



# Data Aggregation for Increase Performance of Wireless Sensor Networks Using Learning Automata Approach

Maryam Tamiji<sup>1</sup> · Saeed Nasri<sup>2,3</sup>

Published online: 20 April 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Wireless sensor networks, a new generation of networks, are composed of a large numbers of nodes and the communication between nodes takes place wirelessly. The main purpose of these networks is collecting information about the environment surrounding the network sensors. The sensors collect and send the required information. There are many challenges and research areas concerned in the literature, one of which is power consumption in network nodes. Nodes in these networks have limited energy sources and generally consume more energy in long communication distances and therefore run out of battery very fast. This results in inefficacy in the whole system. One of the proposed solutions is data aggregation in wireless networks which leads to improved performance. Therefore, in this study an approach based on learning automata is proposed to achieve data aggregation which leads to dynamic network at any hypothetical region. This approach specifies a cluster head in the network and nodes send their data to the cluster head and the cluster head sends the information to the main receiver. Also each node can change its sensing rate using learning automata. Simulation results show that the proposed method increases the lifetime of the network and more nodes will be alive.

**Keywords** Wireless sensor network · Learning automata · Data aggregation · Network lifetime · Clustering · Sensing rate

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s11277-019-06395-x>) contains supplementary material, which is available to authorized users.

---

✉ Saeed Nasri  
s\_nasri@iaun.ac.ir  
Maryam Tamiji  
tamiji.1987@gmail.com

<sup>1</sup> Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

<sup>2</sup> Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

<sup>3</sup> Digital Processing and Machine Vision Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran

## 1 Introduction

Today, life without wireless communication is unbelievable. Technology advancement and the creation of circuits has led to the use of wireless circuits in most electronic devices. Therefore, wireless networks have emerged as a revolution in every aspect of our lives. Wireless network has unique characteristics which make them different from the other networks. Due to the reduced size and cost of sensors, Wireless sensor networks (WSNs) have quickly become important in the field of network systems [1].

WSNs are created to receive information from the environment and send data to the base station (BS) which have many applications in military environments, medical care, fire alarm and industries. Due to the high aggregation of data in WSNs, methods should be proposed for the optimum use of wireless sensor to increase the lifetime of the networks. In this regard, various techniques such as clustering algorithms are presented so that members of clusters can send the collected information to the cluster heads and then cluster heads collect these data and submitted to the BS. Therefore, as choosing cluster heads and their distribution in the network require an improved algorithm, the necessity of research in this field is highlighted, thus, the necessity of this piece of research in this field considering the pivotal role of these networks is emphasized [1].

As sensors are tiny and have limited power source, they run out of power due to large power consumption and it leads to reduced efficacy or the failed system [2]. Therefore, designers are constantly looking for ways to reduce power consumption and longer lifetime in this type of networks. One of methods applied to achieve this goal is network clustering [3]. Clustering prevents all nodes send their data to the source and therefore the cluster heads undertake this responsibility. Since nodes are closer to the cluster heads than the original source, they need less power to send data, and it increases the network lifetime [4].

Using clustering technique in WSNs is challenging [4–7]. How some nodes are chosen as cluster heads, or the number of cluster heads in each network, or whether cluster heads should be stable or changing during network functioning are examples of such challenges. To have an effective clustering, improved algorithms should be used. One of the new methods in this regard is the use or duplication of learning in organisms. Learning with the use of changes in the efficacy of a system is defined based on past experiences. An important feature of learning systems is their ability to improve their performance over time. Mathematically we can argue that the goal of a learning system for optimizing is a task that is not well understood. In the present paper it is tried to control energy consumption using learning automata (LA) and an efficient method for clustering WSNs.

An efficient strategy for reducing power consumption in wireless networks is optimizing routing algorithms. In new routing algorithms the shortest route is chosen for sending data. But this policy leads to traffic flow in a series of nodes [8]. These algorithms reduce the power consumption of the whole system but they increase the consumption in some nodes, too. Therefore, these nodes lose their function in the system faster than other nodes and cause system failure [2]. Thus finding a way to keep the load balanced across the entire network is essential and it can have an enormous impact on network lifetime.

And an random environment that automata is related to that. The second one is a learning algorithm A LA is made up of two parts. The first one is an random automata with limited number of actions which automata learns how to act efficiently.

In this paper, the LA approach is used for clustering and sensing (LAACS). This clustering algorithm optimizes nodes' power and increases the lifetime of the network by changing the cluster head during the operation of the network and the sampling rate of the

nodes from the consumption environment. Since clustering algorithm is based on LA, they reform themselves over time and thus they can select the best clusters. So the performance of these algorithms is shown over time. It is shown that they increase the lifetime of the network up to 11%.

## 2 Background Research

In the following section, some methods introduced in conjunction with data aggregation and routing algorithms in WSNs, as well as information about LA and their role in WSNs are presented.

In 2015, Sarigiannidis et al. [9] introduced a method to share the bandwidth between the sending and receiving link in WiMAX networks based on LA. In this method, automata divide the bandwidth periodically and update its choice probability function based on feedback received from the node. It is assumed that bandwidth can include 42 symbols. Therefore, the automata's performance includes all the possible ways of dividing 42 symbols between the sending and receiving link (sending 21 symbols, receiving 21 symbols, sending 20 symbols, receiving 22 symbols). The total probability of all automata performance equals to one. Thereof the probability of choosing each action depends on the number of actions in automata and in fact depends on the different ways of dividing bandwidth between two sending and receiving links.

In 2014, Dehkordi et al. [10] introduced a method to allocate communication channel in clustering ad hoc networks based on LA. In this method, the cluster heads are randomly selected and each cluster head has a unique LA. Total performances of each automaton equal to one operation for each node of the cluster. In this method automata choose a node based on action probability vector in automata. If the chosen node has a packet to send to the cluster head, the choice is rewarded. Therefore, automata will learn gradually which node has more packets to send. Another method which works base on LA is the one introduced in [11].

Kashanaki et al. [12] has used LA to control the traffic flow in wireless mesh networks. In this plan, each routing node uses its LA to open a gateway to send its packets. If it is an appropriate choice, automata reward the choice. Otherwise the choice receives penalty. The appropriateness of the choice depends on the amount of traffic flow the gateway experiences relative to other gateways.

In 2011 Sarigiannidis et al. [13] determined the ratio of sending link to the receiving link in IEEE 802.16e wireless networks using automata learning. In this approach, automata update the total performance according to the fullness of each link (sent or received) relative to the other one. For example, automata choose the increased sent link ratio aver the received one. If automata find that the bandwidth of the sent link is not used as much as the received link, automata will punish the choice of increased nodes' ratio.

Heinzelman et al. [14] introduced LEACH which is a self-organizing protocol with dynamic clustering. This approach uses a random method to balance energy consumption between nodes. Based on his approach, the nodes organize themselves into local clusters and a single node undertakes the role of a local BS 2 or the head of the group. If the head of clusters is chosen based on a stable priority and the lifetime of the system stays the same all the time, it is apparent that wrong nodes are chosen and they will die soon. Therefore, LEACH [15] uses random rotation of heads between the nodes to prevent battery dead in a particular node.

Each sensor node determines its cluster using the cost of minimum energy required to communicate. After all nodes identified their cluster, each leading cluster determines a plan for the head nodes in its cluster. This allows nodes to turn off their radio components except for the scheduled time and thereby it minimizes the energy used in conventional sensors. When a cluster head receives the information from all nodes, it aggregates the data, and then sends the compressed data to the BS. Since the central station may be far away from the leading cluster, this step will require a lot of energy, but it will affect a small number of nodes.

A large number of algorithms such as TEEN [16], HEED [17], PEGASIS [18], access point TEEN [19] and EEPSC [20] have been presented to develop LEACH method.

### 3 The Proposed Method

In the following section, the proposed method, which includes clustering and the sensing rate of sensor, is described. Each section is reviewed in detail.

#### 3.1 Clustering

In this part, the environment of the hypothetical scenario is divided into several equal sections and the goal is to determine a node as a cluster head in each section. All nodes in the network have the capability to become cluster head or normal node. As normal nodes in each section send their information to the cluster head in the same section, they consume less energy and thus they can work longer (Fig. 1).

As the radius sense of the sensor is considered 20 m, nodes are distributed in  $160 \times 160$ . Considering the worst case, if we divide the environment into 16 equal parts, each normal node should send its information within 56 yards. Automata select the node's status. The automata run every  $1 \text{ s} + 0.1 \text{ s}$  automata select the state in accordance with the state

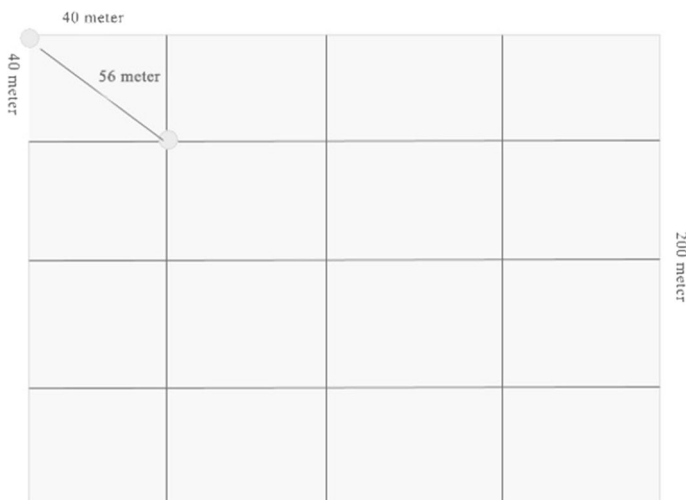


Fig. 1 The environment of a hypothetical scenario

probability table. After selecting the state, automata evaluation function evaluates the selection. If the choice is appropriate, it rewards the choice and the probability of choosing that state raises for the next time. Otherwise, the choice receives penalty and its probability to be chosen decreases for the next time.

### 3.2 Sensing Rate in Sensors

Each node, acting as a sensor in the network, changes its sensing rate according to the background information received from the environment. This reduces the energy consumption in nodes when changes in the environment happens gradually. This is done by automata in nodes and nodes amends their sensing rate over time to achieve an appropriate value. The general procedure is shown in Fig. 2.

### 3.3 Node's Automata

The selected automata are P model automata with normal and cluster heads to determine node's state. Coefficients of reward and punishment in automata equal to 0.07. The coefficients are obtained after testing the algorithm on a hypothetical scenario.

Usually the initial probability of each mode in automata should be equal. But as it is estimated that not more than 16 node changes to cluster heads, the table of initial probability of automata choices is set according to Table 1. This helps that not more than 20 nodes out of the first 100 nodes turn to cluster heads and automata achieve a stable situation faster.

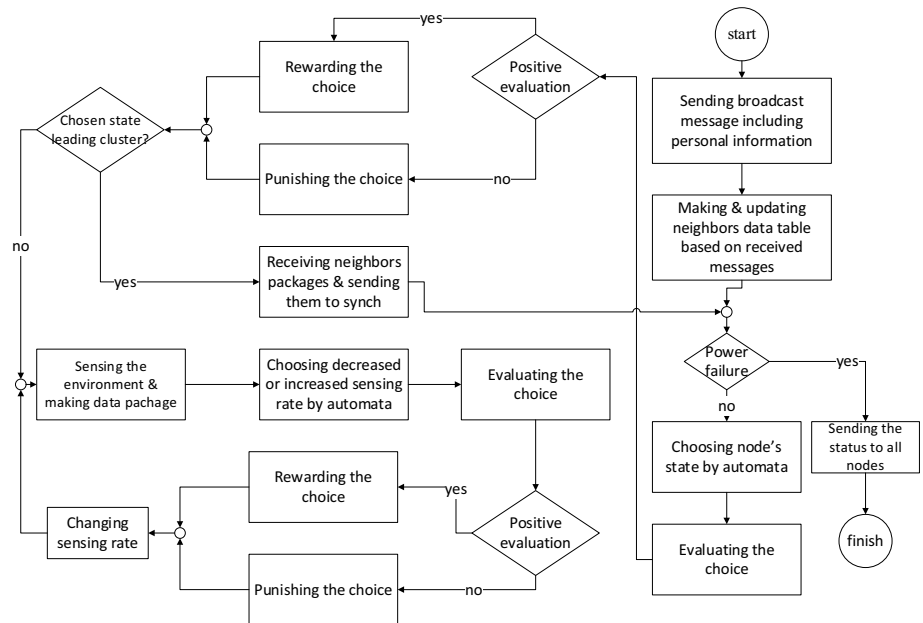


Fig. 2 The general procedure in the proposed method

**Table 1** Automata probability determining node's state

Cluster head	Sensor
0.2	0.8

Automata evaluation function decides upon the neighbor's power and state based on factors of node's remaining power. This function is based on the Eq. (1). The automata evaluation function's values are determined using the values in Table 2

$$rate = \begin{cases} aP + bP_o + cH \text{ cluster head's selected mode} \\ dP + eP_o + fH \text{ normal node's selected mode} \end{cases} \quad (1)$$

In Eq. 1, coefficients are filled using the values in Table 3.

These values are determined after multiple tests with different values for each coefficient. Figure 3 shows the evaluation function codes.

Reward and penalty codes in automata modes are shown in pseudo-codes in Figs. 4 and 5. In this code, **\_b** specifies the selected mode, **p** represented automata's probability table, **β** shows penalty and **α** shows automata's reward.

As in some cases the answer of divisions is not a real number, the total of cell arrays may not equal 1. To solve the problem, in the second circle the total of cell arrays are computed. If the result is less than 1, the shortage will be added randomly to an array.

### 3.4 Sensors' Automata

The automata increase or decrease the sensing rates of the environment by nodes. Total states of automata include decreased and increased rates which both have the probability of 0.5. The automata run every  $1\text{ s} + -0.1\text{ s}$  and select a state randomly. If the selected state is an increasing rate and in the five final senses the same values are sensed by the node, automata punish the choice, otherwise automata reward the choice, penalty and reward's values for these automata are considered 0.05.

## 4 Tests

In the experiment, to aggregate data on two hypothetical scenarios in the simulator NS2, simulations and the results of these tests have been compared with previous approaches to integrate data in wireless sensor networks. First to test and evaluate the plan, two scenarios are defined in NS2 Simulator. In the first scenario, 100 nodes are selected randomly in  $200 \times 200$  environment and distributed consistently and in the second scenario 100 nodes are distributed non-uniformly in the  $200 \times 200$  environment. Other parameters, set in the simulator, are shown in the table. Each scenario is simulated 50 times. They are also simulated 10 times to examine other compared algorithms, like LEACH, HEED, EADC, ECDC

The parameters of simulations are list in Table 4.

**Table 2** Automata evaluation function's values

Parameter	Cluster head	Normal node
$P$	1. If the remaining power is more than $1/3$ of the initial node's power Otherwise, it equals 0	1. If the remaining power is less than $1/3$ of the initial node's power Otherwise, it equals 0
$P_o$	1. If the remaining power is more than average remaining power of the neighbor nodes Otherwise, it equals 0	1. If the remaining power is less than average remaining power of the neighbor nodes Otherwise, it equals 0
$H$	1. If there is no neighbor leading cluster Otherwise, it equals 0	1. If there is a neighbor leading cluster Otherwise, it equals 0

**Table 3** Values of coefficient of automata evaluation function, determining node's state

Coefficient	Values
<i>a</i>	0.3
<i>b</i>	0.4
<i>c</i>	0.3
<i>d</i>	0.1
<i>e</i>	0.2
<i>f</i>	0.7

**Fig. 3** Evaluation function codes

```

bool MyAgent::rate(result r)
{
    double ra=0;
    if(r==clustermode){
        if(findheader()==0)
            ra+=0.3;
        if(Power>=FirstPower/3)
            ra+=0.3;
        if(Power>neighbourpower())
            ra+=0.4;
        if(ra>=0.9)
            return true;
        return false;
    }
    else{
        if(findheader())
            ra+=0.7;
        if(Power<FirstPower/3)
            ra+=0.1;
        if(Power<neighbourpower())
            ra+=0.2;
        if(ra>=0.1)
            return true;
        return false;
    }
}

```

#### 4.1 Test Evaluation Parameters

In the tests, two important parameters, i.e. network lifetime and the number of alive nodes, are considered for performance efficacy per time unit. Generally, the network lifetime is equal to the length of time that at least 70% of network nodes are on duty.



**Fig. 4** Automata's penalty, determining node's state

```

void Automata::penalty(int _b){
    for(int i=0;i<count;i++){
        if(_b==i)
            p[i]= (1-β)*p[i];
        else
            p[i]=β/(count-1)+(1-β)*p[i];
        }
    double temp=0.0;
    for(int i=0;i<count;i++){
        temp+=p[i];
        }
    if(temp<1){
        p[rand()%count]+=(1-temp);
        }
}

```

**Fig. 5** Automata's reward, determining node's state

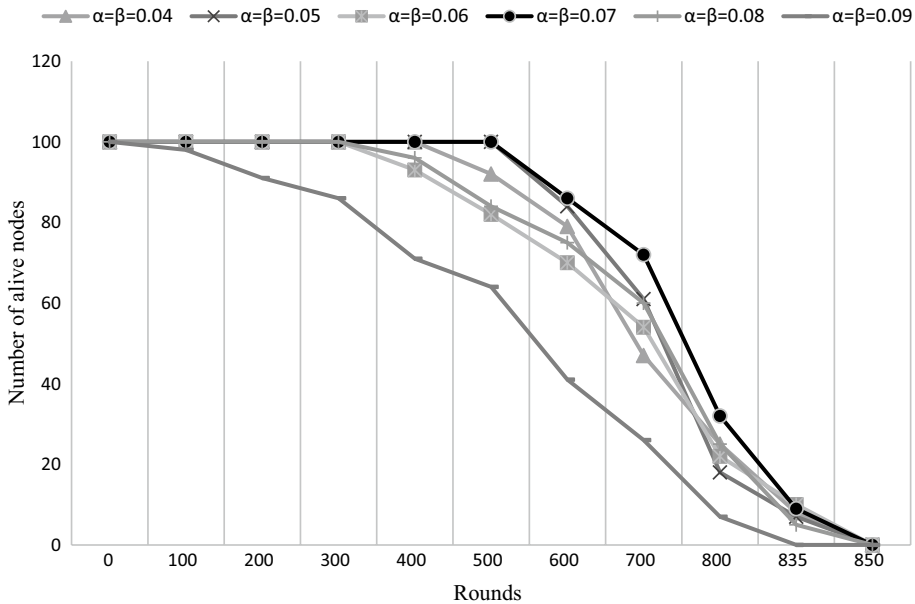
```

void Automata::reward(int _b){
    for(int i=0;i<count;i++){
        if(_b==i)
            p[_b]= p[_b]+α*(1- p[_b]);
        else
            p[_b]=(1-α)*p[_b];
        }
    double temp=0.0;
    for(int i=0;i<count;i++){
        temp+=p[i];
        }
    if(temp<1){
        p[rand()%count]+=(1-temp);
        }
}

```

**Table 4** Parameters of the simulation

Parameter	Value
Sensor field	200 m × 200 m
BS location	(250 × 100)
Number of nodes	100
Initial energy of nodes	1 mJ
Data packet size	500 bytes
Packet header size	25 bytes
Control message size	25 bytes
Node communication range ( $r_c$ )	20 m
Sensing range ( $R_s$ )	20 m
Sensing energy ( $E_{sen}$ )	0.5 nJ/(bit signal)
Communication energy ( $E_{com}$ )	5 nJ/(bit signal)



**Fig. 6** The results of running the algorithm in different values for automata's reward and punishment

## 4.2 Results of Pilot Project

In Fig. 6, a series of tests were carried out between 0.1 and 0.9 to determine the amount of penalty and reward, but only values between 0.4 and 0.09 are shown in the figure for better understanding. As it is evident, when the amounts of penalty and reward in the automata responsible for choosing the node's status are 0.7, the highest number of alive nodes per unit of are seen in the network.

In Fig. 7, in automata evaluation function, 6 parameters of a, b, c, d, e, f are shown to which specific values are assigned. This figure shows "e" value, for which 0.1–0.4 were run on algorithm. It shows that when "e" equals 0.2 in evaluation function of the automata responsible for choosing the node's status, the highest number of alive nodes per unit of are seen in the network. The values of 5 other parameters were also found by testing on algorithm.

In Figs. 8 and 9, the network lifetime for the pilot project in scenarios one and two are shown when the network is working 100–70%.

As it is shown, the proposed plan performed weaker in 100% and 90% status than previous plans. This is because the used automata in the network were not trained and their choices were random. But the performance of the network will enhance after fulfilling the training time.

The reason for showing up to 70% of alive nodes in figures is that the network lifetime is considered only up to time when 70% of nodes are alive. In scenario 1, 100% nodes are alive approximately after 525 s. Therefore, the first node turned off after 525 s, but in scenario 2 the first node turned off after 440 s.

In Fig. 10, a comparison is made between the two scenarios. It is concluded that when nodes are distributed more uniformly in the environment, they have shown better performance.

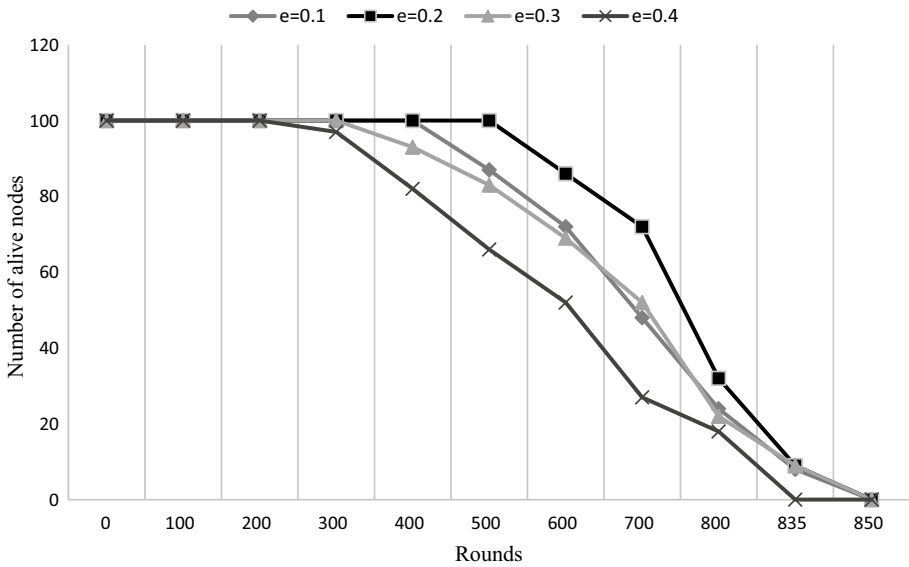


Fig. 7 The results of running the algorithm in different values for efficacy coefficient of the remaining energy in the node to its neighbors (automata evaluation function)

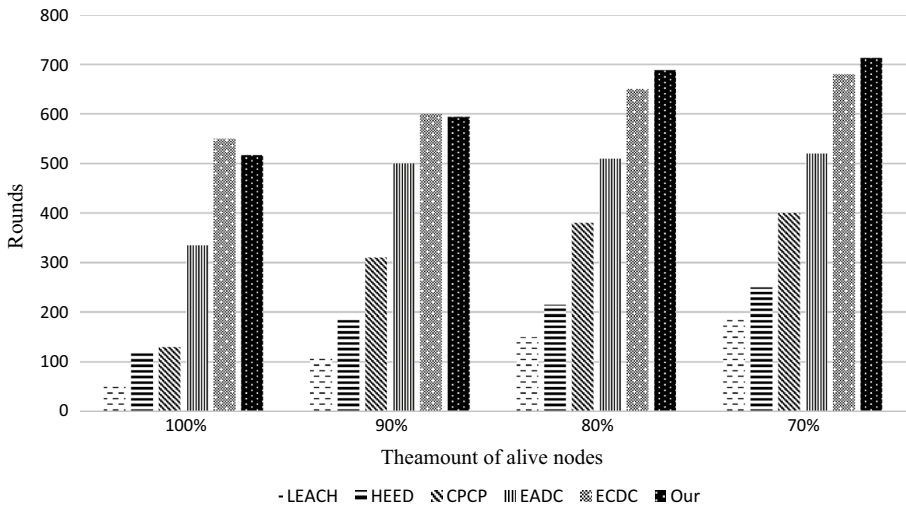


Fig. 8 Network lifetime in scenario 1

Since the network lifetime was 713 s in the first scenario and 699 s in the second scenario. As it is evident, in the 70% state relative to the best algorithm, namely ECDC, it increases the network lifetime by 5% in the first scenario and 11% in the second scenario.

Figure 11 shows the number of alive nodes in the network per time unit. As it is evident, using the proposed plan, the network can have alive nodes up to 850 s after simulation. Which is 1.7% higher than in testing scenario with algorithm ECDC.

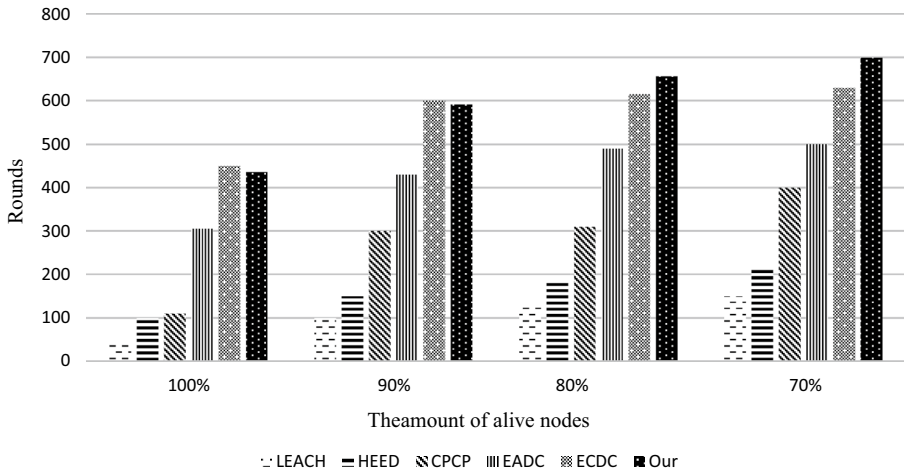


Fig. 9 Network lifetime in scenario 2

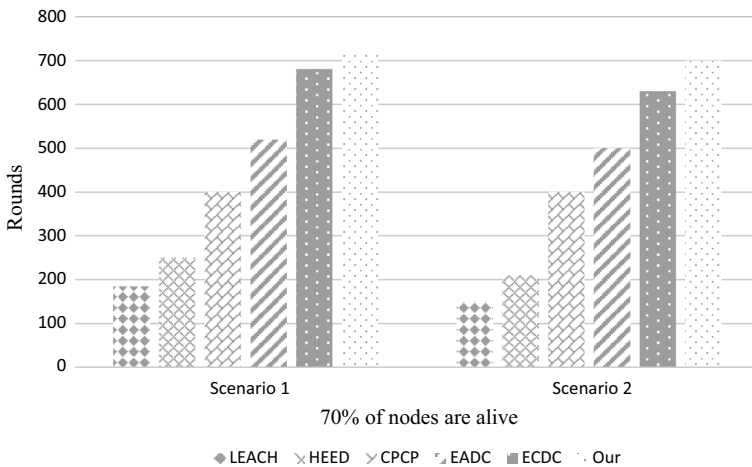


Fig. 10 The network lifetime in both scenarios

Figure 12 makes a contrast between the number of alive nodes in the network in scenario 1 and 2. As it is predictable, the plan in scenario 1 acts better than scenario 2.

### 5 Conclusion

This paper proposes a new method for data aggregation in WSNs based on LA. The plan is divided into clustering and the sensing rates of sensors. In clustering, using LA, leading clustering nodes in each area are specified and each node senses the rate of change in the environment using its automata and therefore changes its sensing time. Both of them cause reduced power consumption in the network and therefore increased network lifetime. Finally,

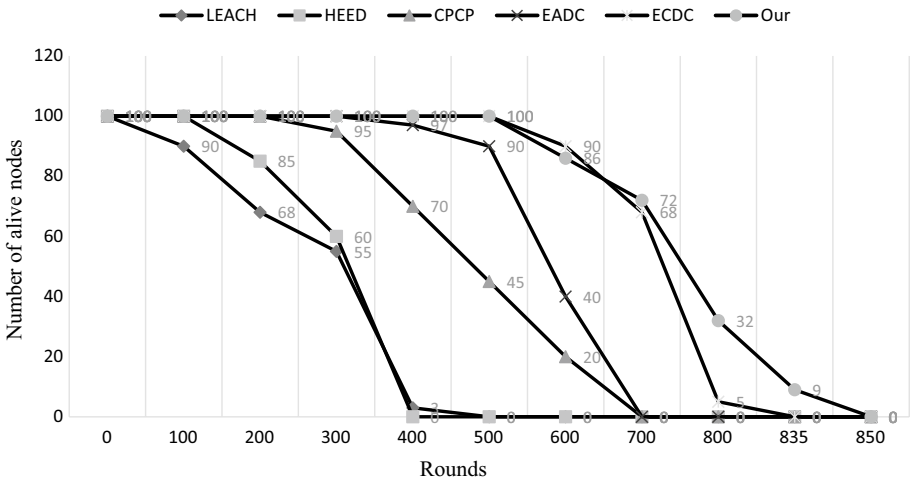


Fig. 11 Alive nodes in scenario 1

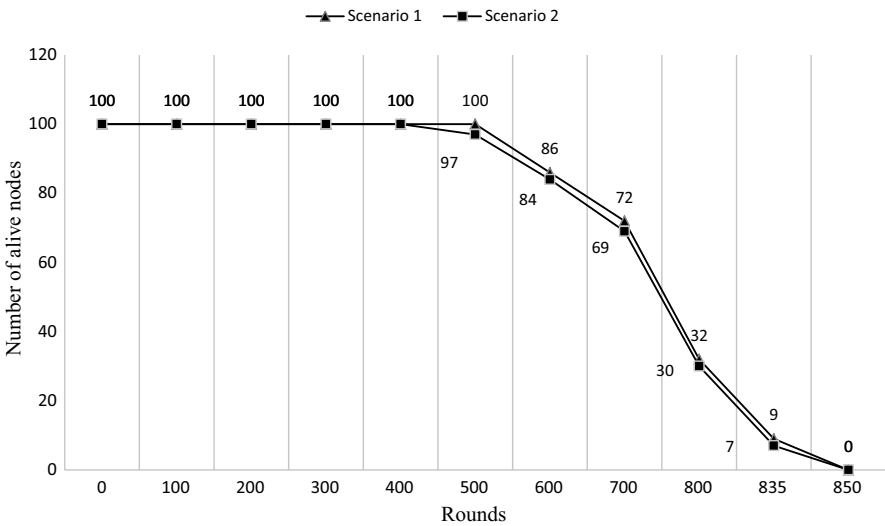


Fig. 12 Alive nodes in both scenarios

using NS2 simulation several tests were carried out and significant results were presented. The results of these tests showed the increased network lifetime in comparison to previous plans and also more alive nodes per unit of time.

## References

1. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12), 2292–2330.
2. Nakamura, E. F., Loureiro, A. A., & Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys (CSUR)*, 39(3), 9.
3. Younis, O., Krunz, M., & Ramasubramanian, S. (2006). Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20(3), 20–25.
4. Abbasi, A. A., & Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14–15), 2826–2841.
5. Li, F., & Wang, Y. (2008). Circular sailing routing for wireless networks. In *INFOCOM 2008. The 27th conference on computer communications*. IEEE (pp. 1346–1354). New York: IEEE.
6. Casari, P., Nati, M., Petrioli, C., & Zorzi, M. (2006). ALBA: An adaptive load-balanced algorithm for geographic forwarding in wireless sensor networks. In *Military communications conference, 2006. MILCOM 2006*. IEEE (pp. 1–9). New York: IEEE.
7. Yi, Y., Kwon, T. J., & Gerla, M. (2001). A load aware routing (LWR) based on local information. In *2001 12th IEEE international symposium on personal, indoor and mobile radio communications (Vol. 2, p. G)*. New York: IEEE.
8. Yang, J., Lin, Y., Xiong, W., & Xu, B. (2009). Ant colony-based multi-path routing algorithm for wireless sensor networks. In *2009. ISA 2009. International workshop on intelligent systems and applications (pp. 1–4)*. New York: IEEE.
9. Sarigiannidis, A. G., Nicopolitidis, P., Papadimitriou, G. I., Sarigiannidis, P. G., Louta, M. D., & Pomportsis, A. S. (2015). On the use of learning automata in tuning the channel split ratio of WiMAX networks. *IEEE Systems Journal*, 9(3), 651–663.
10. Dehkordi, S. S., & Torkestani, J. (2014). Channel assignment scheme based on learning automata for clustered wireless ad-hoc networks. *Journal of Computer Science and Information Technology*, 2, 149–172.
11. Beheshtifard, Z., & Meybodi, M. R. (2014). *Online channel assignment in multi-radio wireless mesh networks using learning automata*. Preprint. [arXiv:1405.3448](https://arxiv.org/abs/1405.3448).
12. Kashanaki, M., Beheshti, Z., & Meybodi, M. R. (2012). A distributed learning automata based gateway load balancing algorithm in wireless mesh networks. In *2012 International symposium on instrumentation and measurement, sensor network and automation (IMSNA) (Vol. 1, pp. 90–94)*. New York: IEEE.
13. Sarigiannidis, A., Nicopolitidis, P., Papadimitriou, G., Sarigiannidis, P., & Louta, M. (2011). Using learning automata for adaptively adjusting the downlink-to-uplink ratio in IEEE 802.16 e wireless networks. In *2011 IEEE symposium on computers and communications (ISCC) (pp. 353–358)*. New York: IEEE.
14. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), 660–670.
15. Junping, H., Yuhui, J., & Liang, D. (2008). A time-based cluster-head selection algorithm for LEACH. In *IEEE symposium on computers and communications, 2008. ISCC 2008 (pp. 1172–1176)*. New York: IEEE.
16. Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *Null (p. 30189a)*. New York: IEEE.
17. Younis, O., & Fahmy, S. (2004). HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4), 366–379.
18. Lindsey, S., & Raghavendra, (2002). C. S. PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings. IEEE (Vol. 3, p. 3)*. New York: IEEE.
19. Manjeshwar, A., & Agrawal, D. P. (2002). APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Ipdp (p. 0195b)*. New York: IEEE.
20. Zahmati, A. S., Abolhassani, B., Shirazi, A. A. B., & Bakhtiari, A. S. (2007). An energy-efficient protocol with static clustering for wireless sensor networks. *International Journal of Electronics, Circuits and Systems*, 1(2), 135–138.



**Maryam Tamiji** was born in Aligoudarz–Iran in 1987. She received the B.Sc. degree in Computer Science from Aligoudarz University, Iran, in 2012 and M.Sc. degree in Computer Engineering from Najafabad Branch, Islamic Azad University, Najafabad, Iran, in 2016 respectively. Her interested research is Computer Networks.



**Saeed Nasri** was born in Isfahan–Iran in 1962. He Received the B.Sc. degree in Electronic engineering in 1988, and the M.Sc. degree in Communication Engineering in 1991 from the Isfahan University of Technology (IUT) and the Ph.D. degree in Communication Engineering from the Science and Research Branch, Islamic Azad University (SRBIAU), Tehran, Iran, in 2013. Since... he has been with Department of Electrical Engineering, Najafabad Branch, Islamic Azad University as an assistant professor and Faculty Member of Digital Processing and Machine Vision Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran. His interested research is Machine Vision and Image Processing and Computer Networks.