An Intelligent Protocol to Channel Assignment in Wireless Sensor Networks: Learning Automata Approach

Vahid Moghiss
Department of Computer Engineering
Islamic Azad University-Qazvin branch
Qazvin, Iran
Moghiss@gmail.com

Mohammad Reza Meybodi[1] and Mehdi Esnaashari[2]
Soft Computing Laboratory, Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran, Iran
{[1] mmeybodi, [2] esnaashari}@aut.ac.ir

***Abstract*— The optimal consumption of energy is one of the most important aims in Wireless sensor networks. Among influential factors in reducing the energy consumption ratio, we can mention intelligent allocation of channel to node and sending the data with the least rate of interference with other nodes. The successful execution of this act by MAC sub-layer depends on determining optimal values for factors such as percentage of duty cycle, random time to retest the channel an etc. Duty cycle, defined as the ratio of listening period to total period of listening and sleeping, is of great importance as the parameter of regulating node sleeping and listening time for ideal consumption of energy. There is presented an algorithm based on learning automat, called LA-MAC, which tries to reduce energy consumption by regulating optimally the duty cycle value, thereby increasing the lifetime of network. In this algorithm, each node is equipped with a learning automat which determines the value of duty cycle in terms of residual energy and preconfigured lifetime. The results of simulation in the environment of NS simulator confirm the efficiency of LA-MAC from four factors of the death time of the first node, end-to-end delay, average network lifetime and the number of survived nodes at the end of simulation, performed in two different scenarios while compared with A-MAC and S-MAC with various duty cycles.**

Keywords: wireless sensor networks; learning automat; duty cycle; channel assignment

I. INTRODUCTION

Wireless sensor network is a special kind of Ad-hoc networks consisted of small nodes which are able to sense the surrounding with the certain target, information processing, storing, information exchange with other nodes and compatibility against changes (topology and …). Usually, all the nodes are identical and satisfy the general aim of network by cooperating with each other. The main aim of WSN is to monitor and control the climatic change (physical or chemical) in a specified area [1]. In sensor networks, each node must be simple and cheap without abundant resources of processor, memory or battery. As the highest rate of energy consumption is formed in sending and receiving data, sensing the environment and processing in each node respectively, the time of sending and receiving data must be chosen so intelligently that the packet sender can catch the channel in the least time and its receiver is potentially ready to receive it.

The coordination among the nodes of a network to handle the channel is performed in the MAC sub-layer. The aims to be considered while we design a protocol for MAC layer in wireless sensor networks are the increase of throughput, increase of network lifetime and delay reduction. There appear some conflicts in this sub-layer responsible for energy waste in WSN such as collision, overhearing, overmitting, protocol overhead and idle listening [2].

When a node receives more than one packet simultaneously, there will appear collision so that the packets must be removed and the resending is requested. This wastes the receiving costs at destination node and sending costs at source node.

Overhearing is defined as receiving a packet whose destination is elsewhere. it is clear that this case exists in wireless media which are intrinsically broadcasting amongst the case which waste the node energy. Overmitting is formed when the receiver is not ready while the packets arrive. Protocol overheads such as RTS/CTS contracts or request packets in allocation of demand and in addition, overheads which are formed for each packet as header and tailer cause waste of node energy. Idle listening means that the node listens to an empty channel to receive probable traffic. The idle node is ready to receive the packet which brings some costs and can be useless if there is low load of network.

Regarding the limitation of battery available in each sensor node and lack of its interchangeability, attention to the above troubles and attempt to reduce their effects are the most important problems posed in designing protocols of MAC layer for WSN.

In this article, there is presented an intelligent algorithm of MAC layer based on learning automat named LA-MAC which is the generalization of S-MAC algorithm [3]. Duty cycle, defied as the ratio of listening period to the total period of listening and sleeping, is an effective factor in the ratio of energy consumption. In S-MAC algorithm, this parameter is regulated manually only once before the implementation in an interval from 1% to 100% while we cannot change it at all. In LA-MAC algorithm, a learning automat is used to regulate the value of duty cycle in each sensor node so that it learns the appropriate value gradually having received the feedback from the environment. Based on our definition, the appropriate value of duty cycle is the rate of listen/sleep through passing it we can attain the preconfigured lifetime for each node. The results of simulation show that LA-MAC has increased the death time of the first node, the average network lifetime and the number of nodes survived at the end of simulation and decreased the end-to-end delay, compared with A-MAC and S-MAC [4].

The rest of this paper has been organized as the following. In section II we discuss some of related works. In section III the problem is explained. Learning automata is described briefly in part A of section IV and protocol LA-MAC will be explained in part B of section IV. The simulation results are presented in section V. Finally conclusion is the output of section VI.

II. RELATED WORKS

So far, several algorithms have been introduced for wireless sensor networks MAC layer each of which has focused on one or another troubles having been posed in the previous section (collision, overhearing,...)[12,13]. Among the presented algorithms, S-MAC is one of the most renown and common algorithms have been introduced.

S-MAC is medium access control protocol based on competition designed for WSN[3]. This protocol operates in terms of recurrent listening in that each node performs listening and sleeping period with a fixed length according to a schedule. Low duty cycle is considered as default for all nodes. Low duty cycle reduces energy consumption but increases the delay. As S-MAC supposes a fixed duty cycle for all nodes, it can not adjust its operation in terms of residual energy rate of each node and traffic load in it. This reduces throughput of network.

DS-MAC [2] [5] has added the dynamic duty cycle feature to S-MAC protocol following the reduction of delay for specific applications. In this protocol, when the receiver node notices the high average value of immediate delay, it reduces its sleeping time and announces it in SYNC packet. Sensor nodes increase the duty cycle when the delay is great for each hop and decreases it when the traffic loads return to low levels. To estimate the present traffic loads in network each source calculates the delay in standing queue from receiving the packet to sending it and puts this additional field in SYNC. The observed delay and the average consumption of power for each packet are improved in this protocol compared with S-MAC.

A-MAC, an MAC protocol based on competition for wireless sensor networks guarantees the preconfigured lifetime and also reduces the delay in different times. Each sensor node regulates duty cycle in terms of traffic load and residual energy value so that the energy consumption in each node approaches its ideal thereby leading the node to survive the preconfigured life time. If the traffic load of the node decreases, we can reduce sleeping delay by increasing duty cycle. In A-MAC, the duty cycle is varied from 1% to 99%.

Most of proposed MAC protocols for WSN consider the nodes to be static after the expansion. In scenarios containing moving nodes, the performance of network decreases as there may be the possibility of rapid formation of new connections for moving nodes. MS-MAC [5-6] has developed S-MAC to support moving nodes. In this protocol, the severity of received signal from each neighbour is recorded to estimate the move ability in sensor node and announces any change indicating motion in SYNC packet.

Each node, based on level of received signal from periodic SYNC packets of neighbour, notices the motion in neighbour nodes. For more than one moving neighbour, SYNC packet contains the maximum estimated speed of neighbours.

III. PROBLEM STATEMENT

Consider a two dimensional sensor network which consists of N sensor nodes S_1 to S_N. All of the sensors have similar sensing range (Rs) and radio transmission range (R_t). The initial energy of all nodes is taken constant and equal to E_{init}. There is not considered any mobility for nodes in this network.

Definition 1. Preconfigured lifetime: it's the least time we expect the nodes to be survived in the network, dependent on our needs. In fact, it can be said that if all nodes survive during this period, we are in our ideal state.

Definition 2. Duty cycle: its value is obtained from relationship 1.

$$DutyCycle = \frac{ListenPeriod}{ListenPeriod + SleepPeriod} \quad (1)$$

One of the mechanisms to reduce energy consumption is to follow listen/sleep method in which the node practically turns off antenna and switches it in the storage state. The factor involved mainly in this operation is duty cycle. In some algorithms, this parameter is static and regulated once in the starting of network. It is evident that the operation of each node does not depend on the current situation of network such as decrease or increase of traffic load or the changes in its topology if the duty cycle is fixed.

Having listen/sleep approach, we aim to establish a dynamic duty cycle in proportion with the changes taken place in network so that we can create an appropriate duty cycle for each node in the time regarding the current situation of network.

IV. THE PROPOSED DYNAMIC CHANNEL ASSIGNMENT ALGORITHMS

A. 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 1 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 of action α_i. An environment in which β can take only binary values 0 or 1 is referred to as P-model environment. A further generalization of the environment allows finite output sets with more than two elements that take values in the interval [0, 1]. Such an environment is referred to as Q- model. Finally, when the output of the environment is a continuous random variable which assumes values in the interval [0, 1], it is referred to as an S-model. Learning automata are classified into fixed-structure stochastic, and variable-structure stochastic. In the following, we consider only variable- structure automata.

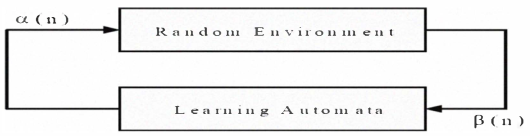

Figure 1. Relationship between learning automata and its environment

A variable-structure automaton is defined by the quadruple $\{\alpha,\beta,p,T\}$ in which $\alpha=\{\alpha_1,\alpha_2,...\alpha_r\}$ represents the action set of the automata, $\beta=\{\beta_1,\beta_2,..\beta_r\}$ represents the input set, $p=\{p_1,p_2,...p_r\}$ represent 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 equations (2) for favorable responses, and equations (3) for unfavorable ones.

$$\begin{aligned} &p_i(n+1) = p_i(n) + a.(1-p_i(n)) \\ &p_j(n+1) = p_j(n) - a.p_j(n) \qquad \forall_j \quad j\neq i \end{aligned} \tag{2}$$

$$\begin{aligned} &p_i(n+1) = (1-b).p_i(n) \\ &p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \qquad \forall_j \quad j\neq i \end{aligned} \tag{3}$$

In these two equations, a and b are reward and penalty parameters respectively. For a = b, learning algorithm is called L_{R-P}, for b<<a, it is called $L_{R\varepsilon P}$, and for b=0, it is called L_{R-I}. For more information the reader may refer to [7] [8]. The only application of learning automata to sensor networks has been reported in [9].

B. Explanation of LA-MAC algorithm

The common aim of researches about wireless sensor networks is to maximize the lifetime of network [13]. When the present energy in sensor nodes is exhausted, the nodes die out. MAC protocols must increase network performance through the reduction of energy waste. Therefore, the attempt to survive node for preconfigured life time is a good idea to make the energy consumption ideal. Figure 2 shows the ideal rate of energy consumption.

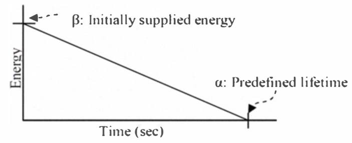

Figure 2. Example of an ideal energy consumption rate.

If the node spends its energy based on this consumption pattern, we can attain the preconfigured lifetime of network. Our proposed algorithm performs in terms of periodic listen/sleep schedule in S-MAC. Each sensor node regulates duty cycle itself dynamically based on the rate of energy consumption. In contrast, in S-MAC the duty cycle is fixed for each node and determined in node configuration time.

Figure 3 shows the scenario in which the data is sent from source node A to sink node E. The value of duty cycle for each node is specified alongside it. The node without data to send awakens in the listen period, meaning the node with a data to send, awakens in the listen periods of next hop`s node and sends the data for its neighbour in the next hop having sent RTS/CTS packets. Consequently, data is relayed to the sink node.

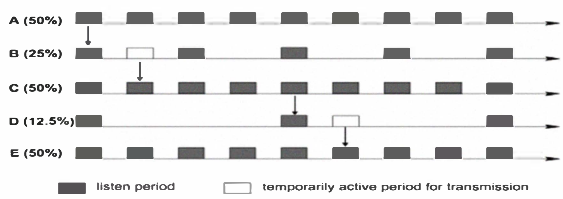

Figure 3. Example of data transmission.

In each node, when the neighbour with the lowest duty cycle awakens, the listen/sleep scheduling is broadcast in terms of SYNC packet. The continuous sending of SYNC packet leads to consequent collision, thereby reducing the network performance and resource waste. To prevent collision, each node broadcasts SYNC packet only once in N frames of the lowest duty cycle of its neighbours when two or more nodes send the SYNC packet simultaneously. The value of n is 10 in our experiments. Also, to reduce the problem of sudden sending of SYNC packets, each node chooses its sending time randomly among n frames.

Each node is equipped with a learning automata with three actions; the first is to increase duty cycle, the second to reduce it and the third is to fix the duty cycle. Initially, due to the same initial energy level of all nodes, the probability of fixing duty cycle is 0.5 and that of two other actions is 0.25.

As mentioned, LA-MAC uses the periodic listen/sleep mechanism. The listen period is divided into two sections. The first one SYNC in which the packets called SYNC are exchanged among the nodes to solve the problem of simultaneity among the neighbor nodes. The second section, called data, is considered for sending data packet. The frame format of LA-MAC is shown in Figure 4.

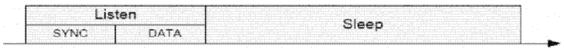

Figure 4. Frame format of LA-MAC

Similar to S-MAC before sending the data, the sender must check the channel and both parties involved must announce their preparation using RTS/CTS packets. If previous stages are successful, the sending starts at the beginning of DATA period and continues to the next sleep period. The listen period length of LA-MAC dependent on the success of its previous stages and sleep period length dependent on the regulation of duty cycle will differ.

1) *self-Organization*

In this section, we introduce the set of operations performed at the beginning of activity of a network or at the arrival of a new node to the network and its own introduction to other nodes.

Similar to S-MAC, in each node there is a table called schedule table in which the information related to listening and sleeping time of neighbour nodes exists. When a node enters networks, it listens to channel for a fixed time (equal to SYNC period) called initial listening. There can be posed two cases.

1. No SYNC packet has been received in this period. In this case the node chooses a listen/sleep period randomly in terms of its id number, enters it in its schedule table and broadcasts it in the next SYNC period in terms of as SYNC packet for its neighbours.

2. SYNC packet has been received. In this case, the node regulates its schedule based on the first received SYNC, add this to schedule table and finally broadcasts the packet in the next SYNC period as a SYNC packet for its neighbours.

In each SYNC period, each node must broadcast its schedule in terms of a SYNC packet for all of its neighbours. A SYNC packet contains three fields: sender's address, the next listen time and listen/sleep schedule (duty cycle).

After the end of self organization procedure, the network starts to perform its activities to regulate the duty cycle.

2) *Network Operation & Duty Cycle Adaptation*

In the working phase of a network, learning automata of each node regulates its duty cycle dynamically in terms of residual energy and remaining time to the estimated lifetime as following.

In each cycle of information gathering, node learning automata chooses one of its actions randomly based on probability rector of actions. Based on chosen action, duty cycle of the node considered changes according to relationship 4.

$$\textit{New Duty Cycle} = \textit{Old Duty Cycle} \pm 0.6 \qquad (4)$$

The value of 0.6 was obtained optimally in simulation and different tests. In each stage of information gathering, the duty cycle is determined for the next stage. At the end of each stage of information gathering each node calculates the value of based on relationship 5.

$$\partial = \frac{T_{elap}}{T_{pred}} - \frac{E_{cons}}{E_{init}} \qquad (5)$$

In this relationship, T_{elap} is the elapsed time from the onset of network activity, T_{pred} is the preconfigured lifetime for network, E_{cons} consumed energy to the due time (E_{init}-E_{curr}) and E_{init} is the initial energy of node. If ∂ is bigger than zero and more than a upper threshold, the residual energy in node is more than required energy of preconfigured lifetime. Therefore, the learning automata reward its own action according to relationship1.

If ∂ is less than zero and lower threshold, we can not reach the preconfigured lifetime with this procedure. Therefore, learning automata penalties its chosen action according to relationship 2. Consequently, learning automata of each node learns the optimal value of duty cycle over the time.

V. SIMULATION RESULTS

To evaluate the performance of the presented algorithm, we used NS2 simulator [10]. The simulation was performed on a 16 node topology with 4*4 grid of features in Table I. To calculate the energy consumption, we used the energy model in NS2.

The death time of the first node, end-to-end delay, average Network lifetime, and the number of nodes survived at the end of preconfigured lifetime are considered as efficiency parameters. The average network lifetime (ALTN) is obtained from relationship 6.

TABLE I. SIMULATION PARAMETERS

Simulation parameters	Values	Simulation parameters	Values
Radio Transmission Range	30m	Distance Between Nodes	20 m
Radio Interference Range	60m	Difference Upper bound	0.1
Radio Bandwidth	20Kbps	Difference Lowe bound	0
Data Packet Size	50 byte	Transmit Power	0.660W
Control Packet Size	9 byte	Receive Power	0.395W
Duration of Listen Interval	112ms	Idle Power	0.350W
Contention Windows for SYNC	15 slot	Sleep Power	0.001W
Contention Windows for Data	31 slot	Initial Energy	300J
Minimum Duty Cycle	10%	Preconfigured Lifetime	4000s
Maximum Duty Cycle	90%	Duty Cycle Inc/ Dec	0.6

$$ALTN = \frac{\sum_{i=1}^{N-m} t_i + (m \times T)}{N} \qquad (6)$$

In which we have:

t_i: the death time of ith node

N: total number of network nodes

m: the number of survived nodes at the end of simulation

T: preconfigured lifetime of network

We simulated both single-flow with one sender node and sink node and multiple-flow with several senders and only one sink node, in different intervals of sending packets and compared the proposed algorithm with S-MAC and A-MAC algorithms in different duty cycles.

A. *Single-flow Network*

In Single-flow network, there is a sender and a sink node. Figure 5 shows the death time of the first node with time variation between sending packets from 2 to 12 seconds. According to the figure, when the distance between the sending increases, the death time of the first node is delayed. The proposed algorithm has longer death time compared with S-MAC and A-MAC so that in sending interval of 12 second for simulation time of 3500 seconds, no node is dead.

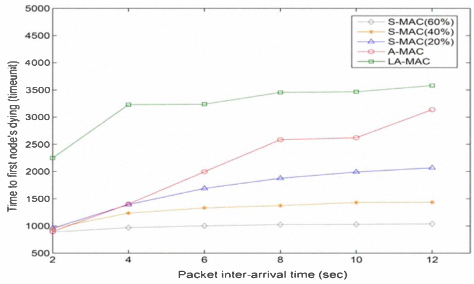

Figure 5. Time to first node dying versus changing packet inter-arrival time (Single-flow)

Figure 6 shows the average network lifetime. As is shown in figure, network lifetime in our algorithm is greater than 97% meaning to attain its aim of increasing network lifetime.

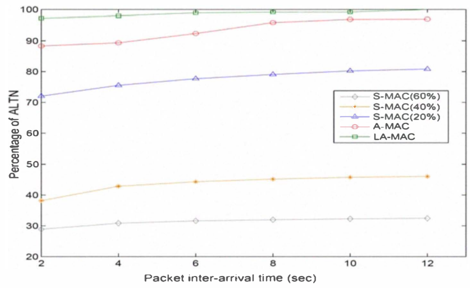

Figure 6. ALTN versus packet inter-arrival time (Single-flow)

Figure 7 shows the number of survived nodes at the end of preconfigured network lifetime. In S-MAC algorithm with all duty cycles, no node was alive at the end of preconfigured lifetime. The effect of long listening of S-MAC on reduction of node lifetime and of network lifetime can be seen. If every 10 seconds, the sender sends a packet to sink, A-MAC has five alive nodes at the end of preconfigured life time while the proposed algorithm with the same traffic pattern has 14 alive nodes out of 16.

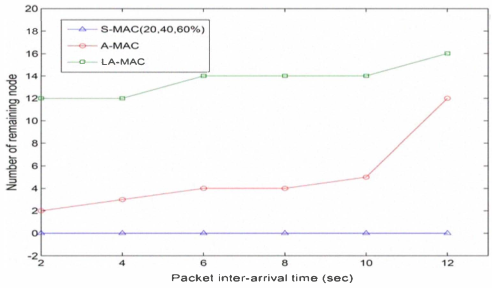

Figure 7. Number of remaining nodes at the end of preconfigured lifetime (Single-flow)

Figure 8 shows the end-to-end delay among the above algorithms. As seen, algorithm S-MAC has much less delay due to the fact that nodes are listen most of the time. This reduction leads to the increase of energy consumption cost and reduction of node lifetime. In our algorithm, the rate of end-to-end delay is less than that in A-MAC algorithm having listen/sleep schedule for nodes.

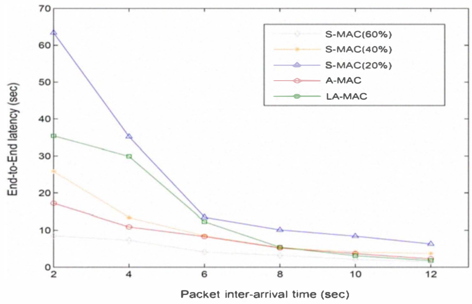

Figure 8. End-to-End delay versus packet inter-arrival time (Single-flow)

B. *Multiple-flow Network*

In multiple-flow network, sink node is node 15 and other nodes (0-14) are senders that randomly send packet to sink. Figure 9 shows the death time of the first node with change in the time of sending packets. Similar to single-flow node and in different duty cycle proposed algorithm increases the survival of the nodes compared with S-MAC and A-MAC.

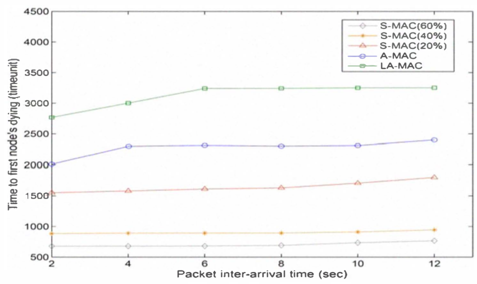

Figure 9. Time to first node dying versus changing packet inter-arrival time (Mutiple-flows)

The average network lifetime is shown in Figure 10. Similar to single-flow case, yet with relative decrease in all algorithms, our proposed algorithm has remarkable increase of average network lifetime in the new situation compared with S-MAC while it has had a better performance against A-MAC.

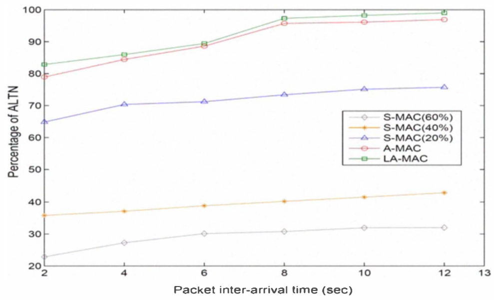

Figure 10. ALTN versus packet inter-arrival time (Multiple-flows)

Figure 11 shows the number of survived nodes at the end of simulation time considered to be 2500. We considered the simulation time less than previous case to show the difference of proposed algorithm with other algorithms. If we considered the time as 3000, no node would be alive in A-MAC.

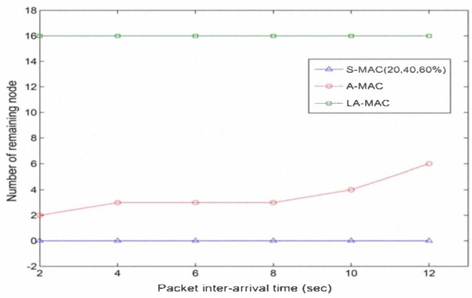

Figure 11. Number of remaining nodes at the end of preconfigured lifetime (2500 sec, Multiple-flows)

In our proposed algorithm, all nodes are alive in all sending intervals. The rate of end-to-end delay is shown in Figure 12. According to figure, in terms of end-to-end delay in sing-flow condition, the performance of proposed algorithm has significant growth so that in the pattern of a 2 second interval of sending, our delay is less than S-MAC. The distance between the delay of proposed algorithm is 3.5 times the same as that of A-MAC algorithm.

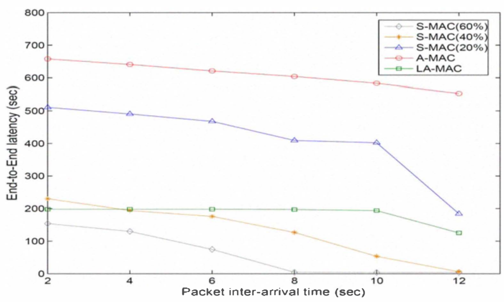

Figure 12. End-to-End delay versus packet inter-arrival time (Multiple-flows)

VI. CONCLUSION

In this paper, we presented a new method based on learning automata to assign a channel in wireless sensor networks. We tried to regulate the rate of duly cycles to improve listen/sleep period of each node by equipping each node with a learning automata which has three different actions (increase, decrease and fix of duty cycle) with different probabilities. The results showed that our proposed algorithm has a more reflexible performance in terms of different traffic loads, due to learning from environment, so that, in parameters of death time of the first node, the number of survived nodes at the end of preconfigured lifetime and the average network lifetime have been in better situation compared to those of A-MAC and S-MAC algorithms in different duty cycles.

REFERENCES

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks", in: IEEE Communications Magazine, August 2002.

[2] Ilker Demirkol, Cem Ersoy, and Fatih Alagöz, Bogazici University, "MAC Protocols for Wireless Sensor Networks: A Survey", in: IEEE Communications Magazine, April 2006.

[3] Wei Ye, Member, IEEE, John Heidemann, Member, IEEE, and Deborah Estrin, Fellow, IEEE, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", in: IEEE/ACM transactions on networking, Vol.12, No.3, June 2004.

[4] Yongsub Nam, Taekyoung Kwon, Hojin Lee, Hakyung Jung, Yanghee Choi, "Guaranteeing the network lifetime in wireless sensor networks: A MAC layer approach", in: Computer Communications 30, June 2007.

[5] Kurtis Kredo, Prasant Mohapatra," Medium Access Control in Wireless Sensor Networks", in: Computer Networks, June 2006.

[6] H. Pham, S. Jha, "An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC)", in: Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), pp. 214–226, 2004.

[7] M. A. L. Thathachar, P. S. Sastry, "Varieties of learning automata: An overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.

[8] K. S. Narendra, M. A. L. Thathachar, "Learning automata: An introduction", In Proceedings of the Prentice Hall, 1989.

[9] M. Golipour and M. R. Meybodi, "LA-Mobicast: A Learning Automata based Mobicast Routing Protocol for Wireless Sensor Networks", Sensor Letters, Vol. 6, No.2, pp. 305-311, April 2008.

[10] ''Network Simulator'' [Online] http://www.isi.edu/nsnam/ns.

[11] Holger Karl and Andreas Willig," Protocols and Architectures for Wireless Sensor Networks" John Wiley & Sons, Ltd, 2005.

[12] Sunil Kumar,Vineet S. Raghavan,Jing Deng, "Medium Access Control protocols for ad hoc wireless networks: A survey", in: Ad Hoc Networks 4, November 2004.

[13] Kaushik R. Chowdhury, Nagesh Nandiraju, Pritam Chanda, Dharma P. Agrawal, Qing-An Zeng, "Channel allocation and medium access control for wireless sensor networks", in: Elsevier Ad Hoc Networks, March 2009.