LA-Mobicast: A Learning Automata Based Mobicast Routing Protocol for Wireless Sensor Networks

M. Gholipour[1] **M. R. Meybodi**[2,3]

[1]Computer Engineering and Information Technology Department
Islamic Azad University, Qazvin, Iran
[2]Computer Engineering and Information Technology Department
Amirkabir University of Technology, Tehran, Iran[*]
[3]Institute for Studies in Theoretical Physics and Mathematics (IPM), School of Computer Science
Tehran Iran

Abstract: The spatiotemporal character of mobicast in sensor networks relates to obligation to deliver a message to all the nodes that will be present at time *t* in some geographic zone *Z*, where both the location and the shape of the delivery zone are the functions of time over some interval (t_{start}, t_{end}). In this paper a learning automata based mobicast protocol for sensor networks to support applications which require spatiotemporal coordination has been proposed. The proposed protocol which we call it LA-Mobicast uses the shape and the size of the forwarding zone to achieve high predicted accuracy. The proposed protocol use learning automata to adaptively determine the location and the shape of the forwarding zone in such away that the same number of wake-up sensor nodes be maintained. The proposed protocol is a fully distributed algorithm which requires lesser communication overhead in determining the forwarding zone and the mobicast message forwarding overhead. In order to show the performance of the proposed protocol, computer simulations have been conducted and the results obtained are compared with the results obtained for five existing mobicast protocols. The results of comparison show that the proposed protocol outperforms existing mobicast protocols in terms of slack time, message exchange, node involved and guarantee percent.

Keywords: Sensor Networks, Mobicast, Adaptive Protocol, Learning Automata

[*] Corresponding author; E-mails: gholipur@gmail.com,mmeybodi@aut.ac.ir

1. Introduction

Sensor networks are large-scale distributed embedded systems composed of a large number of small-sized, low-cost, and low-power devices that integrate sensor, actuators, wireless communication and microprocessor. Sensor networks function as a key infrastructure for a broad range of applications including precision agriculture, intelligent highway system, emergent disaster recovery and surveillance[1,2]. Many sensor networks such as habit monitoring[3] and intruder tracking[4] need to handle physical entities that move in the environment. Only sensors which are close to an interesting physical entity should participate in the aggregations of data associated with the entity as activating sensors that are far away wastes precious energy without improving sensing fidelity. To continuously monitor a mobile entity, a sensor networks must maintain an active sensor group that moves at the same velocity as the entity. The combination of entity mobility and spatial locality, introduces unique spatiotemporal constraints on the communication protocols.

This paper focuses on mobicast, a new class of multicast with spatiotemporal semantics tailored for sensor networks[5,6]. Mobicast allows applications to specify their spatiotemporal constraints by requesting a mobile delivery zone, which in turn enables the application to build a continuously changing group configuration, according to their spatial and temporal locality. In this way, mobicast provides a powerful communication abstraction for supporting local coordination and data aggregation in sensor networks. For example, the service for maintaining a dynamic sensor group for tracking a mobile entity can be easily implemented on top of mobicast. When an interesting entity is discovered and a group is initiated, a group leader initiates a mobicast session to a delivery zone that moves according to the estimated velocity of the mobile entity. The mobicast message includes the location and time of the discovery of the entity. A node joins the group immediately upon reception of the message and leaves the group after the delivery zone moves away. Applications involving sensor and mobile networks require both spatial and temporal constraints to be satisfied simultaneously, that is, data need to be served at the right time and also at the right location.

Formally, a mobicast session is specified by a four tuple, (m, Z[t], Ts, T). m is the mobicast message. Z[t] is the delivery zone area where m should be disseminated at time t. As the delivery zone Z[t] evolves over time, the set of recipients of m changes as well. Ts and T are the sending time and duration of the mobicast session, respectively. A mobicast protocol should provide a spatiotemporal guarantee that all nodes that fall into a delivery zone within the lifetime of a mobicast session must receive the message m before they enter the delivery zone Z[t].

Previous works on mobicast[5-9] have explored several different approaches. The first Huang's mobicast protocol[5] which in this paper we call it "Delivery-zone constrained" (DZC) protocol employs a hold-and-forward strategy, and only nodes on the path of the delivery zone will participate in message forwarding. In this protocol only nodes that find themselves in the delivery zone path will join the forwarding. This delivery-zone constrained forwarding keeps the forwarding overhead at a minimum. Yet, the protocol fails to deliver the mobicast message to delivery zone nodes that are not directly connected to the source through a path fully contained in the area which the delivery zone covers over time. The main Huang's mobicast protocol presented in[5,6] handles random network topologies by limiting message re-broadcasting to a mobile forwarding zone, whose size depends on the compactness of the underlying geometric network. An absolute spatiotemporal guarantee can be achieved by configuring the forwarding zone based on the global minimum compactness value which captures the notion of the worst case "hole" that might appear anywhere in the network. However, this protocol has two drawbacks due to its dependence on global knowledge about the network-wide minimum compactness. First, it cannot scale well to large and dynamic networks where the network compactness can change over time. Second, it can introduce high overhead because the forwarding zone is often unnecessarily large due to the pessimistic configuration based on minimum compactness. Huang *et al* in[6] explored two other approaches to address the above problems. To solve the first problem, a simple adaptive protocol was designed to dynamically change the size of the forwarding zone based on the local compactness of a node's (multi-hop) neighborhood. To address the second problem, Huang found the broadcasting overhead can be reduced significantly by slightly relaxing the delivery guarantees. However, the latter two approaches do not provide guarantees on the spatiotemporal delivery of mobicast. More recently, Huang *et al.*[7] proposed a reliable mobicast protocol via face-aware routing (FAR). Face-aware, exploits ideas adapted from existing applications of face routing to achieve reliable mobicast delivery. The face-aware approach is originated by Bose *et al.*[10]. They consider the routing problems in ad hoc wireless networks modeled as *unit graphs* (faces) in which nodes are points in the plane and two nodes can communicate if the distance between them is less than some fixed unit. Bose and Morin in[11] proposed an algorithm for enumerating all the faces, edges, and vertices of a connected embedded planar graph *G* without the use of mark bits or a stack. Chen *et al*. in[8] proposed a variant-egg (VE)-based mobicast routing protocol in sensor networks. The VE-mobicast protocol can adaptively and efficiently determine the location and shape of the message forwarding zone in order to maintain the same number of waken-up sensor nodes. VE-mobicast by using the moving speed and the direction of the movement has been able to improve the prediction accuracy of the forwarding zone. However, the message delivery method of VE-mobicast is node oriented[9]. This method is not sufficiently efficient in terms of energy consumption. In existing protocols, when the prediction of the path of a forwarding zone is inaccurate, the nodes that were woken up earlier in the forwarding zone also waste much energy. More recently Chen *et al.*[9] proposed a cluster based approach called HVE-mobicast routing protocol. This protocol comparing to existing protocol is a more power-efficient mobicast routing protocol. This is mainly achieved by improving the guarantee percent, especially by considering different moving speeds and directions. However, this protocol is a centralized method that needs the cluster-head node.

In this paper a learning automata based mobicast protocol for sensor networks to support applications which require spatiotemporal coordination has been proposed. The proposed protocol which we call it LA-Mobicast uses the shape and the size of the forwarding zone to achieve high predicted accuracy. LA-Mobicast use learning automata to adaptively determine the location and the shape of the forwarding zone in such a way that same number of wake-up sensor_nodes to be maintained. The proposed protocol distinguishes itself from previous mobicast protocol by providing a distributed and adaptive method to calculate the forwarding zone using learning automata. The proposed protocol unlike VE-mobicast and HVE-mobicast protocols is not a location aware protocol. In order to study the performance of the proposed protocol, computer simulations have been conducted. Simulation results have shown performance enhancements in slack time, message exchange, nodes involved and guarantee percent, compared to existing mobicast protocols. The remainder of the paper is organized as follows. In section 2 mobicast and in section 3 the subject of learning automata is briefly reviewed. The proposed protocol is presented in section 4. Section 5 gives the simulation results and section 6 concludes the paper.

2. Mobicast

While mobicast is an interesting and useful abstraction for information dissemination in sensor network applications, implementation challenges are significant, especially when one desires high delivery guarantees. Providing spatiotemporal guarantees in mobicast introduces several key technical challenges. Since many sensor networks need to be deployed in an ad hoc fashion (*i.e.*, dispersed from an airplane or vehicles), a mobicast protocol must achieve reliable and timely delivery to a dynamic set of nodes over random network topologies where routing voids are prevalent. Fig.1 illustrates an example in which the delivery zone is expected to move across a hole on its path. Two nodes which are close in physical space can be relatively far away in logical network space (in terms of network hops). One can see there are many holes of varying sizes. The potential existence of holes in the network poses a challenge for mobicast. A mobicast session might be stopped prematurely because of a hole too big on its path.

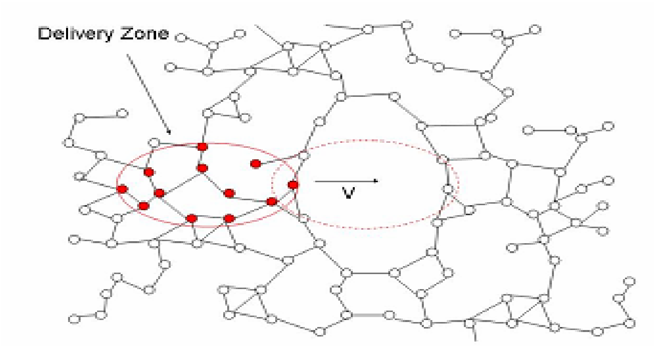

Figure 1: A random sensor network

From the drawbacks of this protocol we can see that in order to guarantee mobicast delivery for all delivery zone nodes, some nodes that are not in the delivery zone have to participate in message forwarding that we call them “forwarding zone nodes”, which move at some distance (headway distance) ahead of the delivery zone. Initial mobicast protocol assumes that the delivery zone moves at a fixed velocity, nodes are fixed, and communication has bounded one-hop latency during a mobicast session. The forwarding zone guarantees that all nodes entering the delivery zone as long as the network is not partitioned will receive the message in advance. The forwarding zone also serves to limit the retransmission to a bounded space and minimize energy consumption. The nodes which are in a forwarding zone retransmit the mobicast message, immediately after receiving it. While other nodes which are not in forwarding zone until becoming a member of the forwarding zone, do not retransmit the message. This hold-and-forward behavior by the nodes that receive the message early ensures the “just-in-time” feature of the mobicast propagation policy. An important question that all researches about mobicast try to answer is how to determine the shape and the size of the forwarding zone.

3. 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.

Environment can be defined by the triple $E = \{\alpha, \beta, c\}$ where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ represents a finite input set, $\beta = \{\beta_1, \beta_2, ..., \beta_r\}$ represents the output set, and $c = \{c_1, c_2, ..., c_r\}$ is a set of penalty probabilities, where each element c_i of c corresponds to one input action α_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.

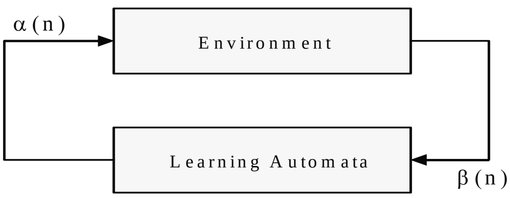

Figure 2. Relationship between learning automata and its environment

A variable-structure automaton is defined by the quadruple $\{\alpha, \beta, p, T\}$ in which $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ represents the action set of the automata, $\beta = \{\beta_1, \beta_2, \cdots, \beta_r\}$ represents the input set, $p = \{p_1, p_2, \cdots, p_r\}$ 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 α_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.

$$p_i(n+1) = p_i(n) + a.(1 - p_i(n)) \qquad (1)$$
$$p_j(n+1) = p_j(n) - a.p_j(n) \qquad \forall j \quad j \neq i$$
$$p_i(n+1) = (1-b).p_i(n) \qquad (2)$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \qquad \forall j \quad j \neq i$$

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 $L_{R\varepsilon P}$ [2], and for *b* = 0, it is called L_{R-I} [3]. For more information about learning automat the reader may refer to [12].

4. The Proposed Protocol (LA-Mobicast)

In this section we propose a learning automata based mobicast protocol for sensor networks to support applications which require spatiotemporal coordination. The proposed protocol use learning automata to adaptively determine the location and the shape of the forwarding zone in such away that the same number of wake-up sensor nodes be maintained. The proposed protocol is a fully distributed algorithm which requires lesser communication overhead in determining the forwarding zone and the mobicast message forwarding overhead. A set of learning automata are used to adaptively increase the size of the forwarding zone in order to achieve higher performance in terms of slack time, message exchange, node involved and guarantee percent.

Before describing the proposed protocol, we give some definitions. Immediate neighbors of a node are the nodes which are directly connected to that node and the number of such nodes is called the degree of the node. Forwarding nodes of a node is a set of immediate neighbors of a node which is placed in front of the moving delivery zone. The number of such nodes is called forwarding node's degree. For example in figure 3, node A has six immediate neighbors B, C, D, E, F and P and if the direction of the delivery zone is to the right then the forwarding nodes are B, C and D.

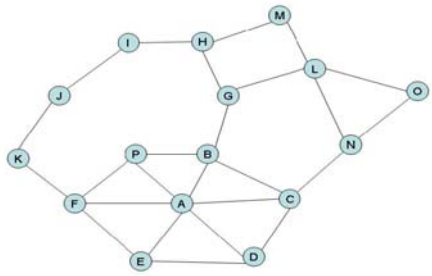

Figure 3: An example graph

In the proposed mobicast protocol each sensor in the network is equipped with a learning automaton. The task of each learning automaton is to determine the size of forwarding zone for forwarding the message. Each learning automaton has r actions $\{\alpha_1, \alpha_2, ..., \alpha_r\}$. Each action is a positive or negative integer value used by a learning automaton to change the radius of the forwarding zone in order to expand or shrink the forwarding zone. The elements of the action probability vector of the node that initiates the mobicast are equal and set to 1/r.

In the proposed protocol the learning automaton of the node that initiates the mobicast, selects one of its actions using its action probability vector. The selected action which is a positive or a negative integer value is then added to the radius of the forwarding zone to obtain the new radius for forwarding zone and the forwarding node's degree. A header for the mobicast packet is formed and then broadcasted locally by the node. The header for the mobicast packet contains forwarding node's degree, the action selected by the automaton, actions probability vector for the node's automaton, delivery zone radius, delivery zone velocity and direction (x, y components) and sender location (x, y coordinates).

When a node in the forwarding zone receives the packet, it compares its forwarding node's degree with the forwarding node's degree in the header of the received packet. If its forwarding node's degree is greater than the forwarding node's degree in the header then one of the following is performed.

- If the value of the selected action in the header is negative then this action is rewarded and the action vector probability in the header is updated by the learning automaton residing in the node according to the L_{R-P} learning algorithm. (The action selected by the sender node is an

[1] Linear Reward-Penalty
[2] Linear Reward epsilon Penalty
[3] Linear Reward Inaction

appropriate action and therefore must be rewarded because the sender node had shrunken the forwarding zone and the current node is being encountered a more compact area.)

- If the value of selected action in the header positive then this action is penalized and the action vector probability in the header is updated by the learning automaton residing in the node according to the L_{R-P} learning algorithm. (The action selected by the sender node is not an appropriate action and therefore must be penalized because the sender node had expanded the forwarding zone and the current node is being encountered a more compact area.)

After updating the action vector probability, the learning automaton in the node chooses one of its action using updated action vector probability and like before (like the initiator node) the header of the packet is formed and then forwarded to next nodes. This process is continued until the mobicast session is finished.

The LA-mobicast protocol is described in more details in Figure 4. The pseudo code shows the operation which is performed by a node upon hearing a mobicast message $\tilde{m}$ (if **t <** t_0 **+T)**.

if ($\tilde{m}$ **)** is new and t < t_0 +T **then**
    if (I am in F[t]) **then**
        execute the learning program;
        broadcast $\tilde{m}$ immediately; *// fast forward*
        if (I am in Z[t]) **then**
            deliver the message data D to the application layer;
        else
            compute the earliest time t_{in} for me to enter the delivery zone;
            if t_{in} exists and t_{in} < t_0 +T **then**
                schedule delivery of data D to application layer at t_{in};
            end if
        end if
    else
        compute the earliest time t' for me to enter the forwarding zone;
        if t' exists **then**
            if $t_0 \le t' \le t$ **then**
                execute the learning program;
                broadcast $\tilde{m}$ immediately; *// catch-up*
            else if $t < t' < t_0 + T$ **then**
                execute the learning program;
                schedule a broadcast of $\tilde{m}$ at t'; *//hold and forward*
            end if
        end if
    end if
end if

The learning program is given below.

Learning program:

(*NIN*: The number of neighborhood nodes in front of a node**)**

(*NIM*: The number of in front of neighborhood in the $\tilde{m}$ **)**

compute *NIN* and *NIM*
if *NIN* > *NIM* **then** *// more compact*
    if the selected action by the sender node is a negative number **then** *//shrinking the forwarding zone*
        reward the action;
    else
    if the selected action by the sender node is a positive number or zero **then**
        penalize the action;
    end if
end if
if *NIN* < *NIM* **then** *// less compact*
    if the selected action by the sender node is a positive number **then** *//expanding the forwarding zone*
        reward the action;
    else
    if the selected action by the sender node is a negative number or zero **then**
        penalize the action;
    end if
end if
if *NIN* = *NIM* **then** *// equal compact*

```
    if  the selected  action by the sender node is zero then  // do not the forwarding zone
            reward the action;
    else
    if the selected  action by the sender node is a positive number or negative number then
            penalize the action;
    end if
end if
```

Figure 4: Learning Automata based Mobicast Protocol

5. Simulation Results

In order to study the performance of the proposed protocol, Five existing protocols, DZC Mobicast[5], Huang's Mobicast[5,6], FAR[7], VE-Mobicast[8], HVE-Mobicast[9] and the proposed protocol are simulated using GloMoSim simulator[13] and the results obtained are compared in terms of four performance metrics defined below.

- **Slack time**: measures how early the message is delivered to a node with respect to its requisite deadline (to be at the specific node).
- **Message exchange:** the total number of messages that every sensor node transmits.
- **Node involved:** the number of waken-up nodes in forwarding zone.
- **Guarantee percent:** Guarantee percent is the percentage of those nodes entering the delivery zone and have received the mobicast message in advance, even if some of them are not directly connected.

The simulations were carried out in 1000*400 areas, with 400 sensor nodes which were set up at random. The communication radius of the sensor node is 50meters. The delivery zone where the spatiotemporal application takes place is circular, the velocity is 20 Km/h, from left to right, and the radius of delivery zone is 50 meters. In order to minimize the dependency of the simulation results on the network configuration the experiments were run on ten different network configurations generated via uniformly distributing 400 sensor nodes on a 1000*400m area. Each result reported is the average taken over the results obtained for ten network configurations.

The results of simulations are presented in tables 1 through 4. The first three simulations whose results are given in table 1 through 3 are conducted in order to study the effect of the learning parameter *a* of learning automata on the performance of the proposed protocol. Simulation results when the action sets are {-5,-2,0,2,5}, {-4,-2,0,2,4} and {-3,0,3} for different values of learning parameter, are given in tables 1, 2 and 3, respectively. From the results reported in these tables we can say that for all action sets, LA-Mobicast protocol performs the same when $a=b=0.1$ with guarantee percent equals to 54.08. The reason for such a low guarantee percent is that LA-Mobicast can not traverse the large hole that exist in the network as shown in figure 1. The best result for LA-Mobicast which is given in table 1 is obtained for action set {-5,-2,0,2,5} and $a=b= 0.5$.

In the forth simulation, the best result obtained for the proposed method (LA-Mobicast with action set {5, 2, 0,-2,-5} and a=0.5) is compared with the results obtained for five existing mobicast protocols, ZC Mobicast, Main Huang's Mobicast, FAR Mobicast, VE-Mobicast, and HVE-Mobicast. The result of comparison show that the proposed protocol outperforms the existing mobicast protocols in terms of total slack time, total message exchange, total node involved, average slack time, number of node in delivery zone and guarantee percent.

Table 1: Comparison of LA-Mobicast with action set {5, 2, 0,-2,-5} and different values of *a*

Reward and penalty parameter (*a*=*b*) / Performance Metrics	**0.1**	**0.2**	**0.3**	**0.5**	**0.7**	**0.99**
Total slack time	57.32	136.149	170.613	130.397	144.059	158.30
Total message exchange	357	770	934	688	849	878
Total node involved	90	170	195	166	182	188
Average slack time	1.08	1.39	1.74	1.33	1.47	1.62
Guarantee percent	54.08	100	100	100	100	100

Table 2: Comparison of LA-Mobicast with action set {4, 2, 0,-2,-4} and different values of *a*

Reward and penalty parameter (*a*=*b*) / Performance Metrics	0.1	0.2	0.3	0.5	0.7	0.99
Total slack time	60.12	146.143	188.415	138.192	151.217	161.7
Total message exchange	371	810	981	702	876	898
Total node involved	98	170	195	166	182	188
Average slack time	1.13	1.39	1.74	1.36	1.47	1.62
Guarantee percent	54.08	100	100	100	100	100

Table 3: Comparison of LA-Mobicast with action set {3,0,-3} and different values of *a*

Reward and penalty parameter (*a*=*b*) / Performance Metrics	0.1	0.2	0.3	0.5	0.7	0.99
Total slack time	61.6	101.143	198.415	141.192	150.113	153.71
Total message exchange	357	527	1003	709	863	870
Total node involved	90	116	232	171	185	190
Average slack time	1.16	1.40	2.02	1.44	1.54	1.56
Guarantee percent	54,08	100	100	100	100	100

Table4: Comparison of LA-Mobicast with action set {5, 2, 0,-2,-5} and *a*= 0.1 with existing protocols

Reward and penalty parameter (*a*=*b*) / Performance Metrics	LA-Mobicast	HVE-Mobicast	VE-Mobicast	FAR Mobicast	Main Haung's Mobicast	DZC-Mobicast
Total slack time	130.397	146.25	153.5	154.2	166.295	57.32
Total message exchange	688	721	781	746	884	172
Total node involved	166	176	181	184	197	55
Average slack time	1.33	1.49	1.56	1.57	1.7	1.04
Guarantee percent	100	100	100	100	100	54.08

Another experiment has been conducted in order to study the performance of the proposed protocol and the existing protocols in the presence of large hole in the network such as the network shown in figure 5. In the network of figure 5 there is a hole with width almost equal to the width of the network in the beginning of the network area. The results of this experiment are given in tables 5 and 6. As shown in table 5, DZC mobicast can cause premature termination of a mobicast session due to network hole. This method terminates at beginning of the first hole. The Huang's method introduces excessive flooding overhead in the network because the forwarding is unnecessarily large due to the pessimistic configuration based on minimum compactness. As shown in table 5, LA-Mobicast outperforms all existing protocols. Table 6 shows the performance of LA-Mobicast with action set {5, 2, 0,-2,-5} and different values of learning parameter *a* for the network of figure 5. As it is seen the best result is again obtained for *a*=0.5.

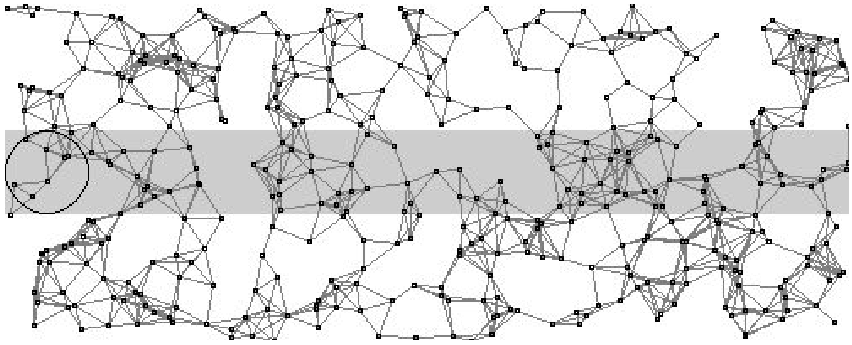

Figure 5: A network with large holes

Table5: Comparison of LA-Mobicast with action set {5, 2, 0, -2, -5} and *a*=*b*= 0.5 with existing protocol for the network figure 5

Reward and penalty parameter (*a*=*b*) / Performance Metrics	LA-Mobicast	HVE-Mobicast	VE-Mobicast	FAR Mobicast	Main Huang's Mobicast	DZC-Mobicast
Total slack time	139.23	150.2	160.1	167.2	173.6	11.2
Total message exchange	696	743	790	783	1736	32
Total node involved	173	181	191	231	388	6
Average slack time	1.42	1.53	1.63	1.7	1.77	2.2
Guarantee percent	100	100	100	100	100	6.1

Table 6: Comparison of LA-Mobicast with action set {5, 2, 0,-2,-5} and different values of *a* for the network figure 5

Reward and penalty parameter (*a*=*b*) / Performance Metrics	0.1	0.2	0.3	0.5	0.7	0.99
Total slack time	13.8	147.23	182.2	139.23	159.4	172.1
Total message exchange	43	770	959	696	901	892
Total node involved	10	184	197	173	192	212
Average slack time	2.3	1.5	1.85	1.42	1.62	1.75
Guarantee percent	6.1	100	100	100	100	100

6. Conclusion

In this paper, a mobicast protocol based on learning automata for sensor networks to support applications which require spatiotemporal coordination was proposed. The proposed protocol uses learning automata to adaptively determine the location and the shape of forwarding zone in such away that the same number of wake-up sensor nodes be maintained. The results obtained from simulations showed that the proposed protocol outperforms the existing mobicast protocols DZC Mobicast, Huang's Mobicast, FAR, VE-Mobicast and HVE-Mobicast in terms of slack time, message exchange, node involved and guarantee percent.

References

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, IEEE Communications Magazine, No. 40, pp.102–114 (**2002**).
2. D. Estrin et al., National Academy Press, Computer Science and Telecommunications Board (CSTB) Report (**2001**).
3. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao, ACM SIGCOMM

Workshop on Data Communications in Latin America and the Caribbean (**2001**).

4. D. Li, K. Wong, Y. Hu and A. Sayeed, IEEE Signal Processing Magazine, Vol. 19, No. 2 (**2002**).

5. Q. Hang, C. Lu, and G. C. Roman, Proceedings of the ACM Conference on Embedded Networked Sensor Systems (Sensys), pp. 205-217 (**2003**).

6. Q. Hang, C. Lu and G. C. Roman, Telecommunication Systems, Special Issue on Wireless Sensor Networks, Vol. 26, No. 2-4, pp.129-160 (**2004**).

7. Q. Hang, C.Lu, and G.-C. Roman, Proceeding of the IEEE International Conference on Computer Communications (INFOCM) (**2004**).

8. Yuh-Shyan Chen and Shin-Yi Ann, ACM Wireless Networks (SCI) (**2006**).

9. Yuh-Shyan Chen and Shin-Yi Ann, Proceedings of IEEE Wireless Communications and Networking Conference (IEEE WCNC 2006), Las Vegas, NV, USA, 3-6 (**2006**).

10. P. Bose, I. Morin, I. Stojmenovic and J. Urrutia, ACM Wireless Networks, Vol. 3, No. 4, pp.609–616 (**2001**).

11. P. Bose and P. Morin, International Journal of Foundations of Computer Science, pp. 297–308 (**2002**).

12. K. S. Narendra and M. A. L. Thathachar, Prentice Hall (**1989**).

13. http://www.pcs.ucla.edu/projects/glomosim/.