A Differentiated Real-Time Routing Protocol Along with Re-Routing Policy in Wireless Sensor Networks

Sayyed Majid Mazinani, *Member, IACSIT*, Elham Sadr, and Ali Naderi

***Abstract*—Many wireless sensor network (WSN) applications require real-time communication. One of the most important and challenging issues in real-time applications of resource-constrained WSNs is providing end-to-end delay requirement. To address such an issue a few QoS routing protocols have been proposed. Also, in many applications, the delay level required by the data packets is different. In this paper, we focus on building a real-time routing protocol called DRTR which routes packets towards the destination node by classifying data into differentiated classes. DRTR improves real-time performance by means of reducing the packet dropping in routing decisions. It is a power-aware routing protocol which takes into account both power transmission costs and residual energy of routers to achieve power efficiency. Moreover, DRTR employs a new policy called re-routing policy which allows the packets of a specific class to be routed as the packets of a lower/higher real-time class in particular situations.**

***Index Terms*—Differentiated real-time routing, power-aware routing, re-routing policy, wireless sensor networks.**

I. Introduction

Wireless sensor networks (WSNs) have drawn the attention of the research community in the last few years. This growing interest can be largely attributed to new applications enabled by large-scale networks of small devices capable of harvesting information from the physical environment, performing simple processing on the extracted data and transmitting it to remote locations[1].

Among several important aspects of WSNs such as architecture and protocol design, energy conservation, and location algorithm, supporting Quality of Service (QoS) in WSNs is still a largely unexplored research field [2].

Depending on different applications, generated packets call for diverse Quality of Service (QoS) supports. The commonly accepted QoS metrics include bandwidth, delay, delay jitter (delay variation), reliability (packet loss rate), etc[3]. Although energy efficiency is usually the primary concern in WSNs, the requirement of low latency communication is getting more and more important in new applications. Out-of-date information will be irrelevant and may even lead to negative effects to the system monitoring and control. Examples of real-time (RT) sensor applications can be found in many military or environmental surveillance systems [4].

Manuscript received June 10, 2013; revised August 15, 2013.

Sayyed Majid Mazinani is with the Department of Electrical Engineering, Khayyam Higher Education Institute, Mashhad, Iran (e-mail: mazinani@ieee.org).

Elham Sadr and Ali Naderi are with the Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Iran (e-mail: sadr_elham@yahoo.com, a.n.lasr@gmail.com).

Additionally, in many applications, the delay constraint required by the data packets is different. So, the necessity of a real-time routing protocol which can classify data into differentiated classes is comprehensible. From a layered point of view, the MAC should provide channel access delay (single-hop) guarantee, while in the network layer the routing protocol should bound the end-to-end (multi-hop) transmission time. One may also adopt a cross-layer design to have a joint optimization. Several attempts have been recently made to propose real-time routing protocols.

In SPEED [5] is a real-time routing protocol for soft end-to-end deadline. This protocol guarantees deadline by maintaining a packet delivery speed across the network which should be greater than or equal to the desired velocity. The required velocity defined by the ratio of straight line distance from source s to target t over the required deadline. In SPEED if there is no neighboring node which can support the desired velocity, the protocol probabilistically drops packets to regulate the workload.

RPAR [6] is a real-time power-aware routing (RPAR) protocol that is proposed to achieve application specified communication delay at low energy cost by dynamically adjusting transmission power and routing decisions. It allows the application to control the tradeoff between energy consumption and communication delay by specifying packet deadlines. Both RPAR and SPEED operate based on the one-hop neighborhood information.

THVR [7] is newly proposed real-time protocol. It has also adopted the approach of mapping packet deadline to a velocity like SPEED, which is known as a good metric to delay constrained packet delivery. However, its routing decisions will be made based on two-hop neighborhood information. It has achieved lower end to- end deadline miss ratio and higher energy utilization efficiency because using more neighbor information for routing result in better performance. Both THVR and SPEED have one important drawback. If the forwarding unit cannot find any node that can provide required velocity, the packet will be dropped. It forces some delay and energy consumption overhead to network and can cause problem when the packet is mission critical. So, the selected path is not optimal.

PATH [8] is also a power -aware routing protocol which utilizes the concept of two-hop neighbor information and power-control mechanism. PATH dynamically adjusts transmitting power in order to reduce the probability of packet dropping. Also, it addresses practical issue like network holes, scalability and loss links in WSNs.

In this paper, we focus on building a routing protocol called DRTR which routes packets towards the destination node in a power-aware real-time manner through a modular approach. In DRTR, depending on the real-time class which packets belong to, they are routed through different paths being able to provide the demanded real-time level by that class.

Considering both required energy to forward a packet and the remaining energy of intermediate nodes in addition to real-time level required by the packets, DRTR tries to find the best satisfying nodes in order to forward packets to them as the next nodes of paths towards the sink.

Also, in order to improve the delivery ratio in the whole of the network in absence of a suitable next node, DRTR employs a new policy called *re-routing policy* which allows the packets of a specific class to be routed as the packets of a lower/higher real-time class. The remainder of the paper is organized as follows. Section II explains the network model and assumption. Protocol description are presented and discussed in Section III. Section IV describes performance analysis. Finally, Section V concludes our work, and discusses some future directions.

II. Network Model and Assumption

In the following the network is represented by a set V of nodes. We note $dist(v_i, v_j)$ as the linear distance between two nodes $v_i, v_j \in V$. Each node should be aware of its own coordinates. Sensors shipped with the GPS receivers, can readily sense their location information. This position serves as the network (global) address. In addition, the node should be aware of its current battery state $B(v_i)$ (also termed residual energy). We assume that nodes have the same and spherical transmission power range P_{range}, and that each node can control its transmission power. The set of nodes in v_i's vicinity denoted by $N(v_i)$ is called v_i's neighboring nodes defined by $N(v_i) = \{v_j : dist(v_i, v_j) \leq P_{range}\}$. In addition to $N(v_i)$,we define the set of neighboring nodes providing positive progress for node v_i, towards the sink, denoted by $N_{prog}(v_i)$, as the set of neighboring nodes that are closer to the sink than v_i. It is given by: $N_{prog}(v_i)= \{v_j \in N(v_i) : \mathrm{dist}(v_j, sink) \leq dist(v_i, sink)\}$. Also, DRTR uses the progressive value between two nodes v_i and v_j denoted by $prog(v_i, v_j)$, which is the distance from one node to the other node in the direction of the vector from the source to the sink. Like all geographic routing protocols, each node needs to know about the positions of its neighboring nodes as well as the destination node (sink). A HELLO protocol is executed between neighboring nodes allowing mutual update of the neighboring nodes' list and several parameters, as in [9], [10]. For localized routing to be effective, nodes are supposed to be stationary. Node density is supposed to be high enough to prevent void situation, in which a router cannot find a closer node to the destination amongst its neighboring nodes.

III. Protocol Description

As mentioned earlier, DRTR running on each node in a distributed manner, determines next nodes of the paths from a source node to the sink based on a modular approach along with a re-routing policy for special situations.

A. Real-Time Module

Depending on the application, it is possible to define n differentiated real-time classes denoted by RTC_k ($1 \leq k \leq n$), each one is requested by some packets. A set of neighboring nodes of node v_i which can support a specific real-time class RTC_k denoted by $N_{prog}^{RTC_k}$, is formally defined as: $N_{prog}^{RTC_k}(v_i) = \{v_j \in N_{prog}(v_i) : d_k \leq delay(v_i, v_j) \leq d_{k+1}\}$; where d_k and d_{k+1} are two possible successive values for delay(values of d_k and d_{k+1} depend on the application).

This module uses the packet velocity approach given in [10] that has the advantage of not requiring any synchronization between nodes. The main difference from [10], however, is the use of a simple but memory and time-efficient estimation method (EWMA) instead of Jacobson's algorithm, and particularly the consideration of waiting time at the next hop's queue. Assume a delay sensitive packet has a delivery deadline, *dd*, specified by the upper layers and indicating the time the packet should be delivered to the sink node. We define two velocities to be used; required velocity (speed), S_{req}, and offered (actual) velocity, $S_{off}(v_j)$, for every node v_j in $N_{prog}(v_i)$. Upon receiving a packet the recipient node stamps the corresponding reception event locally. To account for all the possible delays in the node, i.e., queuing, contention, retransmission, etc., it updates the deadline prior to each transmission in the MAC layer to account for the delay from receiving the packet until it reaches its final transmission. If the reception time is denoted, t_{rec}, the time of last transmission, t_{tr}, the bandwidth, *bw*, and the packet size, size, then the time remaining to the deadline, *rt*, is updated at node, v_i, as:

$$rt = rt_{req} - (t_{tr} - t_{rec} + size / bw), \qquad (1)$$

where rt_{req} is the value of, *rt*, at time of reception, and (t_{tr}-t_{rec} + $size/bw$) gives the entire delay from the reception of the packet at v_i until the transmission of the last bit. It includes both queuing delay ($t_{tr}-t_{rec}$) and data transfer delay ($size/bw$). Propagation delay can smoothly be added but it is omitted since it can be negligible. Upon reception of the packet at v_i, the required speed is calculated using both the remaining time to the deadline (stamped in the packet either by the previous node or the upper layer) and the remaining distance to the destination as given:

$$S_{req} = \frac{d(v_i, sink)}{rt}. \qquad (2)$$

This way we propose a solution to handling the end to-end deadline as local problem of satisfying the required velocity at each hop. Furthermore, no global time stamping is used but only relative time, which does not require clock synchronization.

To achieve the required velocity, the real-time module at node, v_i, calculates the velocity offered by every candidate, using EWMA-based estimations provided by the neighbor manager. These estimations include waiting time at the queue of node v_i, say $wt(v_i)$, transmission time to the next node, $dtr(v_j)$, and waiting time at the queue of the latter, $wt(v_j)$. Many of the previous solutions in the literature haven't considered the waiting time at next node's queue. Note that delay due to transmission, $dtr(v_j)$, includes estimation of the time interval from the packet becomes head of v_i's transmission queue until its reception at v_j. This includes all delays due to contention (channel sensing, RTS/CTS if any, slots, etc. depending on the used MAC protocol) and data transfer delay. It is updated after each packet transmission with EWMA, using its delay ω as a sample, given by: $\omega = t_{ACK} - size(ACK)/bw - t_0$; where t_0 denotes the time the packet is ready for transmission (becoming the head of transmission queue), t_{ACK} the time of *ACK* reception, *bw* the bandwidth and *size*(*ACK*) the size of the *ACK* packet. The estimated velocity for node, v_j , is given by:

$$S_{off}(v_j) = \frac{dist(v_i, sink) - dist(v_j, sink)}{wt(v_i) + dtr_{v_j} + wt(v_j)}. \tag{3}$$

After computing velocities of all candidate nodes, the real-time module calculates the set of nodes supposed to meet the required deadline, $N_{prog}^{S_{req}}(v_i)$ as,

$$N_{prog}^{S_{req}}(v_i) = \{v_j \in N_{prog}(v_i) : S_{off}(v_j) \geq S_{req}\}.$$

This set is then transferred to the power-efficiency module to extract the most power-efficient node.

B. Power-Efficiency Module

In this module, DRTR tries to find the most energy-efficient next node from the qualified candidates selected by the real-time module. Both power transmission costs and residual energy of routers should be considered to achieve power efficiency. So, DRTR uses a new metric called cost function as a key means to make decision during routing in this module. The energy consumed by the physical layer in transmitting and receiving one bit of data over a distance d can be formulated as follows:

$$E = 2E_{elec} + E_{amp}d^2, \tag{4}$$

where E_{elec} is the energy consumed by the electronics [Joules/bit], E_{amp} is the energy used in transmitting 1 bit over 1 meter [Joules/bit/n] and d is the distance between the nodes.

Upon receiving a packet at a node v_i, it must calculate the value of cost function for all candidates in its vicinity chosen by the real-time module as follows:

$$\cos t(v_i, v_j) = \left\lceil \frac{dist(v_i, sink)}{prog(v_i, v_j)} \right\rceil \times \frac{\left(2E_{elec} + E_{amp} \times dist(v_i, v_j)^2\right)}{RE(v_j)}, \tag{5}$$

where $RE(V_j)$ is the residual energy of the potential next node v_j in the vicinity of v_i. DRTR at node v_i selects the node v_j which provides the lowest value of cost function.

C. Re-Routing Policy

Normally, if there is not any node which can provide the favorite real-time class, the packets must be dropped [5][7]; but as mentioned earlier, in order to improve the delivery ratio in the whole of the network, we employ a new policy called re-routing policy which makes possible the packets of a specific class to be routed as a lower/upper real-time class's packets through another neighboring node supporting that class.

If a packet inevitably must be re-routed as a lower class packet, the total latency would be increased. In order to prevent from this increase in total latency, in remaining of the path, the packet will be treated as a packet of higher classes and consequently routed through lower latency nodes. Each time a packet needs to be re-routed as a packet of lower/higher classes, DRTR determines the possibility of re-routing and also the suitable new class for the packet so that the original total latency can be provided in the remainder of the path. In order to achieve these goals, DRTR acts in the following way.

At first, suppose that a packet generated at source node v_s travels the entire path towards the sink in its original real-time class like RTC_x. Therefore, the total delay (D_{total}) experienced through the path which is equal to sum of each hop delay, has to meet the following condition:

$$D_{total} = \sum_{v_s}^{sink} D_i, \tag{6}$$

where D_i is the delay of each single hop of the path.

Since DRTR routes the packets hop by hop, until a packet reaches the sink by traversing all nodes on the path, the real value of total latency is not available in the middle of the path. In practice, we use the maximum permissible value for D_{total} in the calculations related to re-routing policy.

Now, suppose that during the routing, an intermediate node of the path, v_j fails to find a satisfactory next node in its vicinity to which it forwards the packet; in other words, a packet requesting the real-time class, RTC_x, can proceed its way towards the sink until it stops at node v_j for which $N_{prog}^{RTC_x}(v_j)$ is empty.

In such a situation, some protocols drop the packet [9][10], but DRTR tries to continue routing by means of re-routing policy in order to prevent a reduction in total delivery ratio in the network. Let $D_{gain}(v_j)$ be the acquired latency of a packet originating from v_s until reaching v_j (until now that it has reached).

Also $D_{rem}(v_j)$ is the remaining delay which totally can be tolerated by the packet in the rest of path from v_j to the sink so that the expected total latency is visited finally. D_{rem} is calculable using the following equation:

$$D_{rem} = D_{total} - D_{gain}. \tag{7}$$

When $N_{prog}^{RTC_x}(v_j)$ is empty, v_j is allowed to select a next node from $N_{prog}(v_j)$ denoted by v_{j+1} which provides a lower or higher real-time level belonging to a real-time class other than RTC_x .

If the real-time class supported by v_{j+1} is lower than RTC_x, the packet will be sent to it and then in the rest of path from v_{j+1} to the sink it can shift to its original class RTC_x again. On the other hand, if DRTR is permitted to choose a node providing a real-time class lower than RTC_x, then it has to check to which class the packet may be shifted so that the increase of total latency is compensated in the remaining path from v_{j+1} to sink. The lowest possible level of real-time which at least v_{j+1} must support denoted by λ, is needed to meet the following condition:

$$\sum_{i=j+2}^{j+HC(v_j)-1} D_i \le D_{rem}(v_j) - \lambda \Rightarrow \lambda \le D_{rem}(v_j) - \sum_{i=j+2}^{j+HC(v_j)-1} D_i, \quad (8)$$

where $HC(v_i)$ is the estimated number of required hops for routing the data to the sink. If node v_i chooses its neighbor v_j as the next node to transfer data to it, then v_i can estimate the total hop count needed to route the data to the sink as follows:

$$HC(v_i) = \frac{dist(v_i, sink)}{prog(v_i, v_j)}. \quad (9)$$

We use the average value of latency for a link in the network denoted by D_{min} for D_is in the mentioned relation. Thus, it is applicable as:

$$\lambda \le D_{rem}(v_j) - \left((HC(v_j) - 2) \times D_{min} \right). \quad (10)$$

D. Neighbor Manager

The neighbor manager runs the HELLO protocol, manages neighbor table, and implements EWMA-based estimations described before. This enables it to provide the decision-making modules with the required information for routing. Neighbor table assigns an entry for each neighbor node, which includes all information related to the node such as position, residual energy, estimated hop count to sink, required transmission energy towards it and etc. The HELLO protocol consists of periodical broadcast of HELLO packets. These packets are used to update existing entries, and delete entries when neighboring nodes break down, which can be detected in case of not receiving HELLO packets after a defined period of time (timeout). Neighbor manager is the first module that receives the packet from the higher layers. It provides the routing modules with all information it needs such as the set of nodes ensuring positive progress (N_{prog}) and current values of its required parameters.

IV. Performance Analysis

In this section, we evaluate the performance of our algorithm via simulation. We implemented a simulation framework using MATLAB and C++.The goal of the simulation is to show that DRTR can provide an improved performance in a differentiated data model. The results are compared with three real-time routing protocols for WSNs named SPEED [5], THVR [7], and PATH [8].

A. Simulation Model

The same network setup is used to compare the four routing protocols. Table I summarizes the network characteristics. Each node is equipped with a total amount of energy 5J at the beginning of the simulation. We apply the same radio model introduced in [11] and used by several papers. In this radio model, the energy consumed in transmitting and receiving k bits of data over a distance d can be formulated as follows.

$$EnT(k) = E_{elec}.k + E_{amp}.k.d^2, \quad (11)$$

$$EnR(k) = E_{elec}.k, \quad (12)$$

where E_{elec}= 50nj/bit and E_{amp} = 100pj/bit/m^2.

TABLE I: Network Characteristics

Simulation Area (Terrain)	200 m×200 m
Radio Range	40 meters
Number of Nodes	200 nodes
Node Deployment	Uniform random
Data Packet Size	100 Bytes
Buffer Size	10 packets

We used a traffic scenario, where four source nodes at the left side of the terrain send periodic data to the sink at the right side. Normally, each source node generates data units with the rate of 40packets/s. Three real-time classes denoted by regular data, soft real-time data and hard real-time data were assigned to the data packets. Real-time packet rate was varied from 0.1 to 1, and the remaining rate to 1 was allocated to regular packets, i.e. the overall traffic load is fixed for all scenarios.

B. Simulation Results

We ran the simulation with several parameters, including real-time packet rate and time. Simulation results are obtained from multiple runs and results are averaged over the runs. In order to investigate the effect of re-routing policy, we implemented a base-line model of DRTR without re-routing policy called Normal Routing comparing with DRTR. Fig. 1 shows the relation of the end-to-end delay over the time for different packet types. From this figure, we can find that the latency is approximately fixed over the time in DRTR. In two other scenarios, we used only two types of packet. In Fig. 2, end-to-end delay over the time has been showed for the case we have only regular packets and soft real-time packets. Until 40s, there is no real-time packets in the network so both Normal Routing and DRTR are the same while after 40s when source nodes start generating real-time packets the latency decrease remarkably in DRTR. Fig. 3 also shows the relation of the end-to-end delay over the time but there are only regular packets and hard real-time packets. After 40s, the real-time packets are injected in the network, so the end-to-end delay decreases in DRTR in comparison with Normal Routing. Here, the latency is even less than the previous scenario because DRTR forwards the hard real-time packets through the least latency paths. Fig. 4 compares the percentage of packets delivered within the deadline in our scheme with SPEED, THVR and PATH over different real-time packet rates. PATH is relatively less affected buts its performance is less than DRTR, whose performance even increases with the real-time packets' rate, and thus a tremendous improvement. This improvement results mainly from re-routing policy utilized in DRTR instead of dropping packets used in other protocols.

V. Conclusion

In this paper, we focused on designing a general energy efficient, and differentiated real-time routing protocol named DRTR which routes packets by classifying data into differentiated classes. In DRTR, the routing decision is made through a double module scheme. Also, in order to improve the delivery ratio in the whole of the network in absence of a suitable next node, DRTR employs a new policy called re-routing policy which allows the packets of a specific class to be routed as the packets of a lower/higher real-time class. We evaluated the performance of our proposed protocol through simulation under different scenarios. DRTR was compared with SPEED, THVR, and PATH protocols by investigating the effects of real-time packet rate and time which exhibits a better performance than others. As a future work, we intend to implement our new re-routing policy for optimization of tree-based routing protocols.

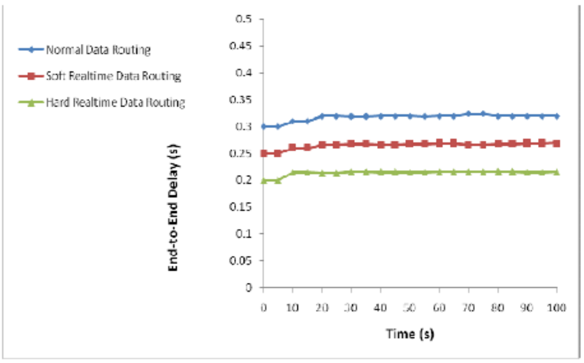

Fig. 1. End-to-end delay vs. time.

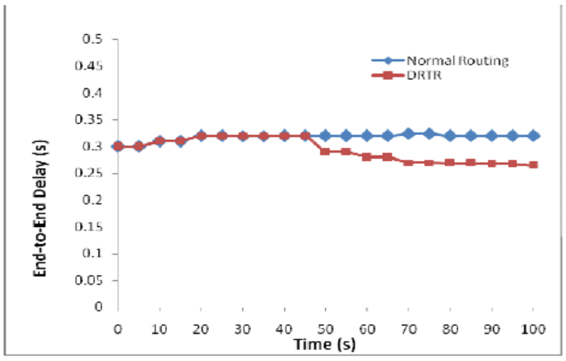

Fig. 2. Delay vs. real-time packet rate.

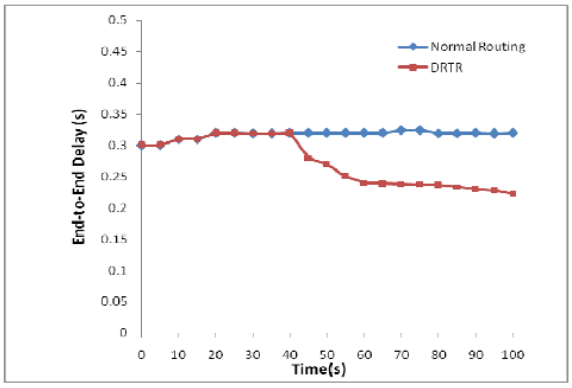

Fig. 3. Delay vs. real-time packet rate.

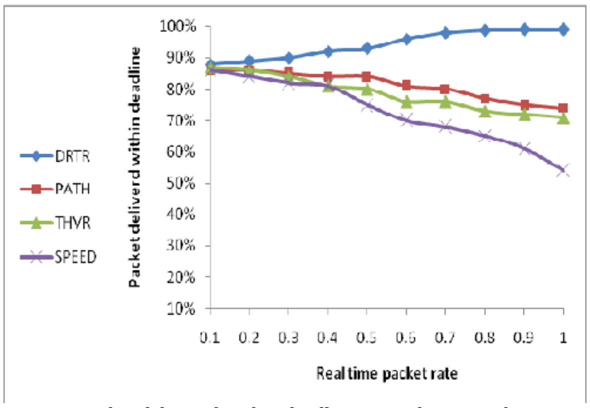

Fig. 4. Packet delivered within deadline vs. real-time packet rate.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks (Elsevier) Journal*, vol. 51, no.4, pp. 921-960, March 2007.

[2] W. Chen, W. Li, H. Shou, and B.Yuan, "A QoS-Based adaptive clustering algorithm for wireless sensor networks," in *Proc. IEEE International Conference on Mechatronics and Automation*, June 2006.

[3] X. Yin, S. member, X. Zhou, M. Pan, and S. Li, "Admission control with multi-constrained qos providing in wireless sensor networks," in *Proc. International Conference on Networking Sensing and Control (IEEE ICNSC)*, April 2010, pp. 524-529.

[4] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proc. IEEE*, vol. 91, no. 8, pp. 923-933, November 2006.

[5] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless protocol for real-time communication in sensor networks," in *Proc. IEEE International Conference on Distributed Computing Systems*, 2005, pp. 46–55.

[6] C. S. Chen, Y. Li, and Y. Q. Song, "An exploration of geographic routing with one-hop based searching in wireless sensor networks," in *Proc. Chinacom*, Aug. 2008, pp. 376-381.

[7] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proc. SECON '04*, 2004.

[8] P. Rezayat, M. Mahdavi, M. GhasemZadeh, and M. AghaSarram, "A novel real-time routing protocol in wireless sensor networks ," in *Proc. IEEE International Conference on Current Trends in Information Technology (CTIT)*, Dec. 2009, pp. 1 – 6.

[9] E. Felemban, C. G. Lee, and E. Ekici, " MMSPEED: Multipath multi-SPEED Protocol for QoS Guarantee of reliability and timeliness in wireless sensor networks," in *Proc. IEEE Transactions on Mobile Computing*, vol. 5, no. 6, June 2006.

[10] K. Zeng, K. Ren, W. Lou, and P. J. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.

[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishanan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Comunications*, no. 4, pp. 660-670, October 2002.

Sayyed Majid Mazinani was born in Mashhad, Iran on 28 January 1971. He received his Bachelor degree in Electronics from Ferdowsi University, Mashhad, Iran in 1994 and his Master degree in Remote Sensing and Image Processing from Tarbiat Modarres University, Tehran, Iran in 1997. He worked in IRIB from 1999 to 2004. He also received his phD in Wireless Sensor Networks from Ferdowsi University, Mashhad, Iran in 2009. He is currently assistant professor at the faculty of Engineering in Imam Reza University, Mashhad, Iran. He was the head of Department of Electrical and Computer Engineering from 2009 to 2012. His research interests include Computer Networks, Wireless Sensor Networks and Smart Grids.

Elham Sadr is a master of Computer Software Engineering student at Islamic Azad University of Mashhad (IAUM), Iran. She received her BSc degree in Computer Software Engineering from Islamic Azad University of Mashhad (IAUM), Iran. She is a faculty member in the Department of Computer Science, Islamic Azad University of Neishabour, Iran. She has cooperated in translating several books in computer science field. Her current research interests include wireless sensor networks and QoS support in wireless networks.

Ali Naderi was born in 1981 in Mashhad, Iran. He received his BS in Computer Engineering from Yazd University in 2005 and his MS in Computer Engineering from Islamic Azad University of Mashhad in 2012. Now, he is lecturer in High Education Institution of Kashmar, Iran. His research interests are computer networks, Quality of Service and wireless sensor networks.