Distributed QoS Routing Algorithm in Large Scale Wireless Sensor Networks

Mohammad Sadegh Kordafshari[1], Azadeh Pourkabirian[2], Mohammad Reza Meybodi[3] and Ali Movaghar [4]

[1] Department of Computer Engineering, Science And Research Branch, Islamic Azad University (IAU), Tehran, Iran
[2]Department of Computer Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[3]Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran
[4]Department of Computer Engineering, Sharif University of Technology, Azadi Ave. Tehran, Iran.

***Abstract*-This paper presents a novel routing protocol based on the Learning Automata method for large scale Wireless Sensor Networks (WSNs) codenamed DRLR (distributed reinforcement learning routing). In this method, each node is equipped with learning automata so that it can learn the best path to transmit data toward the sink. The approach proved to be efficient, reliable, and scalable. It also prevents routing hole by considering network density and average of energy levels available. The approach also increases network lifetime by balancing energy consumption. We compared our approach to two other methods (MMSPEED and EESPEED) and the simulation results show our algorithm to better meet end-to-end delay and reliability requirements and to improve network lifetime more**

I. INTRODUCTION

A sensor network consists of a large number of smart sensing devices, referred as sensor nodes. Large amounts of these small-size and low cost sensor nodes can be rapidly deployed in a region of interest and form a loosely-coupled distributed networking system called Wireless Sensor Network (WSN). WSN's [1] have emerged as a popular area of research. With the continuously growing wireless technologies Recent use of WSN's in real time applications lead to the need of focusing research towards routing protocols. In these applications, timely and reliable delivery of the data is very important for positive results as out-dated data will lead to disaster effects. Many mechanisms have been proposed for routing QoS constrained real-time multimedia data [2-4][22][23].

In [5] proposes the first architecture (RAP) that handles deadline issues pertaining to soft real time systems. SPEED [6] provides soft end-to-end deadline for real time systems. It combines feedback control and non deterministic QoS-aware geographic forwarding such that each packet can be routed without global topology information. In addition, it doesn't consider energy metric and reliability in its routing protocol.

MMSPEED [7] extends SPEED to support different velocities and level of reliability for probabilistic QoS guarantee in WSN. Geographic forwarding and multiple QoS levels are used to provide scalability and differentiated QoS options in timeliness and reliability domains. Advantages of this protocol are that it is scalable to very large and dense networks. On the downfall side energy metric is not taken into account and it fails when packet reaches the void in network topology.

In EE_SPEED [8], QoS routing is based on a weight function, which is a combination of the three factors: Delay, Energy and Speed. MCBT [9] propose a distributed algorithm to create a stable backbone by selecting the nodes with higher energy or degree as the cluster heads.

There are some special considerations concerning routing protocol design for WSNs. First of all, because the individual sensor devices have limited power and battery replacement or recharging is typically not practical, any routing protocol must work in an energy-efficient manner. In addition, the nodes in the network are mostly randomly deployed and the position information is not available without a Global Positioning System (GPS) service for the sake of economic cost reduction.

Furthermore self organization and adaptive collaboration among sensors become key properties to provide survivable structure in the network level [24]. Collaboration and self organization among sensor units allow network to route physical information from the observed environment to the base station via multi-hop routes. The features such as low costs of the WSN nodes deployed into inaccessible regions and long life without any maintenance enable sensor networks to be used in a wide range of applications.

Especially in large-scale WSNs where the numbers of nodes can reach thousands or even more, the scalability objective of the routing protocol to handle the long distance which the sensed data must travel from sensors to collection nodes and the huge amount of network overhead must be taken into consideration.

Therefore, there is no way to use classical routing approaches in WSNs and there is need for new routing approaches. As such, these routing approaches emerged as Learning based schemes.

Routing based on Learning techniques have gained in significance because of their flexibility and performance potential [25]. Recently, a few attempts are have been made to apply learning to sensor networks for mobicast routing [18], multicast routing [26][27][19], data aggregation [20], and clustering [16][21].

978-1-4673-0158-9/12/$31.00 ©2012 IEEE

In this paper we propose a real-time routing protocol which is based on the Learning Automata model which and is reliable and energy-aware. The protocol can increase network lifetime by balancing energy consumption. It can also prevent the occurrence of routing holes by considering network density. Furthermore, the proposed algorithm is scalable because it makes all the routing decisions on the basis of local information.

The organization of the remainder of this paper is as follows. The proposed method is given in Section II. Finally, simulation results and conclusions are given in Section III and Section IV, respectively.

II.THE PROPOSED PROTOCOL

In this section, we propose a real-time, reliable, and energy-aware routing algorithm based on Learning Automata which is called distributed reinforcement learning routing (DRLR). This routing protocol finds the best path for forwarding packets toward the sink by considering QoS requirements such as delay, energy, and reliability. DRLR also solves the routing hole problem in WSNs.

The proposed algorithm consists of two major phases: initial phase and selection phase. In the first phase, nodes receive their neighbor's information and store it in their routing tables. This phase is performed only when the sink is activated for the first time and it is also repeated when the sink moves to another location. In the second phase, each node selects the next hop based on local information in its routing table. The two phases are described below.

A. Initial phase

The sink initiates this phase by sending out a Hello packet containing residual energy, Node ID,Time stamp, hop count = 0, and its Sink ID to all the nodes in its range. Upon receiving the Hello packet, a node updates the information of its routing table, increases the hop count, and inserts its residual energy and ID into the Hello packet. Then it rebroadcasts the Hello packet toward its neighbors as soon as possible. The process is repeated for the remaining nodes in the network.

At the end of this phase, each node S_i in the network has two sets $Nset(S_i, SinkID)$ and $Fset(S_i, SinkID)$. The first one contains all neighbors of node S_i which sent it the Hello packet. The second one contains the neighbors of s_i which are one hop closer to the sink than is S_i. The relationship is calculated by Equation (1) below.

```
If ( $S_j \in Nset(S_i,Sink1)$ and Hop-count ($S_j$) < Hop-count ($S_i$) )
Then        add $S_j$ to Fset ($S_i$,Sink1);                                (1)
```

B. Selection phase

In this phase, each node selects the next best hop to forward packets toward the sink. In fact, the node selects the best neighbor using an LA (Learning Automaton[10][11]). Each node is equipped with an LA which gradually learns the next best hop on the path according to local information of the node. The learning automaton associated with node S_i is named LA_i that has $|Nset(S_i, SinkID)|$ actions. An action $\alpha_{ij}(S_j \in Nset(S_i, SinkID))$ of the LA_i means that the next hop will be node S_j in $Nset(S_i, SinkID)$. The action is used for forwarding packets of S_i toward the sink. The probability set of each action by LA_i is initialized through Equation 2:

$$P_{ij}(1) = \begin{cases} \left|\frac{1}{|Fset(S_i, SinkID)|}\right| & \{S_j \in Fset(S_i, SinkID)\} \\ 0 & otherwise \end{cases} \quad (2)$$

where $P_{ij}(1)$ is the probability of selecting action α_{ij} by LA_i in the initial step. The other components of LA_i (reinforcement signal, reward, and penalty) will be described once the two packets defined for this phase are discussed. The packets are Data packet and Reply packet.

1. Data packet

Data packet x contains Data (x), deadline (x), and $T^{elapsed}$. Data (x) has information that will be forwarded toward the sink. Deadline (x), is the point in time at which a message must be received in the sink. It is assigned a value by the source node according to the type of service. $T^{elapsed}$ which determines the remaining time of deadline (x) in each intermediate node can be calculated by Equation (3) [7].

$$T^{elapsed} = T^{Departure} - T^{Arrival} + T^{TransDelay} \quad (3)$$

where $T^{Arrival}$, which is tagged by the MAC layer, is the time when S_i receives the last bit of packet x, $T^{Departure}$ is the time when node S_i transmits the first bit of packet x to the physical layer, and $T^{TransDelay}$ is the transmission delay of packet x, which can be computed using the transmission rate and packet length. The next node S_j can determine the remaining time from the deadline with $T^{elapsed}$ without using a globally synchronized clock. Thus, once node S_j successfully receives from node S_i the packet containing the correct measurement of the $T^{elapsed}$, it can update the remaining time from the deadline through Equation (4).

$$deadline(x) = deadline(x) - (T^{elapsed} + T^{propDelay}) \quad (4)$$

where $T^{propDelay}$ is the propagation delay between two neighbor nodes S_i and S_j, which is negligibly small.

2. Reply Packet

The receiving node (S_j) sends a reply packet to the sender (S_i). The reply packet contains $E_j, D(j, SinkID), T_{processing\ Rep}, |Fset(S_j, SinkID)|$. In this packet, E_j is the residual energy of S_j, $T_{processing\ Rep}$ is the duration for processing a reply packet, $|Fset(S_j, SinkID)|$ is the number of forward neighbors node S_j toward the sink, and $D(j, SinkID)$ is the minimum estimated delay between node S_j and the destination (Sink ID) calculated by Equation (7).

3. Delay estimation

Node S_i calculates the round-trip single-hop delay using a timer. The process is as follows: S_i sends packet x to S_j, which sends back a reply packet. $Delay(i, j)$ which is the delay between S_i and S_j, can be calculated by Equation (5)

$$delay(i, j) = round\ trip\ single\ hop\ delay(i, j) - T_{ProcessingREP} \quad (5)$$

The current delay was estimated by combining the newly measured delay with previous delays via the exponential weighted moving average (EWMA) [6].

$Delay_j\,(i\,,SinkID\,)$ is the delay between node S_i and the sink via node S_j which is calculated through Equation (6).

$$delay_j\,(i\,,SinkID\,) = delay\,(i\,,j\,) + D\,(j\,,SinkID\,) \qquad (6)$$

where D(j, SinkID) is the minimum delay between S_j and the destination node (Sink ID) and is a received field in the reply packet which is calculated at node S_j.

$$D\,(j\,,SinkID\,) = \min_{\forall k \in Nset(s_j\,,SinkID\,)} \{delay_k\,(j\,,SinkID\,)\} \qquad (7)$$

The initial value of $\mathrm{D(j, SinkID)}$ is calculated in the initial phase by Time stamp parameter.To compute the current amount of delay, the EWMA method was applied, and then its value was put in the routing table.

4. Reinforcement signal

After receiving the reply packet and updating the routing table, node S_i calculates the reinforcement signal for packet x using function $\mathrm{F_i(x, Sj)}$ (Equation (8)).

$$\mathrm{F_i(x,Sj)} = \varphi(\mathrm{E}_{\mathrm{avg_i}} - \mathrm{E_j}).\,\varphi\left(\mathrm{delay_j(i, SinkID)} - \mathrm{deadline(x)}\right).\,\varphi(\mathrm{AcceptReliability} + \mathrm{PER_j} - 1)$$

(8)

where

$$\varphi(\mathrm{z}) = \begin{cases} 1, & \mathrm{z} < 0 \\ \gamma, & \mathrm{z} \geq 0 \end{cases}, \quad 0 \leq \gamma \leq 1 \qquad (9)$$

E_{avg_i} is the average residual energy of nodes in $Fset(S_i, SinkID)$ in the routing table.

AcceptReliability is the reliability defined for the packet according to the type of service.

$PER_{i,j}$ is the packet error ratio which is calculated from Equation (10):

$$\mathrm{PER_{i,j}} = \frac{\mathrm{The\_number\_of\_lost\_packets\_sent\_toward\_S_j}}{\mathrm{all\_packets\_sent\_toward\_S_j}} \qquad (10)$$

If $F_i(x, Sj) \geq \theta$, it is proved that α_{ij} was an appropriate action. Thus, $\mathrm{P_{i,j}}$ is rewarded according to Equation (11)

$$\begin{cases} \mathrm{p_{i,j}(n+1) = p_{i,j}(n) + \alpha.\,(RS_{i,j}).\left(1 - p_{i,j}(n)\right)} \\ \mathrm{p_{i,k}(n+1) = p_{i,k}(n) - \alpha.\,(RS_{i,j}).\,p_{i,k}(n)} \qquad \forall \mathrm{k} \;\; \mathrm{k} \neq \mathrm{j} \end{cases} \qquad (11)$$

And if $F_i(x, Sj) < \theta$, α_{ij} is proved to have been an inappropriate action. Thus, $\mathrm{P_{i,j}}$ is penalized according to Equation (12)

$$\begin{cases} \mathrm{p_{i,j}(n+1) = \left(1 - \beta(1 - RS_{i,j})\right).\,p_{i,j}(n)} \\ \mathrm{p_{i,k}(n+1) = \dfrac{\beta(1 - RS_{i,j})}{r - 1} + \left(1 - \beta(1 - RS_{i,j})\right).\,p_{i,k}(n)} \qquad \forall \mathrm{k} \;\; \mathrm{k} \neq \mathrm{j} \end{cases} \qquad (12)$$

In Equations (11) and (12), $\mathrm{RS_{ij}}$ is a coefficient which determines the amount of reward and penalty for the probability of each action according to the feedback received from the environment. This coefficient is here called the reinforcement signal and is computed as follows:

$$\mathrm{RS_{i,j}} = \mathrm{U}\,(\mathrm{F_i(x, Sj)}.\,(1 - \left|\frac{1}{\mathrm{Fset(S_j, SinkID)}}\right|))$$

$$\text{Here,} \quad \mathrm{u(x)} = \begin{cases} 0, & x < 0 \\ \mathrm{x}, & x \geq 0 \end{cases} \qquad (13)$$

5. Scheduling a queue of packets reaching a node

If more than one packet is received by a node, the primary concern would be to schedule the packets in the queue. For this purpose, the Earliest Deadline First (EDF) Algorithm was used. According to this scheduling algorithm, the packet with the earliest deadline will be sent before the other packets in the queue. It is worth noting that high-priority packets are queue-jumped using a prioritized MAC as in RAP [5].

III. Simulation and Results

All simulations were performed using the omnet++ simulator. Nodes are placed randomly on a 200(m)*200(m) two-dimensional area. Energy model specified in [19] was used for estimating the amount of energy consumed for packet transmissions. Each sensor node was set to generate packets each being 1000 bits in size. Simulations were performed on 10, 50, 100, 200, 300, 600, and 800 nodes. In all experiments, α and β for the proposed algorithm were set to 0.1. AcceptReliability and γ were set at 0.7 and 0.5, respectively. The results were averaged over 20 runs.

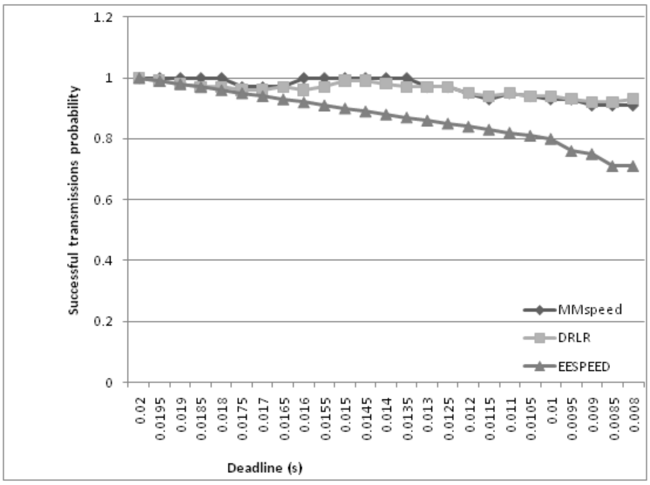

fig. 1. Ratio of successful transmissions

A. Reliability Analysis

For the reliability analysis, account was taken of the number of successful transmissions for different timing constraints and network loads. Fig. 1 shows the percentage of successful transmissions. The DRLR protocol proved to perform significantly better than EESPEED, particularly at shorter packet deadlines.

Fig. 2 shows the relationship between the rate of successful transmissions and the network load. The proposed traffic model was adjusted to Poisson distributed packets with λ = 0.02s and the deadline of packets equal to 0.01s. All protocols worked well when there was a single flow, but increasing the number of flows and thus the network traffic worsened the performance of all protocols. However, the decline in performance was more marked with either MMSPEED or EESPEED than with the proposed algorithm.

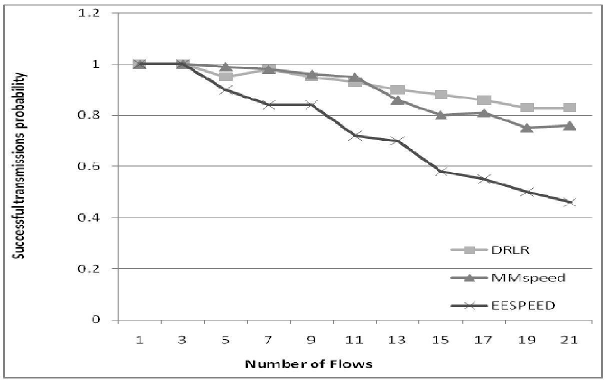

fig. 2. Effect of the network load on reliability

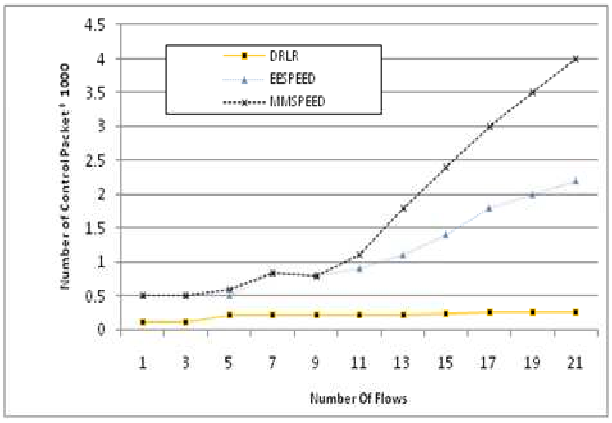

Fig. 3. Packet Overhead Control

B. *Packet Overhead Control*

A comparison was drawn between the MMSPEED and EESPEED protocols and the approach advocated in this work in terms of packet overhead. Two types of overhead were considered arising from sending location and reliability packets. MMSPEED and EE-SPEED send a certain number of packets in order to update location information. However, the proposed algorithm sends no packet for this purpose as it does not need any location information. MMSPEED and EE-SPEED send reliability packets, an action which incurs unwanted overhead. Fig. 3 shows the total number of control packets generated by each protocol during the simulation.

C. *Network Lifetime*

Network lifetime can be defined as the time it takes for the first node or a fraction of the nodes in the network to be depleted of their energies. Since a network tends to dispatch its traffic through the shortest hop paths toward the destination, the nodes on those paths die sooner, causing a reduction in network lifetime.

To increase network lifetime, the proposed approach balances energy consumption across the entire network by considering the energy parameter in routing decisions. More specifically, the DRLR protects nodes with lower energy bands by assigning fewer packets to them. This seems to be an improvement to the previous approaches as MMSPEED does not care about the energy parameter, and EESPEED selects the node with the highest energy level and speed. Fig.4 compares the three algorithms in terms of network lifetime.

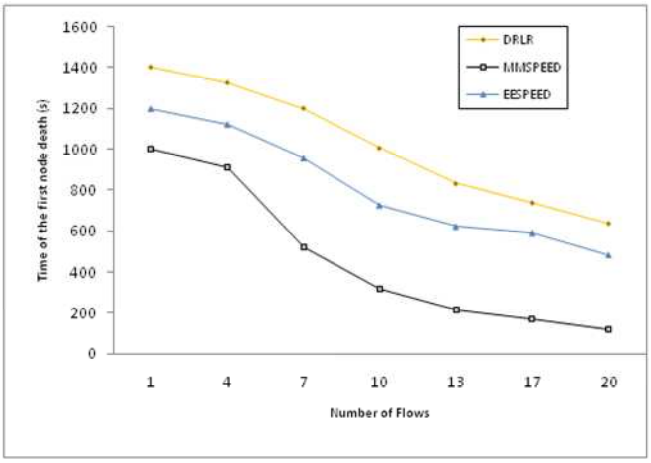

Fig. 4. A comparison of network lifetime on DRLR, MMSPEED, and EESPEED

IV.Conclusion

Real-time routing protocols need to take account of delay and reliability. Such protocols should be able to trade off between end-to-end delay and energy consumption parameters. This paper proposed a real-time routing protocol which is based on the Learning Automata model which and is reliable and energy-aware. The protocol can increase network lifetime by balancing energy consumption. Additionally, it calculates end-to-end delay without using the global clock. It can also prevent the occurrence of routing holes by considering network density. Furthermore, the proposed algorithm is scalable because it makes all the routing decisions on the basis of local information. These features make the proposed protocol a practical alternative to be used in actual networks.

References

[1] H. Karl and A. Willig, "A short survey of wireless sensor networks, Technical Report TKN-03-018," *Telecommunication Networks Group, technical University Berlin, October 2003.*

[2] Y. Li, C. Chen and Y. Song, "Real-time QoS support in wireless sensor networks: a survey," *in the Proceedings of 7th IFAC IntConf on Fieldbuses & Networks in Industrial & Embedded Systems Toulouse, France, Nov 2007.*

[3] P. Rezayat, M. Mahdavi, M. Ghasemzadeh, M. AghaSarram, "A Novel Real-Time Power Aware Routing Protocol in Wireless Sensor Networks," *IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010.*

[4]H.J.Korber, H. Wattar, and G. Scholl, "Modular wireless real-time sensor/actuator network for factory automation applications," *IEEE Transact ions on Industrial Informatics, vol. 3, no. 2, pp. 111–118, 2007.*

[5] C. Lu, B. Blum and T.Abdelzaher , " RAP: A Real-time Communication Architecture for Large-Scale Wireless Sensor Networks," *in the*

Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium , 2002.

[6] T. He, J. Stankovic and C. Lu , " SPEED: A stateless protocol for real-time communication in sensor networks," *in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.*

[7] E. Felembane, C. Lee and E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks,"*IEEE Transactions on Mobile Computing, 5(6), pp. 738-754, 2006.*

[8] M. S. Kordafshari, A. Pourkabirian, K. Faez, A. M. Rahimabadi, "Energy-Efficient SPEED Routing Protocol for Wireless Sensor Networks," *Fifth Advanced International Conference on Telecommunications, pp. 267-271, 2009.*

[9] I. Shin, M. Kim, M. W. Mutka, H. Choo, T. J. Lee, " MCBT: Multi-Hop Cluster Based Stable Backbone Trees for Data Collection and Dissemination in WSNs," *MDPI Sensors, vol. 9, pp. 6028-6045, 2009.*

[10] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An introduction," *Prentice Hall, 1989.*

[11] M. A. L. Thathachar and 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.*

[12] M. Haleem, R. Chandramouli, " Adaptive downlink scheduling and rate selection: a cross layer design,"*Special issue on Mobile Computing and Networking, IEEE Journal on Selected Areas in Communications,* vol. 23, no.6, June 2005.

[13] P. Nicopolitidis, G. I. Papadimitriou, A. S. Pomportsis, "Exploiting Locality of Demand to Improve the Performance of Wireless Data Broadcasting ," *IEEE Transactions on Vehicular Technology,* vol.55, no.4, pp. 1347-1361, July 2006.

[14] P. Nicopolitidis, G. I. Papadimitriou, A. S. Pomportsis, "Learning Automata Based Polling Protocols for Wireless LANs," *IEEE Transactions on Communications,* vol.51, no.3, pp. 453-463, March 2003.

[15] B. V. Ramana, C. S. R. Murthy, "Learning-TCP: A Novel Learning Automata Based Congestion Window Updating Mechanism for Ad hoc Wireless Networks," *in Proceedings of 12th IEEE Intl. Conf. on High Performance Computing,* December 2005, pp. LNCS 454-464.

[16] J. A. Torkestani and M. R. Meybodi, "An Efficient Cluster-Based CDMA/TDMA Scheme for Wireless Mobile Ad-Hoc Networks: A Learning Automata approach," *Network and Comp. Apps.*, Elsevier, vol. 33, no. 4, July2010, pp. 477–90.

[17] H. Beigy, M. R. Meybodi, "Learning Automata based Dynamic Guard Channel Algorithms," *Journal of High Speed Networks, 2008*

[18] 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.*

[19] J. A. Torkestani and M.R. Meybodi, "Mobility-based Multicast Routing Algorithm for Wireless Mobile Ad-Hoc Networks: A Learning Automata Approach," *Comp. Commun., Elsevier, vol. 33, no. 6, Apr. 2010, pp. 721–35.*

[20] M. Esnaashari and M. R. Meybodi, "Data Aggregation in Sensor Networks using Learning Automata," *Wireless Networks*, Springer, vol. 16, no. 3, Apr. 2010, pp. 687–99.

[21] M. Esnaashari, M. R. Meybodi, "A Cellular Learning Automata based Clustering Algorithm for Wireless Sensor Networks," *Sensor Letters,* Vol. 6, No. 5, pp. 723-735, October 2008.

[22] A. M. Ortiz, F. Royo, T. Olivares, J. C. Castillo, L. Orozco-Barbosa, P. J. Marron, "Fuzzy-logic based routing for dense wireless sensor networks," *Telecommun Syst*, Springer, Sept. 2011.

[23] C. Barenco Abbas, R. González, N. Cardenas, L.J. García Villalba, "A proposal of a wireless sensor network routing protocol," *Telecommun Syst*, Springer, March 2008, pp. 61–68.

[24] J. N. Al-Karaki, A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: a Survey," *Wireless Communications, IEEE, 2004, pp. 6-28.*

[25] A. Forster, "Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey," *in Proc. of the 3rd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2007.*

[26]A. Forster, A. L. Murphy, " FROMS: Feedback Routing for Optimizing Multiple Sinks in WSN with Reinforcement Learning," in *Proc. of the 3rd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP),* 2007.

[27]A. Forster, A. L. Murphy, "Balancing Energy Expenditure in WSNs through Reinforcement Learning: A Study," *in Proc. of the 1st Int. Workshop on Energy in Wireless Sensor Networks (WEWSN),* 2008.