Real-Time Routing Protocols for Wireless Sensor Networks

Yanjun Li, Chung Shue Chen, Ye-Qiong Song

Real-Time Routing Protocols for Wireless Sensor Networks

Yanjun Li[‡§], Chung Shue Chen[*§], Ye-Qiong Song[§†]
[‡]State Key Laboratory of Industrial Control Technology, Zhejiang University, China
[§]LORIA-Nancy University, Viller Les Nancy, France
[*]Norwegian University of Science and Technology, Norway
Email: yanjun.li@loria.fr, cschen@iet.ntnu.no, song@loria.fr

Abstract

Developing real-time routing protocols under energy constraint is one of the key points for providing end-to-end delay guarantee in multi-hop wireless sensor networks. In this paper, we give, at first, an overview of the existing real-time routing protocols and point out some potential approaches to improve them. To enhance existing protocols, one way is to make routing decision based on multi-hop rather than 1-hop neighborhood information. We study the asymptotic performance of a generic routing metric as the quantity of information a priori increases and propose then a 2-hop neighborhood information based real-time routing protocol. As an example, the approach of mapping packet deadline to a velocity is adopted as in SPEED; however, our routing decision is made based on the 2-hop velocity. An energy efficient probabilistic drop is proposed to improve energy utilization efficiency. When packet deadline requirement is not stringent, a design is integrated to release nodes from heavy consumption. Energy balance over nodes is thus improved. Simulation results show that, compared with protocol SPEED that only utilizes 1-hop information, the proposed scheme leads to lower deadline miss ratio and higher energy efficiency.

1. Introduction

For emerging applications of wireless sensor networks (WSN) in surveillance, industrial control, medical care and inventory tracking systems, real-time quality-of-service (QoS) is desired as these applications are often in nature time sensitive. Different from existing best-effort service which may not have stringent packet timeliness requirement and can tolerate a significant amount of packet loss, these real-time (RT) applications are much more demanding. Out-of-date data are often irrelevant and may even lead to negative impacts to the system control and performance [1,2]. QoS and timeliness guarantee in WSN is therefor favorable in newly required RT service.

Supporting real-time QoS in WSN can be addressed from different layers and domains [2]. For example, the medium access control (MAC) is capable of supporting channel access delay guarantee but in single-hop manner, while a routing protocol can help to provide end-to-end or multi-hop transmission guarantee in network layer. Cross-layer optimization has a potential of some further improvements. Recently, in-network data aggregation strategy has attracted more and more attentions in complementing routing protocols for reducing data redundancy and alleviating network congestion. Meanwhile, proper middleware design will help to bridge application and lower layers efficiently so as to support system abstraction and mutual coordinations. In this paper, we will focus on routing protocol which has always played a very significant role in supporting end-to-end QoS.

Considering system simplicity, most existing routing protocols utilize 1-hop neighborhood information. It is potential that multi-hop information may lead to improved performance in issues such as routing, message broadcasting, and channel access scheduling [3–5]. It is very likely that a system can perform better if more information is available and effectively utilized. Here, we first study the asymptotic performance of a generic routing decision when the quantity of information a priori increases. Based on the preliminary results, a 2-hop information based RT routing protocol is proposed and its improvement is shown over 1-hop protocol such as SPEED [6]. The choice of two hops is a tradeoff between performance improvement and complexity. The following work will focus on how to use or integrate the 2-hop information effectively so as to improve both energy and real-time performance.

The rest of the paper is organized as follows. Section 2 discusses related routing protocols in WSN. Section 3 studies a generic routing metric and shows its performance as information a priori increases. Section 4 presents the design of a 2-hop information based routing protocol. Simulation and comparison are reported. Meanwhile, potential enhancement is addressed. Finally, Section 5 concludes the paper.

2. An overview of RT routing protocols

Real-time service has been considered in some existing WSN protocols. Akkaya and Younis [7] have proposed an

[*]This work was partially carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.
[†]Corresponding author.

energy-aware QoS routing protocol which finds energy-efficient path and by which the end-to-end delay requirement can be met. It is suggested to have a classifier in each node to check incoming packet type and divert traffics into different RT and non-RT priority queues. The delay requirement is converted into bandwidth requirement. By using an extended version of Dijkstra's algorithm, the protocol finds a list of least cost paths and picks a path from the list which can meet the delay requirement.

In [8], Ergen *et al.* present an energy efficient routing method with delay guarantee for WSN. They first exclude the delay constraint and formulate the lifetime maximization as a linear programming (LP) problem aiming to determine optimal routing path and maximize the minimum lifetime of nodes. The result of LP is first implemented in a centralized way and then approximated by a distributed iterative algorithm. Then, delay guarantee is included by limiting the length of node-to-sink routing path.

In [9], Boughanmi and Song have proposed a routing metric for evaluating path efficiency. It is defined by the ratio of energy efficiency to end-to-end delay, where the energy efficiency is specified by considering link failure and packet retransmission. The end-to-end delay is defined by the hop count between source and sink, collected by routing response messages in initialization phase. The new routing metric is applied in AODV routing protocol with IEEE 802.15.4 [10] MAC sublayer. Result shows that it can improve the network lifetime and end-to-end delivery ratio when compared to traditional AODV and the metric in [8].

Pothuri *et al.* [11] design a heuristic solution to find energy-efficient path for delay-constrained data in WSN. A set of paths between source and sink nodes are identified and indexed in the increasing order of their energy consumption. End-to-end delay is estimated along each of the ordered paths and the one with the lowest index that satisfies the delay constraint is selected. Their proposed framework achieves a good balance between latency and energy consumption. However, the solution is based on the assumption that nodes are equipped with two radios: a low-power radio for short-range and a high-power radio for long-range communication such that each node can reach the sink directly using its long-range radio.

ZigBee alliance [12] defines the network and application layers on the top of physical and MAC layer standardized by IEEE 802.15.4. The network layer uses a modified AODV by default and hierarchical tree routing (HTR) as last resort. In [13], Nefzi and Song have analyzed and compared the performance of AODV and HTR in terms of end-to-end delay and energy consumption. It is found that the network with HTR has smaller average end-to-end delay and longer lifetime than that with AODV. However, AODV does better in end-to-end delay in the worst case performance. Besides, the energy consumption of AODV is more uniformly distributed. An improvement is made by using a neighborhood table in routing decision in order to improve the worse-case delay in HTR by shortening the worst-case routing path. To some extent, end-to-end hop number implies end-to-end delay. Thus, HTR is suitable in guaranteeing the delay time by simply measuring the hop count from source to sink.

In addition, several RT routing protocols use velocity assignment policy, including SPEED [6]. The packet deadline is mapped to a velocity in terms of the distance to destination. A packet is forwarded by a node if it can meet the required velocity. When there is no neighbor node which can meet the requirement, the packet is dropped probabilistically while regulating the workload. Back-pressure packet re-routing in large-delay link is carried out to switch and reduce packets directed to a congested region. In [14], MM-SPEED extends SPEED. It provides multiple delivery velocities for packets with different deadline requirements for supporting different QoS. RPAR [15] offers another improved version of SPEED. The required velocity is based on the progress towards the destination and the packet's remaining time before the deadline. A node will dynamically adjust its transmission power to meet the required velocity in the most energy-efficient way. If no node can meet the velocity, the transmission power will be adjusted to attempt a new discovery. It is worth noting that all the above protocols are based on 1-hop neighborhood information.

In our proposed scheme to be described in Section 4, we also adopt the approach of mapping packet deadline to a velocity. The concept has been shown effective in [6, 14, 15]. However, our routing decision is made based on 2-hop neighborhood information and the corresponding metrics. It is therefore named as Two-Hop SPEED (TH-SPEED) here. For computing 2-hop neighborhood information in wireless ad hoc and sensor networks, some distributed algorithms and efficient information exchange schemes are reported in [16,17]. For a network of n nodes, the complexity analysis presented in [16] has shown that every node can obtain the knowledge of 2-hop neighborhood by a total of $O(n)$ messages, while each message has $O(\log n)$ bits.

3. Multi-hop information based routing and metric: a case study

In this section, we report the performance comparison of a 1-hop, 2-hop, and 3-hop neighborhood information based routing metric, aiming to have a general idea of how much the performance can be improved provided that one can have more routing information a priori. The new routing metric considers both the advance in distance and link quality, incorporated in the routing decision at each hop. The performance based on 1-hop information is compared to those with multi-hop information.

3.1. Analytical link model

In the study, we adopt the lossy WSN link layer model derived in [18], which is built on aggregate statistical measures for realistic time-varying channels. The packet reception rate (PRR), $0 \leq p \leq 1$, of a wireless link is modeled as:

$$p(d) = \left(1 - \frac{1}{2}\exp\left(-\frac{\gamma(d)}{2}\frac{1}{0.64}\right)\right)^{8f} \quad (1)$$

where d is the transmitter-receiver distance, $\gamma(d)$ is the signal-to-noise ratio (SNR), and f is the frame size[1], with respect to Mica2 Motes [19] in standard non-coherent FSK. This model takes into account both distance-dependent path loss and log-normal shadowing in characterizing the wireless link. For transmitting power P_t,

$$\gamma(d)_{dB} = P_{t\,dB} - PL(d)_{dB} - P_{n\,dB} \quad (2)$$

where $P_{t\,dB}$ is set to 0 dBm while the noise floor $P_{n\,dB}$ is at -115 dBm in reference to Mica2 radios. The path loss $PL(d)_{dB}$ will adopt the following commonly used model, e.g. [18], with respect to the channel statistics in [20]:

$$PL(d)_{dB} = PL(d_0)_{dB} - 10n\log_{10}(d/d_0) + X_{\sigma\,dB} \quad (3)$$

where n denotes the path loss exponent, d_0 is the reference distance (at 1 meter), while X_σ has a log-normal distribution with zero mean and variance σ^2. In reference to [18, 21], we adopt the "sandy flat beach" model [20] to simulate common WSN application environment. Performance in other environment models is also evaluated. Results have shown similar tendency and support same conclusion. Due to the scope of this paper, we only present those obtained from the sandy flat beach model with $n = 4$, $\sigma = 4$, and $PL(d_0) = -40.8$ dBm. Following [18], the frame size f is set as 50 bytes.

3.2. Forwarding metric analysis

As reported in [21], the product of PRR and the distance progress towards destination is a more effective forwarding decision metric than a purely distance based routing algorithm in systems of time-varying links. For instance, packet loss also occurs in forwarding between nodes in short distance due to the fact of link uncertainty while $0 < p(d) < 1$. Instead of simply taking the product of PRR and the distance progress as decision metric, we consider the following new metric. Some technical definition are required in advance.

For each node i, $\mathbf{N}(i)$ is used to denote the set of its direct neighbors. $\mathbf{F}(i)$ is used to denote the set of node i's potential forwarders which will make a progress towards the destination. In other words,

$$\mathbf{F}(i) \triangleq \{j | d(i,D) - d(j,D) > 0, j \in \mathbf{N}(i)\}. \quad (4)$$

Moreover, $\mathbf{F_2}(i)$ is used to denote the set of forwarding nodes in 2-hop. Consequently,

$$\mathbf{F_2}(i) \triangleq \{k | d(j,D) - d(k,D) > 0, j \in \mathbf{F}(i), k \in \mathbf{N}(j)\}. \quad (5)$$

Remarks:

1. The distance $d_{i,j}$ between pair nodes i and j is assumed measurable.
2. The PRR, $p(d_{i,j})$, of each distance $d_{i,j}$ is modeled and given by (1). The PRR statistics are assumed known, including path loss exponent n, log-normal shadowing standard deviation σ, and the path loss reference $PL(d_0)$.
3. The progress from node i to its candidate forwarder j is identified by its expected distance, $\overline{d}_{j,D}$, to the destination node (says, D), when chosen:

$$\overline{d}_{j,D} \triangleq p(d_{i,j})d_{j,D} + (1 - p(d_{i,j}))d_{i,D}. \quad (6)$$

   This can also be interpreted as the expected position after 1-hop routing despite the fact that a packet cannot be buffered at the expected point. However, $\overline{d}_{j,D}$ can serve as a metric for choosing the suitable one.
4. The node among $\mathbf{F}(i)$ with smallest $\overline{d}_{j,D}$ is selected as the packet forwarder, denoted by $j*$.
5. The packet is forwarded to node j^*. However, this does not mean the transmission is surely successful since the PRR between nodes i and j^* could be smaller than 1. In general, $0 < p(d_{i,j^*}) < 1$. Thus, it has a success rate of $p(d_{i,j^*})$ when sending a packet from i and j^* in the coming forwarding. If a transmission fails due to this link uncertainty, the buffered packet at current node i will initiate a new forwarder selection based on new or updated network topology and link reliability. One may consider it as an ARQ retransmission with possibly updated PRR and links.

Consider a 2-hop information based routing[2], similarly to (6), we select the one among j with smallest $\overline{d}^{(2)}_{j,D}$ defined below as the next-hop forwarder. The following metric is to identify the best potential candidate with expected progress in 2 hops:

$$\begin{aligned}\overline{d}^{(2)}_{j,D} \triangleq\; & p(d_{i,j})p(d_{j,k})d_{k,D} + (p(d_{i,j})\tilde{p}(d_{j,k}) + \\ & \tilde{p}(d_{i,j})p(d_{j,k}))d_{j,D} + \tilde{p}(d_{i,j})\tilde{p}(d_{j,k})d_{i,D} \quad (7)\end{aligned}$$

where $\tilde{p}(\cdot) \triangleq 1 - p(\cdot)$ for notational convenience.

Iteratively, consider 3-hop information based routing. Says, a neighbor node of k is node l. Similarly to (6) and (7), we define the best candidate of next-hop forwarder by its expected progress in 3-hop neighborhood with respect to the following metric:

$$\begin{aligned}\overline{d}^{(3)}_{j,D} \triangleq\; & p(d_{i,j})p(d_{j,k})p(d_{k,l})d_{l,D} + \\ & (p(d_{i,j})p(d_{j,k})\tilde{p}(d_{k,l}) + p(d_{i,j})\tilde{p}(d_{j,k})p(d_{k,l}) \\ & +\tilde{p}(d_{i,j})p(d_{j,k})p(d_{k,l}))d_{k,D} + \\ & (p(d_{i,j})\tilde{p}(d_{j,k})\tilde{p}(d_{k,l}) + \tilde{p}(d_{i,j})p(d_{j,k})\tilde{p}(d_{k,l}) \\ & +\tilde{p}(d_{i,j})\tilde{p}(d_{j,k})p(d_{k,l}))d_{j,D} + \\ & \tilde{p}(d_{i,j})\tilde{p}(d_{j,k})\tilde{p}(d_{k,l})d_{i,D}. \quad (8)\end{aligned}$$

[1] It includes preamble, packet payload and CRC.

[2] Here, we simply assume the 2-hop information is available.

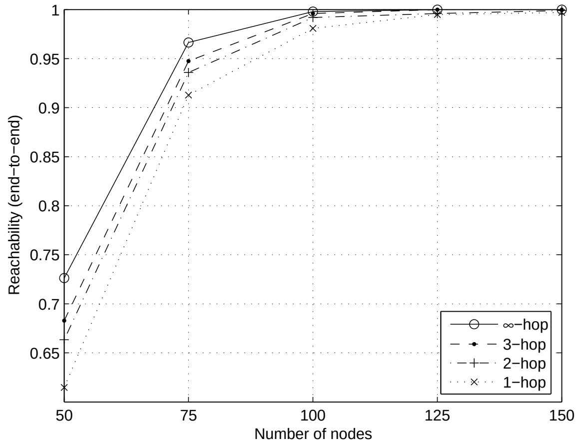

Figure 1. A comparison of reachability with different depths of neighborhood information.

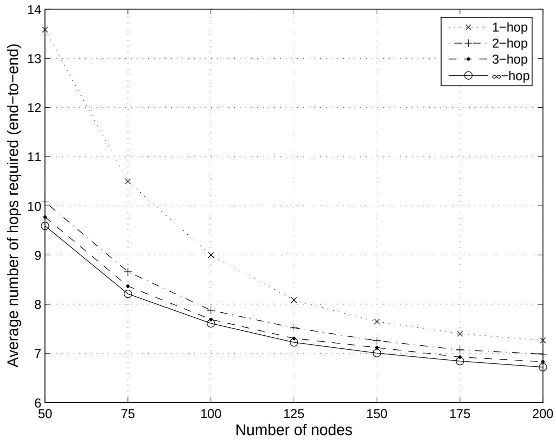

Figure 2. A comparison of average number of hops required from source to sink in different depths of neighborhood information.

3.3. Simulation and performance

Nodes are uniformly distributed in a geographical area of (200m × 200m). The source and destination nodes are located at (30m, 30m) and (170m, 170m) respectively. Following (1)–(3), given distance d between a pair of nodes, the corresponding PRR, $p(d)$, is drawn with respect to the log-normal shadowing each time during packet routing. The simulation model can be interpreted as in time-varying wireless links with dynamic network topology and connectivity[3].

In the performance evaluation, the number of hops required in routing from source to destination is highly concerned. It implies the routing delay. Here, we do not take into account MAC layer delay. This will help to identify the performance only due to routing issues and isolate from impacts of other factors. A more complete study of a 2-hop based routing including MAC and WSN experimental link model will be presented later in Section 4. Meanwhile, we also look at the reachability of routing that is defined by the percentage of runs in which packets from source can reach the destination. If a packet is always delayed at a specific node or possibly looped in an isolated region and thus cannot go to the destination after a large number of hops, it will be considered as unreachable, i.e. routing failure.

Here, simulation results are obtained by 3000 runs each. Fig. 1 and Fig. 2 show the reachability and average number of hops required from end-to-end respectively. In Fig. 1, reachability generally increase as node density increases as expected. Besides, enhancement is observed generally from the k-hop to $k + 1$-hop based routings when more neighborhood information is allowed.

[3]A connectivity is often defined [18] by PRR great than or equal to 0.9, while the transitional region is in PRR values between 0.1 and 0.9.

Note that the gain since from 2-hop to 3-hop is relatively marginal, while that from 1-hop to 2-hop based routing is more attractive. It should be noted that when the number of nodes in the WSN is large, the improvement of reachability from k-hop to $k+1$-hop based routing is small. This is due to the fact that, as pairs of nodes are getting closer and closer, the PRR will be high for most cases. Thus, even if we just follow the 1-hop information, a packet can easily reach the destination by a simple path searching. This can also be observed in Fig. 2 that the gap in-between the four curves of average number of hops required is getting smaller. However, the difference from 1-hop to 2-hop based routing is still quite significant, particularly when the number of nodes is relatively small.

Note that the ∞-hop result is provided as a benchmark, which refers to an ideal and optimistic performance. In Fig. 2, the average number of hops required is taken from runs in which a packet is routed from source to destination. It indicates the statistical result of successful forwarding. As shown in Fig. 2, as the number of nodes in the WSN increases, statistically the number of hops required will decrease since now more and better forwarding options are likely available. This helps a better progress and reduces the number of hops experienced.

4. TH-SPEED: another case study

4.1. Algorithm design

Referring to the study above, we may have a general idea that 2-hop information based routing better improves the routing path decision. This is the motivation of TH-SPEED design. TH-SPEED primarily aims at lowering packet deadline miss ratio for demanding real-time WSN. However, it also considers energy utilization efficiency which has not been addressed in SPEED and MM-SPEED. Similarly to SPEED, we assume each node is aware of its geographic location possibly by some localization techniques [22], or just using the mechanism specified in the IEEE 802.15.4a amendment [23]. Suppose location information can be further exchanged among 2-hop neighbors [17, 24]. Therefore, each node is aware of its immediate and 2-hop neighbors, and their locations. To estimate the packet delivery speed to next hop, we adopt the velocity concept used in SPEED.

TH-SPEED has four core components: (i) 2-hop velocity based forwarding strategy, (ii) delay estimator, (iii) energy-efficient probabilistic drop, and (iv) optional residual energy cost function for node energy consumption balancing. Basically, our protocol uses a 2-hop packet delay estimation to compare with the required velocity and thus decides which node should be the forwarder. If there is no suitable one, the packet will be dropped by in a probabilistic mechanism. By the 2-hop information, holes or congestions in the network could be predicted at an early time. Meanwhile, a more promising path can be identified after considering more possibilities. The cost is that TH-SPEED requires more neighborhood information for a better decision. Besides, some more computations are conducted in decision making. We assume the increment is affordable and will discuss in Section 4.3 one possible solution to reduce the overhead.

4.1.1. Two-hop velocity based forwarding

To begin with, some technical definitions are required. The source and destination nodes are labeled by S and D respectively. End-to-end packet delivery velocity for a required deadline, t_{set}, is defined as:

$$S_{set} = d(S, D)/t_{set}. \tag{9}$$

In SPEED, the core stateless non-deterministic geographic forwarding (SNGF) works as follows. Upon receiving a packet, node i calculates the velocity provided by each of the forwarding nodes in $\mathbf{F}(i)$ expressible as:

$$S_i^j = \frac{d(i, D) - d(j, D)}{Delay_i^j} \tag{10}$$

where $j \in \mathbf{F}(i)$ and $Delay_i^j$ denotes the estimated hop delay between nodes i and j. If there exists j such that $S_i^j > S_{set}$, the one with largest velocity is selected.

In the proposed TH-SPEED, similarly, by 2-hop information, node i will calculate the velocity provided by each of the 2-hop forwarding node pairs, $\{\mathbf{F}(i), \mathbf{F}_2(i)\}$. That is,

$$S_i^{j \to k} = \frac{d(i, D) - d(k, D)}{Delay_i^j + Delay_j^k} \tag{11}$$

where $j \in \mathbf{F}(i)$ and $k \in \mathbf{F}(j)$. If there exists node pairs $\{j, k\}$ such that $S_i^{j \to k} > S_{set}$, the one that can provide the largest velocity is preferred. Therefore, node j, the

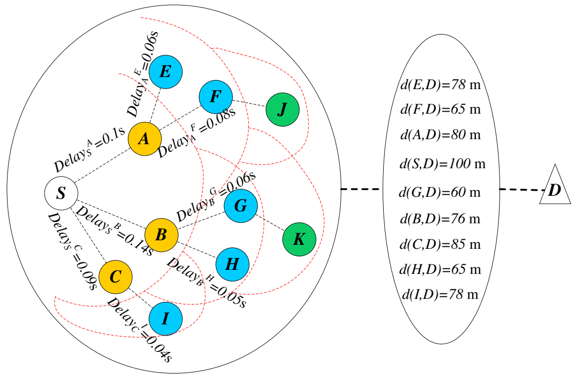

Figure 3. A case study to show their difference: SPEED and TH-SPEED.

parent of node k, will be chosen as the immediate forwarder. Then, node j will relay the packet and takes the role of node i. The mechanism continues and repeats at node j with its 2-hop neighborhood so as to find the next forwarding node iteratively.

Fig. 3 gives an example. Suppose a packet is to be sent from S to D. Here, $\{A, B, C\} \in \mathbf{F}(S)$, $\{E, F\} \in \mathbf{F}(A)$, $\{G, H\} \in \mathbf{F}(B)$, $\{I\} \in \mathbf{F}(C)$, $\{J\} \in \mathbf{F}(F)$ and $\{K\} \in \mathbf{F}(G)$. Let the end-to-end deadline be 0.65s. Following (9) and the numerical values given in Fig. 3,

$$S_{set} = 100m/0.65s = 154\ m/s.$$

According to SPEED, by (10),

$$\begin{aligned} S_S^A &= (100m - 80m)/0.1s = 200\ m/s, \\ S_S^B &= (100m - 76m)/0.14s = 171.4\ m/s, \\ S_S^C &= (100m - 85m)/0.09s = 166.7\ m/s. \end{aligned}$$

Thus, node A will be chosen as the forwarder since it can provide the largest velocity higher than S_{set}. Iteratively, node A will choose node F as its forwarder since S_A^E = (80m−78m)/0.06s = 33.3 m/s, while S_A^F = (80m−65m)/0.08s = 187.5 m/s.

However, according to TH-SPEED, node S will search among its 2-hop neighborhood and calculate the velocity provided by each 2-hop pair. Following (11), node pair $\{B, G\}$ can provide velocity:

$$\begin{aligned} S_S^{B \to G} &= \frac{d(S, D) - d(G, D)}{Delay_S^B + Delay_B^G} \\ &= (100 - 60)/(0.14 + 0.06)\ m/s \\ &= 200\ m/s \end{aligned}$$

which is greater than S_{set} and is the largest one among all the pairs, as $S_S^{A \to E}$ = 137.5 m/s, $S_S^{A \to F}$ = 194.4 m/s, $S_S^{B \to H}$ = 184.2 m/s, and $S_S^{C \to I}$ = 169.2 m/s respectively. Therefore, node B will be chosen as the immediate forwarder. By TH-SPEED, it is expected that the sender will have a forwarding node pair that can provide the largest velocity in 2-hop neighborhood. However, by SPEED, it is only one-hop optimized. Besides, if there is a topology hole after the first forwarding node, SPEED can get a problem. However, by TH-SPEED, this can be alleviated.

Inherently, TH-SPEED has 1-hop more prediction capability as using a "telescope" while finding the path. General speaking, even if the starting choice is not the globally optimized one, it may have a better chance to gradually be corrected due to the deeper sight of view when compared with SPEED.

4.1.2. Delay estimator

In (11), it is observable that the delay estimation from a sender to its available forwarders has played a significant role in the velocity metric. The delay of a packet from node i to its forwarder j is comprised of the MAC delay, transmission time (including acknowledgement time) and the transmission count[4], denoted by $Delay_{MAC}$, $Delay_{tran}$ and C_i^j respectively.

$$Delay_i^j = (Delay_{MAC} + Delay_{tran}) \times C_i^j. \quad (12)$$

The transmission time of a packet and its acknowledgement can be considered as constant determined by the packet size and network bandwidth. That is,

$$Delay_{tran} = \frac{packet\_size + ack\_size}{bandwidth}. \quad (13)$$

Our delay estimator follows the classical method used for round trip time (RTT) estimation in TCP protocol [25], via the following updating equation:

$$R \leftarrow \alpha R + (1-\alpha)M \quad (14)$$

where R is the average RTT estimate, M is the RTT measurement from the most recently received packet, and α is filter gain. It is shown efficient in [25] and [15]. Following the same concept, we estimate $Delay_i^j$ by the joint consideration of the history average delay and the most recent value from the former transmission. However, if the packet fails to be transmitted after exceeding the maximum number of retransmissions according to ARQ mechanism, the measurement M_i^j for node pair (i, j) in current time will be set to a large value to avoid selecting the path for a certain number of rounds. Estimate of $Delay_i^j$ at time t can thus be expressed as:

$$Delay_i^j(t) = \frac{\alpha}{t-1}\sum_{k=1}^{t-1} Delay_i^j(k) + (1-\alpha)M_i^j(t-1). \quad (15)$$

The link delay of a packet is measured by the sender, which will stamp the time a packet is sent out and compare it with the time an ACK is received. Assume that

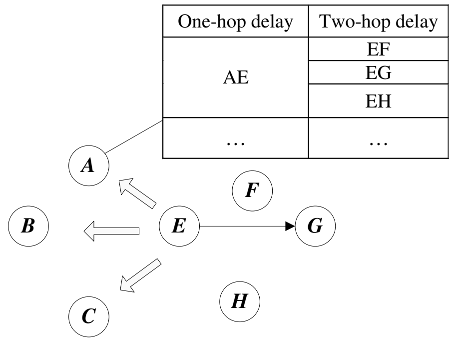

Figure 4. Two-hop delay update.

the ACK is transmitted in a parallel channel without collision and loss, the single-hop delay can be approximated by the RRT since the propagation time of ACK is negligible. To update the link delay information to corresponding nodes in the routing path, after receiving the ACK with delay information from its forwarder, a node will multicast a feedback packet, which contains the updated delay of the forwarding link, to its parent nodes. Fig. 4 shows an example of the link delay update after node G is chosen as the forwarder of node E. $Delay_E^G$ is updated at E after receiving ACK from G and then feedback to A, B and C. Accordingly, the delay field EG in their records, e.g. a 2-hop delay table, will be updated to (15) with the new information.

4.1.3. Energy-efficient probabilistic drop

A policy of energy-efficient probabilistic drop is taken when no node in the 2-hop forwarding set can provide the required velocity. The packet drop probability is proportional to its distance apart from the destination. That is, a node closer to the source will have a higher probability to drop the packet than the node closer to the destination when there is no forwarder which can meet the required velocity. Since a packet near the destination has already traveled a long way and a lot of nodes have consumed energy to forward it. It is worthwhile to try the best to see whether we can finally deliver it successfully. Although the current hop may not be able to meet the required velocity, it is possible to meet the end-to-end requirement finally if the coming hops have relatively short delays. However, if the node near the source cannot meet the velocity, from the point of view of energy utilization efficiency, it will be more efficient to drop earlier and look for a better chance in the coming retransmission.

Details of the probabilistic drop policy are described below. Suppose node i searches among its 2-hop neighborhood and cannot find a forwarder that can maintain the required velocity, it will drop the packet by a probability equal to $\frac{d(i,D)}{d(S,D)}$, where $d(i, D)$ is the distance from node i to destination and $d(S, D)$ is the distance between the source and destination respectively.

We will show in Section 4.2 the consequent difference

[4] ARQ is adopted thus if the packet fails to be transmitted due to collision or bad links, retransmission will be initiated.

of the energy-efficient probabilistic drop with respect to two other methods: (i) all packets will be forwarded via the nodes which provide the largest velocity even when they cannot meet S_{set}, and (ii) once there is no node that can provide the required velocity, the packet will be dropped immediately. The policy of energy-efficient probabilistic drop has outperformed the other two under a joint consideration of deadline miss ratio and energy efficiency.

4.1.4. Energy balancing

In SPEED, no strategy for energy balance is considered. Some nodes will frequently be chosen as forwarders due to their significant positions in the geographical area. This can be observed from the simulation result reported in Section 4.2. If a tradeoff between packet delay and node energy consumption balance is allowed or the deadline requirement is not very stringent, it may not be necessary to always choose the node that can provide the largest velocity as forwarder. Instead, we choose the one which has the largest joint metric, ve, defined in terms of the velocity and residual energy below. Provided that the velocity is still higher than S_{set}, a certain amount of the expected velocity is sacrificed to have energy consumption balance by looking at the residual energy and velocity together:

$$ve_i^{j\rightarrow k} = \frac{c_v \times \frac{S_i^{j\rightarrow k}}{S_{set}} + c_e \times \frac{residual\_energy_j}{initial\_energy_j}}{c_v + c_e} \quad (16)$$

where c_v and c_e are the weights of velocity and energy respectively. A larger c_v value tends to prefer nodes which can provide greater velocity and thus less delay. However, it may lead to concentrative energy consumption. A larger c_e will direct traffics to more nodes and consequently lead to a better load balancing but possibly increased packet delay. The tradeoff between c_v and c_e depends on the link quality and traffic distribution. We will leave the investigation as future work and currently set $c_v = c_e = 1$.

4.2. Performance evaluation

The effectiveness of TH-SPEED is evaluated in the following simulation studies. To be close to practical WSN and realistic implementation, we set the MAC layer, link quality model and energy consumption parameters based on Mica2 Motes. Here, we will focus on the conventional many-to-one traffic model commonly adopted in environmental monitoring WSN. A number of 200 nodes are randomly distributed in a 200m × 200m area. For comparison, results from a number of 400 nodes will be discussed as well. To simulate multi-hop transmissions with a large enough number of hop counts, we locate the sources in the left lower area of the region and uniformly distributed within a circle of radius 30m centered at (30m, 30m), while the sink is fixed at position (200m, 200m).

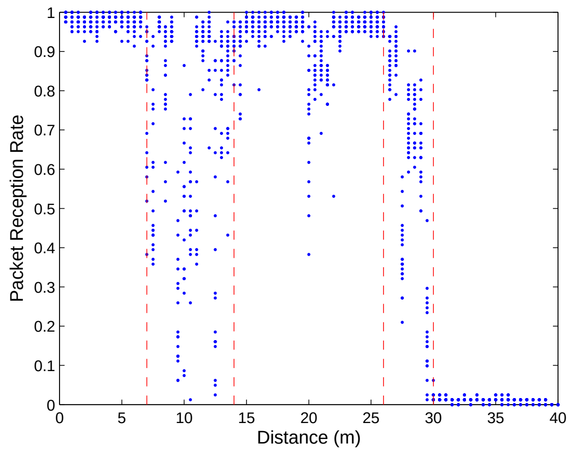

Figure 5. Packet reception rates at difference distances.

4.2.1. MAC setting

Following the default CSMA scheme in Mica2 Motes, to initiate a packet transmission, a sensor node will generate a random initial backoff time uniformly distributed in the range of [15, 68.3] ms and start a timer. Upon timer expiration, the channel is sensed. If it is found idle, a packet is transmitted. If the channel is busy, the sensor node will generate a further random time because of the congestion. The time is uniformly distributed in the range of [12.08, 193.3] ms. The backoff timer starts again. To improve delivery reliability, ARQ is employed here. If the total number of transmission count and MAC backoff count is great than 7, the packet is dropped.

4.2.2. Experimental link model

Here, we adopt the link model from an experiment based on Mica2 Motes. A sequence of sensor nodes are deployed linearly in spacing of 0.5 m from one another. The packet loss rate between pairs of nodes at different distance is measured. At any time, there is always one transmitter and the remaining nodes will count the number of packets successfully received. Each node is scheduled to transmit 80 packets at 10 packets/s in one around and finally the average reception rate is counted. Nodes are deployed on the ground of an open tennis court. The transmission power is set at 0 dBm. Fig. 5 shows the result of scatter diagram and how link quality varies with distance. The study in [26] has shown similar pattern.

By Fig. 5 and collected statistics, the link quality can be described as a piecewise function of distance d and modeled by a random variable $r(d, \mu, \sigma^2)$ in normal distribution with mean μ and variance σ^2 in the range of respective distance [27]. Table 1 shows the model parameters. For our simulation, a random number x is generated each time and then compared to $r(d, \mu, \sigma)$. If $x < r(d, \mu, \sigma^2)$, the packet is supposed successfully transmitted. Otherwise, it is considered lost and retransmission will be initiated. Therefore, a bad link will generally lead to a greater delay with more retransmissions.

Table 1. Link quality model following Fig. 5

Distance d (m)	Mean μ	Variance σ^2
0-7	0.97	0.02
7-14	0.70	0.14
14-26	0.93	0.06
26-30	0.53	0.08
30-40	0.01	0.005

4.2.3. Energy consumption model

Table 2 shows the energy model based on Mica2 Motes [28]. When the node is sending a packet, the CPU is in active state and the current consumption equals to $8.0 + 8.5 = 16.5$ mA with a time duration of 0.5 ms. When receiving a packet, the CPU is in active state and the current consumption is $8.0 + 7.0 = 15.0$ mA with a duration of 0.5 ms. When the node is just listening, the current consumption is only counted by the CPU's consumption, i.e. 8.0 mA. In sleeping mode, the CPU is in idle state and the current consumption is only 3.2 mA. Initial energy in each node is assumed the same. The voltage supply is by default 3V and constant.

Table 2. Mica2 Motes based energy model

Operation	Time (ms)	I (mA)
CPU active	N/A	8.0
CPU idle	N/A	3.2
Transmit (0 dBm)	0.5	8.5
Receive	0.5	7.0

4.2.4. Simulation and performance

In supporting real-time QoS, we are particularly interested in the packet delay performance and their deadline miss ratio. Note that the following definitions are all in end-to-end sense.

(i) Deadline miss ratio (DMR) is defined by the number of packets which miss their deadlines over the number of initiated packets.

(ii) Energy consumed per packet (ECP) is defined by the total energy consumed divided by the number of packets successfully transmitted.

(iii) Packet average and worst-case delays are defined by the mean of packet delay and the largest value experienced by the successfully transmitted packets.

First, we will show the effectiveness of energy-efficient *probabilistic drop* strategy employed in a comparison to the two other approaches previously mentioned: (i) all packets will be forwarded via nodes with largest velocity even when they cannot meet the required velocity, namely as *best-effort* forwarding, thus no packet will be dropped, and (ii) once there is no node that can provide the required velocity, the packet will be dropped immediately, namely as *hard-decision* drop.

The comparison of their DMRs under a same network topology with 200 nodes and 25 sources is given in Fig. 6. Best-effort forwarding has a slightly lower DMR when the deadline is relatively tight. However, when the deadline is increased and greater than 700 ms, the performance of best-effort forwarding is worse than that in the probabilistic drop because packet congestion occurs and the best-effort forwarding does not drop packets. Consequently, it suffers higher loss. Hard decision drop has a much higher DMR than the other two when the deadline is small since it is incapable of taking the benefit of statistical diversity gain during the multi-hop propagation, for example, in the best-effort forwarding.

As energy utilization efficiency is also one of the major concerns, we compare that in the three strategies. Fig. 7 shows their energy consumed per successfully transmitted packet. It is observed that Fig. 7 has quite similar characteristics and tendency as those shown in Fig. 6. The best-effort forwarding has a slightly lower energy consumption than the probabilistic drop scheme when the deadline is very tight. However, since deadline greater than 700 ms, probabilistic drop is generally much more energy-efficient via dropping packets with a consideration of the routing progress. On the other hand, hard decision may underestimate the capability of meeting the deadline later even when the packet has propagated to a location close to the destination and thus lead to a certain level of energy inefficiency. Despite the probabilistic drop scheme is not always the best among the three, by comparing their DMRs and energy consumption, it can reach an overall better performance and is more adoptable.

In the following, a detailed performance study of TH-SPEED is conducted and compared with SPEED. Fig. 8 shows the DMR of TH-SPEED in a WSN of 200 nodes, in which there are 10 source nodes. The result is plotted against different deadline requirements from 600 ms to 3000 ms. As expected, the DMR decreases as the deadline increases. It is observable that under TH-SPEED, when the deadline is large enough, the DMR converges to zero. In comparison, as shown in Fig. 8, SPEED has a much higher DMR generally. Besides, even when the deadline is up to 3000 ms, SPEED has only tended to a DMR level of 0.1. Comparatively, the DMR in TH-SPEED drops much faster than that in SPEED. The result has clearly indicated the effectiveness of TH-SPEED with the proposed 2-hop based routing strategy.

The energy efficiency of TH-SPEED is compared to SPEED in Fig. 9. As expected, the energy consumed

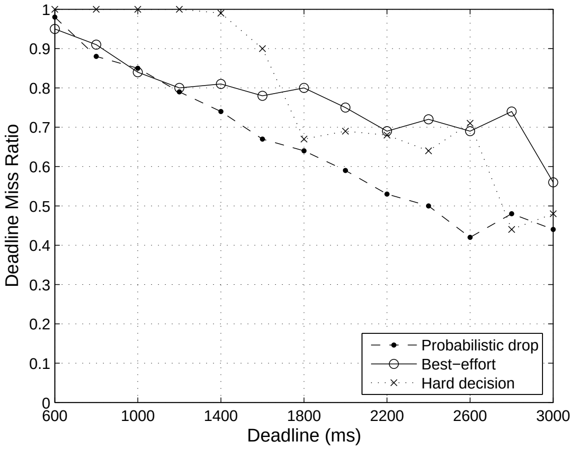

Figure 6. A comparison of packet deadline miss ratio among the three strategies.

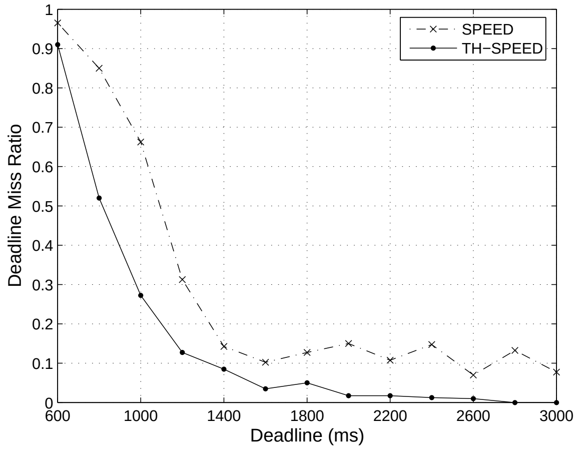

Figure 8. DMR under different deadlines requirements. Number of nodes = 200. Number of source nodes = 10.

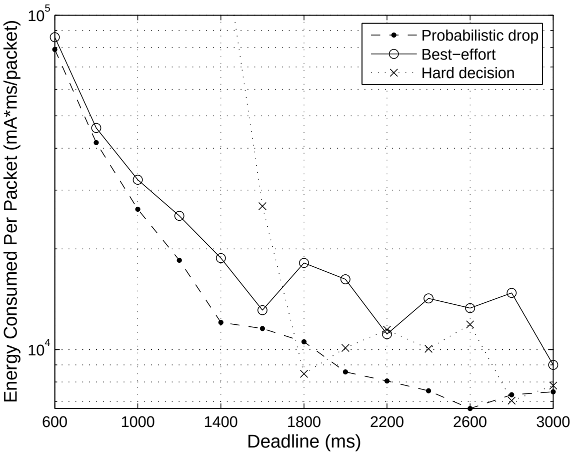

Figure 7. A comparison of energy utilization efficiency among the three strategies.

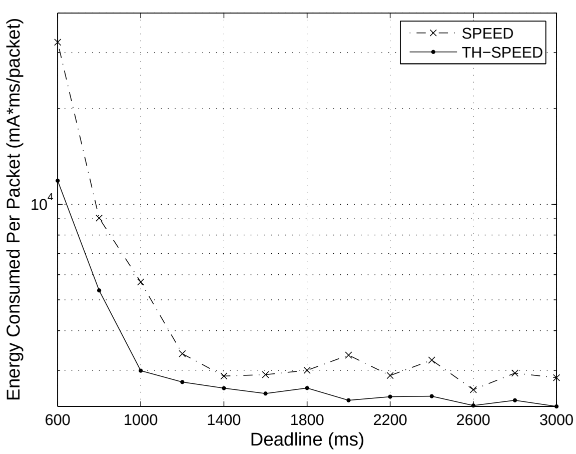

Figure 9. ECP in comparison.

per successfully transmitted packet decreases as the deadline increases since more packet can be finally forwarded to the destination due to a longer allowable time for the packet delivery. Compared to SPEED, TH-SPEED has consumed less energy. In other words, it has a higher energy efficiency. One of the major reasons is that TH-SPEED can achieve a lower DMR. It is observable that Fig. 9 has similar tendency and convergence characteristics as those in Fig. 8. Generally, TH-SPEED outperforms SPEED and can converge to a lower energy consumption level as deadline increases.

Fig. 10 shows the packet end-to-end average and worst-case delays respectively. It is observed that TH-SPEED and SPEED have quite close performance. Generally speaking, when there are several routing paths with delays which can satisfy the required velocity, TH-SPEED will have a better chance to go into a shorter path and have lower end-to-end delay due to the 2-hop routing optimal selection[5]. As shown in Fig. 8, it is able to successfully deliver more packets from end to end. However, note that they will include some packets from relatively bad network topology scenarios or large routing delay situations in which SPEED may have already dropped the packets. Therefore, it is possible that the worst-case or average delay in TH-SPEED may be higher than those in SPEED by the measurements. This phenomenon is observable in Fig. 10. However, more importantly, the worst-case delay is always bounded by the deadline requirement.

Furthermore, the number of nodes is increased from 200 to 400 in the same area and with same number of source nodes. While comparing TH-SPEED and SPEED in DMR, ECP and packet delay in 400 nodes, simulation

[5]TH-SPEED finds a routing path that can meet the required velocity in terms of 2-hop knowledge.

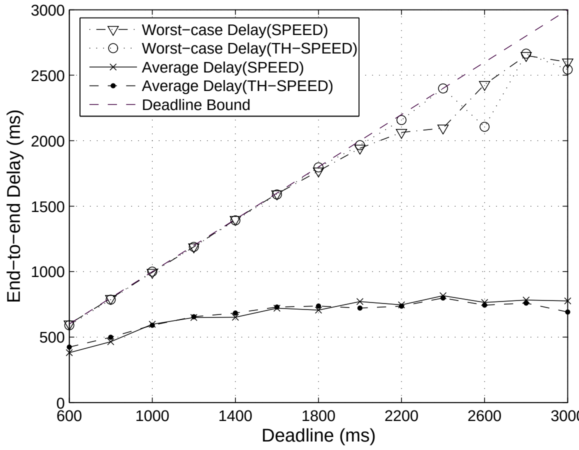

Figure 10. End-to-end delay comparison.

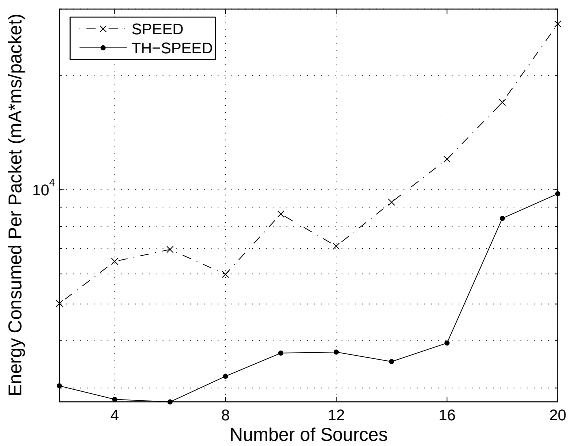

Figure 12. ECP in comparison.

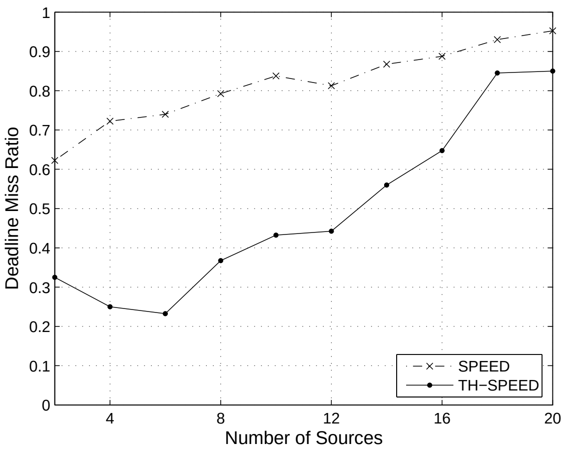

Figure 11. DMR under different number of source nodes. Number of nodes = 200.

results obtained show that the performance tendency and characteristics are very similar to those in Fig. 8, Fig. 9 and Fig. 10 respectively. Due to a lack of space, we will present elsewhere. However, it is worth noting that, in both network sizes, TH-SPEED outperforms SPEED in DMR and also energy utilization efficiency indicated by Fig. 8 and Fig. 9. Meanwhile, their packet average and worst-case delays have very similar performance.

Moreover, we investigate the performance of TH-SPEED under different workload. Fig. 11 shows the DMR as the number of sources is increased from 2 to 20, while the deadline requirement is fixed at 800 ms. In both SPEED and TH-SPEED, it is observed that the DMR increases as the number of sources increases and so is the energy consumption as indicated in Fig. 12 respectively. The increase in DMR is resulted by the increased channel busy probability, packet collisions at MAC and network congestion by the increased number of sources and consequent traffics. However, compared to SPEED, TH-SPEED has a lower DMR and energy consumption as shown in Fig. 11 and Fig. 12 respectively. This reflects the general improvement by TH-SPEED. The packet average and worst-case delay performance in both schemes is approximately at the same level. Plots are omitted due to the similarity.

Last, we study the performance of the residual energy cost function, which is an optional add-on for node energy consumption balance in case packet deadline requirement can be relaxed and a relatively large value is allowed. The motivation is that if there are several nodes who can serve as forwarding nodes and provide a velocity greater than the required velocity, instead of simply choosing the one that has the largest velocity, we can take into account the residual energy of nodes for a better balancing. Among those who can meet the velocity requirement, a node with higher residual energy will be favorable.

Fig. 13 shows the node distribution and their locations in this study. There are totally 200 nodes including 4 source nodes. The sources are located in the lower left area inside the circle, while the sink is fixed at the upper right point. The deadline is set to a large value of 3000 ms. Fig. 14 and Fig. 15 show the node energy consumption distribution in SPEED and TH-SPEED respectively after 200 runs. Nodes that have consumed much more energy than the other are highlighted in solid and dashed rectangles referring to SPEED and TH-SPEED respectively.

As observed in SPEED, some nodes along the path from sources to sink are frequently chosen as forwarders and consume much more energy than the other, while in TH-SPEED only nodes close to the sources and sink consume relatively high energy. The latter is natural and unavoidable especially as there may not be many good forwarding options near the sources and sink. Besides, by comparing Fig. 15 to Fig. 14, energy consumption in TH-SPEED is more evenly distributed among those between source and sink. It can be expected that TH-SPEED will have a longer system lifetime due to the balancing. However, the cost is the tradeoff in packet delay performance.

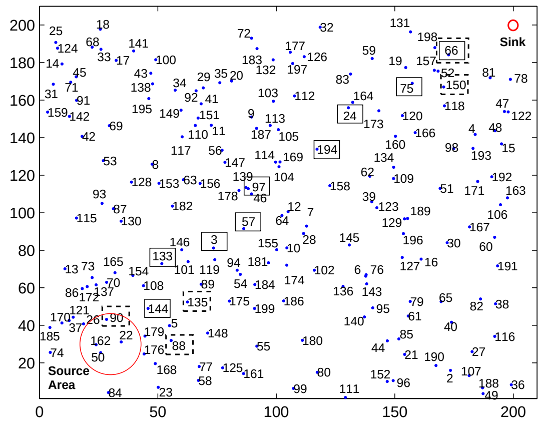

Figure 13. The topology of 200 nodes in the study of energy consumption distribution. Solid and dashed rectangles indicate SPEED and TH-SPEED respectively.

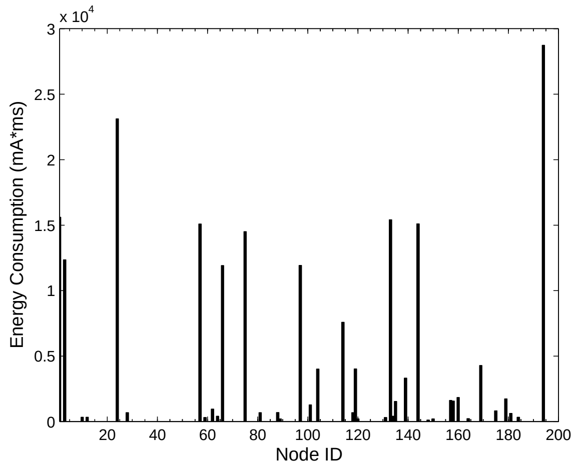

Figure 14. Node energy consumption in SPEED. Number of nodes = 200.

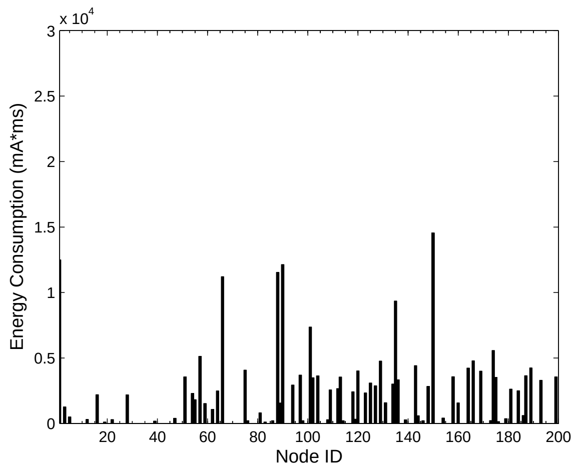

Figure 15. Node energy consumption in TH-SPEED. Number of nodes = 200.

As shown in Table 3, in this WSN, TH-SPEED will have a larger packet average delay by the residual energy consideration. However, even now, the DMR in TH-SPEED is still smaller than that in SPEED. That is, the DMR which is highly concerned in real-time service has not been sacrificed in the node energy consumption balancing.

Table 3. Performance of TH-SPEED after including residual energy consideration. The result is compared to SPEED.

Routing Protocol	SPEED	TH-SPEED
Deadline Miss Ratio	17%	0%
Average Delay (ms)	603.92	963.15
Energy Utility (mA×ms/packet)	2472.3	2486.8

4.3. Discussions

It is worth noting that, in the current design, the 2-hop link delay updating will generally lead to more overheads than that required for conventional 1-hop information updating. More feedback packets will be sent to the corresponding parent nodes. However, one can consider to reduce the overheads by piggybacking the updated information in ACK. These data will be sent together only when an ACK is to be sent. This can help to keep in a small number of feedback packets although the packet size will be larger. A drawback is that the 2-hop delay information may not be updated frequently enough. However, since the link delay estimation is based on the combination of history average value and the recent one, there could be minor difference to the estimation performance even if the update is not immediate and especially in WSN with low mobility. An investigation is expected in a future work.

In our simulation, the deadline requirement is assumed constant in all nodes. For different deadlines in different packet types, MM-SPEED [14] has designed a prioritized MAC and multi-SPEED routing to provide service differentiation. In [7], RT and non-RT packets are separated with classifier and assigned different bandwidth according to different priorities. It is also possible to integrate cross-layer method with priority scheduling to our current design which does not include service differentiation with prioritized MAC mechanisms.

5. Conclusion

The idea of using multi-hop neighborhood information to make routing decision is investigated in this paper.

Through the asymptotic performance study of a routing decision obtained from different hops of neighborhood information a priori, we see a significant improvement potential from 1-hop to 2-hop based routing decision, which inspires us to design a 2-hop neighborhood information based real-time routing protocol for WSN. We adopt the approach of mapping packet deadline to a velocity as SPEED; however, the routing decision is made based on the 2-hop velocity. An energy efficient probabilistic drop is designed to improve energy utilization efficiency. When packet deadline requirement is not stringent, a design is integrated to release nodes from heavy consumption. Energy balance over nodes is thus improved. Simulation results show that, compared with existing protocol SPEED that only utilizes 1-hop information, TH-SPEED achieves lower end-to-end deadline miss ratio and higher energy efficiency. In future work, we are interested to see how to support differentiated QoS and keep the required information exchange in a minimum necessary amount. The results reported here may also lead to other interesting protocols and designs.

References

[1] J. Stankovic, T. Abdelzaher, C. Lu, L. Sha, and J. Hou, "Real-time communication and coordination in embedded sensor networks", *Proc. IEEE*, vol. 91, no. 7, pp. 1002–1022, July 2003.

[2] Y. Li, C. S. Chen, Y. Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey", in *Proc. IFAC FET*, November 2007, pp. 373–380.

[3] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithm with guranteed delivery for wireless networks", *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1023–1032, October 2001.

[4] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks", *IEEE Trans. Mobile Comput.*, vol. 1, no. 2, pp. 111–122, April-June 2002.

[5] L. Bao and J. J. Garcia-Luna-Aceves, "Channel access scheduling in ad hoc networks with unidirectional links", in *Proc. Workshop on Discrete Algorithms and Methods for Mobility*, 2001, pp. 9–18.

[6] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks", in *Proc. ICDCS*, May 2003, pp. 46–55.

[7] K. Akkaya and M. Younis, "An energy-aware QoS routing protocol for wireless sensor networks", in *Proc. ICDCS*, May 2003, pp. 710–715.

[8] S. Ergen and P. Varaiya, "Energy efficient routing with latency guarantee for sensor networks", *ACM Wireless Networks*, vol. 13, no. 5, pp. 679–690, October 2007.

[9] N. Boughanmi and Y. Q. Song, "A new routing metric for satisfying both energy and delay constraints in wireless sensor networks", *J. VLSI Signal Processing*, 2007.

[10] IEEE Std 802.15.4, "Part 15.4: Wireless Medium Access (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE-SA Standards Board, September 2006.

[11] P. Pothuri, V. Sarangan, and J. Thomas, "Delay-constrained energy efficient routing in wireless sensor networks through topology control", in *Proc. IEEE Int. Conf. Netw. Sensing Control*, April 2006, pp. 35–41.

[12] Z. Alliance, (), [Online]. Available: http://www.zigbee.org

[13] B. Nefzi and Y. Q. Song, "Performance analysis and improvement of Zigbee routing protocol", in *Proc. IFAC FET*, November 2007, pp. 199–206.

[14] 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 Comput.*, vol. 5, no. 6, pp. 738–754, June 2006.

[15] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor network", in *Proc. IWQoS*, June 2006, pp. 83–92.

[16] G. Calinescu, "Computing 2-hop neighborhoods in ad hoc wireless networks", in *Proc. AdHocNow*, 2003, pp. 175–186.

[17] V. Rajendran, K. Obraczka1, and J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", *Wireless Networks*, vol. 12, no. 1, pp. 63–78, Febuary 2006.

[18] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links", in *Proc. IEEE SECON*, 2004, pp. 517–526.

[19] Crossbow, (2002), Mica2 Motes datasheet [Online]. Available: http://www.xbow.com.

[20] K. Sohrabi, B. Manriquez, and G. J. Pottie, "Near ground wideband channel measurement in 800-1000 MHz", in *Proc. IEEE VTC*, July 1999, pp. 571–574.

[21] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks", in *Proc. Sensys*, 2004, pp. 108–121.

[22] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks", in *Proc. IEEE INFOCOM*, 2001.

[23] IEEE P802.15.4aD/4 (Amendment of IEEE Std 802.15.4), "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE-SA, July 2006.

[24] L. Bao and J. Garcia-Luna-Aceves, "Transmission scheduling in ad hoc networks with directional antennas", in *Proc. MobiCom*, 2002, pp. 48–58.

[25] V. Jacobson, "Congestion avoidance and control", in *Proc. ACM SIGCOMM*, 1988.

[26] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks", in *Proc. ACM Sensys*, November 2003, pp. 14–27.

[27] Y. Li, Z. Wang, and Y. Sun, "Analyzing and Modeling of the Wireless Link for Sensor Networks", *Chinese Journal of Sensors and Actuators*, vol. 8, pp. 1846–1851, 2007.

[28] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and A. Welsh, "Simulating the power consumption of large-scale sensor network applications", in *Proc. ACM Sensys*, 2004, pp. 188–200.