Equalization of Energy Consumption in Ad Hoc Networks using Learning Automata

Faezeh Akbari†, Seyed Majid Noorhosseini††, Mahnaz Imanazari Bonab†††
faezehakbari@gmail.com majidnh@aut.ac.ir Mahnaz62@gmail.com
†, †††Computer Engineering and Information Technology Department, Islamic Azad University, Qazvin, Iran
††Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

Summary

One of the main constraints in mobile ad hoc networks is high probability of failure due to energy-exhausted nodes. So, if some nodes die early due to lack of energy, they cannot communicate with each other. Therefore, inordinate consumption of nodes' energy should be prevented. In fact, nodes energy consumption should be balanced in order to increase the network lifetime. In this paper we balance the energy consumed by ad hoc network nodes by clustering the nodes based on their remaining energies. Clusters form by learning automata dynamically and change based on lifetime prediction of clusters member as well as the number of their neighbors. The simulation results show that our proposed method outperforms MARI and flat topology both in prolonging network lifetime and in balancing energy consumption of nodes.

Key words:
Ad hoc Network, Energy consumption, QOS, Learning Automata, Maximizing Network Lifetime

1. Introduction

Mobile Ad hoc Networks (MANETs) consist of a collection of wireless mobile hosts (called nodes), recently have received increasing attention. Independence from central network administration, ability for being self-configured, self-healing through continuous reconfiguration, scalability and flexibility are the distinguished reasons to deploy such networks [1]. MANETs require no fixed infrastructure or central administration. Mobile nodes in an ad hoc network work not only as hosts but also as routers, and communicate with each other via packet radios.

Since most wireless nodes in ad hoc networks are not connected to a power supply and battery replacement may be difficult, optimizing the energy consumption in these networks has a high priority and power management is one of the most challenging problems in ad hoc networking.

Energy consumption in ad hoc node can be due to either useful or wasteful source. Useful energy consumption can be due to (1) transmitting /receiving data, (2) processing query requests, and (3) forwarding queries/data to neighboring nodes. Wasteful energy consumption can be due to (1) idle listening to the media, (2) retransmitting due to packet collision, (3) overhearing, and (4) generating/handling control packets. In general, radios in an ad hoc network node can operate in four distinct modes of operation: transmit, receive, idle, and sleep [2]. Transmit and receive modes are for transmitting and receiving data. In the idle mode, the radio can switch to transmit or receive mode. Idle is the default mode for an adhoc environment. The sleep mode has extremely low power consumption. Therefore, taking advantage of the sleep mode is very important in energy efficient protocols.

As noted above, energy conserving is important and necessary. Therefore it is imperative that at any moment some specific number of nodes be active and the rest remain in inactive mode. We keep number of active nodes in desirable way, so network lifetime will be prolonging by far. If active nodes can cover desirable level of network, less number of active nodes will be required in total network and will not be the empty space of active node. We balance energy consumption of nodes by means of replacing cluster heads and forwarding data. Therefore the number of active nodes and also network coverage by them can be main factors in improving QOS.

Several MAC protocols attempt to reduce energy consumption. Many protocols [3-6] have controlled the network topology by determining which nodes should participate in the network operation (be awake) and which should not (remain asleep).

In AFECA [3], nodes are given a sleep interval that is related to the number of its neighbors. However, each AFECA node is awake for a fraction of time which is roughly 2/(2+N) (assuming each node has an accurate measurement of its neighborhood size N). Since AFECA assuming a store-and-forward routing mechanism, messages must wait for nodes to wake before making further progress. An AFECA node does not know whether it is required to listen in order to maintain connectivity, so to be conservative AFECA tends to make nodes listen even when they could be asleep. Our proposed method never keeps a node awake unless it is essential for connecting two neighbor clusters.

Span [4] constructs a backbone network structure to establish all node communications. The node, owning the backbone network creation is called a coordinator. The coordinator is always active. Other nodes can sleep when they are not needed. In addition, the sleeping nodes periodically check if they could become the coordinator in the subsequent time period. This coordinator changing mechanism balances the power consumption of each node, and extends the network lifetime. A non-coordinator node announces to be a coordinator if it discovers that two of its neighbors cannot reach each other either directly or via one or two coordinators. It may be a node with critical energy. While in our proposed method cluster heads make decision which node is forwarding node. Since they know complete information of all member nodes, they select a proper node.

Subramanyam [6] presents a schema such that each node in the network makes local decisions on whether to sleep or stay awake and as a MARI node participate in the forwarding backbone topology. The MARI nodes have the maximum power level among their one hop neighbors and all non-MARI nodes are within the transmission range of MARI nodes. The gateway nodes are selected which are having power so that they can forward packets between MARI nodes. Both MARI nodes and gateway nodes are continuously awake to route the packets to the other member nodes within the transmission range. All members of nodes will go to sleep mode, if they do not have to transmit or receive data. The wake up time for each node is calculated from a pseudo-random manner, such that the MARI node and neighbor nodes know the waken time each other.

In this paper, we present new method, which are based on learning automata for supporting QOS according to residual node energy and its coverage measurement. This schema of topology management runs above the MAC layer and interacts with the routing protocol. In this method in order to access to desirable coverage, first we use network clustering. The main task of the cluster head is forwarding and controlling active nodes in its own cluster. Other nodes of this cluster gain active or inactive state according to received feedbacks. If cluster head may not communicate with neighbor cluster head, it will select one of its member nodes as forwarding node. This selection would be according to above factors and also location of this node in forwarding path. We will show that this method improves QOS desirably.

The remainder of the paper is organized as follows. In Section 2 the subject of learning automata is briefly reviewed and section 3 is the proposed method. Section 4 gives the simulation results and Section 5 concludes the paper.

2. Learning Automata

Learning automata [11] is an abstract model which randomly selects one action out of its finite set of actions and performs it on a random environment. Then environment 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.

The environment can be defined by the triple E={α,β,c} where $\alpha=\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$ represents a finite input set, $\beta=\{\beta_1,\beta_2,\ldots,\beta_r\}$ represents the output set, and $c = \{c_1,c_2,\ldots,c_r\}$ 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. A variable-structure automaton is defined by the quadruple {α,β,p,T} in which $\alpha=\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$ represents the action set of the automata, $\beta=\{\beta_1,\beta_2,\ldots,\beta_r\}$represents the input set, $p=\{p_1,p_2,\ldots,p_r\}$ represents the action probability set, and finally p(n + 1) = T[α(n),β(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, the automaton updates its action probability set based on Eq. (1) for favorable responses, and Eq. (2) for unfavorable ones.

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

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

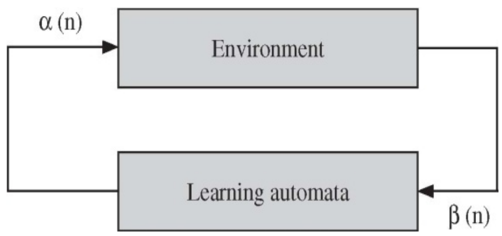

Figure 1.Relationship between learning automata and its environment.

In these two equations, a and b are reward and penalty parameters respectively. For $a = b$, the learning algorithm is called L_{R-P} , for $a < b$, it is called $L_{R\varepsilon P}$, and for $b = 0$, it is called L_{R-I} . For more information about learning automat the reader may refer to [7].

3. Proposed Method

In order to save more energy, it is necessary that specific number of nodes be active and other nodes be inactive state. We keep number of active nodes desirable level. So network lifetime will be prolonging by far. If active nodes can cover desirable level of network, less number of active nodes will be required in total network and the empty space of active node will not be in the network. Before describing the proposed protocol, we give some definitions. The 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.

In this algorithm, first we need to do network clustering. There are several proposed algorithms [8, 9] for clustering. We use HEED clustering algorithm that present in [10]. First, we briefly review HEED protocol. HEED protocol follows several goals: prolonged network lifetime, terminated clustering phase after the number of finite and specific iteration, minimizing control overhead and the distribution of uniform cluster head across the network. Cluster head selection is primarily based on the residual energy of each node. Also this selection can be a function of neighbor proximity or cluster density. HEED uses the primary parameter to probabilistically select an initial set of cluster heads, and the secondary parameter is used when node falls within the range of more than one cluster head, which including the situation when two tentative cluster heads fall within the same range.

If the neighbor has been as a member of other cluster that residual energy of its cluster head is lower than residual energy of new cluster head, the neighbor joints to new cluster. In addition, if the residual energy level of cluster head be lower than residual energy level of introduced cluster, so it is member of new cluster head.

If a node becomes a cluster head, it sends an announcement message to its own cluster members. If a node completes clustering execution without selecting a cluster head, it announces itself to be a cluster head. A node can be a cluster head at consecutive clustering intervals if it has high residual energy.

In end phase of clustering, network will be dividing to the number of clusters. Each node is a member of a cluster and each cluster has a cluster head. Each cluster head is equipped with a learning automaton. Each learning automaton has two actions (active, inactive). The value of the action probability of cluster heads is equal and set to 0.5. At every round of information collection, Learning automata of head clusters select whether member node be active or inactive. In this method each node can give four statuses: cluster head node, gateway node, member node, and dead status.

Cluster heads and gateway nodes are continuously awake in current round and all the member nodes go to sleep mode, if they do not have to transmit or receive data.

The packets destined to the nodes in the sleep mode can be buffered at this head cluster. When the node is awake, it can retrieve these packets from buffering its cluster head node.

After the clustering phase, each node sends information to its own cluster head and its neighbors. This information contains: node id, its status, its lifetime, its residual energy, degree of the node, its current cluster head and its neighboring cluster (if it is in its range). At every round, cluster head makes decision, based on acquired probabilities and costs, whether remains cluster head or this responsibility abdicates to others. Each cluster head may not able to communicate directly with neighboring cluster head, so one of the member nodes which have cost in optimal way, will perform forwarding of data packet.

Nodes wake up 'n' times in a period 'T' and during this time, it gives reward or penalty. This received reward or penalty will determine that node should be sleep or awake in next period.

In every T second, nodes predict its own lifetime. Each node monitors its energy consumption and estimates its lifetime based on current and past interval. It calculates how much average energy is consumed by node i per t second during the interval. This value represents how long the remaining energy can keep up the connections with these conditions.

Note that, since the status of the node can change over time due to variation in lifetime of nodes, the activation of a new node depends on its lifetime. In order to apply this method, all nodes should periodically obtain its own lifetime (Eq.3).

$$L(i) = C * \left(\frac{E_r(t)}{E_C(r)} \right) + (1 - C) * \left(\frac{E_R(t-1)}{E_C(r-1)} \right) \qquad (3)$$

Where:
L (i): Lifetime of node i.
E_R (t): Residual energy of node i at time t
E_C (r): Consumption energy of node i at current round
E_R (t-1): Residual energy of node i at time t-1
E_C (r-1): Consumption energy of node i at past round

In this work, each node i estimates its lifetime based on two values, history of past and current round. We use C=0.7, thus giving higher priority to the current history to reflect the current condition of energy expenditure of nodes in best way. In proposed mechanism, we suppose if the cluster member node sends packet in time t then it probability sends packet in time t+1.

Each node predicts its lifetime (Eq.3) and sent it to cluster head. The cluster head gives penalty and reward according to average lifetime of all member nodes. Probability values determine the status of each node. Dead status is defined in each node independently. Once the remaining energy of node reaches the threshold, the node informs its status to cluster head as dead. We set threshold to consumption more than 80% of initial energy level.

The cluster head gives penalty and reward to its member nodes, gateway node and even itself using lifetime values that has received. Since the cluster head plays as receiver or transmitter in t current interval, its energy consumes more than others, so its status must change and this responsibility should be transferred to other nodes with sufficient qualification.

We consider node lifetime and its degree. So cluster head learns which one of the member nodes node covers more space. As see in three following equations both factors are important as the same. The parameter has been multiplied in lifetime to decreasing variation of number of neighbors and lifetime measurement.

The method of giving penalty and reward to nodes:

_if node lifetime is lower of 50% the resulting average lifetime of member nodes, learning automaton gives penalty with coefficient of 0.3 and obtains penalty value with follow equation:

$$\beta = \frac{|N_h - N_i| + L_i * b_1}{N_h + (L_i + b_1)} \qquad (4)$$

b1, b2, b3 is three parameters that take the values to decreasing variation of the lifetime and degree of each node. N_h is degree of the current cluster head. N_i is the degree of the node i. L_i is lifetime of node i and L_{avg} is average lifetime of all member nodes in current cluster. We consider both parameters have the same dimension.

Coefficient value of penalty is tentative.

_ If node lifetime is more than 50% and lower than 80% the resulting average lifetime of member nodes, the learning automaton gives penalty with coefficient of 0.1 and obtains penalty value with follow equation:

$$\beta = \frac{L_i}{L_{avg}} * \frac{|N_h - N_i| + L_i * b_2}{N_h + (L_i * b_2)} \quad (5)$$

_if node lifetime is more than 80% of the resulting average lifetime of member nodes; the learning automaton gives reward and obtains reward value with follow equation:

$$\alpha = \frac{|L_{avg} - L_i| * b_3 + N_i}{L_{avg} * b_3 + N_h} \quad (6)$$

After updating the probability value of each node, the learning automaton in the cluster head chooses one of its member nodes as a new cluster head. Each of nodes has higher probability value, it has higher chance for sleeping and each of nodes has lower probability value, it has higher chance for awake and give cluster head or gateway node.

4. Simulation Results

In this section, we conduct simulations to evaluate our proposed method and compare its performance with the existing implementations of MARI topology [6] and Flat.

Performance Metrics. We monitor the performance of the algorithms with the use of the following metrics. The *Network lifetime* is the total elapsed time from the state of network connectedness to a state in which the first node of network die. The *Energy consumption* is the average energy dissipated by the node in order to transmit a data packet from the source to the destination.

The simulations were carried out in 800*800 environments, with different nodes which were set up at randomly. The communication radius of the each node is 150 meters. In order to minimize the dependency of the simulation results on the network configuration the experiments were run on several network configurations generated via uniformly distributing nodes on an 800m*800m area. Each result reported is the average taken over the results obtained for several network configurations.

We set the initial energy level of each node between 1400 and 1500 unit energy randomly and assume energy consumptions for transmission and reception are 660mW and 395mW respectively.

Network lifetime can be defined as different methods. Lifetime can be consider as dead of first node, can be as dead of a fixed percentage of the nodes or dead of all nodes. We consider lifetime as dead of first node. In this scenario, number of nodes is variable. Figure 2 shows the network lifetime. As the figure 2 shows lifetime of our proposed scheme outperforms other two methods. This presented result is the taken over the results obtained for 20 network configurations.

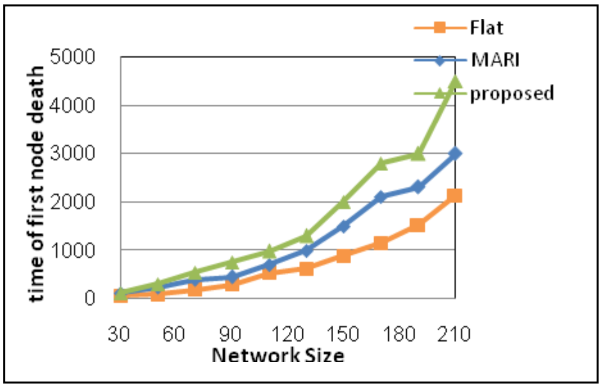

Figure2. Lifetime for various network sizes where lifetime is defined as from time 0 until first node is dead.

We define the network lifetime as the total elapsed time from the state of full battery charge for all nodes in the network to a state in which a fixed number of nodes in the network die due to energy source exhaustion. As defined above maximizing the network lifetime is equivalent to minimizing the variance of the residual energy of the network nodes. Figure 3 shows the ratio of a number of nodes that are alive to the total number of nodes during part of the simulation time at different time instances. As are seen in Flat topology starts dying out sooner. In MARI, the nodes start dying later but die more rapidly. But our proposed scheme performs better. As lifetime increases, more nodes are alive and delivery ratio also goes up.

The energy consumption of node measures the average energy dissipates by the node in order to transmit a data packet from the source to the destination. The same metric is used in [12] to determine the energy efficiency level. It is calculated as follows:

$$Node\,energy\,consumption = \frac{\sum_{i=1}^{N}(e_{i,init} - e_{i,res})}{N\sum_{i=1}^{S} dataM_j} \quad (7)$$

Where N is defined as the number of nodes, $e_{i,init}$ and $e_{i,res}$ are respectively the initial and residual energy levels of node i ,S is the number of nodes that participate in routing and $dataM_j$ is the number of data packets, which received by destination j.

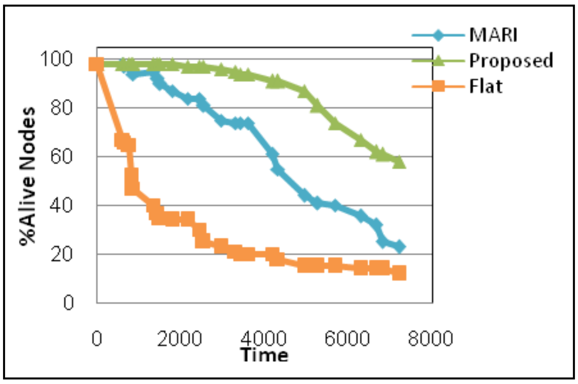

Figure 3.Ratio of alive nodes to total number of nodes to time.

This presented result is the taken over the results obtained for a hundred network configurations. Figure 4 shows energy consumption of node under different sizes of network. We can observe that there is lower node energy consumption in our proposed scheme over the other schemes. The Flat is the most costly method.

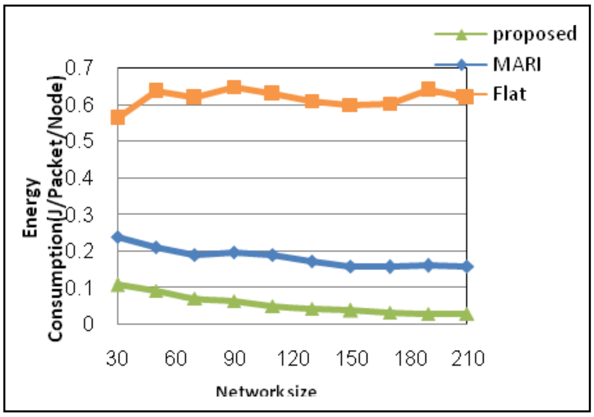

Figure4. Average node energy consumption.

5. Conclusion

In this paper, we propose a new method based on learning automata for ad hoc networks to support QOS. The QOS parameter considered in this paper is balancing the energy consumption among the nodes by clustering, to increases the network lifetime. In fact, we minimize the variance of the remaining energies of the nodes in the network. The clusters are dynamically formed by learning automata and change based on residual energies of nodes. The proposed schema uses learning automata to determine node status in the current round. The simulations results show that our proposed topology management schema outperforms MARI and Flat topology in terms of energy consumption and network lifetime.

References

[1] G. Breed,” Wireless ad hoc networks: Basic Concepts,” In High Frequency Electronics, pages 44–46, 2007.
[2] M. W. Subbarao,” Ad Hoc Networking Critical Features and Performance Metrics,” Wireless Communications Technology Group, NIST, 1999.
[3] Y. Xu, J. Heidemann and D. Estrin,”Adaptive Energy Conserving Routing for Multihop Ad Hoc Network, ”USC/Information Sciences Institute, 2000.
[4] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris,”Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks,” in Proceedings of 7th Int. Conference on Mobile Computing and Networking, pp. 85-96, 2001.
[5] Nanancharaiah, B. Sudha, G.F. Murthy, ”A Scheme for Efficient Topology Management of Wireless Ad hoc Networks using the MARI algorithm,”16th IEEE International Conference on Network, 2008.
[6] M.V.Subramanyam and K.Satya Prasad,”Quantitative Analysis of An Ad hoc Wireless Network Proposed topology” ,In: Asian journal of Information Technology5(5), pp.491- 496, 2006.
[7] K.S.Narendra and M.A.L.Thathachar,Prentice Hall (1989).
[8] M. Chatterjee, S. K. Das, and D. Turgut,”WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,” Cluster Computing, pp. 193–204, 2002.
[9] M. Gerla, T. J. Kwon, and G.Pei,”On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering,” in Proceeding of WCNC, 2000.
[10] O. Younis and S. Fahmy, ”Distributed Clustering in Adhoc Sensor Networks: A Hybrid, Energy-Efficient Approach,” in Proceedings of IEEE INFOCOM, vol. 1, pp. 629–640, 2004.
[11] M. Gholipour and M.R. Meybodi,”LA-Mobicast: A learning automata based distributed protocol for mobicast in sensor networks,” Proceedings of CSICC2007, pp. 1154-1161, 2007.
[12] C. Intanagonwiwat, R. Govindan, and D. Estrin,”Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in Proc. of ACM MobiCom, Boston, MA, USA, pp. 56–67, 2000.

Faezeh Akbari received B.E. degree from Kashan Azad University in 2004. She received M.E. degree from Qazvin Azad University in 2009. Her research interests include network management as well as sensor and ad hoc networks.

Seyed Majid Noorhosseini received the B.Sc. and M.Sc degrees from Amirkabir University of technology in 1986 and 1989, respectively. He received his PhD degree from McGill University in Montreal, Canada in 1994. He was a senior scientist at Nortel Networks in Canada and U.S during 1996-2005, working in different areas of network management. He has a US patent 6707795 in alarm correlation method and system. He is now with the Department of Computer Engineering and Information Technology at Amirkabir University of Technology.

Mahnaz Imanazari Bonab received her B.E. degree from Payamnoor University in 2006. She received her M.E. degree from Qazvin Azad University in 2009. Her research interests include network management, databases and image processing.