a sensor network with 100 nodes and the central node need k^* =65 in each period. So, 65 nodes send continuously in all periods. Therefore the quality of service is prepared ideally in these periods. But immediately after emptying the energy of this 65 nodes and going out of the network, remain 35 nodes will not be able to prepare the quality of service and it is the end of network lifetime while these 35 nodes have not still consume most of their energy. As a result, cooperating these nodes in the previous periods and balancing the energy consumption of the sensor nodes results in increasing the network lifetime considerably.

3. Proposed Method

In traditional methods the only point being paid attention to was to guarantee quality of service by activating the k^* node and sleeping other nodes in order to reduce energy consumption and prolong the network lifetime. These methods have two weaknesses. The first one is that they select k^* node regardless of energy of nodes, but it will be better that the nodes with more energy be selected. The second weakness is the possibility of improper distribution of the active k^* nodes. It means that the active nodes compression is more or less than required value at some locations.

In this paper, we propose an energy efficient scheduling method called Energy Signal Scheduling (*ESS*) with rapid convergence towards activating k^* nodes. Also, we will improve *ESS* by dividing the sensor network to some clusters and applying the *EES* for each cluster separately. Thus, we propose a scheduling method called QoS-aware Energy Signal Scheduling (*QoS-ESS*) which is able to remove both weaknesses mentioned above.

3.1 Energy Signal Scheduling (ESS) Method

The proposed scheduling method is scheduled into periods. In each period, the nodes send its remained energy together with their data to central node. The central node creates a list of the sensor nodes. The central node sorts its list descending based on the remaining energy of the sensor nodes. The energy of k^*th node is called boundary energy (see Figure 3). After that, the central node broadcasts this energy value as Energy Signal (S_{energy}) to sensor nodes.

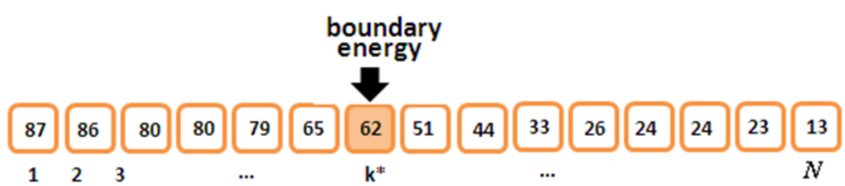

Figure 3. The central node sorts its list descending based on the remaining energy of the sensor nodes. The energy of *kth node is called boundary energy.**

The sensor nodes will send data at next period with probability α (α is a number near to 1), if their remaining energy is equal to or more than S_{energy}. But, the nodes which their remaining energy is less than the S_{energy} will send data at next period with probability β. If the sensor node do not sends until next period, it will become inactive (the receiver of node will be inactive until the end of period) and in the next period will decide according to the same probability. As a result, there will be no waste of energy as *Ack Automata* and unlike *Gur Game* method. As mentioned before, in *Gur Game*, the receiver of all nodes should wait for receiving the reward amount until the end of period. This will decrease the network lifetime.

The relationship between the number of nodes and k^* is shown in (1).

$$\alpha k^* + \beta (N - k^*) = k^* \qquad (1)$$

Where N is the number of nodes and α is a value near 1 (e.g. α=0.9). Therefore the optimum probability β is calculated as (2):

$$\beta = ((1-\alpha)\ k^*) / (N-k^*) \qquad (2)$$

It is considerable that sending is based on probability. If some node which don't sends data and don't receives new S_{energy} value, send once only, regarding its energy which is more than S_{energy}, it will become a constant sender at next periods. It means that regarding its more energy in comparison with S_{energy}, it will send data in probability α (α is a number near 1). This is shown excessively in Figure 4.

Three sample nodes are shown in Figure 4 (a). After sending data to central node and receiving S_{energy} value, the right node becomes inactive often (see Figure 4 (b)). After a while, the left node energy will be consumed gradually, so its energy and boundary energy decreases slowly (see Figure 4 (c)). Moreover, it makes boundary energy be increased and even causes left node to be inactive. But finally the right node also sends data and regarding its great energy it will join to the nodes that their energy is more than S_{energy} for a long time (see Figure 4 (d)).

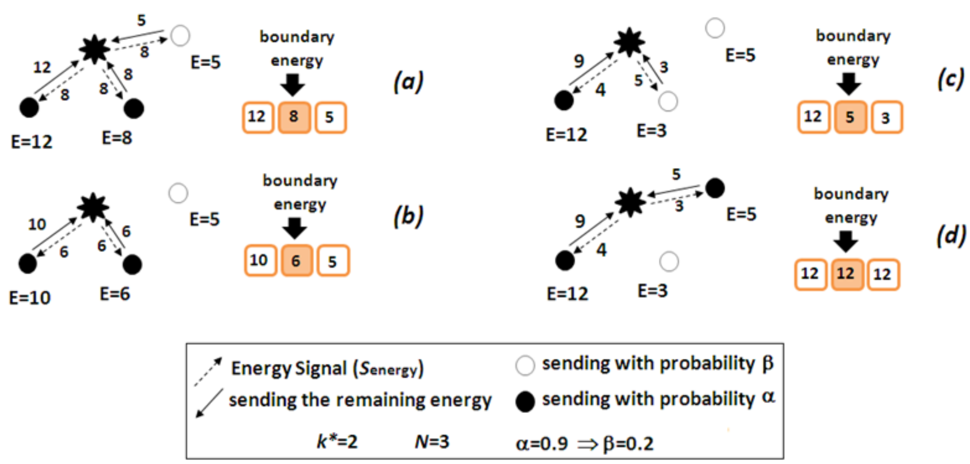

Figure 4. Switching the Nodes between Active and Sleep Mode based on their Remaining Energy

We give the pseudo-code for *ESS* method as follows:

The pseudo-code for central node:

```
Initialize N and k*
In each period do {
   -wait for receiving all data and remaining energy of nodes
   -Update list with received energies
   -Sort list descending
   -calculate boundary energy (i.e. k* th  energy in list)
   -broadcast boundary energy with Senergy signal
}
```

The pseudo-code for each sensor node:

```
Initialize S energy = 0
In each period do {
  -If (remaining energy ≥ S energy):  send data with probability α
   Else: send data with probability β
  - If node sends data: wait for receiving new S energy signal (receiver is on)
    But if don't send data: go to sleep mode (receiver is off)
}
```

3.2 QoS-aware Energy Signal Scheduling (QoS-ESS)

As mentioned before, the weakness of traditional methods and also *ESS* is regardless of evenly distribution of both active and inactive nodes. In this section, we try to apply a more proper distribution of both active and inactive nodes in the sensor network. We divide the sensor network to some clusters and calculate the boundary energy value for each cluster separately. In fact, the *ESS* is a special manner of *QoS-ESS* in which the number of clusters equals one. The *QoS-ESS* consists of two phases as startup and stable phase. We assume that each one knows its own location which can be achieved by using some location system [4, 12].

3.2.1 Startup Phase: At first, the central node broadcasts the special packet ("*please-register"* packet) to all nodes. This packet consists of geographic location of the sensor network and also the number of regions (r).

Each sensor nodes after receiving this packet, calculates its cluster number based on (3) to (7). After that, it informs the central node of its existence by sending a packet called "*registerMe*". This packet consists of its ID and cluster number. This process is shown in Figure 5.

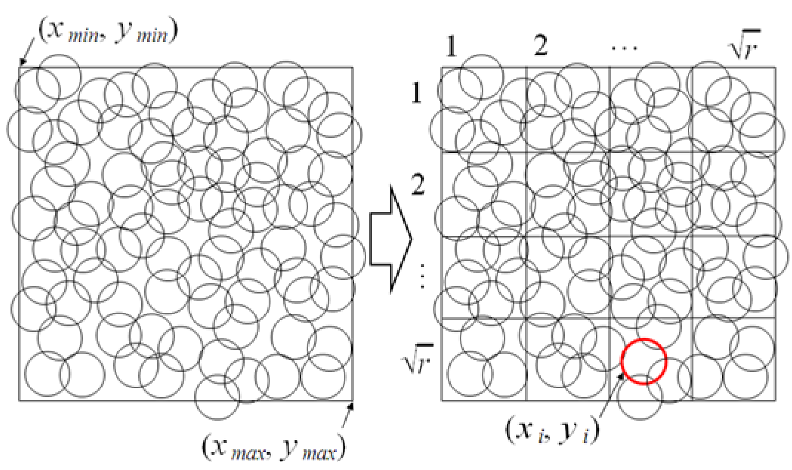

Figure 5. Dividing the Sensor Network to Some Clusters

$$width = \lceil (x_{max} - x_{min}) / \mathrm{sqrt}(r) \rceil \quad (3)$$

$$height = \lceil (y_{max} - y_{min}) / \mathrm{sqrt}(r) \rceil \quad (4)$$

$$row = \lfloor (y_i - y_{min}) / height \rfloor \quad (5)$$

$$column = \lfloor (x_i - x_{min}) / width \rfloor \quad (6)$$

$$Cluster_i = row \times \mathrm{sqrt}(r) + column \quad (7)$$

Where, (x_{min}, y_{min}) and (x_{max}, y_{max}) is network coordinates. Also, (x_i, y_i) is the sensor node coordinates, r is the number of clusters, $Cluster_i$ is cluster number of which the sensor node is belong to it.

After receiving all *"registerMe"* packets, the central node will be informed of the number of all nodes in the sensor network and the number of nodes in each cluster. So, regarding these values, it determines the optimum number of active nodes in each cluster which is necessary for guaranteeing the quality of service. This process is done based on (8) to (10) as follow:

$$N_1+N_2+N_3+\ldots..+N_r = \Sigma^{r}_{i=1} N_i = N \quad (8)$$

$$k_i^* = (k^*/N) \times N_i \quad (9)$$

$$k_1^* + k_2^* + k_3^* + \ldots + k_r^* = \Sigma^{r}_{i=1} k^*_i = k^* \quad (10)$$

Where, N is the number of all nodes, N_i is the number of nodes in i th cluster. k^* is the total number of required active nodes to guarantee the quality of service and k_i^* is the number of required active nodes in i th cluster.

3.2.2 Stable Phase: After performing the startup phase, stable phase is started. In this phase, the nodes send their data during the periods. The first period begins immediately after startup phase. Each node sends its cluster number in addition to its remaining energy amount. The central node places data separately in a descending range based on their remaining energy and determines k_i^* th energy value as $S_{energy}\hat{}_i$ for each i th cluster. Then it broadcasts these boundary energies to sensor nodes. If the remaining energy of the sensor nodes in i th cluster is equal to or more than $S_{energy}\hat{}_i$, they will send their data with probability α at next period. The nodes that their remaining energy is less than $S_{energy}\hat{}_i$ will send their data with probability β_i at next period. If the nodes don't send by next period, they will become inactive and they decide again in the same probability at next period. β_i and $S_{energy}\hat{}_i$ are determined based on (11) and (12):

$$\alpha\, k_i^* + \beta_i (N_i - k^*_i) = k^*_i \quad (11)$$

$$\beta_i = ((1-\alpha) \times k^*_i) / (N_i - k^*_i) \quad (12)$$

The clustering effect on improving the quality of service is shown in Figure 6.

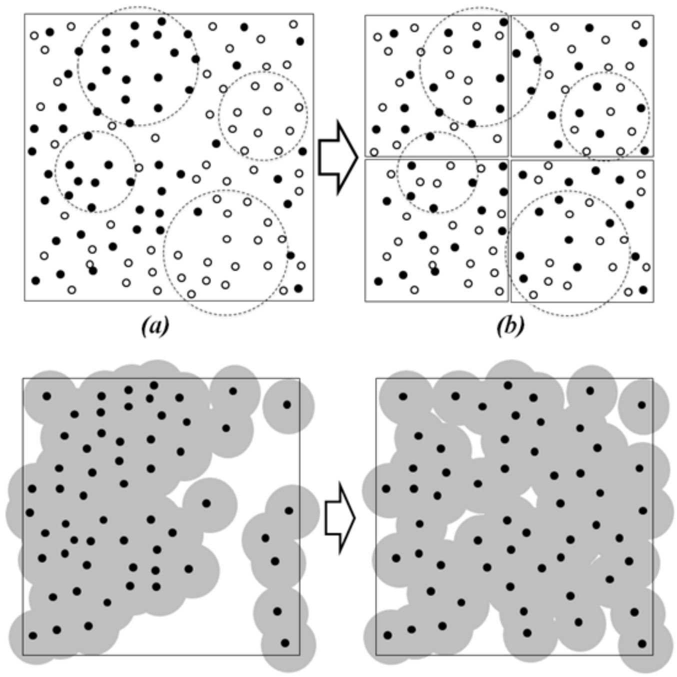

Figure 6. The Clustering Effect on Improving the Quality of Service

In Figure 6 (a), it is shown that half of nodes are active and half of them are inactive. But due to improper distribution of active nodes, performing the *ESS* method will not get the favorite result and both uncovered and redundancy are observed at some points. But this problem is solved by applying *QoS-ESS* method as shown in Figure 6 (b).

4. Experiment Results

In this section, our proposed method is simulated and compared to *Gur Game* and *Ack Automata* methods using NS2 simulator. We use the discussed scenario in papers [9] and [11] in order to compare the proposed *ESS* and *QoS-ESS* methods to *Gur Game* and *Ack Automata* methods. It is supposed that 100 nodes are distributed in the environment randomly. The central node receives packets from nodes in a single-hop manner. Suppose that the central node needs a receiving rate equal to 65 packets in each period (k^* =65). In each period, the nodes send environmental data to the central node and then regarding the number of sender nodes and required quality of service, the central node broadcasts the S_{energy} signal (in *ESS* and *QoS-ESS* methods) and control signal (in *Ack Automata* and *Gur Game* methods).

This experiment is done in 1000 epoch. Also, we supposed that α =0.9, r =9 and N =100. The parameters value for simulation is shown in Table 1.

Table 1. The Parameters Value for Simulation

Epoch	k^*	α	R	N
1000	65	0.9	9	100

Figure 7 and Table 2 show the number of received packets in each epoch. The results show that in about 890 epochs, the *ESS* method receives 65 packets. Also, the *QoS-ESS* method receives 65 packets in about 800 epochs. But, due to death of many sensors, *Gur Game* and

Ack Automata methods are not able to send the required numbers of packets. The *QoS-ESS* attends to the distribution of active nodes in the sensor network. Thus, in *QoS-ESS* method, the quality of service is prepared during fewer epochs in comparison with *ESS* method.

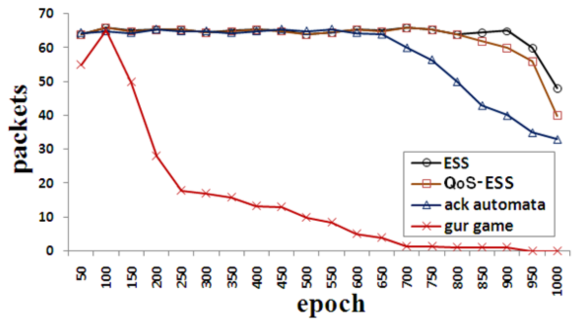

Figure 7. The Number of Received Packet in each Epoch

Table 2. The Number of Received Packet in each Epoch

Epoch	*ESS*	*QoS-ESS*	*Ack Automata*	*Gur Game*
50	64.0	64.0	64.5	55
100	66.0	66.0	65.0	65.2
150	65.0	65.0	64.5	50
200	65.5	65.5	65.5	28
250	65.5	65.5	65.0	18
300	64.5	64.5	65.0	17
350	65.0	65.0	64.5	16
400	65.5	65.5	65.0	13.2
450	65.0	65.0	65.5	13
500	64.0	64.0	65.0	10
550	64.5	64.5	65.5	8.5
600	65.4	65.4	64.5	5
650	65.0	65.0	64.0	4
700	66.0	66.0	60.0	1.5
750	65.5	65.5	56.5	1.3
800	64.0	64.0	50.0	1.1
850	64.5	62.0	43.0	1
900	65.0	60.0	40.0	1
950	60.0	56.0	35.0	0
1000	48.0	40.0	33.0	0

In Figure 8 and Table 3, we compared the number of nodes going out of the network. The important point of this diagram is the difference between the numbers of died nodes in our both proposed method, *Gur Game* and *Ack Automata* methods. In *Ack Automata* method, when the energy of first node depletes, the number of nodes going out of the network increases at almost a constant slope. While in the proposed methods, at first, the number of died nodes increase slowly but increase suddenly at the last epochs. Thus, in the proposed methods, the node's energy depletes almost simultaneously and the energy consumption of the sensor nodes is balanced. This can be evaluated in details in Figures 9 and 10 (Tables 4 and 5). These figures show the minimum, maximum and average of node's energy in different epochs. The noticeable point is that the minimum and maximum energy values are nearer to the average value in the proposed method. It means that in the proposed method, the nodes participation for sending data is equal and so, the energy of all nodes is consumed equally.

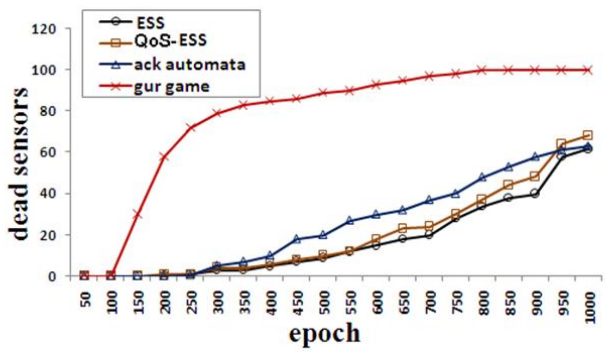

Figure 8. Number of Nodes Going Out of the Sensor Network

Table 3. Number of Nodes Going Out of the Sensor Network

Epoch	*ESS*	*QoS-ESS*	*Ack Automata*	*Gur Game*
50	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0
150	0.0	0.0	0.0	30.0
200	0.0	1.0	0.0	58.0
250	1.0	1.0	0.5	72.0
300	3.0	4.0	5.0	79.0
350	3.0	4.0	7.0	83.0
400	5.0	6.0	10.0	85.0
450	7.0	8.0	18.0	86.0
500	9.0	10.0	20.0	89.0
550	12.0	12.0	27.0	90.0
600	15.0	18.0	30.0	93.0
650	18.0	23.0	32.0	95.0
700	20.0	24.0	37.0	97.0
750	28.0	30.0	40.0	98.0
800	34.0	37.0	48.0	100
850	38.0	44.0	53.0	100
900	40.0	48.0	58.0	100
950	58.0	64.0	61.0	100
1000	62.0	68.0	63.0	100

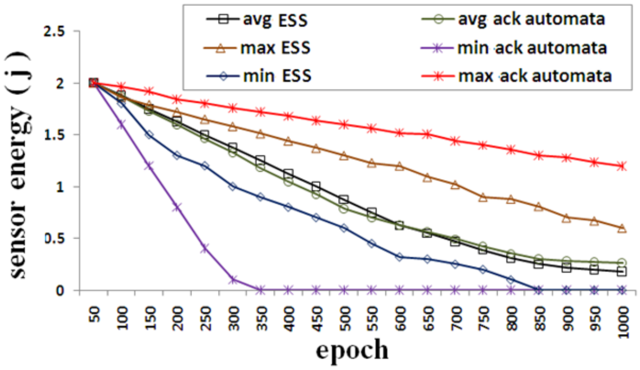

Figure 9. The Minimum, Maximum and Average of Node's Energy in Different Epochs for *ESS* and *Ack Automata* Methods

Table 4. The Minimum, Maximum and Average of Node's Energy in Different Epochs for *ESS* and *Ack Automata* Methods

Epoch	*ESS*			*Ack Automata*		
	Min	Ave.	Max	Min	Avg.	Max
50	2	2	2	2	2	2
100	1.8	1.875	1.86	1.6	1.865	1.96
150	1.5	1.75	1.79	1.2	1.73	1.92
200	1.3	1.625	1.72	0.8	1.595	1.84
250	1.2	1.5	1.65	0.4	1.46	1.8
300	1	1.375	1.58	0.1	1.325	1.76
350	0.9	1.25	1.51	0	1.18	1.72
400	0.8	1.125	1.44	0	1.045	1.68
450	0.7	1.0	1.37	0	0.92	1.64
500	0.6	0.875	1.3	0	0.785	1.6
550	0.45	0..75	1.23	0	0.7	1.58
600	0.32	0.625	1.2	0	0.625	1.52
650	0.3	0.55	1.09	0	0.56	1.51
700	0.25	0.47	1.02	0	0.49	1.44
750	0.2	0.39	0.9	0	0.42	1.4
800	0.1	0.31	0.88	0	0.35	1.36
850	0	0.25	0.81	0	0.3	1.3
900	0	0.22	0.7	0	0.28	1.28
950	0	0.2	0.67	0	0.27	1.24
1000	0	0.18	0.6	0	0.26	1.2

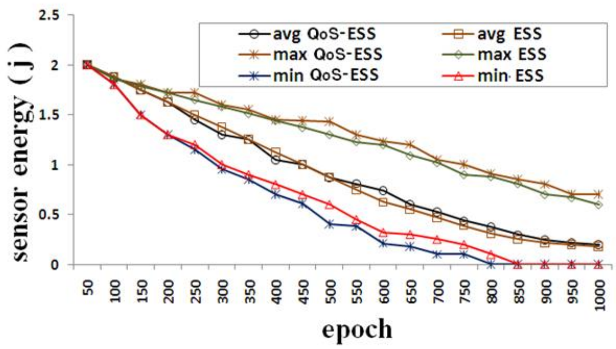

Figure 10. The Minimum, Maximum and Mean Energy of Node's Energy in Different Epochs in *ESS* and *QoS-ESS* Methods

Table 5. The Minimum, Maximum and Mean Energy of Node's Energy in Different Epochs in *ESS* and *QoS-ESS* Methods

Epoch	*ESS*			*QoS-ESS*		
	Min	Ave.	Max	Min	Avg.	Max
50	2	2	2	2	2	2
100	1.8	1.875	1.86	1.8	1.876	1.86
150	1.5	1.75	1.79	1.5	1.74	1.8
200	1.3	1.625	1.72	1.3	1.623	1.72
250	1.2	1.5	1.65	1.15	1.45	1.71
300	1	1.375	1.58	0.95	1.3	1.6
350	0.9	1.25	1.51	0.85	1.25	1.55
400	0.8	1.125	1.44	0.7	1.05	1.45
450	0.7	1.0	1.37	0.61	1	1.44
500	0.6	0.875	1.3	0.4	0.875	1.43
550	0.45	0..75	1.23	0.38	0.81	1. 3
600	0.32	0.625	1.2	0.21	0.74	1.23
650	0.3	0.55	1.09	0.18	0.6	1.2
700	0.25	0.47	1.02	0.1	0.53	1.05
750	0.2	0.39	0.9	0.1	0.44	1
800	0.1	0.31	0.88	0	0.38	0.91
850	0	0.25	0.81	0	0.3	0.85
900	0	0.22	0.7	0	0.25	0.8
950	0	0.2	0.67	0	0.22	0.7
1000	0	0.18	0.6	0	0.2	0.7

The experiments results show that the proposed method can lead to increase network lifetime in comparison with other methods, because the energy consumption in the proposed methods is balanced.

5. Conclusion

One design challenge in sensor networks is to save limited energy resources to prolong the network lifetime. Regarding the sensors redundancy, we can improve the network lifetime by activating enough number of nodes and making the others sleep. Also, proposing a method that uses the nodes energy equally can prolong the network lifetime. Moreover, the definition of Quality of Service (QoS) and the metrics to evaluate the performance of a Wireless Sensor Network (WSN) are different from traditional networks. There are different Metrics for quality of service in wireless sensor networks and one of the most important of them is the number of active nodes. In this paper, we have proposed an energy efficient scheduling method called Energy Signal Scheduling (*ESS*). Also, we have presented QoS-aware Energy Signal Scheduling (*QoS-ESS*). The *QoS-ESS* method improves *ESS* by dividing the sensor network to some clusters and applying the ESS for each cluster separately. We have simulated our proposed method and compared its functionality to some other methods. Simulation results show high efficiency of the proposed method.

References

[1] Y. Li, C. Ai, Z. Cai and R. Beyah, "Sensor scheduling for p-percent coverage in wireless sensor networks", Springer, Science Business Media, LLC, **(2009)** May.

[2] Y. Tian and Z. Tang, "Wireless Meter Reading Based Energy Balanced Steady Clustering Routing Algorithm for Sensor Networks", Advances in Electrical and Computer Engineering (AECE), vol. 11, no. 2, **(2011)**, pp. 9-14.

[3] J. Jia, J. Chen, G. Chang and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm", Elsevier, Computers and Mathematics with Applications, vol. 57, **(2009)**, pp. 1756-1766.

[4] J. Jia, J. Chen, G. Chang, Y. Wen and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius", Elsevier, Computers and Mathematics with Applications, vol. 57, **(2009)**, pp. 1767-1775.

[5] X. Zhang, X. Ding, S. Lu and G. Chen, "Principles for Energy-Efficient Topology Control in Wireless Sensor Networks", In: WiCom '09. 5th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, **(2009)** September 24-26.

[6] T. Qiu, F. Xia, L. Feng, G. Wu, and B. Jin, "Queueing theory-based path delay analysis of wireless sensor networks", Advances in Electrical and Computer Engineering (AECE), vol. 11, no. 2, **(2011)**, pp. 3-8.

[7] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks", IEEE Infocom, **(2001)**, Anchorage, Alaska, USA, pp. 1380-1387.

[8] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks", International Conference on Mobile Computing and Networking, **(2001)**, Rome, Italy, pp. 139-150.

[9] R. Iyer and L. Kleinrock, "QoS control for sensor networks", International Conference on Communications, **(2003)**, Anchorage, Alaska, USA, pp. 517-521.

[10] B. Tung and L. Kleinrock, "Using Finite State Automata to Produce Self-Optimization and Self-Control", IEEE Transactions on Parallel and Distributed Systems, vol. 7, **(1996)**, pp. 439-448.

[11] J. Frolik, "QoS Control for Random Access Wireless Sensor Networks", Wireless Communication and Networking Conference, **(2004)**, Atlanta, Georgia, USA, pp. 1522-1527.

[12] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less low-cost outdoor localization for very small devices", IEEE Personal Communications, vol. 7, **(2000)**, pp. 28-34

Authors

Hamid Shokrgue received the B.S. degree from Jahrom University, Iran, in 2009 and M.S. degree from Islamic Azad University, Dezful, Iran in 2012, both in computer engineering. His research interests include Ad-hoc and sensor networks.

Arash Nikdel received the B.S. degree in Computer Engineering from the Islamic Azad University and the M.S. degree in Computer Engineering from Ahvaz Science and Research Branch in Iran, in 2007 and 2011, respectively. His research interests include computer networks and evolutionary algorithms.