

Energy-efficient and QoS-aware geographic routing protocol for wireless sensor networks

Ali Ghaffari^{a)}, Amirmasoud Rahmani, and Ahmad Khademzadeh

Department of Computer Engineering,

Science and Research Branch, Islamic Azad University, Tehran, Iran

^{a)} A.Ghaffari@iaut.ac.ir

Abstract: In this paper, we propose a new energy-efficient and QoS-aware geographic routing (EQGR) protocol for wireless sensor networks. EQGR maximizes the network lifetime and uses optimum cost function to select the best neighbor node. For minimizing inter-node and intra-node timeliness, we use two neighbor information routing tables for reliability and real-time domains. To minimize the queue waiting time for time-sensitive packets, scheduling policy considers different priorities for different types of data according to their criticality. Simulation results show that EQGR can achieve reliable data forwarding with energy-efficiency, low missed-deadline data delivery, low end-to-end latency, and extremely low control signal overhead.

Keywords: wireless sensor networks, quality of service, geographical routing

Classification: Science and engineering for electronics

References

- [1] Y. Xue, B. Ramamurthy, and M. C. Vuran, "A service-Differentiated Real-time Communication Scheme for Wireless Sensor Networks," *Proc. 33th IEEE Conf. Local Comput. Netw.*, pp. 748–755, 2008.
- [2] E. Felemban, C. Lee, and E. Ekici, "MMSPEED: Multipath Multi-speed Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Network," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 738–754, June 2006.
- [3] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks," *Proc. 28th Annual IEEE Conf. Local Comput. Netw.*, Los Alamitos, CA, USA, pp. 406–415, Oct. 2003.
- [4] A. A. Ali and F. Norsheila, "A Real-time Routing Protocol with Load Distribution in Wireless Sensor Networks," *Comput. Commun.*, pp. 3190–3203, 2008.
- [5] G. Park, S. Yi, J. Heo, W. C. Choi, G. Jeon, and C. Shim, "Energy aware routing with dynamic probability scaling," *Lecture in Computer Science*, New York, Springer, vol. 3642, pp. 662–670, 2005.
- [6] M. Zuniga and B. Krishnamachari, "Analyzing the Transitional Region in Low Power Wireless Links," *IEEE Conf.*

- [7] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, “Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks,” *Proc. IEEE Wireless Communications and Networking, Conf., WCNC-2003*, New Orleans, Louisiana, USA, vol. 3, pp. 1918–1922, 16–20 March 2003.
- [8] A. Leon-Garcia, “*A Probability and Random Process for Electrical Engineering*,” 2nd ed., Wesley, 1994.

1 Introduction

End-to-end real-time and reliable communications are important for WSNs to achieve the collaborative sensing task with specific timing constraints and unreliable links [1]. According to the event importance and urgency, the data packets associated with different events can be assigned different end-to-end deadline requirements. Our goal is to provide end-to-end QoS provisioning in real-time and reliability domains while prolonging the lifetime of network via load balancing and local decisions without end-to-end route discovery and maintenance. In real-time domains, each node assigns proper priority to the data packets depending on the information urgency and packet remaining deadline. At each node, a three-level queue scheduler is used to provide the service to the packets, according to their priority and criticality. In reliability domain, we use link quality and multipath forwarding depending on the packet’s reliability. EQGR computes optimal forwarding nodes based on parameters such as residual energy, occupied buffer, link quality, and required energy for packet transmission.

2 Related works

MMSPEED [2] is a routing protocol that considers both timeliness and reliability for path selection. For timeliness domain, multiple QoS levels are supported by providing multiple packet delivery speed guarantees. For reliability, multiple reliability requirements are supported by probabilistic multipath forwarding. However, MMSPEED fails to consider energy issues. ReInForm [3] addresses end-to-end reliability by sending multiple copies of a packet along multiple edge-disjoint paths from the source to the sink. The protocol does not consider path delays while selecting multiple paths. RTLD [4] take into account link quality, packet delay, and remaining power of the next hop nodes. However RTLD fails to consider priority queue.

3 Protocol overview

In this section, we describe proposed protocol with more details.

3.1 Neighbor table construction

Sensor nodes use *HELLO* packets to create a neighbor table (NT) as shown in Fig. 1 (a) and Fig. 1 (b). Neighbor nodes receive *HELLO* message and, by comparing their geographical position with the source node, decide to reply

Position (x,y)	Node ID	Energy Cost	Used Buffer	Link Quality	Residual Energy
-------------------	------------	----------------	----------------	-----------------	--------------------

(a)

Position (x,y)	Node ID	Speed
-------------------	------------	-------

(b)

Fig. 1. EQGR tables. (a) *HELLO* packet response and (b) Real-time routing table.

to the *HELLO* message response if the neighboring node is closer to the sink than it is. We choose a threshold for residual energy (E_{th}) and every node having more energy than E_{th} will be permitted to add in the table.

For neighbor nodes i and j , the geographical distances from i and j to the sink k are $dist(i, k)$ and $dist(j, k)$ respectively. Suppose that node i sends a packet to node j . This sending makes $dist(i, k) - dist(j, k)$ geographic progress toward the sink k along the virtual direct line from node i to sink k . The progress speed $Speed_{i,j}^k$ from node i to node j toward the sink k is calculated as:

$$Speed_{i,j}^k = \frac{dist(i, \text{sink } k) - dist(j, \text{sink } k)}{delay(i, j)} \quad (1)$$

Single hop delay is measured at the sender, which timestamps the packet entering the node queue and calculates the round trip single hop delay when receiving the *ACK*. When receiving node j replies to source node i , it inserts the packet transmission time in the *ACK*. Once i receives the *ACK*, it subtracts the transmission time from the arrival time to calculate the RTT (Round Trip Time). Propagation delay is negligibly small.

When we select some nodes to send packets to, we consider a timer in each node. These timers decide to send acknowledge to the upstream node or not. By broadcasting this control packet to all neighbors, if the sender exists in their table, all routing tables will update. One of the most important information that can obtain from an acknowledgment signal is residual energy. When the energy level of a node reaches less than E_{th} , the node will be deleted rapidly from the node routing table.

3.2 Cost function

To select the next node among the alternative neighbors providing positive advancement towards the sink, we use the cost function as follow:

$$Cost(i, j) = \alpha(E_{cost}(i, j)) + \beta(B_{used}(j)) + \gamma\left(\frac{1}{LQ(i, j)}\right) + \delta\left(\frac{1}{E_{res}(j)}\right) \quad (2)$$

Where $E_{cost}(i, j)$ is required energy for sending a packet from node i to node j directly. $B_{used}(j)$ and $E_{res}(j)$ is the used buffer and residual energy of j node. $LQ(i, j)$ is the link quality between (i, j) . α , β , γ and δ are weight factors, and $\alpha + \beta + \gamma + \delta = 1$. EQGR selects a path that satisfies packet

required reliability and that requires low energy and low delay. Link energy cost can be obtained as follows:

$$E_{cost} = \frac{([dist(i, j)]^2)^\theta}{(E_{res}(j))^\omega} \quad (3)$$

Where $dist(i, j)$ is the distance between node i and node j , θ and ω are weighting factors [5]. Link quality is defined as [6]:

$$LQ(i, j) = \left(1 - \frac{1}{2} \exp^{-\frac{\gamma(d)}{2} \frac{1}{0.64}}\right)^{8F} \quad (4)$$

Where F is frame size and $\gamma(d)$ is the signal-to-noise ratio.

3.3 Service differentiation

Multipath routing with Load balancing extends the network lifetime and avoids congestion problems. To select a set of nodes from the N available nodes, the number of required paths is calculated as [7]:

$$paths = x_\alpha \cdot \sqrt{\sum_{i=1}^N p_i(1 - p_i)} + \sum_{i=1}^N p_i \quad (5)$$

Where, P_i corresponds to the probability of successfully delivering a message. Eq. (5) gives the number of required paths with the overall success probability of α . Where x_α [7] is the corresponding bound from the standard normal distribution.

We suppose all nodes are location-aware, so we can estimate hop count h from source node i to the sink as follows:

$$h = \frac{dist(i, \sin k)}{dist(i, j)} \quad (6)$$

This local estimation is based on this assumption that, for each following hop, the geographic progress to the destination will be similar to the current progress. With localized information, we can calculate the deadline for real time packet:

$$Deadline(x) = h \times delay(i, j) \quad (7)$$

We attach this deadline to the event reporting packet. EQGR selects the most desired velocity for a packet x as follows:

$$desiredSpeed(x) = \frac{dist(i, \sin k)}{Deadline(x)} \quad (8)$$

We must choose a neighbor node from a real-time table, the speed of which is equal to or higher than the desired speed. If there is no node inside the routing table that can satisfy the single hop speed, the packet is dropped for reducing the congestion, and a control signal is issued to source node. One of the advantages in our algorithm is that we have two routing tables. Consequently, when we have different timeliness and reliability requirements, EQGR first classifies the packet into the proper required speed based on the end-to-end deadline and Euclidean distance to the destination. Then EQGR finds multiple forwarding nodes from the real-time table that satisfies the packet timeliness requirement. Then we seek these nodes in the other table. After affecting the cost function, we select proper nodes.

3.4 Packet deadline updating

To piggyback the elapsed time, the MAC layer updates the elapsed time $t_{elapsed}$

Just before it actually transmits the packet x as follows:

$$t_{elapsed} = t_d + t_t - t_a \quad (9)$$

Where, t_d is the time when intermediate node transmits the first bit of packet, t_t is the transmission delay which can be computed using the bit rate and the packet length and t_a is arrival time of the packet.

Now, each intermediate node can update the remaining time to deadline as follows:

$$Deadline_x^{new} = Deadline_x^{old} - t_{elapsed} \quad (10)$$

3.5 Queuing manager

Depending on different applications, the traffic in the wireless sensor networks may be mixed with time-sensitive, reliability-sensitive and normal data packets.

To obtain low latency when routing delay-sensitive packets, higher priority should be given to these packets than normal and reliability-sensitive packets. We propose to use three level queues. The packet classifier checks the type of the incoming packets and sends it to appropriate FIFO queues each corresponds to a priority level. The scheduler, schedules queues based on priority of the queues from the highest priority queue to the lowest one. The queue service time is unknown but can follow any general distribution. It is, however, independent of the queue and represented by a unique random variable for all queues. Suppose queues' sizes are high enough to hold all the receiving packets. The model is thus a queuing system of type M/G/1 [8], as shown in Fig. 2.

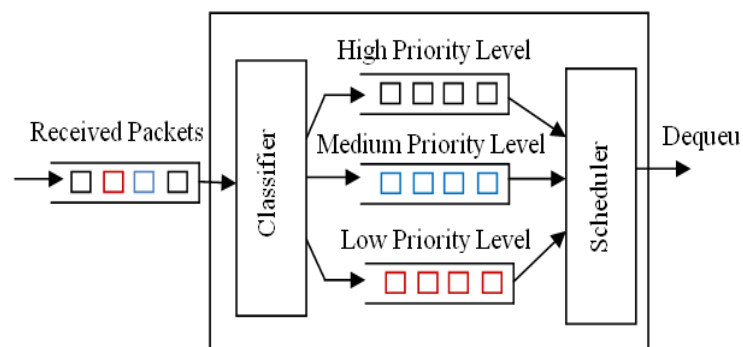


Fig. 2. Queuing Model for EQGR.

This packet classifying strategy prevents a highest priority packet from being delayed by lower priority packets.

4 Performance evaluation

We evaluate the performance of EQGR by comparing it with RTLD and MMSPEED protocols with C++. In our simulation 100 nodes with 20m radio range are randomly distributed in a 400 m × 400 m region with sink node at the upper right corner of the simulation field. Initial energy of each node is 3.3 J, transmission power 1 mw and energy threshold is 0.1 J. Packet

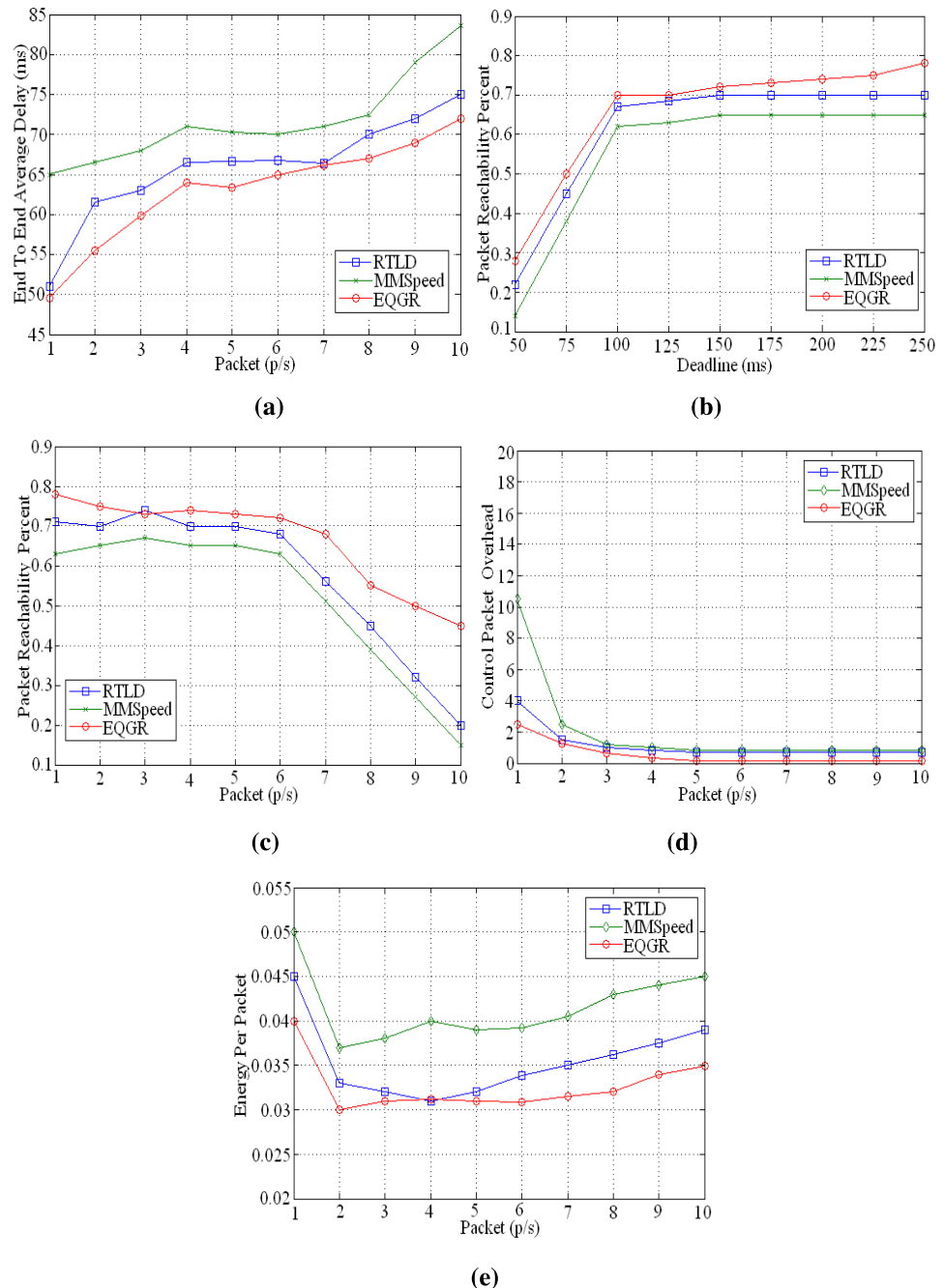


Fig. 3. Simulation results. (a) End-to-end average delay, (b) Packet reachability percentage by different deadlines, (c) Packet reachability percentage by different packet rate, (d) Control Packet Overhead, and (e) Energy consumption per packet.

size is 70 byte and data rate is 200 kbps. The trial with 0.3, 0.2, 0.2 and 0.3 for α , β , γ and δ experiences high performance in term of delivery ratio, power consumption and on-time delivery ratio. For energy cost function we use, $\theta = 1$ and $\omega = 50$.

4.1 Average end-to-end delay

From Fig. 3 (a) EQGR provides a short average delay compared to the other two protocols. Because EQGR considers link quality, required speed and uses a three-level priority queue to preventing a real-time packet from being delayed by non-real-time and control packets.

4.2 Packet delivery ratio

Fig. 3 (b) and 3 (c) shows that EQGR experiences a higher delivery ratio than the other two protocols. Because EQGR considers the residual energy, required speed, and multipath forwarding with the best link quality. In addition, Fig. 3 (c) shows that the packet reaching probability decreases as the packet rate increases. This is due to packet loss, because of high packet traffic and network congestion.

4.3 Control packet overhead

The Fig. 3 (d) shows that EQGR has less control packet. This is due to its neighbor discovery, which does not allow the neighbor node to respond if it is not in the direction of the sink node or has lower energy than E_{th} . Therefore, the probability of packet collision is reduced, and control packet overhead is minimized. The reason for high overhead at the starting point is that extra control packets are used to create the neighbor table.

4.4 Energy consumption

Fig. 3 (e) demonstrates that EQGR consumes less power compared to the other two protocols, because the packet overhead in EQGR is less than in the other two protocols. The reduced energy consumption in EQGR results from distributing the load throughout the neighboring nodes.

5 Conclusion

EQGR decreases the end-to-end delay and increases the reliable data delivery by dynamic load balancing and priority-based data forwarding. Simulation results show that EQGR can significantly improve on-time packet delivery rate, the number of packets that meet end-to-end reliability, and, most importantly, extended network lifetimes.

In future work, we intend to study the void problem in EQGR.