

# EEMLA: Energy Efficient Monitoring of wireless Sensor Network with Learning Automata

Habib Mostafaei

Islamic Azad University, Urmia branch  
Urmia, Iran  
e-mail: Habib.Mostafaei@Gmail.com

Mohammad Reza Meybodi and Mehdi Esnaashari  
Soft Computing Laboratory, Computer Engineering and  
Information Technology Department Amirkabir  
University of Technology Tehran  
Tehran, Iran  
e-mail: {mmeybodi, Esnaashari}@aut.ac.ir}

**Abstract**— When sensors are redundantly deployed, a subset of sensors should be selected to actively monitor the field (referred to as a "cover"), while the rest of the sensors should be put to sleep to conserve their batteries. Despite of its potential application, wireless sensor network encounters resource restrictions such as low computational power, reduced bandwidth and specially limited power resource. In this paper we propose learning automata based algorithm for energy-efficient monitoring in wireless sensor networks. Learning Automata are used for choosing the nodes having redundant coverage contribution. The proposed monitoring method in comparison to existing methods uses less number of nodes for monitoring network area. To evaluate the performance of the proposed algorithm several experiments have been conducted. The simulation results establish that the monitoring of sensor nodes with the proposed technique shows better utilization of the resources that effectively leads to an energy efficient maximally covered sensor network topology. Experiments have also shown that the proposed monitoring algorithm in comparison to other existing methods prolongs the network lifetime.

**Keywords:** Wireless sensor networks, Area coverage, energy-efficient, Learning Automata (LA).

## I. INTRODUCTION

Recently, the idea of wireless sensor networks has attracted a great deal of research attention due to wide-ranged potential applications that will be enabled by wireless sensor networks, such as battlefield surveillance, machine failure diagnosis, biological detection, home security, smart spaces, inventory tracking, etc. [1]. A wireless sensor network consists of tiny sensing devices, deployed in a region of interest. Each device has processing and wireless communication capabilities, which enable it to gather information from the environment and to generate and deliver report messages to the remote base station (remote user). The base station aggregates and analyzes the report messages received and decides whether there is an unusual or concerned event occurrence in the deployed area. Considering the limited capabilities and vulnerable nature of an individual sensor, a wireless sensor network has a large number of sensors deployed in high density (high up to 20nodes/m<sup>3</sup>) and thus redundancy can be exploited to increase data accuracy and system reliability [15].

Several applications, such as environmental monitoring, require sensors be redundantly deployed to accommodate unexpected failures and improve the fidelity of received measurements. Redundancy means that some parts of the field are covered by more than one sensor at the same time. If idle sensors are not put to sleep, then redundant node deployment does not necessarily improve the coverage time of the field, defined as the time until the fraction of the area that is monitored by at least one sensor falls below a given threshold. This is because the sensor's radio expends a significant portion of its battery lifetime in idle-listening to support data forwarding, and thus active sensors tend to die at roughly the same time [17]. It was also reported in [14] that in the WINS Rockwell seismic sensor the power consumed in receive and idle-listening modes are 0.36 mW and 0.34 mW, respectively. In contrast, the energy consumed in the sleep mode of the MICA2 is three orders of magnitude less than that during idle-listening.

In this paper, we propose a novel method for addressing the problem of area coverage in wireless sensor networks using learning automata. In the proposed approach which we call it EEMLA, each node in the network is equipped with a learning automaton which learns (schedules) the proper on and off times of that node based on local neighbor's information.

This work proposes an energy efficient maximally covered sensor network algorithm that addresses the requirements of power efficient infrastructure issues for WSN. EEMLA runs on each node in a WSN to decide on the state of the node, either ACTIVE or ASLEEP, and thereby ensures minimal number of active nodes at a particular moment of time to cover the maximal area of the backbone network.

The EEMLA is developed around the modeling tool of learning automata. It is established that the lifespan of a network, monitored by EEMLA, is better than that of other similar state-of-the-art monitoring schemes. Our main purpose is increase of network lifetime.

The rest of the paper is organized as follows. Section II briefly surveys related work. Section III introduces an overview of learning automata. The problem statement is given in section IV. Section V presents the EEMLA algorithm. Section VI reports the performance of our proposed network monitoring scheme. Finally, Section VII concludes the paper.

## II. RELATED WORKS

In [16], a probing-based density control algorithm is proposed to ensure long-lived, robust sensing coverage by leveraging unconstrained network scale. In this protocol, only a subset of nodes are maintained in working mode to ensure desired sensing coverage, and other redundant nodes are allowed to fall asleep most of the time. Working nodes continue working until they run out of their energy or are destroyed. A sleeping node wakes up occasionally to probe its local neighborhood and starts working only if there is no working node within its probing range. Geometry knowledge is used to derive the relationship between probing range and redundancy. In this algorithm, desired redundancy can be obtained by choosing the corresponding probing range. However, this derivation is based on the assumption that all the nodes have exactly the same sensing range. It is hard to find a relationship between probing range and desired redundancy, if nodes have different sensing ranges. Furthermore, the probing-based off-duty eligibility rule can not ensure the original sensing coverage and blind points may appear after turning off some nodes.

In [4], [5] authors computed a number of set covers that maximize the lifetime of the sensor network. They proposed two centralized heuristic techniques for target coverage; one uses linear programming and the other is a greedy approach.

Slijepcevic and Potkonjak [13] proposed a centralized heuristic to compute a disjoint maximal set of covers. The authors suggest a method to convert from area coverage into target coverage as being illustrated in Figure 1. The paper first defines the concept of field which is a set of points that are covered by the same set of sensors. For example, the sensors in Figure 1 partition the monitored area (the dashed-line rectangle) into eight fields. By considering each field as a target, the area coverage problem is easily and accurately transformed into the target problem.

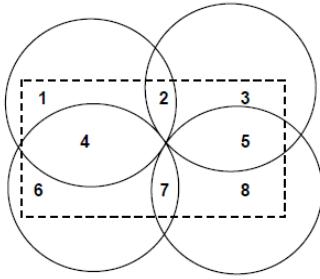


Figure 1. Transformation from area coverage to targets coverage [13]

Tian and Georganas [15] devise an algorithm that ensures complete coverage using the concept of “sponsored area.” Whenever a sensor node receives a packet from one of its working neighbors, it calculates its sponsored area (defined as the maximal sector covered by the neighbor). If the union of all the sponsored areas of a sensor node covers the coverage disk of the node, the node turns itself off.

Zhang and Hou [19] analyze the relationship between complete coverage and connectivity (i.e., if the transmission radio range is at least twice of the sensing range, then coverage implies connectivity), develop several optimal conditions of maintaining coverage, and devise, based on the optimal conditions, a localized method to maintain coverage and connectivity.

Blough and Santi [3] study the upper bound of the network lifetime for cell-based energy conservation techniques.

In [9], [10] authors proposed probabilistically schedules sensing activities according to the sensor's contribution to the sensing coverage of the whole sensor network.

Younis, Krunz, and Ramasubramanian [18] proposed two distributed protocols (LUC-I and LUC-P) that periodically select covers and switch between them to extend "coverage time" and tolerate unexpected failures.

In [2] authors proposed an energy efficient method for monitoring wireless sensor networks using cellular automata. Cellular automata used for detecting redundant nodes.

In this paper we proposed a simple method based on learning automata for detecting redundant nodes in given area and just based on neighbors information's we detect redundant nodes in node sensing range.

## III. LEARNING AUTOMATA

Learning automata is an abstract model which randomly selects one action out of its finite set of actions and performs it on a random environment. Environment then evaluates the selected action and responses to the automata with a reinforcement signal. Based on selected action, and received signal, the automata updates its internal state and selects its next action. Figure 2 depicts the relationship between an automata and its environment.

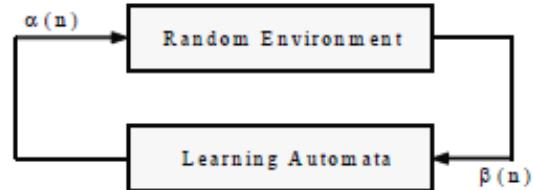


Figure 2. Relationship between learning automata and its environment

Environment can be defined by the triple  $E = \{\alpha, \beta, c\}$  where  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  represents finite input set,  $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$  represents the output set, and  $c = \{c_1, c_2, \dots, c_r\}$  is a set of penalty probabilities, where each element  $c_i$  of  $c$  corresponds to one input of action  $\alpha_i$ . An environment in which  $\beta$  can take only binary values 0 or 1 is referred to as P-model environment. A further generalization of the environment allows finite output sets with more than two elements that

take values in the interval  $[0, 1]$ . Such an environment is referred to as Q-model. Finally, when the output of the environment is a continuous random variable which assumes values in the interval  $[0, 1]$ , it is referred to as an S-model. Learning automata are classified into fixed-structure stochastic, and variable-structure stochastic. In the following, we consider only variable-structure automata.

A variable-structure automaton is defined by the quadruple  $\{\alpha, \beta, P, T\}$  in which  $\alpha = \{\alpha_1, \dots, \alpha_n\}$  represents the action set of the automata,  $\beta = \{\beta_1, \dots, \beta_n\}$  represents the input set,  $P = \{P_1, \dots, P_n\}$  represents the action probability set, and finally  $p(n+1) = T[\alpha(n), \beta(n), p(n)]$  represents the learning algorithm. This automaton operates as follows. Based on the action probability set  $p$ , automaton randomly selects an action  $\alpha_i$ , and performs it on the environment. After receiving the environment's reinforcement signal, automaton updates its action probability set based on equations (1) for favorable responses, and equations (2) for unfavorable ones.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (1)$$

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (2)$$

In these two equations,  $a$  and  $b$  are reward and penalty parameters respectively. For  $a = b$ , learning algorithm is called  $L_{R-P}$ <sup>1</sup>, for  $b << a$ , it is called  $L_{REP}$ <sup>2</sup>, and for  $b = 0$ , it is called  $L_{R-I}$ <sup>3</sup>. For more information the reader may refer to [11][14]. The only application of learning automata to sensor networks has been reported in [6].

#### IV. PROBLEM STATEMENT

The problem is to select a subset of the nodes as active nodes in the network which can cover the entire area of the network. More specifically, if the set of nodes in the network is  $V$ , it is required to select a subset  $V_A \subseteq V$  that covers the entire area covered by  $V$  ( $V_A$  is referred to as a "cover"). The remaining set of nodes  $V_s = V - V_A$  can be put to sleep and later activated to form new covers [18].

Checking the coverage of the entire area of the network is not a simple task. This is because one has to check an infinite number of points in the area to make sure that all points in the area are covered. Therefore, in [3, 13] Authors proposed two methods for transforming the area coverage problem into point (target) coverage problem. In this paper, we use the approach given in [3] for transforming the area coverage into the target coverage problem. In this approach, the network area is divided into small square regions (cells), each having equal size. To check the coverage of the entire area, it is sufficient to check if each of the cells is under the coverage of at least one of the sensor nodes. We assume that sensor nodes are location-aware and are able to locally determine the cells they can cover. Also, we assume that the sensing range of all nodes is equal and the monitoring environment is a rectangular area and Nodes of the network are placed randomly on a two-dimensional area.

#### V. PROPOSED METHOD

Each node  $i$  in the network is equipped with a learning automaton  $LA_i$  which helps the node in determining its suitable state; whether to be active or not. Learning automaton of each node has two actions; *ACTIVE* or *ASLEEP*. At the beginning of the algorithm, *ACTIVE* and *ASLEEP* actions have the same probability equal to 0.5.

At the beginning of the algorithm, each node locally determines the regions it can cover. Then each node broadcasts an advertisement packet in its neighborhood containing its ID, position and cells it can cover. The node then listens to receive advertisement packets from its neighbors.

Network operation is divided into rounds. Each round begins with a *learning* phase, followed by a *monitoring* phase. During the *learning* phase, learning automaton of each node  $i$  randomly selects one of its actions (*ACTIVE* or *ASLEEP*). Node  $i$  then broadcasts an ACTION packet which contains the selected action in its neighborhood. Node  $i$  then wait for certain duration to receive the ACTION packets of its neighbors. When node  $i$  receives all ACTION packets of its neighbors, it operates as follows:

If the selected action of  $LA_i$  was *ACTIVE* then:

1. If all of the regions under the coverage of the node  $i$  are covered by those neighbors whose selected actions are *ACTIVE* then node  $i$  penalizes its learning automaton using (2).
2. Otherwise, node  $i$  rewards its learning automaton using (1).

If the selected action of  $LA_i$  was *ASLEEP* then:

1. If all of the regions under the coverage of the node  $i$  are covered by those neighbors whose selected actions are *ACTIVE* then node  $i$  rewards its learning automaton using (1).
2. Otherwise, node  $i$  penalizes its learning automaton using (2).

<sup>1</sup> Linear Reward-Penalty

<sup>2</sup> Linear Reward epsilon Penalty

<sup>3</sup> Linear Reward Inaction

Each node  $i$  separately stops its *learning* phase if one of the following conditions occurred:

- [1] Action probability of one of the actions of  $LA_i$  exceeds a specified threshold
- [2] Number of action selections by  $LA_i$  exceeds *MaxActionSelection*.

In the *monitoring* phase learning automata of each node selects its best actions and nodes for which the probability of selecting *ASLEEP* action was more than 0.5 switches to *ASLEEP* state. Rest of the nodes will be active and monitor the environment for the whole duration of the *monitoring* phase. In the proposed method, *monitoring* phase lasts for 100 seconds. The next round will be started when the current *monitoring* phase is over.

## VI. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed method several experiments have been conducted and the proposed method is compared with methods given in [15, 18]. We compare EEMLA to a LUC-I and LUC-P [18] and a distributed approach [15] that assume complete knowledge of node locations. The distributed approach in [15] uses a geometric test assuming that relative node locations are known.

For communication energy estimation, we use a first order radio model given in [7]. In this model, energy required for running the transmitter or receiver circuitry is  $E_{elec} = 50 \text{ nJ/bit}$  and the transmitter amplifier requires  $E_{amp} = 100 \text{ pJ/bit/m}^2$ . Energy required to transmit a data packet of size  $l$  bits from node  $i$  to node  $j$  is given by  $T_{ij} = lE_{elec} + lE_{amp}d_{ij}^2$ , where  $d_{ij}$  is the distance between node  $i$  and node  $j$ . Energy required to receive a  $l$  bit packet for any node  $j$  is given by  $R_j = lE_{elec}$ . Also for sleep and idle modes energy consumptions, we use the specifications of MEDUSA II sensor node given in [12]; energy consumed during the sleep and idle modes would be equal to 0.02 mW and 22 mW respectively. Energy required to switch a node from sleep to active mode is assumed to be negligible.

We assume that  $n$  nodes are randomly scattered throughout a 50 x 50 meters field ( $n = 1000$ , unless otherwise specified). We assume that *MaxActionSelection* in learning phase is equal to 75. The initial energy of each node is selected uniformly at random from the range [0, 1]. The comparison will be done based on the following criteria: (1) size of the active set VA and (2) coverage time (network lifetime). Results of the experiments are averaged over 10 runs on different random topologies. Coverage time is defined as the network lifetime.

### A. Experiment 1

This experiment considers the first comparison criterion; size of the active set. Number of nodes in this experiment is equal to 1000 and vary  $R_s$ . Figure 4 depicts

the size of VA as a function of the sensing range. As expected, VA drops for all the compared protocols as  $R_s$  increases. VA of EEMLA always has good results compared to available algorithms.

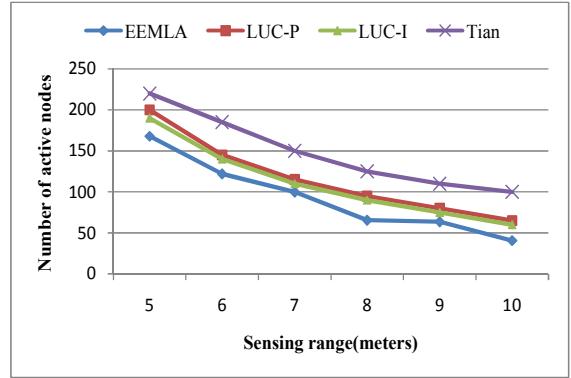


Figure 3. Size of VA ( $n= 1000$ )

This experiment is repeated for different number of nodes. We assess the size of the active set(VA) for the case that nodes have same sensing range and vary number of deployed nodes. We fix  $R_s$  at six meters and we assume that each side of cells has 2.7 meter length. Figure 4 shows the size of VA by each algorithm. It can be seen from this figure that size of VA in the proposed method is smaller than methods given in [15] and [18] respectively. This is because in our approach, learning automaton in each node helps the node to find the redundant nodes in its sensing range.

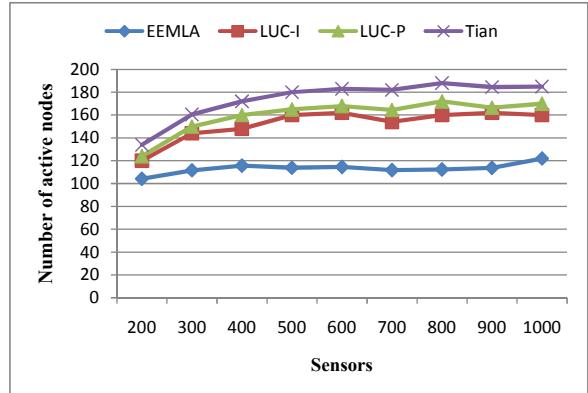


Figure 4. Size of VA with  $R_s=6$  meters

### B. Experiment 2

This experiment considers the second comparison criterion; coverage time. Here not only the coverage quality of the proposed method at the end of the *learning* phase is considered, but also the coverage quality during the *monitoring* phase is taken into account. For this experiment we consider a simple operational scenario in which the energy consumed by a node's radio is computed both in ACTIVE and ASLEEP states. We deduct a fixed amount of energy from the node's battery according to its state. Every node starts with a full battery of 1 Joule. We

take  $R_s = 6$  meters and we assume that each side of cells has 2.7 meter length. Figure 5 shows the fraction of the field that is covered by active nodes when  $n = 500$ . If none of the nodes is allowed to sleep, the network becomes completely uncovered after 100 rounds. The figure shows that our algorithm has superior result in comparison to the other methods. This is due to the fact that in our approach, the chance of a node having more regions under the coverage to become an ACTIVE node is higher than a node having fewer regions under the coverage.

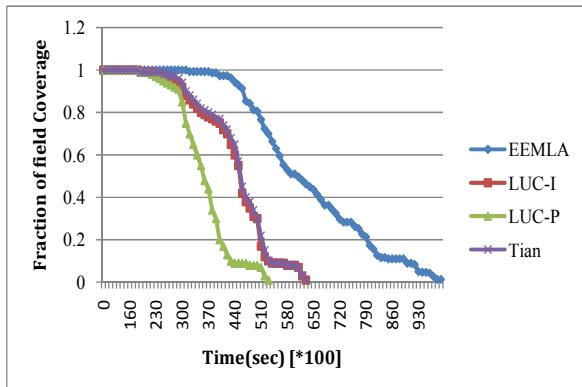


Figure 5. Coverage time in a scenario with  $n=500$  and  $R_s=6$  meters

## VII. CONCLUSION

In this paper we proposed a novel method based on learning automata for area coverage in wireless sensor networks. In this method each node in the network is equipped with a learning automaton. The learning automaton has two actions. The learning automaton for each node helps the node to find the redundant nodes in its sensing range. It was shown through simulations that the proposed method outperforms the existing methods in terms of network's lifetime.

## REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," in Proceedings of the IEEE Communication Magazine, August 2002, pp. 102-114.
- [2] I. Banerjee, S. Das, H. Rahaman, and B. K. Sikdar, "An Energy Efficient Monitoring of Ad-Hoc Sensor Network with Cellular Automata," in IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, October 8-11, 2006, pp. 5100-5105.
- [3] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," in ACM MobiCom Conf, 2002.
- [4] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D. Z. Du, "Wireless sensor networks with energy efficient organization," in Journal of Interconnection Networks 3, 2002, pp. 213-229.
- [5] M. Cardei and J. Wu, "Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," in Computer Communications 29 Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA, 2006, pp. 413-420.
- [6] M. Golipour and M. R. Meybodi, "LA-Mobicast: A Learning Automata based Mobicast Routing protocol for Wireless Sensor Networks," in Sensor Letters, vol. 6, April 2008, pp. 305-311.
- [7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in HICSS 2000.
- [8] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad-hoc sensor networks," in IEEE Trans. Comput., 2003, pp. 753-763.
- [9] J. Lu, L. Bao, and T. Suda, "Probabilistic Self-Scheduling for Coverage Configuration in Wireless Ad-hoc Sensor Networks," in IEEE Computer Communication Workshop (CCW) 2003 and later at International Conference on Sensing Technology (ICST) 2005, pp. 1-18.
- [10] J. Lu, L. Bao, and T. Suda, "Probabilistic Self-Scheduling for Coverage Configuration in Sensor Networks," 2005.
- [11] K. S. Narendra and M. A. L. Thathachar, Learning automata: An introduction: Prentice Hall, 1989.
- [12] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-Aware Wireless Microsensor Networks," in IEEE Signal Processing Magazine, 2002, pp. 40-50.
- [13] S. Slijepcevic and M. Potkonjak, "Power Efficient Organization of Wireless Sensor Networks," in ICC Helsinki, Finland, 2001.
- [14] M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: An overview," in IEEE Transaction on Systems, Man and Cybernetics-Part B: Cybernetics, 2002, pp. 711-722.
- [15] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, GA, Sept. 2002, pp. 32-41.
- [16] J. Wu and S. Yang, "Energy-Efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges," in International Journal of Foundations of Computer Science World Scientific Publishing Company, 2005, pp. 1-15.
- [17] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for longlived sensor networks," in Proc. 10th IEEE Int. Conf. Network Protocols (ICNP'02), Paris, Nov. 2002, pp. 200-201.
- [18] O. Younis, M. Krunz, and S. Ramasubramanian, "Coverage Without Location Information," 2007, pp. 51-60.
- [19] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor network," in Ad Hoc & Sensor Wireless Networks, Jan 2005, pp. 89-124.