QoS Support in Wireless Sensor Networks by Focusing on Coverage Problem

M. Ahmadinia[*], M. R. Meybodi[**], M. Esnaashari[***]

[*]Computer Department, Kerman Branch, Islamic Azad University, Kerman, Iran.
[**],[***] Soft Computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran.

Summary
QoS in the area of wireless sensor networks (WSNs) can be measured based on different criteria: Network lifetime, network coverage, number of active nodes and energy consumption. In this paper, we propose a scheduling solution for activating and deactivating sensor nodes in the network in such a way that all of the above criteria are considered. In the proposed solution, which is based on cellular learning automata, each node is equipped with a learning automaton. The learning automaton of each node decides for the node to be active or not based on the energy level of the node and the activeness of its neighbors. Simulation results show that the proposed algorithm outperforms similar existing methods in terms of energy consumption and network lifetime.

Key words:
Wireless sensor network, Coverage, Cellular learning automata, Quality of service.

1. Introduction

A Wireless sensor network is made up of a number of tiny sensor nodes which are usually scattered throughout the environment randomly. The main objective of such networks is to collect data from the environment and to send collected data towards a central location called the sink. Communication in these networks is performed by means of radio connections in a hop by hop manner.

In wireless sensor networks, energy which comes from sensor nodes' batteries is a vital resource which usually cannot be refilled. This is due to the fact that sensors are usually deployed in remote or hostile environments in which recharging or replacing batteries is very hard if not impossible at all. However, long system lifetime is expected by many monitoring applications of wireless sensor networks. As a result, energy consumption and network lifetime are two important metrics of QoS in WSNs [1].

One another aspect of the quality of service in wireless sensor networks is coverage [1]. The objective of the coverage can be described as follows: each point in the physical environment must be located within the sensing range of at least one sensor node. A generalized version of the coverage problem is k-coverage problem in which the purpose is to monitor each point in the physical environment with at least k sensor nodes. k is referred to as coverage degree.

Sensor networks usually deployed with a degree of redundancy in the number of sensor nodes. This redundancy is needed to ensure the network coverage and prolonging the network lifetime. In such networks, to prolong the network lifetime, it is desirable to allow only a required number of sensor nodes to be active and monitor the environment while the rest of the nodes are in the sleep mode, saving energy.

In this paper a scheduling solution for activating and deactivating sensor nodes in the network based on cellular learning automata will be proposed. In the proposed method, each node is equipped with a learning automaton. The learning automaton of each node decides for the node to be active or not based on the energy level of the node and the activeness of its neighbors. This solution warranties the network coverage while at the same time decreases the energy consumption and increases the lifetime of the network.

The rest of this paper is organized as follows. In section 2 the works done about the coverage of the sensor networks are introduced. Section 3 describes in the cellular learning automata which is as a main learning strategy in proposed algorithm. In section 4 the proposed algorithm is described and in section 5 the results of simulations are represented. The final section is the conclusion.

2. Related work

For supporting QoS in WSNs various works has been performed. QoS in the area of WSNs can be measured based on different criteria: Network lifetime, network coverage, number of active nodes and energy consumption[1]. In any work some of QoS criteria has been considered. In this section, we will briefly overview some of these works that consider coverage problem for wireless sensor networks. The works in [2] and [3] consider a large population of sensors, deployed randomly for area monitoring. The goal is to achieve an energy-efficient design that maintains area coverage. The solution proposed

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

is to divide the sensor nodes into disjoint sets, such that every set can individually perform the area monitoring tasks. These set areas are then activated successively, and while the current sensor set is active, all other nodes are in a low-energy sleep mode. The goal of this approach is to determine a maximum number of disjoint sets, as this has a direct impact on the network lifetime. The solutions proposed are centralized. Slijepcevic and Potkonjak [2] model the area as a collection of fields, where every field has the property that any enclosed point is covered by the same set of sensors. The most-constrained least-constraining algorithm [2] computes the disjoint covers successively, selecting sensors that cover the critical element (field covered by minimal number of sensors), giving priority to sensors that: cover a high number of uncovered fields, cover sparsely covered fields and do not cover fields redundantly. Cardei et al [3] model the disjoint sets as disjoint dominating sets. The maximum disjoint dominating sets computation is NP-complete, and a graph-coloring based algorithm is proposed. Simulations have shown that the number of sets computed is between 1.5 and 2 times greater than using the algorithm in [2], with lapses in area coverage less than 5%, on average.

Another energy-efficient node-scheduling-based coverage mechanism is proposed in [4]. The protocol proposed is distributed and localized. The off-duty eligibility rule determines whether a node’s sensing area is included in its neighbors’ sensing area. A probing-based, node-scheduling solution for the energy-efficient coverage problem is proposed in [6]. Here all sensors are characterized by the same sensing range and coverage is seen as the ratio between the area under monitoring and total size of the network field. The off-duty eligibility rule is based on a probing mechanism. Basically, a sensor broadcasts a probing message PRB within a probing range r. Any working node that hears this message responds with a PRB RPY . If at least one reply is received, the node enters the sleep mode. Probing range is selected based on the desired working node density (number of sensors per unit area) or based on the desired coverage redundancy, whereas the wake-up time is based on the tolerable sensing intermittence. This protocol is distributed, localized, and has low complexity but still does not preserve the original coverage area.

The studies in [7], [8] conduct asymptotic and analytical analysis to provide necessary and sufficient conditions for coverage in various environments. In [9], optimal deployment patterns for different ratios of the communication and sensing ranges are proposed. While these studies provide useful insights and guidelines, which we indeed benefited from, they do not propose specific coverage protocols. Several distributed coverage protocols have been proposed for the disk model. For example, OGDC [10] tries to minimize the overlap between the sensing circles of activated sensors, while CCP [11] deactivates redundant sensors by checking that all intersection points of sensing circles are covered. Other earlier protocols include PEAS [12] and Ottawa [13]. Probabilistic coverage with various sensing models has also been studied in [14][15]. The work in [15] analytically studies the implications of adopting probabilistic and disk sensing models on coverage, but no specific coverage protocol is presented. In [14], the sensing range is modeled as layers of concentric disks with increasing diameters, where the probability of sensing is fixed in each layer. A coverage evaluation protocol is also proposed.

3. CELLULAR LEARNING AUTOMATA

In this section we briefly review cellular automata, learning automata, cellular learning automata and then introduce irregular cellular learning automata [16].

Cellular Automata: Cellular automata are mathematical models for systems consisting of large number of simple identical components with local interactions. CA is a non-linear dynamical system in which space and time are discrete. The simple components act together to produce complicated patterns of behavior. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighborhood .

Learning Automata: Learning Automata are adaptive decision-making devices that operate on unknown random environments. A learning Automaton has a finite set of actions to choose from and at each stage, its choice (action) depends upon its action probability vector. For each action chosen by the automaton, the environment gives a reinforcement signal with fixed unknown probability distribution. The automaton then updates its action probability vector depending upon the reinforcement signal at that stage, and evolves to some final desired behavior.

Cellular Learning Automata: Cellular learning automata, which is a combination of cellular automata (CA) and learning automata (LA), is a powerful mathematical model for many decentralized problems and phenomena. The basic idea of CLA, which is a subclass of stochastic CA, is to utilize learning automata to adjust the state transition probability of stochastic CA. A CLA is a CA in which a learning automaton is assigned to every cell. The learning automaton residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA, there is a rule that the CLA operates under. The local rule of CLA and the

Actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell. The neighboring LAs of any particular LA constitute the local environment of that cell.

Irregular Cellular Learning Automata: An Irregular cellular learning automata (ICLA) is a cellular learning automata (CLA) in which the restriction of rectangular grid structure in traditional CLA is removed. This generalization is expected because there are applications such as wireless sensor networks, immune network systems, graph related applications, etc. that cannot be adequately modeled with rectangular grids. An ICLA is defined as an undirected graph in which, each vertex represents a cell which is equipped with a learning automaton.

The learning automaton residing in a particular cell determines its state (action) on the basis of its action probability vector. Like CLA, there is a rule that the ICLA operate under. The rule of the CLA and the actions selected by the neighboring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell.

4. Problem Statement

Consider a sensor network S consists of N homogeneous sensor nodes s1, s2, ..., sN scattered uniformly and independently at random throughout a rectangular area. Assume that rs is the sensing radius of each node and rc is its transmission radius. Suppose that rc>=2rs. This condition implies that if the network is fully covered, it will be definitely connected [17]. This is a real assumption for many available sensor nodes. For instance, in MICAII [18], transmission radius is almost 1000 meters while the sensor radius is almost 100 meters. Furthermore, assume that sensor nodes are aware of their positions by means of the GPS system with low energy [19]. Each sensor node si has two different modes of operation as follows:

Active: CPU, sensing and communicating units are switched on.

Inactive: CPU, sensing and communicating units are switched off.

At any instance of the time, a sensor node can be only in one of the above two operation modes.

By assuming a high level of redundancy in the number of sensor nodes in the network, the stated problem is to minimize the number of active sensors in the network so that the network area is K-covered, i.e. each point within the area of the network is under the coverage of with at least K different sensor nodes.

5. Proposed Method

Because of the distributive nature of wireless sensor networks and the fact that they usually consist of large number of nodes, centralized approaches are not efficient for them. Therefore, our approach in this paper is distributed. The outline of the proposed method is as follows: each node is equipped with a learning automaton. The learning automaton of each node decides for the node to be active or not. As a consequence to this decision, the node receives reward or penalty from its neighbors indicating that its decision was good or bad. This way, each node gradually learns its suitable state.

The Algorithm

At the beginning, a time-driven asynchronous ICLA which is isomorphic to the sensor network topology is created. Each node si in the sensor network corresponds to the cell i in ICLA. Two cells i and j in ICLA are adjacent to each other if si and sj in the sensor network are close enough to hear each other's signal. The learning automaton in each cell i of ICLA, referred to as LAi, has two actions a0 and a1. Action a0 is to declare the sleep operation mode as the preferred operation mode and action a1 is to declare the active operation mode as the preferred operation mode for the node. The probability of selecting both actions is the same at the startup. Deciding for the state of each node si is performed in a number of iterations. In each iteration, any LAi selects one of its actions randomly according to its action probability vector. This selection shows the tendency of si to be in active or sleep operation mode. The selection of LAi and the remaining energy of the node are then packed into an INFORM packet and are sent to the neighbors of si. si then waits for a specified duration to receive the INFORM packets from its neighbors. After the specified waiting duration, si performs the following operations:

- Computes e_μ which is the average of the remaining energy level of its neighboring nodes.
- Performs the *Redundancy Determination* algorithm which will be given in details in the next subsection.

Based on the selected action of LA_i, the results of the *Redundancy Determination* algorithm, remaining energy of s_i (e_i) and e_μ, the reinforcement signal to LA_i is computed as follows:

1) If the selected action of LA_i is a_1 and s_i is a redundant node then:

if $e_i < e_\mu$, the reinforcement signal (β_i) is 1, otherwise it is 0.6.

2) If the selected action of LA_i is a_0 and s_i is a redundant node then:

If $e_i < e_\mu$, the reinforcement signal (β_i) is 0, otherwise it is 0.3.

3) If the selected action of LA_i is a_1 and s_i is not a redundant node then:

If $e_i < e_\mu$, the reinforcement signal (β_i) is 0.3, otherwise it is 0.

4) If the selected action of LA_i is a_0 and s_i is not a redundant node then:

If $e_i < e_\mu$, the reinforcement signal (β_i) is 0.6, otherwise it is 1.

As stated above, the reinforcement signal (β_i) may take values from the set {0, 0. 3, 0. 6, 1}. In this set, 0 and .3 are reward values and .6 and 1 are penalty values. Based on the values of β_i, one of the following reward or penalize equations must be used:

$$p_i(n+1) = p_i(n) + a_{high}[1 - p_i(n)] \quad (1)$$
$$p_j(n+1) = 1 - p_i(n+1) \qquad \beta_i(n) = 0$$

$$p_i(n+1) = p_i(n) + a_{low}[1 - p_i(n)] \quad (2)$$
$$p_j(n+1) = 1 - p_i(n+1) \qquad \beta_i(n) = 0.3$$

$$p_i(n+1) = (1 - b_{high})p_i(n) \quad (3)$$
$$p_j(n+1) = 1 - p_i(n+1) \qquad \beta_i(n) = 1$$

$$p_i(n+1) = (1 - b_{low})p_i(n) \quad (4)$$
$$p_j(n+1) = 1 - p_i(\text{n+1}) \qquad \beta_i(n) = 0.6$$

In the above equations, alow, ahigh, blow and bhigh are four parameters which control the amount of reward or penalty given to the learning automata. The amount of reward (penalty) given to a learning automaton which receives reward (penalty) by equation (1) (equation (3)) is higher than the amount of reward (penalty) given to a learning automaton which receives reward (penalty) by equation (2) (equation (4)). This is true only if ahigh>alow and bhigh>blow.

The algorithm proceeds separately in each node si until the iteration number in that node exceeds a specified threshold called MaxIteration. At this time, the operation mode of si is determined based on the action probability vector of LAi. If probability of selecting action a1 is greater than a specified threshold (th), then the operation mode of si will be active, otherwise, the operation mode of si will be sleep.

In order to balance the energy consumption of all nodes of the network, the proposed algorithm must be repeated in some predetermined intervals. This way, those active nodes whose energy levels become lower than their sleeping neighbors, change their operation modes to sleep and instead some of their sleeping neighbors become active to cover their surrounding area.

Note that using the above algorithm, a node having higher energy is more probable to become an active node than a node with lower energy. This is due to the fact that a node with higher energy than the average energy of its neighbors receives rewards according to equation (1) and penalty according to equation (4) whereas a node with lower energy than the average energy of its neighbors receives rewards according to equation (2) and penalty according to equation (3).

A. Determining Redundancy

A node can determine if it is redundant based on the locations and operation modes of its neighboring nodes. To do that, a node si places a virtual grid on its sensing region. A virtual grid consists of a number of grid points. For a more precise decision, a node has to increase the number of grid points and for a faster decision, it has to decrease the number of grid points. If all grid points of the sensing region of a node si are under the coverage of at least K active neighboring nodes, si will be a redundant node. We refer to the above procedure as Redundancy Determination algorithm.

2. Simulation

In this section the performance of the proposed algorithm is evaluated through computer simulations. In these simulations, the results obtained from the proposed method are compared with the results obtained from the existing coverage algorithms; CCP [11], ottawa [13] and PEAS [12]. Ottawa and PEAS methods are only able to solve the problem for 1-coverage, but CCP can handle k-coverage as well. We utilize following criteria for comparison of the performance of these methods: 1) the number of active nodes, 2) the ratio of the average energy of the active nodes to the average energy of the inactive nodes (given by equation (5)) and 3) lifetime of the network.

$$\eta = \left(\frac{\sum_{s_i \in AN} e_i \Big/ |AN|}{\sum_{s_i \in IN} e_i \Big/ |IN|} \right) \quad (5)$$

In the above equation, AN is the set of active nodes and IN is the set of inactive nodes. $|AN|$ and $|IN|$ stands for the cardinality of sets AN and IN respectively.

In the proposed experiments the network environment is assumed to be a 50(m) × 50(m) rectangular area through which a number of sensor nodes are deployed uniformly at random. The transmission radius and sensing radius of sensor nodes are assumed to be 20 and 10 meters respectively. The initial energy level of each sensor node is selected uniformly at random from the interval [1.8 , 2]. The size of data packets and control packets are assumed to be 526 and 8 bits respectively. alow and blow parameters are set to 0.1 and ahigh and bhigh parameters are set to 0.2. ts parameter is set to 95%. Experiments are performed using J-Sim [20] simulator for 100, 200, 300, 400 and 500 sensor nodes. The results are averaged over 10 runs.

Experiment 1: This experiment is conducted to find a suitable value for the MaxIteration parameter. If this parameter is selected too small, proper learning will not be done and if it is selected too large, the energy overhead of the learning process will be significantly high. Therefore, it is desired to find a suitable value for this parameter as a trade off between the amount of energy overhead and proper learning. For this study, we change the value of this parameter from 8 to 60 and find the percentage of area under the k-coverage for each value of MaxIteration. The experiment is repeated for different number of nodes deployed in the network. Figure 1 which gives the results of this experiment shows that 50 is a suitable value for MaxIteration parameter. Selecting values below 50 for MaxIteration results in loosing the full coverage. On the other hand, selecting values above 50 for this parameter has no effect on the coverage, but results in energy waste.

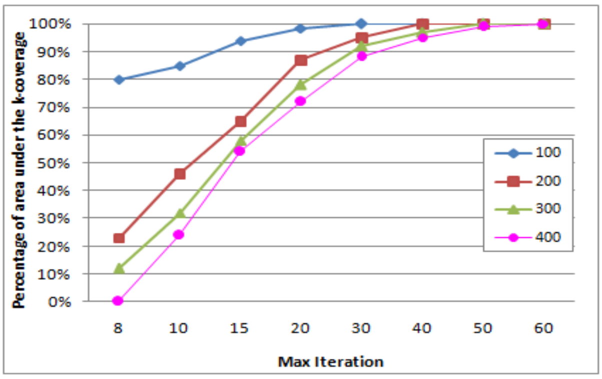


Fig1. Percentage of area under the k-coverage for different values of MaxIteration and different number of deployed nodes in the network

Experiment 2: This experiment is conducted to evaluate the performance of the proposed method in comparison to the mentioned methods in terms of the number of active nodes for different degrees (K) of coverage. Figure 2 gives the result of this comparison for coverage degree 1 and figure 3 gives the result of the comparison for coverage degrees 2 and 3. Note that as it was mentioned earlier, Ottawa and PEAS methods are not designed for coverage degrees above 1 and hence, they are not compared in figure 3. These two figures show that the number of active nodes in the proposed method is noticeably lower than the number of active nodes in Ottawa and PEAS, but the number of active nodes in the CCP method is almost equal to that of the proposed method. Therefore, in terms of the number of active nodes, the proposed method outperforms Ottawa and PEAS and performs almost the same as CCP method.

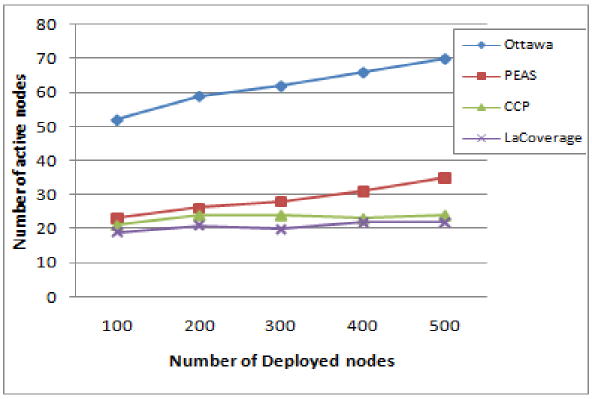


Fig2. Comparison of active nodes in coverage methods with coverage degree 1.

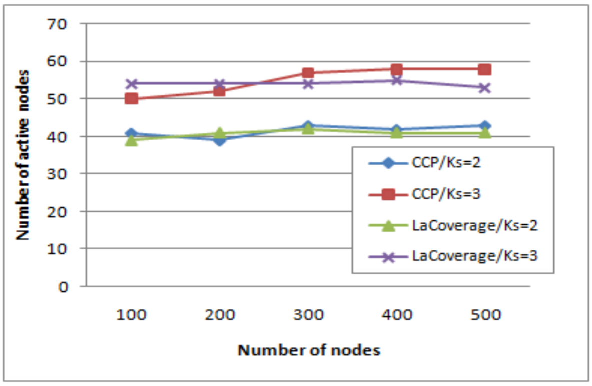


Fig3. Comparison of active nodes in coverage methods with coverage degrees 2 and 3.

Experiment 3: To study the behavior of the proposed method in selection of active nodes according to their residual energy level, in this experiment we plot the value of η for the proposed algorithm and other mentioned algorithms in figure 4. As it can be seen from this figure, in the proposed method unlike other methods, the average residual energy level of active nodes is higher than the average residual energy level of inactive nodes (η >1). This results in a longer lifetime for the network as it will be seen in the next experiment.

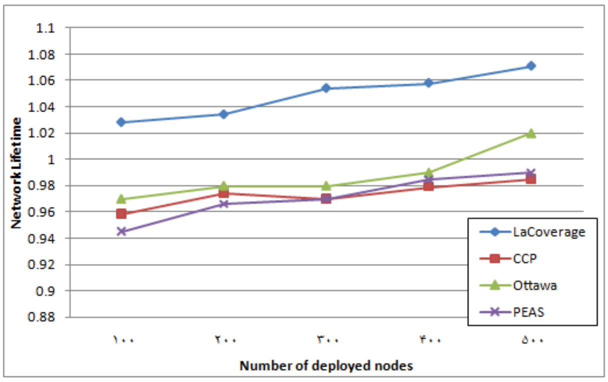
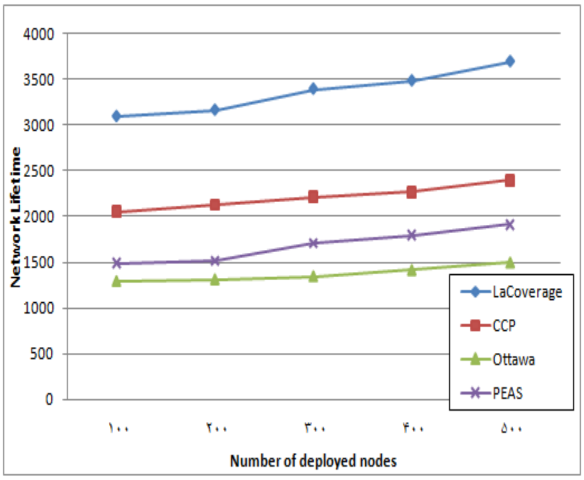


Fig4. Comparison of the ratio of the average energy of the active nodes in to the average energy of the inactive nodes

Experiment 4: In this experiment we study the lifetime of the network when the proposed method and other mentioned methods are used. First node death is used as the definition of the network lifetime in this study [21]. Figure 5 which gives the results of this experiment show that the proposed method outperforms the other methods in prolonging the lifetime of the network. This is due to the fact that in the proposed method, remaining energy of the nodes is considered for selecting the set of active nodes whereas in other mentioned methods, such a consideration is not taken into account.



Fog5.Comparison of the life of the network in different states.

3. Conclusion

In this paper a new method that was based on cellular learning automata was proposed for minimizing the number of active sensors in a densely deployed network in such a way that the network area is k-covered. In the proposed method, a number of quality of service parameters in wireless sensor networks are taken into account simultaneously; network area coverage, energy consumption of the nodes, the number of active nodes and network lifetime. The results of the experiments have shown that this method remarkably increases the lifetime of the network while at the same time, the network area is completely K-covered.

Acknowledgments

The authors acknowledge the support provided by the Office of Vice Chancellor for Research, Kerman Branch, Islamic Azad University, Kerman, Iran.

REFERENCES

[1] D. Chen and K. Varshney, “QoS Support in Wireless Sensor Networks: A Survey” Department of EECS, Syracuse University Syracuse, NY, U.S.A 13244, 2004

[2] S. Slijepcevic and M. Potkonjak, “Power Efficient Organization of Wireless Sensor Networks,” Proc. of IEEE International Conference on Communications, Vol 2, pp 472-476,Helsinki, Finland, June 2001.

[3] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, “Wireless Sensor Networks with Energy Efficient Organization”, Journal of Interconnection Networks, Vol 3, No 3-4, pp 213-229, Dec 2002.

[4] D. Tian and N. D. Georganas, “A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks,” Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications, 2002.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” Proc. of HICSS, Jan 2000.

[6] F. Ye, G. Zhong, S. Lu, and L. Zhang, “Energy Efficient Robust Sensing Coverage in Large Sensor Networks,” Technical Report UCLA, 2002.

[7] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: Coverage, connectivity, and diameter,” Ad Hoc Networks, vol. 3, no. 6,pp. 702–716, November 2005.

[8] S. Kumar, T. H. Lai, and J. Balogh, “On k-coverage in a mostly sleeping sensor network,” in Proc. of ACM MOBICOM’04, Philadelphia, PA, September 2004

[9] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai, “Deploying wireless sensors to achieve both coverage and connectivity,” in Proc. of ACM MobiHoc’06, Florence, Italy, May 2006, pp. 131–142.

[10] H. Zhang and J. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” Ad Hoc and Sensor Wireless Networks: An International Journal, vol. 1, no. 1-2, pp. 89–123, January 2005.

[11] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration for energy conservation in sensor networks,” ACM Transactions on Sensor Networks, vol. 1, no. 1, pp. 36–72, August 2005.

[12] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “Peas: A robust energy conserving protocol for long-lived sensor networks,” in Proc. of International Conference on Distributed Computing Systems (ICDCS’03), Providence, RI, May 2003.

[13] D. Tian and N. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” in Proc. of First ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, September 2002.

[14] N. Ahmed, S. Kanhere, and S. Jha, “Probabilistic coverage in wireless sensor networks,” in Proc. of IEEE Conference on Local Computer Networks (LCN’05), Sydney, Australia, November 2005.

[15] B. Liu and D. Towsley, “A study on the coverage of large-scale sensor networks,” in Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS’04), Fort Lauderdale, Florida, October 2004,

[16] Esnaashari, M. and Meybodi, M. R., "A Cellular Learning Automata based Clustering Algorithm for Wireless Sensor Networks", Sensor Letters, 2008.

[17] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration for energy conservation insensor networks,” ACM Transactions on Sensor Networks, vol. 1, no. 1,pp. 36–72, August 2005.

[18] Crossbow Technology, Inc., http://www.xbow.com/Products/Product_pdf_files /Wireless_pdf/6020-0042-01_A_MICA2.pdf.

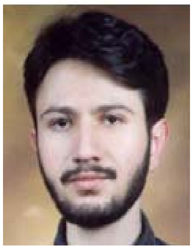
[19] R. Shahani, D. K. Schaeffer, T. H. Lee, ”A 12 mW Wide Dynamic Range CMOS Front-End for a portable GPS Receiver”, Digest of Technical Papers, IEEE International Solid State Circuits Conference, Vol. 40, pp. 368-369, February 1997.

[20] A. Sobeih, W. P. Chen, J. C. Hou, ” J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks”, 2005.

[21] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", In Proc. of IEEE INFOCOM, volume 1, pp. 629-640, March 2004.

Mohammad Ahmadinia received the BS degree in software engineering from Ferdowsi University in Iran, in 2006 and MS degree from Islamic Azad University, Science & Research Branch in Iran, in 2009. Currently, He is a PhD student in Islamic Azad University, Science & Research Branch, in Iran, in Computer Engineering, and he also is a Lecturer in Computer Engineering Department, Islamic Azad University, Kerman Branch, Kerman, Iran. His research interests include Wireless Sensor Network and learning Systems.

Mohammad Reza Meybodi received the BS and MS degrees in Economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the MS and PhD degree from Oklahoma University, USA, in 1980 and 1983, respectively in Computer Science. Currently, he is a full professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to his current position, he worked from 1983 to 1985 as an assistant professor at Western Michigan University,and from 1985 to 1991 as an associate professor at Ohio University, USA. His research interests include channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.



Mahdi Esnaashari received both the B.S. and M.S. Degrees in Computer Engineering from the Amirkabir University of Technology in Iran, in 2002 and 2005, respectively. Currently, he is a Ph.D. student in the Computer Engineering Department at the Amirkabir University of Technology,Tehran, Iran. His research interests include computer networks, learning systems and soft computing.