A two-hop based real-time routing protocal for wireless sensor networks

Yanjun Li, Chung Shue Chen, Ye-Qiong Song, Zhi Wang, Youxian Sun

A Two-Hop Based Real-Time Routing Protocol for Wireless Sensor Networks

Yanjun Li[†‡], Chung Shue Chen[*‡], Ye-Qiong Song[‡], Zhi Wang[†], Youxian Sun[†]
[†]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, {wangzhi, yxsun}@iipc.zju.edu.cn

Abstract

A two-hop neighborhood information based routing protocol is proposed for real-time wireless sensor networks. The approach of mapping packet deadline to a velocity is adopted as SPEED; however, our routing decision is made based on the novel 2-hop velocity. Energy-efficient probabilistic drop is embedded to enhance energy utilization efficiency while reducing packet deadline miss ratio. In case packet deadline requirement is not stringent, a new mechanism is included to release nodes which are frequently chosen as forwarders. Improvement on energy consumption balance throughout the network is observed. The true characteristics of physical and MAC layers are captured in the simulation. A real lossy link model is drawn from extensive experiments through Mica2 Motes. Simulation results show that the new protocol has achieved lower packet deadline miss ratio and higher energy efficiency.

1. Introduction

Real-time (RT) quality-of-service (QoS) that concerns wireless sensor networks (WSN) have led to a substantial amount of research attention [1,2] in recent years after the wide acceptance and popularity of WSN in many emerging and promising applications. For example, in industrial systems [3], sensor devices are equipped to rotating machinery or automated assembly lines to monitor status information and help schedule maintenance tasks. In case of machinery fault or dangerous status, real-time warning message should be transmitted to control center in time so as to take prompt actions [4]. In environmental surveillance, a cost-effective WSN platform can be developed to detect the initiation of wildfire and monitor its spread based on temperature sensors [5]. Different from some existing best-effort services which may not have stringent packet timeliness requirement and can tolerate a significant amount of packet loss, these RT applications are much more demanding. Out-of-date data are usually irrelevant and may even lead to negative effects to the system monitoring and control. QoS control and guarantee in WSN is necessary for supporting today's RT service.

Without loss of generality, the inherent characteristics of WSN have imposed many challenges in the design of QoS-aware protocols [6]. Firstly, the wireless channel is time varying and relatively unreliable, thus a protocol design should not rely on the assumption of perfect channel knowledge. It is hard to have a resource-efficient reservation scheme due to the link quality fluctuation. Secondly, distributed algorithms or protocols are expected instead of centralized control. Thirdly, sensor nodes work with battery supplies. So, the energy efficiency should be taken into account in protocol design although packet delay performance such as deadline miss ratio (DMR) is often the primary concern in a RT system.

Generally speaking, providing real-time QoS in WSN can be addressed from different layers and mechanisms [2]. For example, the medium access control (MAC) is capable of providing channel access delay guarantee in a single-hop manner, while a routing protocol in the network layer can help to guarantee end-to-end or multi-hop transmission time. The approach of cross-layer optimization is able to provide some further improvements. Besides, in-network data aggregation strategy is a good complement to routing protocols for reducing data redundancy and alleviating network congestion. Middleware design can help to bridge the gap between application and lower layers and thus provide abstraction and mechanisms for efficient system coordination. Among the above, routing protocol has always played a crucial role in providing end-to-end QoS. Here, we will focus on this domain.

For system simplicity, most existing routing protocols are based on 1-hop neighborhood information. It is promising that multi-hop information can lead to better performance in many issues including routing, message broadcasting and channel access scheduling [7–10]. For computing 2-hop neighborhood information in wireless ad hoc and sensor networks, some distributed algorithms and efficient information exchange schemes are reported in [11, 12]. In a network of n nodes, computing 1-hop neighbors with $O(n)$ messages is trivial while computing 2-hop neighbors seems to increase the complexity and overheads. However, a complexity analysis reported in [11] has shown that every node can obtain the knowl-

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

edge of 2-hop neighborhood by a total of $O(n)$ messages, each of $O(\log n)$ bits, which could be enough to address the ID and geographic position of nodes.

Intuitively, a scheme can do better if more information is available and can be effectively utilized. In this paper, we will propose a 2-hop information based RT routing protocol for WSN and show its improvement over 1-hop based protocol, deterministic-SPEED (SPEED-S) [13]. The choice of two hops is a tradeoff between performance improvement and the complexity cost. The idea of 2-hop routing is straightforward but how to use or integrate the information properly so as to offer a better solution is generally nontrivial. The resultant design has the following novel features:

1. Compared with existing protocols that utilize only 1-hop neighborhood information, it achieves lower DMR and also higher energy efficiency.

2. In case packet deadline requirement is not stringent, we embed a mechanism that can release nodes which are frequently chosen as packet forwarder. An improvement on energy balance throughout the network is achieved.

3. The design captures the true characteristic of physical and MAC layers. A real lossy link model is drawn from experiments based on Mica2 Motes [14]. The MAC adopts the default CSMA/CA. With settings of the practical platform, the result is more realistic and convincing than existing theoretical ones.

The rest of the paper is organized as follows. Section 2 discusses related routing protocols for providing real-time QoS in WSN and explains the motivations. Section 3 specifies our design and the corresponding details. The performance of proposed protocol is reported in Section 4. Simulations and comparisons have shown its effectiveness. In Section 5, we discuss possible enhancement and potential work. Finally, Section 6 concludes the paper.

2. State-of-the-art of RT routing protocols

Real-time QoS has been considered in some existing routing protocols for WSNs. Akkaya and Younis [15] propose an energy-aware QoS routing protocol that finds energy-efficient path along which end-to-end delay requirement can be met. It is assumed that each node has a classifier to check the type of incoming packets and divert RT and non-RT traffics into different priority queues. The delay requirement is converted into bandwidth requirement. The protocol finds a list of least cost paths by using an extended version of Dijkstra's algorithm and picks a path from the list which can meet the end-to-end delay requirement.

Ergen *et al.* [16] presents an energy efficient routing method with delay guarantee for WSNs. They first exclude the delay constraint and formulate the lifetime maximization as a linear programming (LP) problem with the goal of determining optimal routing path and maximizing the minimum lifetime of each node in the network. The LP solution is first implemented in a centralized way and then approximated by a distributed iterative algorithm with considerations of energy cost. Then, delay guarantee is included by limiting the length of routing path from the node to sink.

Boughanmi and Song [17] propose a routing metric for evaluating path efficiency which is defined as the ratio of the energy efficiency to end-to-end delay, where the energy efficiency is specified in considering link failure and retransmissions. End-to-end delay is measured by hop count between the source and sink, which is collected by routing response messages in the initialization phase. The new routing metric is applied in AODV routing protocol with IEEE 802.15.4 MAC sublayer. It has improved network lifetime and end-to-end delivery ratio when compared to traditional AODV and the metric in [16].

Besides, there are several RT routing protocols that use velocity assignment policy, such as SPEED [13]. Based on the distance towards the destination, the packet deadline is mapped to a velocity. A packet is forwarded by a node which can meet the required velocity. If there is no neighbor node that can meet the requirement, the packet is dropped probabilistically to regulate the workload. Meanwhile, back-pressure packet re-routing in large-delay link is conducted to divert and reduce packets injected to a congested area. MM-SPEED [18] extends SPEED by providing multiple delivery velocities for packets with different deadlines in order to support different QoS. RPAR [19] is another velocity based routing algorithm. The required velocity is based on the progress towards the destination and the packet's remaining time before the deadline. A node will dynamically change its transmission power so as 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. All the above protocols are based on 1-hop neighborhood information.

In our proposed scheme, we also adopt the approach of mapping packet deadline to a velocity. However, our packet routing decision will be made based on 2-hop neighborhood information and corresponding metrics. It is therefore named as Two-Hop Velocity based Routing (THVR) in the paper. The details and routing design will be described in the following section.

3. Design of THVR for RT-WSN

Although 2-hop information based routing is intuitively helpful to improve the routing path decision, an explicit mechanism is necessary. It is worth noting that THVR primarily aims at lowering packet DMR for demanding real-time WSNs but will also consider energy utilization efficiency that has not been explicitly addressed in SPEED and MM-SPEED. Similarly to SPEED, we assume each node in the network is aware of its geographic location possibly via GPS or other localization techniques [20], or using the mechanism specified in the IEEE 802.15.4a standard [21]. The location information can be further exchanged among 2-hop neighbors [12,22]. Thus, 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 for comparing forwarding paths.

THVR is comprised of four components: (i) 2-hop velocity based forwarding strategy, (ii) delay estimation scheme, (iii) energy-efficient probabilistic drop, and (iv) optional residual energy cost function for system-wide energy balancing. Fundamentally, our protocol uses a 2-hop packet delay estimation to compare with required velocity and thus decides which node will be the forwarder. If there is no suitable forwarder, the packet will be dropped by the probabilistic mechanism. By the 2-hop information, holes or congestions in the network topology could be predicted at an early time. Meanwhile, a more promising path can be identified after considering more possibilities. The cost is that THVR requires more neighborhood information for a better decision. Besides, some more computations are conducted in the decision making. We assume the increment is affordable and will discuss in Section 5 a possible method to reduce the overhead. The technical details are explained below.

3.1. Two-hop velocity based forwarding strategy

To begin with, some definitions are defined. For each node i, $\mathbf{N}(i)$ is used to denote the set of its direct neighbors. The source and destination nodes are labeled by S and D respectively. The distance between a pair of nodes i and j is denoted by $d(i, j)$. Consequently, the end-to-end packet delivery velocity for a required deadline, t_{set}, is defined as:

$$S_{set} = \frac{d(S, D)}{t_{set}}. \quad (1)$$

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

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

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

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

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

$$S_i^j = \frac{d(i, D) - d(j, D)}{Delay_i^j} \quad (2)$$

where $j \in \mathbf{F}(i)$ and $Delay_i^j$ denotes the estimated hop delay between i and j. If there exists j such that $S_i^j > S_{set}$, it is chosen as the forwarder with probability $P(j)$ following the discrete exponential distribution below [13]:

$$P(j) = \frac{(S_i^j)^K}{\sum_{j=1}^{N} (S_i^j)^K} \quad (3)$$

where N is the number of candidates in $\mathbf{F}(i)$ and K is a weighing exponent to trade off between load balance and optimal delivery delay. For example, when K is large enough, the node with highest S_i^j will have $P(j)$ tends to 1 and the algorithm will fall into SPEED-S, in which the node that can provide the largest velocity and greater than S_{set} will be definitely chosen as the forwarder.

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

$$S_i^{j \to k} = \frac{d(i, D) - d(k, D)}{Delay_i^j + Delay_j^k} \quad (4)$$

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 will be the preferred forwarding pair. Therefore, node j, the 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 is repeated at node j with its 2-hop neighborhood so as to find the next forwarding node iteratively. As we are more focus on packet DMR and have an alternative design for load and energy balance in Section 3.4, here we will not include the load balance and optimal delay tradeoff function (3) in our protocol, or will simply consider K significantly large. Besides, by the corresponding online hop delay estimator in Section 3.2, overload and congestion will be reflected and avoided for coming routing selection.

By THVR, it is expected that the sender will have a forwarding node pair that can provide the largest velocity in 2-hop neighborhood. However, in SPEED, SPEED-S or SPEED-T[1] [13], it is only 1-hop optimized. For example, if there is a topology hole after the first forwarding node, SPEED-S will get a critical problem and have to activate back-pressure re-routing. However, by THVR, this kind of problems can be alleviated. Inherently, THVR 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 farther sight and view.

3.2. Delay estimation

From (4), we can observe 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

[1]SPEED is non-deterministic and chooses forwarder probabilistically according to (3), while SPEED-S and SPEED-T are deterministic. SPEED-S selects the node that has the maximum single-hop velocity. However, SPEED-T selects the one that has minimum single-hop delay.

from a node i to its immediate forwarder j is comprised of the MAC delay, transmission time (including acknowledgement time) and the transmission count[2], denoted by $Delay_{MAC}$, $Delay_{tran}$ and C_i^j respectively.

$$Delay_i^j = (Delay_{MAC} + Delay_{tran}) \times C_i^j \quad (5)$$

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

$$Delay_{tran} = \frac{packet\_size + ack\_size}{bandwidth}. \quad (6)$$

Our delay estimator follows the classical method used for round trip time (RTT) estimation in TCP protocol [23], via the following updating equation:

$$R \leftarrow \alpha R + (1 - \alpha) M \quad (7)$$

where R is the average RTT estimate, M is the RTT measurement from the most recently received packet, and α is a filter gain constant. It is shown efficient in [23] and [19]. 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 be expressed as follows:

$$Delay_i^j(t) = \frac{\alpha}{t-1} \sum_{k=1}^{t-1} Delay_i^j(k) + (1 - \alpha) M_i^j(t - 1). \quad (8)$$

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 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, i.e., those regard it as a forwarder candidate. Fig. 1 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 by the new information as in (8).

[2] ARQ (Automatic Repeat-reQuest) is adopted thus if the packet fails to be transmitted due to collision or bad links, retransmission will be initiated.

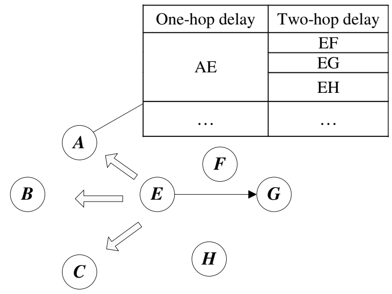

Figure 1. Two-hop delay update.

3.3. Energy-efficient probabilistic drop

If no node in the 2-hop forwarding set can provide the required velocity, a policy of energy-efficient probabilistic drop is taken. Explicitly, the packet drop probability is proportional to its distance apart from the destination. In other words, 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 that can meet the required velocity. Since a packet near the destination has already traveled a long way along the routing path and many nodes have consumed energy to relay it, it is worthwhile to try the best and see whether we can finally deliver it successfully. Despite 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 may 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 it earlier and look for a better chance in the coming retransmission.

The probabilistic drop policy is defined in the following way. 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.

We will show in Section 4 the consequent difference 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 outperforms the other two approaches under a joint consideration of DMR and energy efficiency.

3.4. Cost function for energy balance

In SPEED-S, no strategy for energy balance is considered. Some nodes will be frequently chosen as forwarders due to their significant positions in the geographical area. This can be observed from the simulation result reported in Section 4. 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 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}, we sacrifice a certain amount of the expected velocity to have energy consumption balance by considering the node's residual energy and velocity jointly as:

$$ve_i^{j\to k} = \frac{c_v \times \frac{S_i^{j\to k}}{S_{set}} + c_e \times \frac{residual\_energy_j}{initial\_energy_j}}{c_v + c_e} \quad (9)$$

where c_v and c_e are the weights on 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 balance 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 but currently set $c_v = c_e = 1$.

4. Performance evaluation

The effectiveness of THVR 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. The details are described in the coming sub-sections respectively. 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). Each source generates a CBR flow at the rate of 1 packet/s with a payload of 16 bytes.

4.1. MAC settings

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. Otherwise, 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 will be dropped.

4.2. Link quality model

To model lossy links, we abstract the link model from a real experiment based on Mica2 Motes. A sequence of sensor nodes are deployed linearly. Each has a spacing of 0.5 m with one another. We measure the packet loss rate between pairs of nodes at different distance. Each node is scheduled to transmit 80 packets at 10 packets/s in one round and finally the average packet reception rate is computed. At any time, there is always one transmitter and the remaining nodes will count the number of packets successfully received. For each transmitter, we conduct 15 rounds of tests. The result is shown in Fig. 2, which indicates the scatter diagram of how link quality varies with distance as observed by nodes deployed on the ground of an open tennis court. The default transmission power is 0 dBm. For reference, the study in [24] has shown similar pattern. Due to multi-path effects and environmental noise, the link quality takes on the random (non-monotonically) decreasing trend. Receivers which are able to combat these effects may improve the link quality significantly.

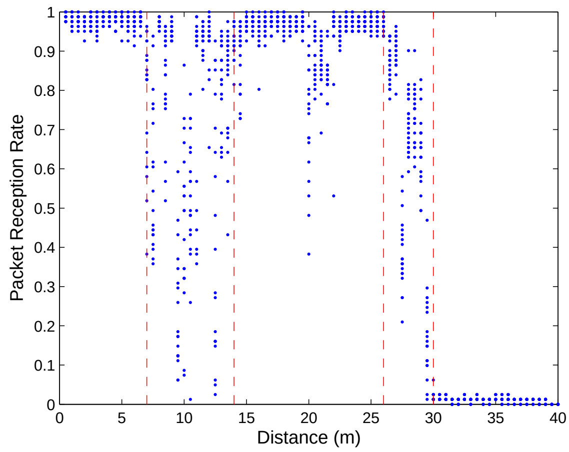

Figure 2. Packet reception rates at different distance.

By the result shown in Fig. 2 and collected statistics, the link quality is a piecewise function of distance d and can be 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 [25]. Table 1 shows the model. 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.

4.3. Energy model

The transmission power is set to 0 dBm. The energy model based on Mica2 Motes [26] is shown in Table 2. When the node is sending a packet, the CPU

Table 1. Link quality model based on Fig. 2

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

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 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 3.2 mA. The respective listening and sleeping time is defined by MAC and depends on the channel state. Initial energy in each node is assumed the same. The energy consumption equals to the product of current, voltage and time duration taken. The voltage supply is by default 3V and assumed 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

4.4. Simulation results

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

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

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

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

To begin with, 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.

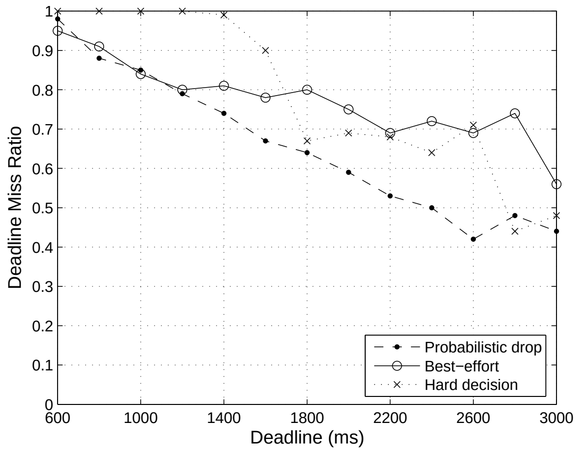

Figure 3. A comparison of packet DMR among the three strategies.

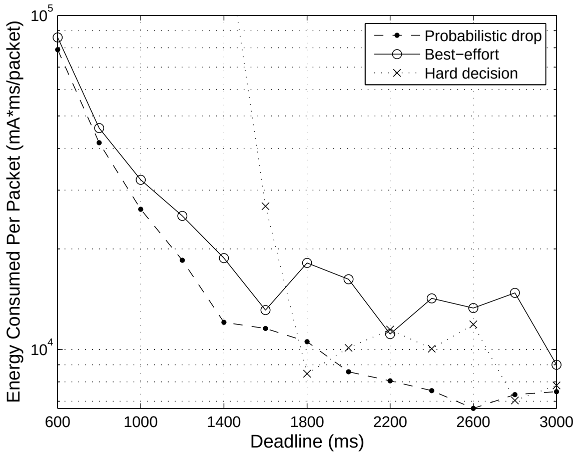

Figure 4. A comparison of energy utilization efficiency among the three strategies.

Fig. 3 shows a comparison of their DMRs under same network topology with 200 nodes and 25 sources. 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. 4 shows their ECPP. It is observed that Fig. 4 has quite similar characteristics and tendency as those shown in Fig. 3. Probabilistic drop is generally much more energy-efficient

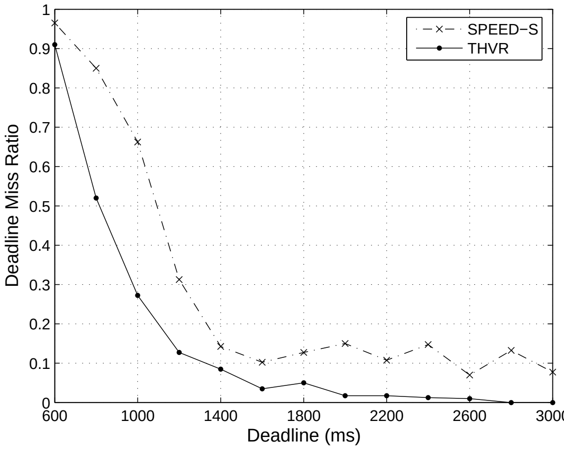

Figure 5. DMR under different deadline requirements. Number of nodes = 200. Number of source nodes = 10.

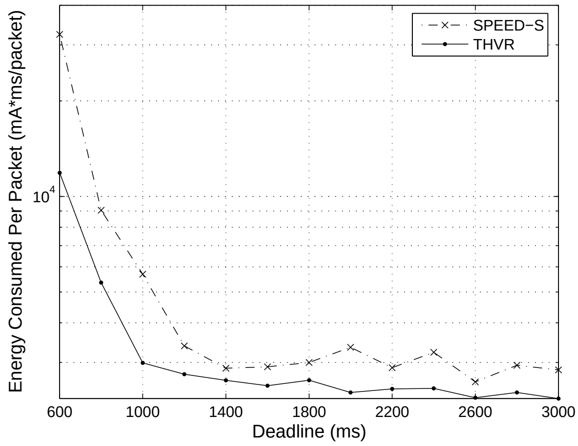

Figure 6. ECPP in comparison.

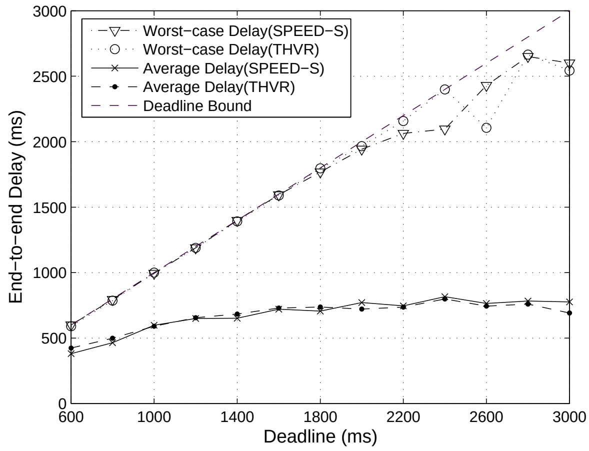

Figure 7. End-to-end delay performance.

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. Although the probabilistic drop scheme is not always the best among the three, by comparing their DMRs and ECPP, it can reach an overall better performance and is more adoptable.

In the following, a detailed performance study of THVR is conducted and compared with SPEED-S. Fig. 5 shows their DMRs 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 THVR, when the deadline is large enough, the DMR converges to zero. In comparison, as shown in Fig. 5, SPEED-S has a much higher DMR generally. Besides, even when the deadline is up to 3000 ms, SPEED-S has only tended to a DMR level of 0.1. Comparatively, the DMR in THVR drops much faster than that in SPEED-S. The result has clearly indicated the effectiveness of THVR upon the 2-hop based routing strategy.

Fig. 6 shows the energy efficiency in THVR compared with SPEED-S. As expected, the ECPP 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-S, THVR has consumed less energy. In other words, it has a higher energy efficiency. One of the major reasons is that THVR can achieve a lower DMR. It is expected and observed that Fig. 6 has similar tendency and convergence characteristics as those in Fig. 5. Generally, THVR outperforms SPEED-S and can converge to a lower energy consumption level as deadline increases.

Furthermore, Fig. 7 shows the packet end-to-end average and worst-case delays respectively. It is observed that THVR and SPEED-S have quite close performance. Generally speaking, when there are several routing paths with delays which can satisfy the required velocity, THVR 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[3]. As shown in Fig. 5, 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-S may have already dropped the packets. Therefore, it is possible that the worst-case or average delay in THVR may be higher than those in SPEED-S by the measurements. This phenomenon is observable in Fig. 7. However, more importantly, as shown in Fig. 7, the worst-case delay is always bounded by the deadline requirement.

To have a comparative study in different network size, we increase the number of nodes from 200 to 400 in the

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

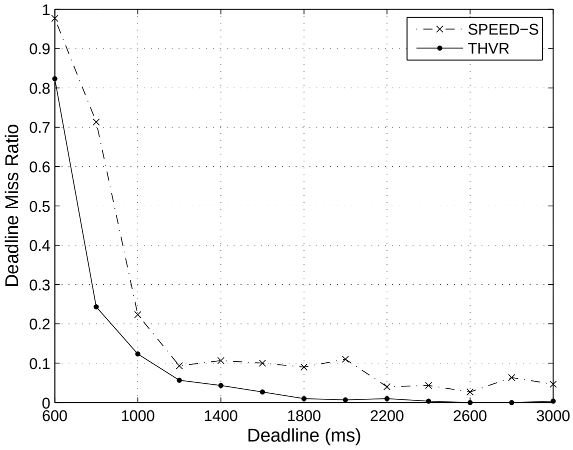

Figure 8. DMR. Number of nodes = 400. Number of sources = 10.

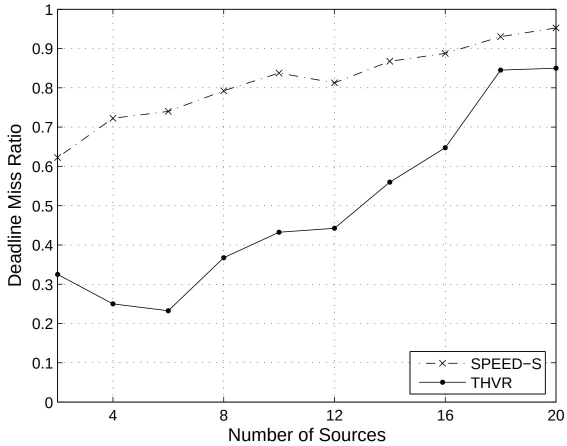

Figure 9. DMR under different number of source nodes. Number of nodes = 200. Deadline is set to 800 ms.

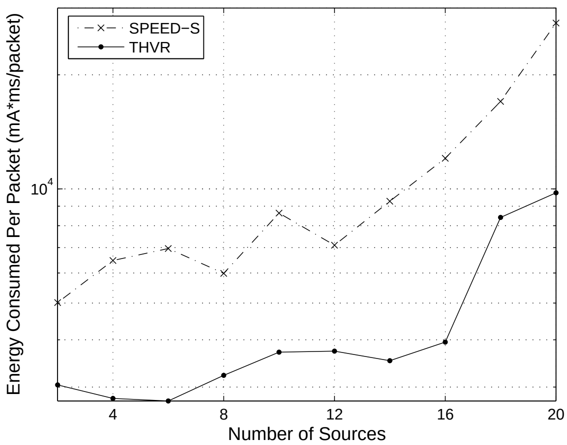

Figure 10. ECPP in comparison.

same area. Meanwhile, the number of source nodes is fixed. Fig. 8 shows the DMR. Results from THVR and SPEED-S are reported. When compared to Fig. 5, Fig. 8 indicates decreased DMR in both THVR and SPEED-S since now there is a higher density of potential forwarding nodes. Intuitively, there are more routing path possibilities to meet the required velocity and consequently the deadline. Moreover, the DMR in Fig. 8 drops faster than that in Fig. 5 and has already converged to a low level at a smaller deadline.

In comparing THVR and SPEED-S in energy consumption and packet delay under 400 nodes, simulation results obtained show that the performance tendency and characteristics are very similar to those in Fig. 5 and Fig. 6 respectively. Due to a lack of space, we will not plot them here. However, it is worth noting that, in both network sizes, THVR outperforms SPEED-S in DMR and also energy utilization efficiency indicated by Fig. 5 and Fig. 6. Meanwhile, the packet end-to-end average and worst-case delay in THVR have very similar performance as in SPEED-S.

In addition, we investigate the performance of THVR under different workload. Fig. 9 shows the DMR as the number of sources is increased from 1 to 20, while the deadline requirement is fixed at 800 ms. In both SPEED-S and THVR, it is observed that the DMR increases as the number of sources increases and so is the energy consumption as indicated in Fig. 10 respectively. The increase in DMR is resulted by the increased channel busy probability, packet collisions at MAC, and network congestion due to the increased number of sources and consequent traffics. However, compared to SPEED-S, THVR has a lower DMR and energy consumption as shown in Fig. 9 and Fig. 10 respectively. This reflects the general improvement by THVR. The packet average and worst-case delay performance in both schemes is approximately at the same level. A plot is thus omitted due to the similarity and a lack of space.

Finally, we investigate the performance of the residual energy cost function, which is an optional add-on for node energy consumption balancing in case packet deadline requirement can be relaxed and a relatively large value is allowed. The motivation is: 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. The effectiveness of the strategy is investigated below.

Fig. 11 shows the node distribution and their locations in the 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. 12 and Fig. 13 show the node energy consump-

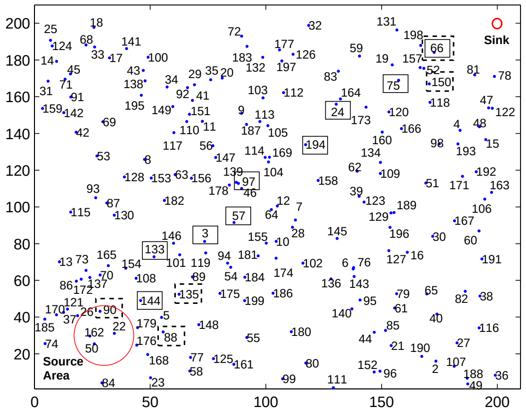

Figure 11. The topology of 200 nodes in the study of energy consumption distribution.

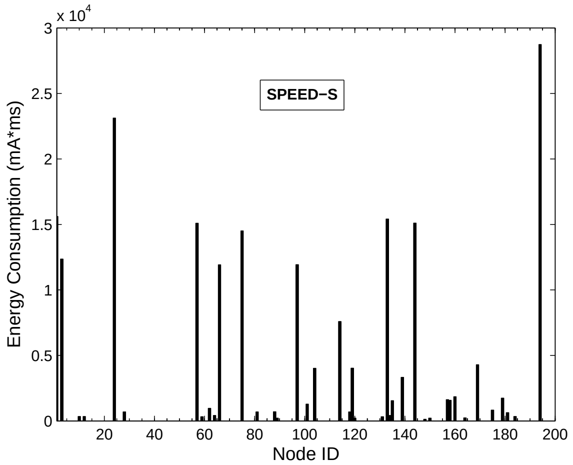

Figure 12. Node energy consumption in SPEED-S. Number of nodes = 200.

tion distribution in SPEED-S and THVR respectively after 200 runs. Some nodes have consumed more energy than the other and we highlight them in solid and dashed rectangles which correspond to SPEED-S and THVR respectively.

It is found that in SPEED-S some nodes along the path from sources to sink are frequently chosen as forwarders and consume much more energy than the other, while in THVR 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. 13 to Fig. 12, energy consumption in THVR is more evenly distributed among those between source and sink. It can be expected that THVR will have a longer system lifetime due to the balancing. However, the cost is the tradeoff in packet delay performance. As shown in Table 3, in this WSN, THVR will have a larger packet average delay by the residual energy consideration. However, even now, the DMR in THVR is still smaller than that in SPEED-S. That is, the DMR which is highly concerned in real-time service has not been sacrificed in the node energy consumption balancing.

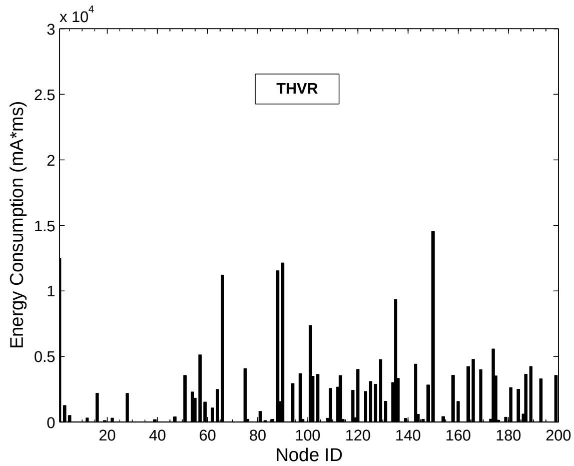

Figure 13. Node energy consumption in THVR. Number of nodes = 200.

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

Routing Protocol	SPEED-S	THVR
DMR	17%	0%
Average Delay (ms)	603.92	963.15
Energy Utility (mA×ms/packet)	2472.3	2486.8

5. Discussion and Future Work

It is worth pointing out 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 despite the fact that 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. A further investigation is expected in a future work.

In our simulation, the deadline requirement is assumed a constant value. For the situation of different deadlines for different packet types, MM-SPEED [18] has designed a prioritized MAC and multi-SPEED routing to provide service differentiation. In [15], RT and non-RT packets are separated with classifier and assigned different bandwidth according to different priorities. Cross-layer method integrated with priority scheduling can be considered to our design which currently does not include service differentiation with prioritized MAC techniques.

6. Conclusion

In this paper, we propose 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 used to save energy while reducing DMR. In case packet deadline requirement is not stringent, a mechanism is embedded that can release the nodes which are frequently chosen as the forwarder. An improvement on energy consumption balance throughout the network is achieved. The true characteristics of physical and MAC layers are captured in the simulation. A real lossy link model is drawn from experiments through Mica2 Motes. Simulation results show that, compared with SPEED-S that only utilizes 1-hop information, THVR achieves lower end-to-end DMR and higher energy efficiency. In a future work, we are interested to see how to support differentiated service and keep the required information exchange in a minimum necessary amount. More comparisons will be addressed. The results reported here may also lead to other interesting design and schemes.

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, 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] N. Aakvaag, M. Mathiesen, and G. Thonet, “Timing and power issues in wireless sensor networks - an industrial test case”, in *Proc. ICPPW*, 2005, pp. 419–426.

[4] N. Ota and P. Wright, “Trends in wireless sensor networks for manufacturing”, *Int. J. Manufacturing Research*, vol. 1, no. 1, pp. 3–17, 2006.

[5] Y. Li, Z. Wang, and Y. Q. Song, “Wireless sensor networks for wildfire monitoring”, in *Proc. WCICA*, June 2006, pp. 109–113.

[6] J. Stankovic, “Wireless Sensor Networks”, *Handbook of Real-Time and Embedded Systems*, CRC, 2007.

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

[8] 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.

[9] 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, 2002.

[10] C. S. Chen, Y. Li, and Y. Q. Song, “An exploration of geographic routing with k-hop based searching in wireless sensor networks”, *submitted*, 2008.

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

[12] 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.

[13] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, “A spatiotemporal communication protocol for wireless sensor networks”, *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 995–1006, 2005.

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

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

[16] 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.

[17] N. Ota and P. Wright, “A new routing metric for satisfying both energy and delay constraints in wireless sensor networks”, *J. VLSI Signal Processing*, 2007.

[18] 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, 2006.

[19] 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.

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

[21] 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.

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

[23] V. Jacobson, “Congestion avoidance and control”, in *Proc. ACM SIGCOMM*, 1988.

[24] 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.

[25] 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.

[26] 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.