

# Learning Automata Based Face-Aware Mobicast

S. M. Safavi · M. R. Meybodi · M. Esnaashari

Published online: 30 January 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

**Abstract** Target tracking is one of the most popular applications of the wireless sensor networks. It can be accomplished using different approaches and algorithms, one of which is the spatiotemporal multicast protocol, called “mobicast”. In this protocol, it is assumed that the area around the moving target, called the delivery zone, is known at any given time during the operation of the network. The aim of the protocol is to awake sensor nodes, which will be within the delivery zone in the near future, to be prepared for tracking the approaching moving target. In this paper, we propose a novel mobicast algorithm, aiming at reducing the number of awakened sensor nodes. To this end, we equipped every sensor node with a learning automaton, which helps the node in determining the sensor nodes it must awaken. To evaluate the performance of the proposed algorithm, several experiments have been conducted. The results have shown that the proposed algorithm can significantly outperform other existing algorithms such as forward-zone constrained and FAR in terms of energy consumption, number of active nodes, number of exchanged packets and slack time.

**Keywords** Wireless sensor networks · Mobicast · Learning automata · Face

## 1 Introduction

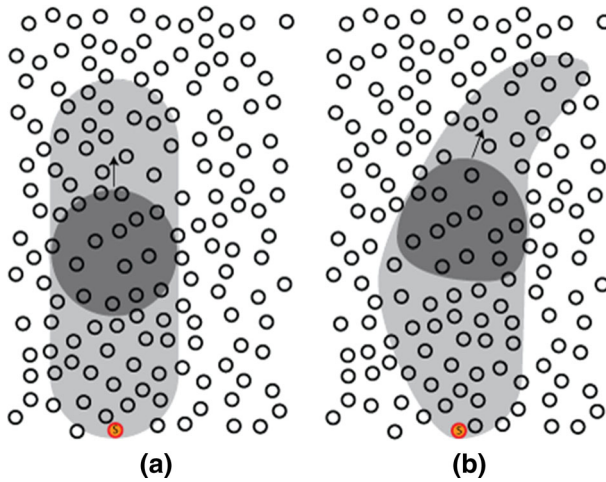
Wireless sensor networks (WSNs) are a hi-technology that researchers have recently focused on. A typical WSN consists of hundreds or thousands of sensor nodes equipped with one or

---

S. M. Safavi (✉) · M. R. Meybodi  
Soft Computing Laboratory, Computer Engineering and Information Technology Department,  
Amirkabir University of Technology, Tehran, Iran  
e-mail: mohsen.safavi@aut.ac.ir

M. R. Meybodi  
School of Computer Science, Institutes for Studies in Theoretical Physics  
and Mathematics (IPM), Tehran, Iran

M. Esnaashari  
Information Technology Department, Cyber-Space Research Institute, Tehran, Iran



**Fig. 1** Mobicast examples

more specific sensors such as temperature sensors or cameras. Also each of these nodes have wireless communicate circuit which can communicate with its neighbor nodes and can be used as a router for sending collected information to the base station which called sink node. The sensor nodes can work in two modes: active or sleep. In the active mode, communicating and sensing circuits are switched on, while in the sleep mode, they are switched off. Energy is a vital resource in WSN, and hence, trying to deactivate the sensor nodes when they are not necessarily needed is a very common practice in such networks [1, 2].

Among the numerous applications of sensor networks, target tracking is one of the most important. Considering the energy constraints of WSN, in target tracking applications, only the nearest sensor nodes to the moving target must participate in the tracking task [3–5]. Other nodes can be in sleep mode, saving their energies. Thus, a sensor node must be awakened before the moving target reaches its sensing area. Without a precise awakening mechanism, the moving target may be lost and the tracking task is failed. For a precise awakening mechanism, it is necessary to have a group of active nodes around the moving target at any time, responsible for not only monitoring the target, but also awakening. An active node must decide on the set of sensor nodes for awakening, considering the moving path of the target.

Huang et al. [6] after studying different aspects of object tracking, concluded that both spatial and temporal (spatiotemporal) constraints must be satisfied simultaneously for precise tracking. The result of their investigation has formed a concept called Mobicast. Mobicast expresses the delivery of the messages to a set of nodes; just in time before the moving target arrives at their sensing region. This is done by considering a mobile delivery zone, which is defined as the set of nodes, which must be awakened (or more precisely, the area within which such nodes present). The delivery zone is mobile, because it moves along the moving path of the target.

Formally, a mobicast session is specified by a quadruple,  $(m, Z[t], T_s, T)$ .  $m$  is the mobicast message.  $Z[t]$  is the group of nodes (delivery zone) where  $m$  should be disseminated. As the delivery zone  $Z[t]$  evolves over time, the set of recipient nodes of  $m$  changes as well.  $T_s$  and  $T$  are the sending time and duration of the mobicast session, respectively. Each node receives this quadruple, can compute location of delivery zone. Figure 1a shows a rectangular delivery zone moving upward. Figure 1b shows a more general scenario where the delivery zone can change its direction, size and shape over time.

Huang et al. [6] proposed a mobicast protocol with 100% tracking guaranty across the network. The main problem of their protocol is its high energy consumption, because of the large delivery zone considered. Huang et al. [7] presented four mobicast protocols. The first protocol is very simple and suffers from many problems. This one is only used for clarifying the concepts. Every node that receives a mobicast message, checks if it resides within the delivery zone of the message, and if this is not the case, it does not deliver the message to the upper layers (specifically application layer). Regardless of this fact, the node rebroadcasts the receiving message in its vicinity. This protocol is not efficient from energy point of view because it has global flooding problem. In addition, it delivers the mobicast messages too early to the nodes, a contradiction with the “just in time” property of mobicast. Average of such early delivery times is referred to as slack time.

The second protocol, named “DZC”, is just like the previous one, with the exception that it uses the concepts of “hold and forward” zones in order to decrease the slack time. A sensor node, within the hold zone, holds received messages and forwards them only when it resides within the forwarding zone.

The third protocol, “FZC”, works in two phases—initialization and operational. This protocol uses a forwarding zone, much bigger than the delivery zone, which always moves ahead of the delivery zone, guarantying the delivery of messages in time. Its main problem is the size of the forwarding zone, which usually contains many, unnecessary, nodes. The last presented protocol is not actually a new one. Indeed, in this protocol the size of the forwarding zone is reduced to some extent, sacrificing guaranty level of message delivery to save more energy.

In Huang et al. [8], proposed the compactness metric, which is a relation between the Euclidean distance and the network spatial distance (hops) among network nodes. Using this concept, they introduced an adaptive mobicast protocol, which locally estimates the compactness metric and uses it for enhancing the size of the forwarding zone in different sensor network fields.

Huang et al. [9] introduced a new mobicast protocol, using the face concept. A face is the subdivision of maximal connected subset of the plane that does not contain a point on an edge or a vertex [9]. They called the nodes around a face as spatial neighbors. In their protocol, every node must be aware of the faces and the spatial neighbors. When a mobicast message arrives at a node, the node checks if any of the faces is in the delivery zone and if so, it rebroadcasts the message locally considering the hold and forward zones concepts.

All above mentioned protocols, have assumed that the mobile target move with the constant velocity in a straight direction. Chen et al. [10] designed a novel mobicast routing protocol, called variant-egg-based mobicast (VE-mobicast), to support a different movement path for the moving target. They defined the concept of predictive accuracy as the percentage of the sensor nodes which are located in both delivery and forwarding zones out of the sensor nodes within the delivery zone. If all sensor nodes in the delivery zone are also in the forwarding zone, then the predictive accuracy is 100%. On the other hand, predictive accuracy is 0% if none of the sensor nodes in the delivery zone are in the forwarding zone. Chen and et al. tried to improve the predictive accuracy in VE-mobicast. At any given time  $t$ , VE-mobicast is divided into two phases. In the first phase, called the egg estimation phase, the variant-egg forwarding zone in time  $t+1$  ( $F[t+1]$ ) is estimated by sensor nodes in  $H[t] = F[t] \cap F[t+1]$ . In the second phase, called the distributed variant-egg-based mobicast, the shape of the next variant-egg forwarding zone is adjusted using the results of the first phase. The shape of the forwarding zone is based on the cassini oval equation [11].

HVE-mobicast routing protocol, presented in [12], extends VE-mobicast so that it can handle different moving speeds for the moving target. In HVE-mobicast, sensor network

is first clustered prior to the target tracking task. During the target tracking task, message delivery in the forwarding zone accomplished by two phase; cluster-to-cluster and cluster-to-node. In the cluster-to-cluster phase, only the cluster-heads and relay nodes are notified to wake up and become active. In the cluster-to-node phase, member nodes of the awakened clusters are notified and awakened by their cluster-heads considering the estimated arrival time of the delivery zone. The main contribution of the HVE-mobicast routing protocol is the speed and accuracy of nodes waking up.

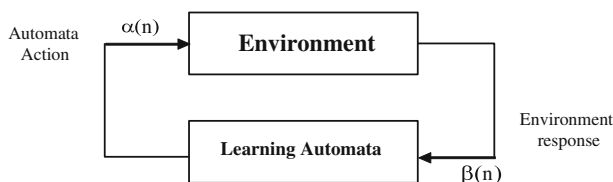
In this paper, we propose a novel mobicast protocol, based on learning automata, for minimizing the activated nodes during mobicast sessions. In this protocol, like the protocol given in [9], when a node  $n_i$  receives a message, it checks if one of its faces is in the delivery zone and if so,  $n_i$  attaches its id to the received message and then rebroadcasts it in its vicinity. In addition,  $n_i$  holds a copy of the message for some certain period. Copies of the message maybe delivered to  $n_i$  during this period from different paths. All such copies will be dropped as soon as they arrived at the node except the last one, which is indeed received from the longest path.  $n_i$  then creates two signals, a positive signal and a negative signal. The positive signal will be sent along the path from which the first message was received, and the negative signal will be sent along the longest path. This way, the probability of receiving mobicast messages from the shortest paths increases eventually. Compared with the other protocols, our protocol significantly reduces the slack time and the number of active nodes during mobicast sessions.

Rest of the paper is organized as follows: In the next section, we will introduce learning automata as an intelligent agent used in the proposed protocol. In Sect. 3, we will give the details of the protocol. Simulation results will be given in Sect. 4. Section 5 concludes the paper.

## 2 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 works based on the selected action and responses to the automata with a reinforcement signal Based on selected action, and based on received signal, the automata updates its internal state and selects next action. Figure 2 depicts the relationship between an automata and its environment.

Environment can be defined by the triple  $E \equiv \{\alpha, \beta, c\}$  where  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  represents a finite input set,  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  represents the output set, and  $c \equiv \{c_1, c_2, \dots, c_r\}$  is a set of penalty probabilities, where each element  $c_i$  of  $c$  corresponds to one input action  $i$ . Environments in which  $\beta$  can take only binary values 0 or 1 are referred to as P-models. 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



**Fig. 2** Relationship between environment and learning automata

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  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  represents the action set of the automata,  $\beta = \{\beta_1, \dots, \beta_m\}$  represents the input set,  $p = \{p_1, \dots, p_r\}$  represents the action probability set, and finally  $p(n + 1) = T[\alpha(n), \beta(n), p(n)]$  the learning algorithm. This automaton operates as follows. Based on the action probability set  $p$ , automaton randomly selects an action  $i$ , and performs it on the environment. After receiving the environment’s reinforcement signal, automaton updates its action probability set based on Eq. (1) for favorable responses, and Eq. (2) for unfavorable ones.

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

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

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  $a \ll b$ , it is called  $L_{R-p}$ ,<sup>2</sup> and for  $b = 0$ , it is called  $L_{R-I}$ .<sup>3</sup>

For more information about learning automata the reader may refer to [13]. Learning automata has been recently used successfully in the area of WSN; LABER: A Learning Automata Based Energy-aware Routing Protocol for Sensor Networks [14], A cellular learning automata based clustering algorithm for WSNs [15], Dynamic point coverage in WSNs: A learning automata approach [16], Data aggregation in sensor networks using learning automata [17], A learning automata based scheduling solution to the dynamic point coverage problem in WSNs [18], Dynamic Point Coverage Problem in WSNs: A Cellular Learning Automata Approach [19], EEMLA: energy efficient monitoring of WSN with learning automata [20], A cellular learning automata-based deployment strategy for mobile WSNs [21], to mention a few.

### 3 Proposed Protocol

The aim of the proposed protocol is at minimizing the number of active nodes as well as the slack time during mobicast sessions. To this end, we make use of Face-aware mobicasting enhanced with learning techniques.

First, we have to precisely define the concept of “spatial neighborhood”. On a planar graph, each node has one or more adjacent faces. We define the “spatial neighborhood” of a node in a planar graph to be the set of nodes in all faces adjacent to that node except the node itself. In the proposed algorithm, we assume that every node knows its spatial neighbors. In what follows, we first describe the message formats used in the proposed algorithm and then, present the algorithm operation.

---

<sup>1</sup> Linear reward-penalty.  
<sup>2</sup> Linear reward epsilon penalty.  
<sup>3</sup> Linear reward inaction.

### 3.1 Message Format

Our protocol has two types of messages. The first one is the mobicast message and the second one is the signal message. Mobicast message contains the following information:

1. Sending time
2. Coordinates of the initial delivery zone: This field contains an ordered sequence of the locations of the initial vertices of the delivery zone, if the zone is a polygon. For a circular delivery zone, the radius and the initial center are recorded instead.
3. Velocity of the delivery zone
4. Message lifetime: It is used for terminating the mobicast session.
5. Message type: Indicates the shape of the delivery zone, e.g., rectangle, pentagon, circle, ellipse, etc.
6. Sequence number
7. A bounded list of last forwarders IDs: This helps nodes in determining the path along which the message traversed towards them.

The coordinates of the initial delivery zone combined with the velocity of the zone and the sending time of the message can be used to determine the location of the delivery zone at any time in the future.

The second message type, i.e. the signal message, is used as the reinforcement signal for the learning automata residing in the sensor nodes. Signal message contains the list of forwarders IDs (just like the mobicast message) and the signal itself (either a positive signal or negative one).

### 3.2 Algorithm Operation

Each node in the network is equipped with a learning automaton. The learning automata embedded in each node has two actions: “sending” and “not sending”. In the beginning, the probability of “sending” action is equal to 1 and the probability of “not sending” action is equal to 0. According to FAR protocol, when a mobicast message  $m$  arrives at a node  $n_i$ , if at least one of the spatial neighbors of  $n_i$  is in the delivery zone,  $n_i$  must rebroadcast the message. But, in the proposed protocol,  $n_i$  does this only if its learning automaton selects the “sending” action. If this action is selected, then  $n_i$  attaches its ID to the list of the last forwarder field, rebroadcasts the message locally considering the hold and forwarding zones and waits for a certain period  $\rho$ . During this period, copies of the message  $m$  may be received at  $n_i$  from different paths.  $n_i$  ignores all of these copies except the last one, which is indeed received from the longest path.  $n_i$  then creates two signal messages; a positive one and a negative one. The positive signal will be sent along the path from which the original message  $m$  was received. On the other hand, the negative signal will be sent along the longest path. When a node receives a signal message, it updates the action probability vector of its learning automata according to Eqs. (1) and (2), if top ID in last forwarder field be equal with its ID. Note that the duplicate messages received after the period  $\rho$  are simply dropped, but considering long enough duration for  $\rho$ , this can be happened very rarely. In Fig. 3, a typical example of the execution of the proposed algorithm (b) and FAR algorithm (a) is depicted. In this example the moving target moves from a similar path multiple of times, so the proposed algorithm converged and only shortest paths remained active.

It's important to know that at the beginning, the proposed algorithm and FAR algorithm perform the same, but when the moving target repeats its moving path, the proposed algorithm

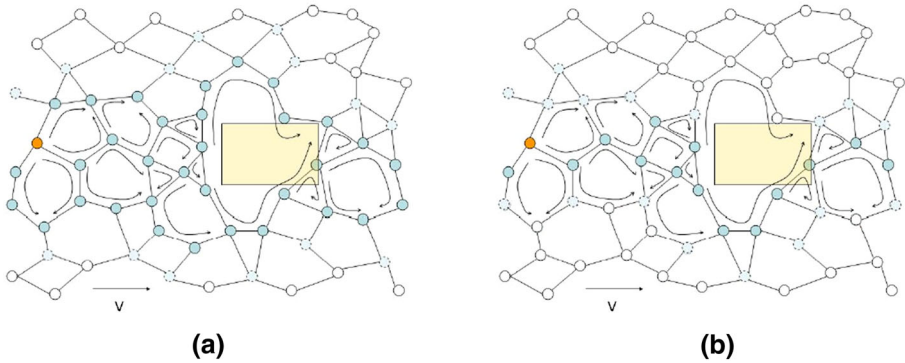


Fig. 3 FAR protocol versus our LA-face aware protocol execution examples

learns that path and thus, become superior to FAR in terms of the number of active nodes. The guaranty level of the proposed algorithm is the same as that of FAR.

### 4 Simulation Results

In order to study the performance of the proposed protocol, multiple protocols and the proposed protocol are simulated using J-SIM. We focus on three metrics to compare protocols: Slack time, Message exchange, and number of active nodes.

The simulation consist of 800 node dispersed randomly with a uniform distribution in an area of  $1,000 \times 600 \text{ m}^2$ . The delivery zone assumed a rectangle with size  $30 \times 10 \text{ m}^2$  that moves with constant velocity  $40 \text{ m/s}$  in a straight line. The mobile entity moves the same path multiple times while our algorithm be converged. So our algorithm in the worst case is like the FAR protocol, but with multiple path of mobile object it converges and results has been so better. Note that the  $a$  and  $b$  parameters of nodes learning automata is considered to be  $0.5$  and  $0.01$  respectively.

Slack time is a metric that shows how much time mobicast packets delivered sooner. Figure 4 depicted comparison of three different protocols. The X axis is the expected delivery

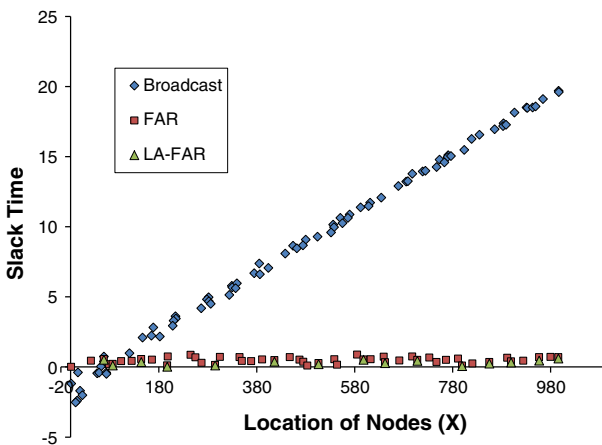


Fig. 4 Slack time comparison of four different protocols

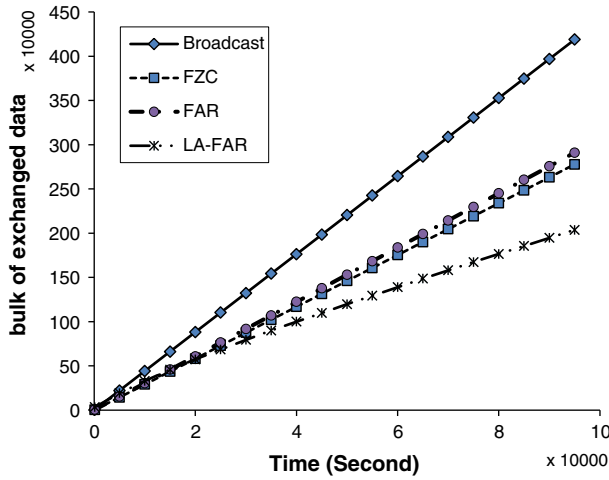


Fig. 5 Message exchange overload compression

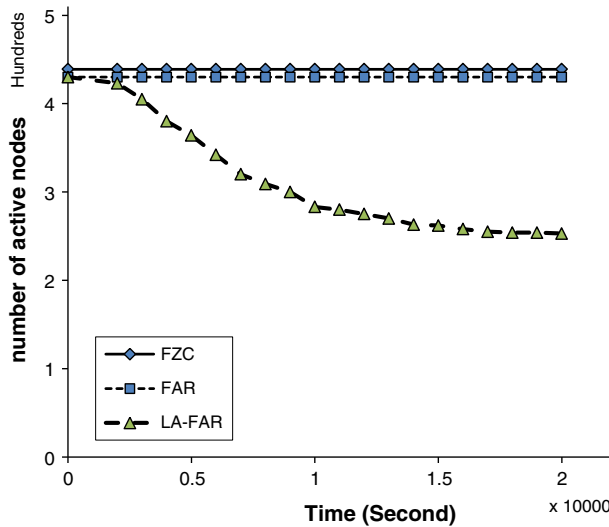


Fig. 6 Number of active nodes compression

deadline for nodes in each location, i.e. the first time they are expected to enter the delivery zone. If a node get the packet before deadline, it was locate in top of X axis, otherwise it was locate under X axis. The Y axis show amount of slack time. Diamond signs shows flooding protocol that broadcast messages to all nodes with as-soon-as possible strategy in network. Square signs shows FAR mobicast protocol. Finally, the triangles shows our algorithm slacktime. The Broadcast delivery has worst slack time because upon a packet received, it forward it. In our protocol because only a small number of nodes are active and only shortest paths are active it's so closed to delivery deadline.

*Message exchange* is the number of packets exchanged between nodes across mobicast session. The message exchange load of forward-zone constrained (FZC) and FAR mobicast



protocols are relatively same over multiple pass of mobile object from same path (because FAR has bigger messages, in Fig. 5, its exchanged data is slightly bigger than FZC protocol) but our protocol in first multiple passes, have higher load, but load is reduced by increasing the same path traversing. After a certain passes and strengthen the shortest paths (learning automata converged), the load of message exchange will be constant.

As presented in Fig. 5, in the few first passes the bulk of exchanged data are high but with removing longest paths from message dissemination process, it reduces significantly. In the end of curve, because only shortest path reminded, bulk of exchanged data become stable.

*Number of active nodes* is number of nodes that activated during mobicast session. As shown in Fig. 6, this is obvious in the first pass of mobile entity, number of nodes that activated in our protocol is exactly equal to FAR protocol. But with repeat passage after a while only shortest paths remain active so number of active nodes in our protocol decrease dramatically.

## 5 Conclusion

We use learning automata as an intelligent tool to decrease number of active nodes in a tracking session. For this work, number of communicated packets increased in short time interval but after this time, it decreases using learning method. When the number of repetition on the same path is high, this protocol has significant advantage. In this work, we suppose the mobile entity have a constant velocity and direct direction as what is supposed in [6–9], so we aren't compare it with recent work like VE-mobicast [10] that supposed mobile entity can change its direction or HVE-mobicast [12] that supposed mobile entity can change both of direction and speed and have bigger energy consumption in compare to former protocols. In the future works, just likes [10, 12] we try to use different methods like clustering methods, so the nodes able optimize shortest path activated nodes in different directions and speeds.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

## References

1. Akyildiz, I., et al. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.
2. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12), 2292–2330.
3. Bhatti, S., & Xu, J. (2009). Survey of target tracking protocols using wireless sensor network. In *Wireless and mobile communications, 2009. ICWMC'09. Fifth international conference on*. IEEE.
4. Ramya, K., Kumar, K. P., & Rao, V. S. (2012). A Survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, 3(4), 93–108.
5. Naderan, M., et al. (2012). Survey of mobile object tracking protocols in wireless sensor networks: A network-centric perspective. *International Journal of Ad Hoc and Ubiquitous Computing*, 11(1), 34–63.
6. Huang, Q., Lu, C., & Roman, G.-C. (2003). Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints. In *Information processing in sensor networks*. Berlin: Springer.
7. Huang, Q., Lu, C., & Roman, G.-C. (2004). Design and analysis of spatiotemporal multicast protocols for wireless sensor networks. *Telecommunication Systems*, 26(2–4), 129–160.
8. Huang, Q., Lu, C., & Roman, G.-C. (2003). Spatiotemporal multicast in sensor networks. In *Proceedings of the 1st international conference on embedded networked sensor systems*. ACM.
9. Huang, Q., Lu, C., & Roman, G.-C. (2004). Reliable mobicast via face-aware routing. In *INFOCOM 2004. Twenty-third annual joint conference of the IEEE computer and communications societies*. IEEE.

10. Chen, Y.-S., Ann, S.-Y., & Lin, Y.-W. (2008). VE-mobicast: A variant-egg-based mobicast routing protocol for sensor networks. *Wireless Networks*, *14*(2), 199–218.
11. Weisstein, E. W. (2005). Cassini ovals. From *MathWorld*. A Wolfram Web Resource. <http://mathworld.wolfram.com/CassiniOvals.html>.
12. Chen, Y.-S., et al. (2009). HVE-mobicast: A hierarchical-variant-egg-based mobicast routing protocol for wireless sensor networks. *Telecommunication Systems*, *41*(2), 121–140.
13. Narendra, K. S., & Thathachar, M. A. (1974). Learning automata—a survey. *Systems, Man and Cybernetics, IEEE Transactions on*, *4*, 323–334.
14. Abolhasani, S., Meybodi, M., & Esnaashari, M. (2007). LABER: A learning automata based energy-aware routing protocol for sensor networks. In *Proceedings of the 3rd information and knowledge technology*.
15. Esnaashari, M., & Meybodi, M. (2008). A cellular learning automata based clustering algorithm for wireless sensor networks. *Sensor Letters*, *6*(5), 723–735.
16. Esnaashari, M. & Meybodi, M. (2009). Dynamic point coverage in wireless sensor networks: A learning automata approach. In *Advances in computer science and engineering* (pp. 758–762). Berlin: Springer.
17. Esnaashari, M., & Meybodi, M. R. (2010). Data aggregation in sensor networks using learning automata. *Wireless Networks*, *16*(3), 687–699.
18. Esnaashari, M., & Meybodi, M. R. (2010). A learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks. *Computer Networks*, *54*(14), 2410–2438.
19. Esnaashari, M., & Meybodi, M. R. (2010). Dynamic point coverage problem in wireless sensor networks: A cellular learning automata approach. *Ad Hoc & Sensor Wireless Networks*, *10*(2–3), 193–234.
20. Mostafaei, H., Meybodi, M. R., & Esnaashari, M. (2010). EEMLA: Energy efficient monitoring of wireless sensor network with learning automata. In *Signal acquisition and processing, 2010. ICSAP'10. International conference on*. IEEE.
21. Esnaashari, M., & Meybodi, M. R. (2011). A cellular learning automata-based deployment strategy for mobile wireless sensor networks. *Journal of Parallel and Distributed Computing*, *71*(7), 988–1001.



**S. M. Safavi** received the B.S. degree in Computer Engineering from the Islamic Azad University of Zanjan in Iran in 2008. Currently, he is a M.S. student in Computer Engineering Department at the Amirkabir University of technology, Tehran, Iran. His research interests include Computer networks, learning systems, Computer architecture and operating systems.



**M. R. Meybodi** received the B.S. and M.S. degrees in Economics from the Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degrees from the 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 current position, he worked from 1983 to 1985 as an Assistant Professor at the Western Michigan University, and from 1985 to 1991 as an Associate Professor at the Ohio University, USA. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.



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