Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 409724, 7 pages
doi:10.1155/2010/409724

Research Article

NQAR: Network Quality Aware Routing in Error-Prone Wireless Sensor Networks

Jaewon Choi,[1] Baek-Young Choi,[2] Sejun Song,[3] and Kwang-Hui Lee[4]

[1] *Department of Computer Engineering, Kyungnam University, 449 Wolyong-dong, Masan, Gyeongnam 631-701, South Korea*
[2] *Department of Computer Science & Electrical Engineering, University of Missouri, Kansas City, MO 64110, USA*
[3] *Department of Engineering Technology & Industrial Distribution, Texas A&M University, College Station, TX 77843, USA*
[4] *Department of Computer Engineering, Changwon National University, 9 Sarim-dong, Changwon, Gyeongnam 641-773, South Korea*

Correspondence should be addressed to Baek-Young Choi, choiby@umkc.edu

Received 1 September 2009; Accepted 21 September 2009

Academic Editor: Benyuan Liu

We propose a network quality aware routing (NQAR) mechanism to provide an enabling method of the delay-sensitive data delivery over error-prone wireless sensor networks. Unlike the existing routing methods that select routes with the shortest arrival latency or the minimum hop count, the proposed scheme adaptively selects the route based on the network qualities including link errors and collisions with minimum additional complexity. It is designed to avoid the paths with potential noise and collision that may cause many non-deterministic backoffs and retransmissions. We propose a generic framework to select a minimum cost route that takes the packet loss rate and collision history into account. NQAR uses a data centric approach to estimate a single-hop delay based on processing time, propagation delay, packet loss rate, number of backoffs, and the retransmission timeout between two neighboring nodes. This enables a source node to choose the shortest expected end-to-end delay path to send a delay-sensitive data. The experiment results show that NQAR reduces the end-to-end transfer delay up to approximately 50% in comparison with the latency-based directed diffusion and the hop count-based directed diffusion under the error-prone network environments. Moreover, NQAR shows better performance than those routing methods in terms of jitter, reachability, and network lifetime.

1. Introduction

Wireless Sensor Networks (WSNs) consist of a large number of battery-powered, low-cost, and tiny sensor nodes, which have the capability of sensing, data processing, and wireless communication. The sensor nodes can be deployed randomly close to or inside the terrain of interest to create a cooperative and self-organizing wireless ad hoc network with minimal provisioning. Unlike the traditional high cost and fixed array of sensor systems, the WSN technology enables countless new applications including environmental hazard monitoring, military surveillance and reconnaissance, and health monitoring applications to name a few.

In WSNs, the sensed data and control messages are exchanged between sensor nodes and the control (sink) nodes relayed by the neighbor sensor nodes via a multihop routing protocol. To build practical services over WSNs, especially considering sensors' limitations in power, computation, and local storage, it is both critical and challenging to support efficient network layer multihop routing protocols. To cope with the characteristics of sensor nodes, various new routing protocols have been proposed in WSNs [1–3]. These protocols are mainly designed (1) to reduce redundant data (data aggregation) and unnecessary controls by using on-demand data centric approaches [4–7], (2) to limit the network scale by using structured approaches such as clustering and hierarchical architectures [8, 9], and (3) to decrease distributed state overheads by using location based approaches [10, 11]. To achieve an efficient resource usage, those routing protocols commonly select routes based upon the static quality factors such as maximum power availability, minimum energy usage, maximum position progress, minimum hop count, or the shortest arrival latency. However, those static quality based parameters have limitations in case of the error-prone and densely deployed WSNs, because they do not take retransmissions due to packet losses and backoffs due to collisions into consideration.

In this paper, we propose a network quality aware routing (NQAR) mechanism to provide an enabling method of the delay-sensitive data delivery over error-prone and densely deployed WSNs. The proposed scheme adaptively utilizes the dynamic network quality factors including link error rates and collision histories. It is designed to avoid the paths with potential noise and collision, which may cause many nondeterministic retransmissions and backoffs. NQAR uses a data centric on-demand method based on a directed diffusion to estimate the minimum cost end-to-end routing path. The NQAR operation steps are as follows. First, each sensor node maintains its network quality information including the packet loss rate, the retransmission rate, and the backoff rate for a certain period. Second, during the interest dissemination period, each node relays interest with its network quality information to its neighbors. Third, each node estimates a single-hop delay based on processing time, propagation delay, packet loss rate, number of backoffs, and retransmission timeouts between two neighboring nodes, which in turn enables the calculations of expected end-to-end delays during the interest dissemination period. Finally, a source node can send delay-sensitive data to a sink node along the shortest expected end-to-end delay path. It is clearly noted that the proposed scheme simultaneously considers the dynamic qualities of wireless network links in addition to the overall static parameters including per-hop processing time and power in the routing decision process.

To the best of our knowledge, NQAR is the first work to simultaneously consider the qualities of wireless links as well as processing time in the routing decision process. We perform extensive simulations under the various qualities of links and show that the NQAR reduces the end-to-end transfer delay up to 50% in comparison with the latency-based directed diffusion [5] and the hop count-based directed diffusion [7] under the error-prone (link error and collision) network environments. Moreover, NQAR performs better than other routing methods in terms of jitter. Since NQAR inherently avoids error-prone links, the reachability (reliability) is improved as well if no packet retransmission is assumed. We also find that NQAR prolongs the network lifetime as it prevents unnecessary energy consumption, resulting from the relative reductions of packet losses and retransmissions.

The remainder of this paper is organized as follows. Section 2 summarizes a background and the related work to this research. Section 3 explains the details of the NQAR algorithm and Section 4 presents the evaluation results. Finally, we conclude our work in Section 5.

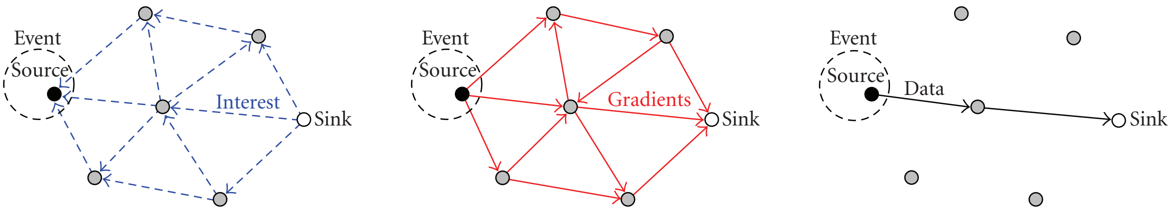
2. Background and Related Work

Routing protocols in WSNs have been designed as power efficient, data-centric, and cooperative approaches to address its unique characteristics (i.e., resource limitations). Sensor Protocols for Information via Negotiation (SPIN) [4] is one of the earliest data centric approaches, which allows any sensor nodes around the information data to initiate interest advertisements. Before sending the real data, it starts negotiation with the collected data description (meta-data). It achieves energy efficiency by reducing redundant data transmission (data fusion). However, if the sensor nodes around the information data are not interested in that data, the sensor node initiated advertisement mechanism cannot ensure the delivery of the information data. Directed diffusion (DD) [5, 6] is one of the most popular data centric approaches, which starts the interest (a network task description) dissemination from the sink nodes. During the interest propagation, each sensor node keeps a time stamped interest to establish the gradients from the data source back to the sink. When the source has data to send, it transmits the data along the lowest latency path. It is energy efficient in that the message propagation and aggregation are done between neighbors. Our proposed routing scheme is based on the directed diffusion, and a simplified schematic of directed diffusion is depicted in Figure 1. It is also better than SPIN from the data coverage point of view. However, it has additional overhead on sensor nodes to handle the control information and does not work well with time-sensitive or continuous data delivery applications due to its interest dissemination model. It also does not consider the global energy-balancing to increase network lifetime.

Energy-efficient Differentiated Directed Diffusion (EDDD) [7] is an extension of the directed diffusion protocol to establish a path between a source and a sink with the minimum hop count and the minimum available energy to enhance the shortcomings of the original directed diffusion. However, both directed diffusion and EDDD do not reflect the error-prone (noise and collision) network link characteristics of WSNs [1]. It causes nondeterministic additional delays due to retransmission and/or reprocessing in the MAC layer.

Low Energy Adaptive Clustering Hierarchy (LEACH) [8] is introduced to achieve an energy efficiency to arrange a structured traffic path by forming clusters. Only a few representative nodes (cluster heads) are involved in the cluster control (assigning transmission time for each sensor node: TDMA) and data transmission (including data aggregation). To support equal energy dissipation, the cluster head roles are evenly alternated among the sensor nodes. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [9] is a network lifetime enhancement work over LEACH protocol. It reduces communication overhead by arranging local coordination among the neighboring sensor nodes and by chaining the communication path to the sinks instead of the cluster formation. Nodes need only to communicate with their neighbors and they take turns in communicating with the sink.

Greedy Perimeter Stateless Routing (GPSR) [10] is an earlier version of the location-based geographic routing protocol. It decreases the number of distributed states and the maintenance overheads by calculating the next forwarding node based only upon the destination location information on each forwarding node. It chooses a routing path according to the best position progression towards the destination. However, it will need an additional location service to map positions and node IDs. Geographic and Energy Aware Routing (GEAR) [11] adds an energy parameter to the geographical progress parameter in calculation of the best



(a) Interest propagation (b) Initial gradients set up (c) Data delivery along reinforced path

Figure 1: Simplified schematic of directed diffusion.

destination path. It refines the next estimated progression cost with the learned cost, which is the feedback information of the previously propagated packet cost to the destination. It also reduces interest dissemination to a certain region to conserve more energy.

All of the above methods, however, use only static factors, and nondeterministic delays due to retransmissions and backoffs that have not been taken into consideration. NQAR is unique in that it uses dynamic network quality parameters as well as static delays in order to estimate the expected event-to-sink path delay.

3. Network Quality Aware Routing (NQAR) Algorithms

An important observation is that packets may be lost due to channel problems such as interference and collision. Then the link layer retransmission is performed after a packet loss is detected, and the time necessary to detect a packet loss is at least twice as much as one-way propagation delay. Furthermore, there can be repeated packet loss and retransmission attempts. Therefore, the problem of selecting a path with the shortest end-to-end propagation delay or minimum hop count is that the performance is significantly affected by packet loss rates of links. In such a case, the existing methods will undergo the additional delays that are not presupposed, and may fail to transmit time-critical data successfully.

We first describe our approach of a path selection that is based on directed diffusion, but takes packet loss rate into account in link costs. We then discuss how the cost can be used to meet various parameters such as delay and energy consumption.

Using the packet loss rate from a node i to its neighboring node j ($r_{\text{loss}(i,j)}$), the expected cost to successfully transmit a packet from i to j ($\text{Cost}_{(i,j)}$) is estimated by (1). $c_{\text{init}(i,j)}$ and $c_{\text{retrn}(i,j)}$ are the initial packet transmission costs, and the additional cost to retransmit the packet in case of packet loss from i to j, respectively. Note that $c_{\text{retrn}(i,j)}$ is multiplied by the expected number of retransmissions given the link error rate. Nodes maintain the local packet loss rates and propagation delays from themselves to their neighboring nodes. This packet loss rate is the recent average of packet losses on a link, and can be maintained by each node in the MAC layer [12],

$$\text{Cost}_{(i,j)} = c_{\text{init}(i,j)} + \frac{r_{\text{loss}(i,j)}}{1 - r_{\text{loss}(i,j)}} \cdot c_{\text{retrn}(i,j)}. \tag{1}$$

Then a node receiving the interest packet i can estimate the end-to-end cost from itself to the sink node 0, ($\text{Cost}_{(i,0)}$) by the sum of hop costs along the path as shown in the following equation:

$$\text{Cost}_{(i,0)} = \sum_{k=1}^{i} \text{Cost}_{(k,k-1)}. \tag{2}$$

The nodes with data matching the query in the interest packet flooded by the sink node become the sources. A source node selects a path (gradient) whose $\text{Cost}_{(i,0)}$ value is the smallest among the $\text{Cost}_{(i,0)}$s of all the directed gradients it is maintaining. Equation (1) can be applied for various considerations. For example, if the expected number of (re)transmissions is considered, we can set $c_{\text{init}(i,j)}$ and $c_{\text{retrn}(i,j)}$ to be 1 as follows:

$$\text{Cost}_{(i,j)} = \frac{1}{1 - r_{\text{loss}(i,j)}}. \tag{3}$$

This enables us to select a path with the smallest number of (re)transmissions. The small number of (re)transmissions is important as excessive retransmissions cause energy depletion early. Moreover, the different number of packet retransmissions over an error-prone link/path adversely affects jitter, in addition to delay. We can also compute the expected energy consumption of a packet over the link between node i and j, by assigning an energy usage $c_{\text{eng}(i,j)}$ to $c_{\text{init}(i,j)}$ and $c_{\text{retrn}(i,j)}$ from (1), as below,

$$\text{Cost}_{(i,j)} = \frac{c_{\text{eng}(i,j)}}{1 - r_{\text{loss}(i,j)}}. \tag{4}$$

Now we focus on the expected end-to-end delay. Let us consider the time necessary for node i to successfully transmit a packet to its neighboring node j. First, the delay should include a packet processing time (t_{prcs}) and propagation time (t_{prop}). Once a packet loss occurs, the packet should be retransmitted after a retransmission timeout (t_{rto}) that is typically set as multiples of the propagation delay and is greater than the round trip time. In this case, the packet should be processed again at node i. Thus, as for 1 hop delay, the initial cost ($c_{\text{init}(i,j)}$) in (1) would be the sum of the processing time ($t_{\text{prcs}(i,j)}$) and propagation delay ($t_{\text{prop}(i,j)}$), and the additional cost ($c_{\text{retrn}(i,j)}$) will become the sum of the processing time ($t_{\text{prcs}(i,j)}$) and retransmission timeout ($t_{\text{rto}(i,j)}$), which leads to the following equation:

$$\text{Cost}_{(i,j)} = \left(t_{\text{prcs}(i,j)} + t_{\text{prop}(i,j)}\right) + \frac{r_{\text{loss}(i,j)}}{1 - r_{\text{loss}(i,j)}} \cdot \left(t_{\text{prcs}(i,j)} + t_{\text{rto}(i,j)}\right). \tag{5}$$

There are several factors involved in a packet processing time (t_{prcs}). When a packet needs to be sent, a clear channel assessment (CCA) is first needed. Then possible queueing and actual transmission times are required. Thus, the processing time (t_{prcs}) is the sum of the channel assessment time (t_{chnasm}), transmission time (t_{trn}), and queueing time (t_{queue}) as follows:

$$t_{\text{prcs}(i,j)} = t_{\text{chnasm}(i,j)} + t_{\text{trn}(i,j)} + t_{\text{queue}(i,j)}. \tag{6}$$

Channel assessment time (t_{chnasm}) varies depending on the channel condition. When a node has a packet to send, its microcontroller observes whether the channel is clear or not. If the channel is clear, the microcontroller signals the radio component such as CC2520 [13] to send out the packet. Otherwise, it will back off for some time (t_{chnbck}) and then test the channel again. The problem is that whether the channel is clear or not is nondeterministic. Thus, we predict the channel condition by the history of backoffs. A microcontroller can count up the number of channel backoffs and calculate the rate of channel busy to channel clear that is referred to as the channel backoff rate ($r_{\text{chnbck}(i,j)}$) in this paper. That is, a channel backoff rate represents the possibility of channel failure in the node. Thus the channel assessment time is estimated by (7). Transmission time (t_{trn}) is deterministic in nature given a packet size and link bandwidth,

$$t_{\text{chnasm}(i,j)} = \frac{r_{\text{chnbck}(i,j)}}{1 - r_{\text{chnbck}(i,j)}} \times t_{\text{chnbck}(i,j)}. \tag{7}$$

By (5)–(7), we can calculate an expected transfer delay of a link. $\text{Cost}_{(i,j)}$ in (5) then means the sum of the expected delays in successfully transferring data over 1 hop considering channel assessment time, transmission time, propagation delay, packet loss rate, and retransmission timeout. This in turn enables us to compute the expected end-to-end delay of a path via (2). Note that NQAR requires that each sensor should estimate and maintain its network quality information such as link loss rate and channel loss rate. Exponentially weighted moving average algorithm can be used for the estimations, and the memory and computation overhead involved is small. In summary, our route selection approach effectively takes into account of dynamic as well as static qualities of link and channel for the performance of delay, jitter, and energy consumption that are analyzed in Section 4.

4. Validation and Performance Study

In this section, we first validate the computation of expected delay with a simple network topology with three paths. We then conduct more extensive simulations to compare the performance results of NQAR with the ones from the original latency-based directed diffusion [5] and the hop count-based directed diffusion EDDD [7]. The performance metrics used include the end-to-end delays, jitters, reachability (thus reliability), and network lifetime.

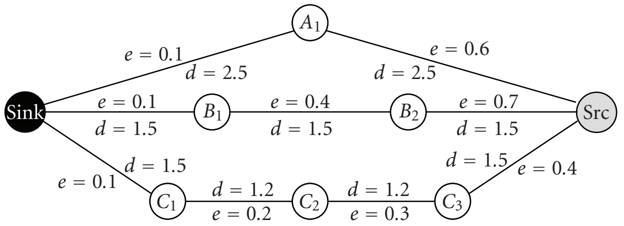


FIGURE 2: A simple topology with three paths.

4.1. Validation. For a simple and concise discussion, we consider error rates (e's) and propagation delays (d's) only in the example network shown in Figure 2, assuming all other delays are identical. In the figure, there are three paths from Src node to Sink node: $P_A(\text{Src} - A_1 - \text{Sink})$, $P_B(\text{Src} - B_2 - B_1 - \text{Sink})$, and $P_C(\text{Src} - C_3 - C_2 - C_1 - \text{Sink})$. There can be transmission failures on links, and link error rates may change over time. Every node maintains only the link error rates and propagation delays with regard to the information from itself to its neighboring nodes.

Let us consider the least hop count path, the least end-to-end latency path, and NQAR route selection scheme. Note that in the example network shown in Figure 2, the least hop count path is P_A, but the transmission delay of path P_A (5.0) is longer than that of the least latency path P_B (4.5). Meanwhile, NQAR selects the path P_C which has the minimum expected delay considering loss rates along the path. By (2) with the specified loss rate and propagation delays, the expected end-to-end transfer delays for three paths in Figure 2 are $C_{P_A} \approx 25.14$, $C_{P_B} \approx 27.83$, and $C_{P_C} \approx 15.31$, respectively. It is interesting that the expected end-to-end transfer delay of the path P_C (C_{P_C}) is the shortest, although the path P_C represents the largest hop count and the longest propagation delay among three paths. To verify this, we performed the experiment of transmitting 1,000 packets from Src node to Sink node over the paths P_A, P_B, and P_C, separately. Figure 3 shows the average delays from this experiment. Two bars on each path indicate the delay expected by (1) and (2), and the actual delay experienced by packets, respectively. It shows that the estimated delay matches the measured delay, thus validates our expected delay computation.

4.2. Simulation Setup. For the more extensive evaluations, we implemented three interest dissemination and gradient generation algorithms of the directed diffusion. The simulations were conducted with 10 × 10 grid network topology. Network conditions including packet loss rates and channel backoff rates are randomly allocated within the range of error rate parameter configuration. For example,

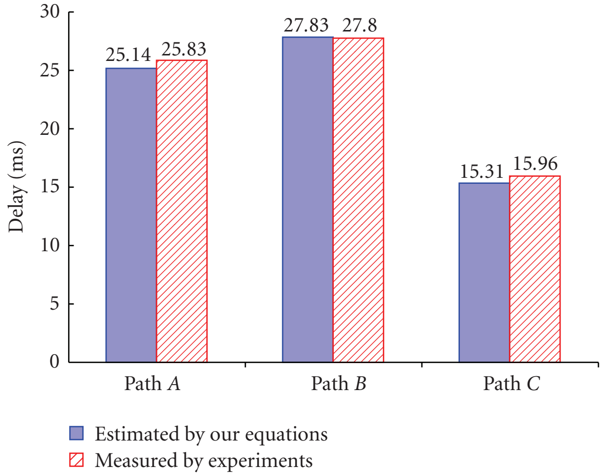


Figure 3: Estimated delay versus measured delay.

Table 1: Configuring of experiment parameters.

Item	Value
Number of nodes	100
Initial energy of each node	1J
Transmitting power	77.4 mW
Receiving power	55.5 mW
Packet size	1,024 bits
Inter-arrival time	100 ms
Bandwidth	250 Kbps

if the range of error rate is 20%, random rate values from 0% up to 20% are assigned to the packet loss rate and the channel backoff rate, respectively. Propagation delays are proportional to the distance between two sensor nodes. We also assume that the retransmission timeout is four times as long as the one-way propagation delay. Each sensor node maintains the history of packet loss rates and propagation delays, and a channel backoff count. As shown in Table 1, we use various simulation parameters the same as the sensor motes implementation by using CC2520 [13] chipset specification. We set the channel backoff time the same as the TinyOS setting that is 6.6 ms. Transmission time is 4.0 ms, resulting from dividing packet size (1,024 bits) by bandwidth (250 Kbps).

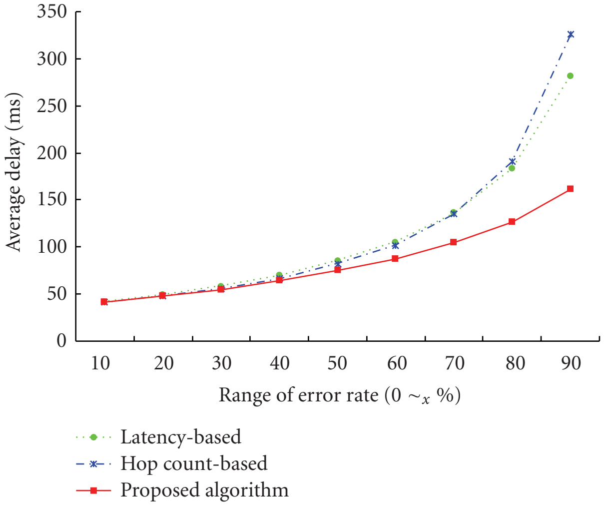


Figure 4: End-to-end delay.

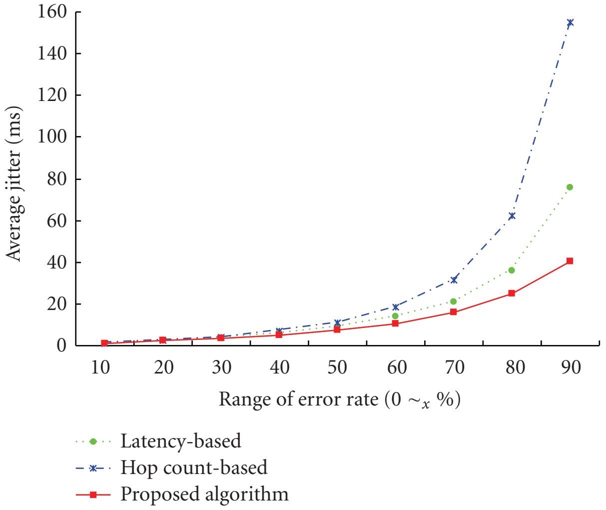


Figure 5: Average jitter.

4.3. Evaluation Results. Figure 4 shows the end-to-end transfer delays of the three routing protocols under different range of error rates. The presented values are the average delays of 1,000 packet transmissions. In Figure 4, we observe that the improvement of delay values of NQAR becomes significant according to the increments of the range of error rates. For example, average delay of NQAR improves by approximately 43% in comparison with the latency-based protocol and by about 51% compared with the hop count-based protocol under 90% of the range of error rates. Usually, if error rates are high, retransmissions and channel backoffs happen more frequently. The results clearly indicate that NQAR works very well, especially in error-prone networks. Figure 5 presents the average jitter values of the three routing protocols under different range of error rates. The values are measured by the standard deviation of the end-to-end delays. NQAR has smaller average jitters and increases the average jitters slower than other protocols along the increments of the range of error rates. For example, average jitter of NQAR is smaller than that of the latency-based protocol by about 47% and that of the hop count-based protocol by approximately 74% under 90% of the range of error rates. The hop count-based protocol shows the worst average jitter values, because it has less consideration on various timing conditions related to the dynamic link quality. The results indicate that NQAR will work better for the various applications time-critical data transmissions and jitter-sensitive streaming data transmissions.

Figure 6 presents the event-to-sink reachability of the three routing protocols under different range of error rates.

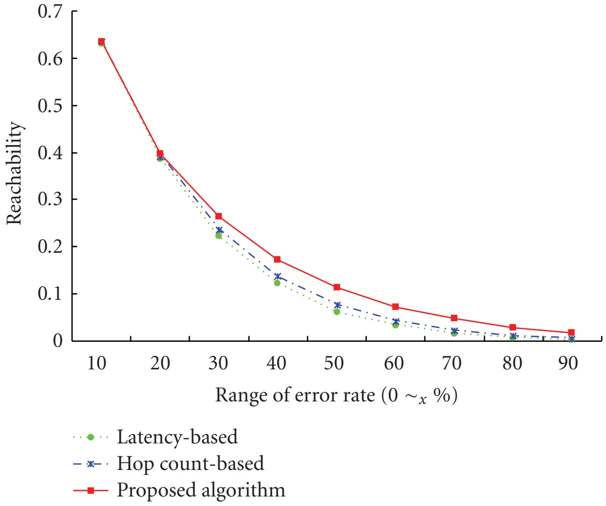


Figure 6: Event-to-sink reachability.

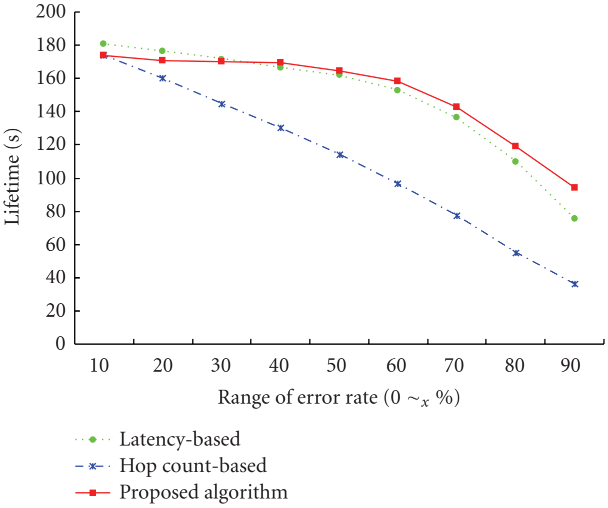


Figure 7: Network lifetime.

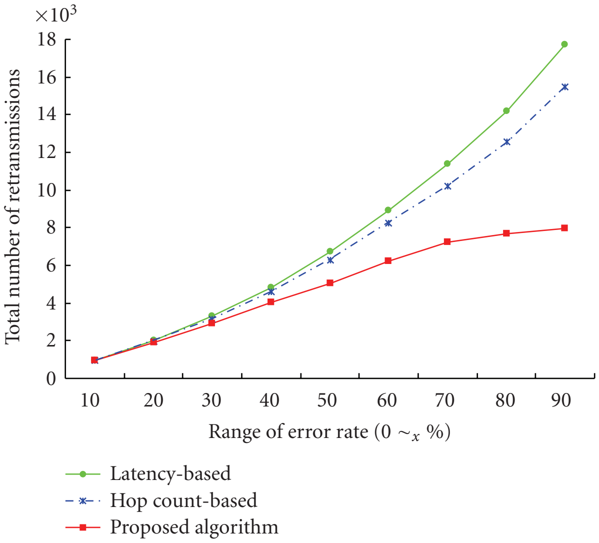


Figure 8: Total number of retransmissions.

The event-to-sink reachability is the ratio of the number of packets arrived at the sink node to the number of packets sent by a source node after assuming that no sensor node performs packet loss detection or retransmission. It is very important to choose a reliable routing path considering the overheads of the loss recovery mechanisms (packet loss detection and retransmission) that include memory overhead for caching, delay increase by retransmission, and additional energy consumption due to memory access and retransmission. As illustrated in Figure 6, NQAR has higher event-to-sink reachability than the other protocols. It is because NQAR protocol selects a path with a lower probability of packet losses, as in (1). Figure 7 exhibits the network lifetimes of the three routing protocols under different range of error rates. We define network lifetime as the operation time until the first sensor node runs out of the energy. From a routing protocol point of view, the network lifetime can be prolonged by reducing the number of unnecessary packet (re)transmissions and by spreading traffic equally over the network. It shows that the network lifetimes of NQAR are longer and the decrements are smoother than the other protocols along the increments of the range of error rates. It is because NQAR efficiently reduces unnecessary retransmissions by selecting a less error-prone routing path. We further analyze the impacts of routing protocols on the network lifetime with the number of retransmissions and standard deviations of the number of transmissions. Figure 8 shows the number of retransmissions of the three routing protocols under different range of error rates. NQAR has much fewer number and slower increments of retransmissions than the other protocols along the increments of the range of error rates, since it takes the link quality into consideration. Compared to the latency-based protocol, the hop count-based protocol has fewer retransmissions, which is because the shorter distance path has the less chance of packet losses. Figure 9 shows the standard deviation of the packet transmissions by each node. The smaller standard deviation means packets were sent evenly by each node. Although it shows similar traffic distribution trends, NQAR has smaller standard deviations than the hop count-based protocol and larger standard deviations than the latency-based protocol. It is because the hop count-based protocol may have a little more chance of selecting a certain node in the diagonal direction path to the sink node.

According to both Figures 8 and 9, for the given similar traffic distribution trends, it is clear that NQAR prolongs the network lifetime in error-prone networks by reducing the chance of retransmissions.

5. Conclusions

We have proposed a network quality aware routing (NQAR) protocol for error-prone and densely deployed WSNs. In addition to the existing routing methods that select routes

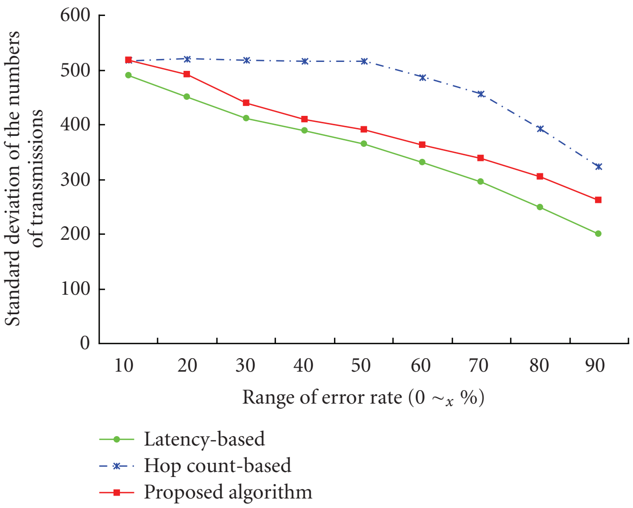


Figure 9: Standard deviation of the packet transmissions.

with the least energy cost, the shortest arrival latency, or the minimum hop count, NQAR adaptively utilizes the network qualities including link error rates and collision histories in the route selection. It is designed to avoid the paths with potential noise and collision that may cause many nondeterministic delays due to backoffs and retransmissions. NQAR uses a data centric approach to estimate a single-hop delay based on processing time, propagation delay, packet loss rate, the number of backoffs, and retransmission timeouts between two neighboring nodes. This in turn enables the source to select the shortest expected end-to-end delay path to send data. NQAR is unique in that it holistically considers the qualities of wireless links as well as processing time in the routing decision process. Through extensive simulations, we have validated that NQAR improves the end-to-end transfer delay performance and decreases jitter significantly under the error-prone (link error and collision) network environments. We have shown that NQAR increases end-to-end reachability (reliability) in case of no data retransmission, because of its inherent nature of avoiding error-prone links. We have also found that NQAR prolongs the network lifetime, as it prevents unnecessary energy consumption resulting from the relative reductions of packet losses and retransmissions.

Acknowledgments

This work was supported in part by the US National Science Foundation under Grant no. 0729197. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US National Science Foundation.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–27, 2004.

[3] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[4] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM ’99)*, pp. 174–185, Seattle, Wash, USA, August 1999.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00)*, pp. 56–67, Boston, Mass, USA, August 2000.

[6] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[7] M. Chen, T. Kwon, and Y. Choi, “Energy-efficient differentiated directed diffusion (EDDD) in wireless sensor networks,” *Computer Communications*, vol. 29, no. 2, pp. 231–245, 2006.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS ’00)*, p. 223, Maui, Hawaii, USA, January 2000.

[9] S. Lindsey and C. S. Raghavendra, “Pegasis: power-efficient gathering in sensor information systems,” in *Proceedings of the IEEE Aerospace Conference*, vol. 3, pp. 1125–1130, Big Sky, Mont, USA, March 2002.

[10] B. Karp and H. T. Kung, “Gpsr: greedy perimeter stateless routing for wireless sensor networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00)*, pp. 243–254, Boston, Mass, USA, August 2000.

[11] Y. Yu, D. Estrin, and R. Govindan, “Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks,” Tech. Rep. UCLA-CSD-TR-01-0023, UCLA Computer Science Department, May 2001.

[12] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural, “Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’05)*, vol. 4, pp. 2646–2657, Miami, Fla, USA, March 2005.

[13] Texas Instruments, “Cc2520 datasheet,” http://focus.ti.com/docs/prod/folders/print/cc2520.html.