Learning Automata Based Face-aware Mobicast

S. M. Safavi [1] , M. R. Meybodi [1,2], M. Esnaashari[3]

[1]Soft Computing Laboratory, Computer Engineering and Information Technology Department
Amirkabir University of Technology, Tehran, Iran
[2] Institutes for Studies in Theoretical Physics and Mathematics (IPM)
School of Computer Science, Tehran, Iran
[3] Information Technology Department
Cyber-Space Research Institute, Tehran, Iran
Mohsen.safavi@aut.ac.ir

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 FZC 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 (WSN) 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 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 1(a) shows a rectangular delivery zone moving upward. Figure 1(b) shows a more general scenario where the delivery zone can change its direction, size and shape over time.
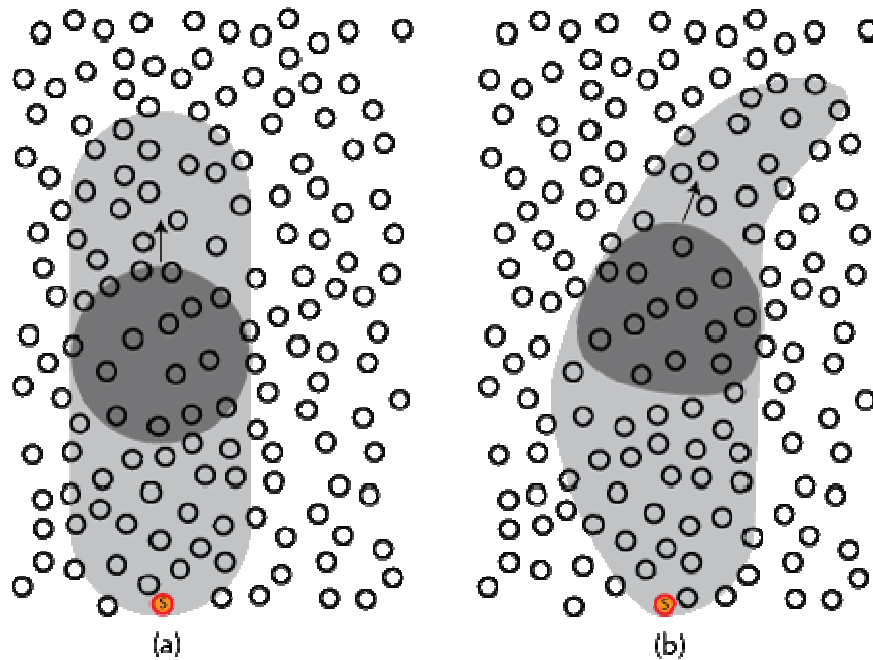
Figure 1: Mobicast examples

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. in [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 [8] Huang et al. 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. in [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. in [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 $H[t] = F[t] \bigcap F[t+1]$. 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 Section 3, we will give the details of the protocol. Simulation results will be given in Section 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.
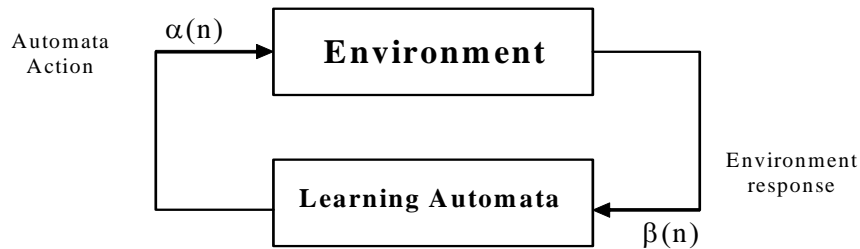

Figure 2: Relationship between Environment and Learning Automata

Environment can be defined by the triple $E \equiv \{\alpha, \beta, c\}$ where $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ represents a finite input set, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ represents the output set, and $c \equiv \{c_1, c_2, ..., c_r\}$

is a set of penalty probabilities, where each element c_i of C corresponds to one input action i. Environments in which β 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 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 $\alpha = \{\alpha_1, ..., \alpha_r\}$ represents the action set of the automata, $\beta = \{\beta_1, ..., \beta_m\}$ represents the input set, $p = \{p_1, ..., p_r\}$ 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 Equation (1) for favorable responses, and Equation (2) for unfavorable ones.

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

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

In these two equations, a and b are reward and penalty parameters respectively. For $a = b$, learning algorithm is called L_{R-P}[1], for $a << b$, it is called L_{R_P}[2], and for $b = 0$, it is called L_{R-I}[3].

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 wireless sensor networks [15], Dynamic point coverage in wireless sensor networks: 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 wireless sensor networks [18], Dynamic Point Coverage Problem in Wireless Sensor Networks: A Cellular Learning Automata Approach [19], EEMLA:

---

[1] Linear reward-penalty.
[2] Linear reward epsilon penalty.
[3] Linear reward inaction.

energy efficient monitoring of wireless sensor network with learning automata [20], A cellular learning automata-based deployment strategy for mobile wireless sensor networks [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.

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 massage, 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).

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 ρ. 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 equations (1) and (2), if top ID in last forwarder field be equal with its ID. Note that the duplicate messages received after the period ρ are simply dropped, but considering long enough duration for ρ, this can be happened very rarely. In figure 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.

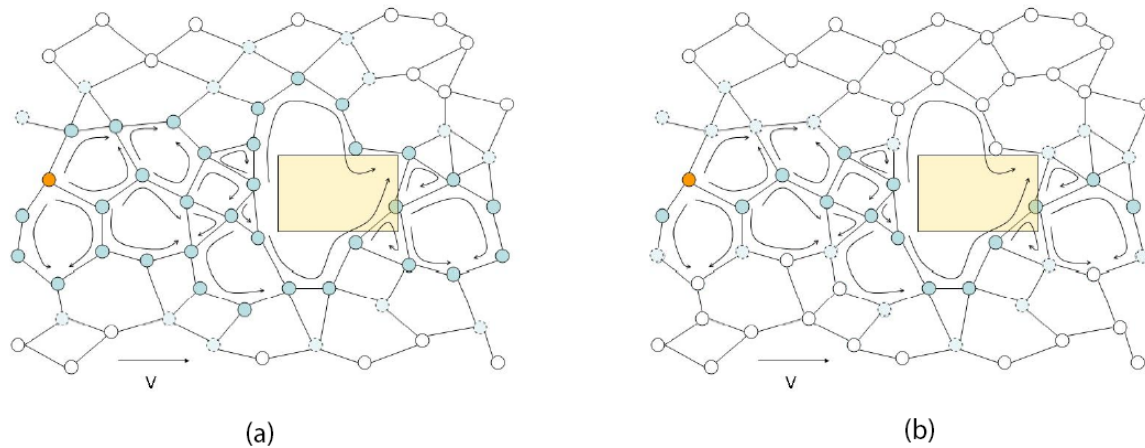(a)   (b)

Figure 3: FAR protocol VS our LA-face aware protocol execution examples

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 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 1000x600 m^2. The delivery zone assumed a rectangle with size 30X10 m^2 that moves with constant velocity 40 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 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.
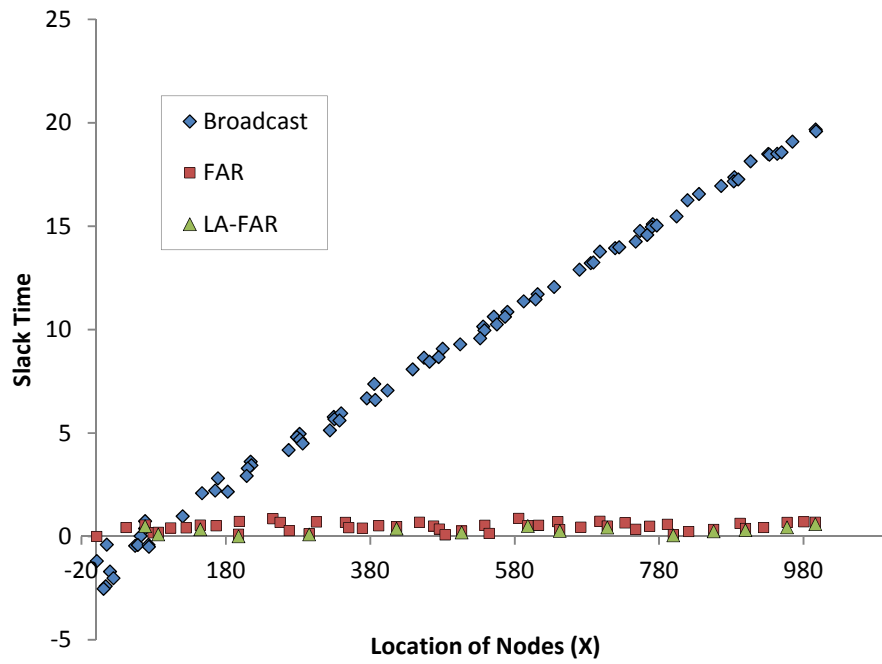
Figure 4: Slack time comparison of four different protocols

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 figure 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.
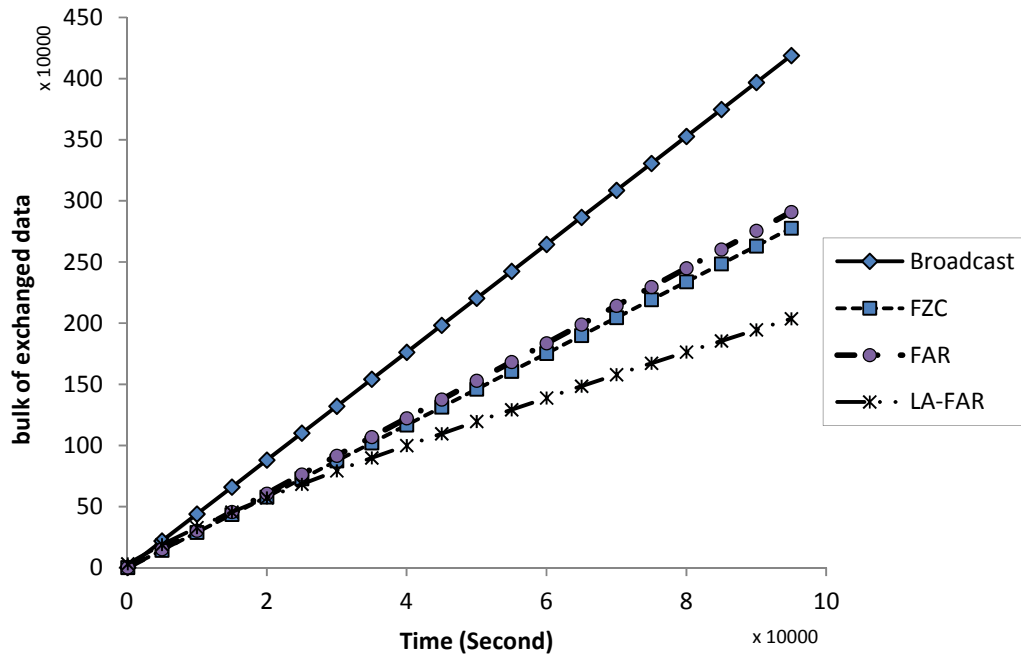
Figure 5: Message exchange overload compression

As presented in figure 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. 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.
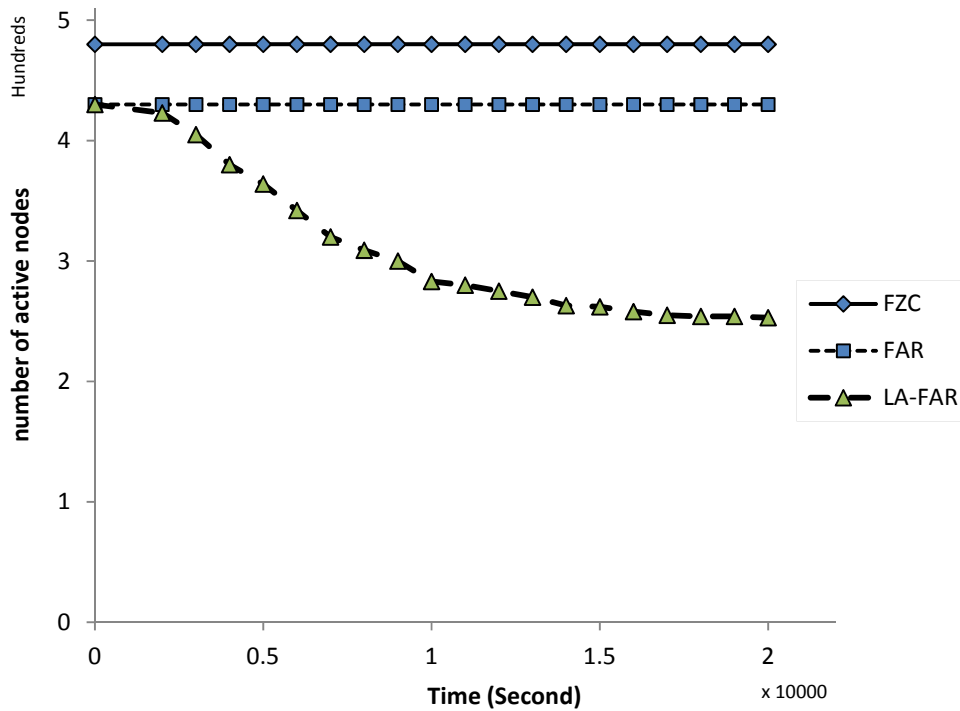
Figure 6: number of active nodes compression

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.

6. References

1. Akyildiz, I., et al., *Wireless sensor networks: a survey.* Computer Networks, Elsevier, 2002: p. 393-422.
2. Yick, J., B. Mukherjee, and D. Ghosal, *Wireless sensor network survey.* Computer networks, 2008. **52**(12): p. 2292-2330.
3. Bhatti, S. and J. Xu. *Survey of target tracking protocols using wireless sensor network*. in *Wireless and Mobile Communications, 2009. ICWMC'09. Fifth International Conference on*. 2009. IEEE.

4.   Ramya, K., K.P. Kumar, and V.S. Rao, *A Survey on Target Tracking Techniques in Wireless Sensor Networks.* International Journal of Computer Science and Engineering Survey, 2012. **3**(4).

5.   Naderan, M., et al., *Survey of mobile object tracking protocols in wireless sensor networks: a network–centric perspective.* International Journal of Ad Hoc and Ubiquitous Computing, 2012. **11**(1): p. 34-63.

6.   Huang, Q., C. Lu, and G.-C. Roman. *Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints*. in *Information Processing in Sensor Networks*. 2003. Springer.

7.   Huang, Q., C. Lu, and G.-C. Roman, *Design and analysis of spatiotemporal multicast protocols for wireless sensor networks.* Telecommunication Systems, 2004. **26**(2-4): p. 129-160.

8.   Huang, Q., C. Lu, and G.-C. Roman. *Spatiotemporal multicast in sensor networks*. in *Proceedings of the 1st international conference on Embedded networked sensor systems*. 2003. ACM.

9.   Huang, Q., C. Lu, and G.-C. Roman. *Reliable mobicast via face-aware routing*. in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. 2004. IEEE.

10.   Chen, Y.-S., S.-Y. Ann, and Y.-W. Lin, *VE-mobicast: a variant-egg-based mobicast routing protocol for sensornets.* Wireless Networks, 2008. **14**(2): p. 199-218.

11.   Weisstein, E.W., *Cassini ovals.* From MathWorld. A Wolfram.

12.   Chen, Y.-S., et al., *HVE-mobicast: a hierarchical-variant-egg-based mobicast routing protocol for wireless sensornets.* Telecommunication Systems, 2009. **41**(2): p. 121-140.

13.   Narendra, K.S. and M.A. Thathachar, *Learning automata-a survey.* Systems, Man and Cybernetics, IEEE Transactions on, 1974(4): p. 323-334.

14.   Abolhasani, S., M. Meybodi, and M. Esnaashari, *LABER: A Learning Automata Based Energy-aware Routing Protocol for Sensor Networks.* Proceedings of the 3rd Information and Knowledge Technology, 2007.

15.   Esnaashari, M. and M. Meybodi, *A cellular learning automata based clustering algorithm for wireless sensor networks.* Sensor Letters, 2008. **6**(5): p. 723-735.

16.   Esnaashari, M. and M. Meybodi, *Dynamic point coverage in wireless sensor networks: A learning automata approach*, in *Advances in Computer Science and Engineering*. 2009, Springer. p. 758-762.

17.   Esnaashari, M. and M.R. Meybodi, *Data aggregation in sensor networks using learning automata.* Wireless Networks, 2010. **16**(3): p. 687-699.

18.   Esnaashari, M. and M.R. Meybodi, *A learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks.* Computer Networks, 2010. **54**(14): p. 2410-2438.

19.   Esnaashari, M. and M.R. Meybodi, *Dynamic Point Coverage Problem in Wireless Sensor Networks: A Cellular Learning Automata Approach.* Ad Hoc & Sensor Wireless Networks, 2010. **10**(2-3): p. 193-234.

20.   Mostafaei, H., M.R. Meybodi, and M. Esnaashari. *EEMLA: energy efficient monitoring of wireless sensor network with learning automata*. in *Signal Acquisition and Processing, 2010. ICSAP'10. International Conference on*. 2010. IEEE.

21.   Esnaashari, M. and M.R. Meybodi, *A cellular learning automata-based deployment strategy for mobile wireless sensor networks.* Journal of Parallel and Distributed Computing, 2011. **71**(7): p. 988-1001.