

## Research Article

# Delay-Driven Routing for Low-Duty-Cycle Sensor Networks

**Zuzhi Fan**

*College of Information Science & Technology, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Zuzhi Fan; [zuzhifan@gmail.com](mailto:zuzhifan@gmail.com)

Received 3 May 2013; Revised 24 July 2013; Accepted 19 August 2013

Academic Editor: Davide Brunelli

Copyright © 2013 Zuzhi Fan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Duty-cycled operation has been introduced as an efficient way to preserve nodes energy and prolong network lifetime for wireless sensor networks. However, such networks are often logically disconnected since there is a limited number of active nodes within a period of time. Traditional routing algorithms, which have been designed for always-awake wireless networks, suffer excessive waiting time incurred by asynchronous schedule of nodes and cannot be applied to these time-dependent sensor networks. In this work, we study the optimization of delivery delay for low-duty-cycle sensor networks. Specially, we theoretically analyze the sleep latency in low-duty-cycle networks and present a new routing metric, which takes both lossy link and asynchronous schedule of nodes into consideration. Based on the metric, we propose delay-driven routing algorithms to find optimal forwarder in order to reduce delivery delay for source-to-sink communication. We compare our design against state-of-the-art routing algorithms derived in wireless networks through large-scale simulations and testbed experiments, which show that our algorithms can achieve a significant reduction in delivery delay.

## 1. Introduction

Wireless sensor network (WSN) is supposed to be a potential solution for many surveillance applications deployed in unmanned or hostile environment, such as habitat and ecological monitoring [1–4], disaster and scientific exploration [5, 6]. Many of these applications are designed for long-term existence, which requires battery-powered sensor to survive several months even years. However, it is too expensive even infeasible to replace or replenish energy-depleted nodes in large scale. To reduce the gap between application requirements and limited energy supply, duty-cycled operation is introduced as an economical way to preserve energy [1], with which sensor nodes wake up briefly and then sleep for a long period of time. Consequently, large amount of energy consumed in idle state, which is a state that transceiver of node keeps on listening to surrounding channel in order to be ready for possible incoming packets, can be greatly reduced.

However, it is impossible to maintain an always-awake communication backbone in such duty-cycled networks. Firstly, most of power management protocols [7, 8] are proposed to conserve energy, where the minimum number

of sensor nodes are scheduled to be active at certain time. Secondly, nodes are often asynchronously scheduled in order to meet the requirement of coverage or connectivity [9, 10]. In other words, the connectivity of network is often related to the spatial-temporal distribution of sensor nodes. Thus, the existing routing schemes are impractical as they are designed based on the assumption that nodes are always awake, and they can overhear channels and receive packets from neighboring nodes. Consequently, sleep latency [8] instead of energy-efficiency is most significant for low-duty-cycle sensor networks.

In this paper, we attempt to study the optimization of delivery delay from the perspective of routing mechanism, which can be applied to randomly scheduled network topology with asynchronously duty-cycled nodes. We first introduce *expected sleep latency* (ESL) as a routing metric for source-to-sink communication in such time-dependable networks. In particular, we theoretically analyze the sleep latency for both single-hop and multihop data forwarding under unreliable link quality and then present the new metric for delay optimization in low-duty-cycle sensor networks. Based on the metric, we propose local-optimized routing and global-coordinated routing algorithm to reduce sleep

latency for source-to-sink communication. To validate our design, extensive simulations and testbed experiments are conducted.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents network model and assumptions. In Section 4, we propose the new metric and routing algorithms. Simulation and testbed results are discussed in Sections 5 and 6, respectively. Section 7 concludes the paper.

## 2. Related Work

Different routing algorithms have been proposed to optimize network performances in wireless networks. In geographic routing [11], data packets are often forwarded via farthest neighboring node to minimize hop distance between source and destination. To handle lossy link in sensor networks, Zamalloa and Seada investigate distance-hop tradeoff for geographic routing and show that the product of packet reception rate and distance traversed to the destination is an optimal value to decide data forwarding path [12]. De Couto et al. introduce expected transmission count (ETX) as a metric to find high-throughput path in multihop wireless communication [13]. However, all these schemes are designed for the purpose of optimizing network performances in always-awake networks.

On the other hand, many power management approaches have been proposed to schedule nodes in order to conserve energy for low-duty-cycle networks. Though these schemes are generally application-dependent, they can be classified into coordinated duty-cycling or random duty-cycling. The prior is often designed to meet specific requirements of application [7, 8]. For example, Cao et al. present both sleep scheduling scheme and streamlined forwarding technique towards the detection of rare events in sensor networks [8]. With randomly wake-up schedule, sink-to-node communication delay is bounded with active-slots augmentation in duty-cycled sensor networks [14]. Gu et al. also argue that delivery delay could be reduced if more sink nodes are introduced into network [15]. Keshavarzian et al. firstly introduce multiparent forwarding mechanism to optimize delay [16]. More recently, Gu and He extend such multiparents forwarding technique to duty-cycled sensor networks [17]. The rationale behind is that a forwarding set consisting of multiple potential candidates rather than single node is selected to relay data packets at each hop, by which waiting time can be reduced in case of transmission failure. Our work is focused on routing strategy for delay optimization in low-duty-cycle sensor network, which is complementary or orthogonal to these works.

Some other works are also focused on schedule or data forwarding problem to trade off delivery delay and energy efficiency [18–24]. Both Su et al. [18] and Lai and Ravindran [19] study the minimum-latency routing problem in time-dependent sensor networks. However, their works are based on the assumption of perfect link and reliable transmission. An opportunistic routing scheme (ORW) [20] is proposed to balance delay and energy consumption, where low-power-listening technique is used. Similar to our work, a metric

referred to as EDC is proposed for opportunistic routing in duty-cycled sensor network [21]. Instead of delay optimization, EDC is presented for the purpose of energy-efficiency improvement.

## 3. Model and Assumptions

In this section, we describe the notation of sleep latency and network model for low-duty-cycle sensor networks.

*3.1. Delay Model.* In duty-cycled sensor networks, all nodes decide and share their own working schedules, which are usually asynchronous among neighboring nodes in order to reduce information redundancy [8]. Guided by such working schedule, a sensor node is either in an active state or a dormant state at a given slot. For effective communication, a node has to transmit a packet when its receiver is in the active state. In this case, the connectivity among nodes becomes time-dependent since their neighboring nodes consistently switch between active and dormant states. Formally, we can denote the duty-cycled network at a given time  $t$  as a time-dependent graph,  $G(t) = (V, E(t))$ , where  $V$  is the nodes set of whole network and  $E(t)$  is the set of directed edges at time  $t$ . A directed edge  $e_{ij}(t)$  belongs to  $E(t)$  if and only if nodes  $i, j$  are within each other's communication range, and the receiver is in the active state so that it can receive packets from the sender [17].

Note that sensor nodes often have periodical schedule for sensing purposes in realistic applications [8]. Without loss of generality, suppose the duration of working period to be  $T$ , we can divide it into a sequence of time slots with equal length  $\tau$ , during which sensor node is in either active or dormant. Consequently, the working schedule for node  $i$ ,  $\Gamma_i$  can be uniquely represented as a list of active slots, that is,  $\Gamma_i = \{t_1^i, t_2^i, \dots, t_K^i\}$ , where  $K$  is the number of active slots within whole period.

We define *sleep latency*,  $s_{ij}(t)$ , as the waiting time from the moment ( $t$ ) that the sender  $i$  has a packet being ready to the moment that the receiver  $j$  is in the receiving state. To elaborate, we give a sample topology shown in Figure 1(a) where nodes A, B, and C wake up and sense periodically according to the schedule depicted in brackets given whole period  $T = 10$ . Figure 1(b) illustrates the communication diagram between node A and nodes B, C. For example, the sender A is ready to send packet at 3 but has to buffer it for a period of time until the receiver B wakes up at slot 7. Thus, the sleep latency between node A and node B is  $s_{AB}(3) = 7 - 3 = 4$  since the packet is actually received by node B at slot 7. Consequently, the sleep latency of whole forwarding path from A to D through B is  $s_{AD}(3) = 5$  if all links are assumed to be reliable. However, the sleep latency from node A to node D via node C could be different since node A has to wait for the chance of transmission in C's next period. In fact, the delay from node A to node D through C is  $s'_{AD}(3) = 15$ .

Note that sleep latency in low-duty-cycle sensor network is usually far more larger than other delivery delay, such as processing, transmission, or propagation delay in traditional sensor networks, which are normally in the order of milliseconds. In this paper, we suppose that sleep latency and

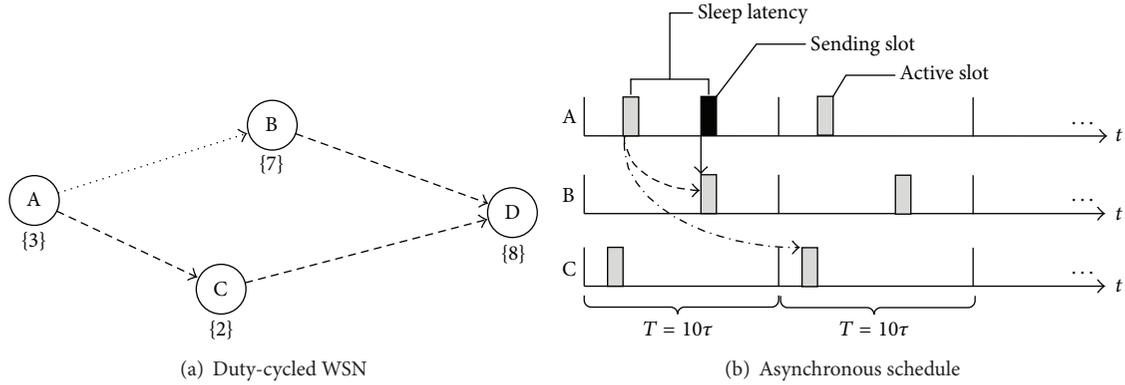


FIGURE 1: Duty-cycled sensor network.

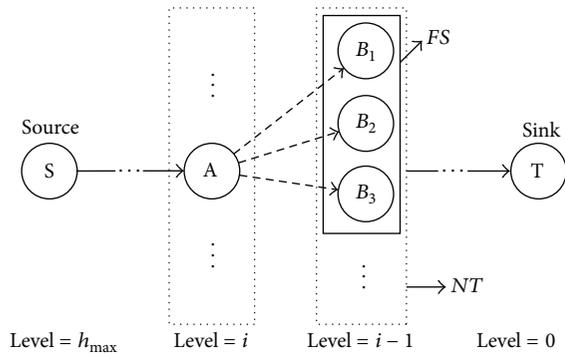


FIGURE 2: Traffic model.

delivery delay are equivalent notions in low-duty-cycle sensor networks.

**3.2. Network and Traffic Model.** Different from traffic model in wireless ad hoc networks, data collection is the main communication model for sensor networks, where data packets converge from many source nodes to a few sink nodes. To assure loop-free data forwarding, we assume a traffic model shown in Figure 2 where data packets always flow from lower level (with larger hop count) to the sink through those nodes in higher level. In particular, we can formulate a forwarding set (FS) for each node, which consists of neighbors with less hop count to the sink. Thus, routing module will only select next hop from the forwarding set. As the example shown in Figure 2, the forwarding set of node A,  $FS_A = \{B_1, B_2, B_3\}$  is actually the subset of neighboring table.

**3.3. Assumptions.** Suppose that the source node has data packets to be sent to the sink node periodically; we make following assumptions.

- (1) The discussion of work scheduling in the sensor network would be irrelevant if there were not some synchronizing mechanisms [25]. We assume that the network is locally synchronized so that a node knows

when it can send packet with the information of neighbors' working schedules.

- (2) It is reasonable to assume that there is low data traffic in low-duty-cycle sensor nodes since the sleep period of nodes is relatively long. Consequently, the delay such as queuing latency due to transmission interference or network congestion can be trivial.
- (3) We assume the existence of wireless transmission failure. The probability of a successful transmission through a wireless communication link is quantified by link qualities, which can be measured by prob-based approaches in [13, 26] or through low-cost piggybacking on normal data traffic. Our design does not depend on the specified link model. For simplicity, a realistic channel model derived in [27] is utilized for the simulations. The model considers several environmental and radio parameters, such as path-loss exponent, log-normal shadowing variance, the modulation, and encoding schemes of the radio.

## 4. Main Design

The main goal of our design is to find optimal forwarder for each node so that the delivery delay from source to the sink node could be reduced.

**4.1. Expected Sleep Latency.** In this section, we present a new metric for routing decision in low-duty-cycle sensor networks. For simplicity, we first consider single wake-up schedule, where each node wakes up only once within the whole period ( $T$ ).

Assuming that the working time of nodes A, B is  $t_A, t_B$ , respectively, shown in Figure 3, the ESL from node A to node B can be represented as the following if the link is reliable (i.e.,  $q = 1$ ):

$$S_1 = \begin{cases} t_B - t_A, & t_B \geq t_A \\ t_B - t_A + T, & t_B < t_A. \end{cases} \quad (1)$$

However, wireless link is notoriously unreliable in extremely dynamic environments. Thus, the estimation of sleep

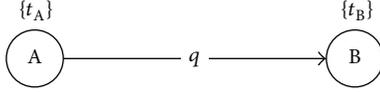


FIGURE 3: Sleep latency over single hop.

latency should take transmission failure into consideration. In case of failure, the sender has to wait for whole working period to get the next chance of retransmission. The sleep latency of the  $k$ th transmission,  $S(k)$ , can be calculated as the following formula:

$$S(k) = S_1 + (k - 1)T, \quad (k \geq 1). \quad (2)$$

Note that the  $k$ th transmission happens only when the first  $(k - 1)$  times of transmission fail. On the other hand, small probability event may happen when the total  $N$  attempts fail, where  $N$  is the maximum number of allowable retransmissions for each hop. In this case, the packet would be dropped by the sender. Given the link quality,  $q$ , the conditional probability of the  $k$ th transmission,  $PR(k)$ , can be represented as follows:

$$PR(k) = \frac{(1 - q)^{k-1} q}{1 - (1 - q)^N}. \quad (3)$$

As a system parameter,  $N$  can be configured to trade off delivery ratio and energy cost. With a larger value of  $N$ , we can achieve a higher delivery ratio at the cost of more energy. In particular, we can confine  $N$  with a parameter  $\delta$  which denotes the expected delivery ratio of one-hop transmission. With given  $\delta$ ,  $N$  should satisfy the following constraint:

$$\sum_{k=1}^{N-1} PR(k) \leq \delta \leq \sum_{k=1}^N PR(k). \quad (4)$$

Thus, The ESL, for a packet to be successfully delivered can be calculated as

$$ESL = \sum_{k=1}^N \{S(k) PR(k)\}. \quad (5)$$

For candidates with multiple wake-up slots, the average waiting time of retransmission decreases since there is no necessary to wait for the whole period. Assuming that neighboring node  $j$  has  $n$  wake-up slots, that is,  $\Gamma_j = \{t_1, t_2, \dots, t_n\}$ , the sleep latency of  $k$ th transmission,  $S(k)$ , can be approximated with

$$S(k) = \begin{cases} |t_m - t_0| + \frac{Tk}{n}, & m \neq 0, \\ |t_n - t_0| + T\left(\frac{k}{n} - 1\right), & m = 0. \end{cases} \quad (6)$$

Here,  $m$  is the reminder of  $k$  divided by  $n$  ( $1 \leq k \leq N$ ). Specially, if wake-up slots of forwarding candidate follow even-distribution, the average waiting time of data transmission is approximated to  $T/n$ .

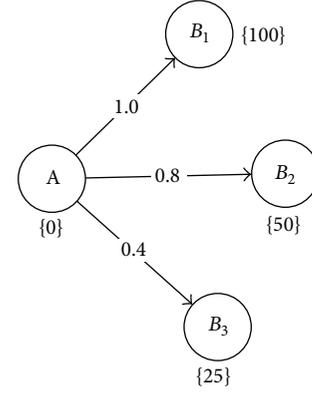


FIGURE 4: Example of routing strategies.

At last, suppose that source node ( $v_s$ ) send packets to the sink ( $v_t$ ) over  $L$ -hops path,  $R^{s,t} = \{v_0, v_1, \dots, v_L\}$ , where  $s = 0$ ,  $t = L$ , and the expected end-to-end delay (EED) can be calculated as follows:

$$EED = \sum_{i=1}^L ESL(i, i + 1). \quad (7)$$

Obviously, the E2E delay is also dependent on the length of routing path ( $L$ ).

**4.2. A Walkthrough for Expected Sleep Latency.** In traditional always-awake networks, a sender can send its packets to next hop once data packets are ready to be sent. Ignoring network congestion and transmission collision, the E2E delay is proportional to the length of routing path. In this case, the shortest path routing is efficient for delay minimization. ETX has been proposed as a metric to find high throughput path in lossy wireless networks [13]. On the other hand, DESS is designed to reduce delivery delay by prioritizing the neighbor with earliest wake-up time [22].

We demonstrate the weakness of these existing routing algorithms by an example as shown in Figure 4, where node A is an always-awake node. Suppose that the wake-up time of its candidates  $B_1$ ,  $B_2$ , and  $B_3$  is 100, 50, and 25 and link qualities are 1, 0.8, and 0.4, respectively, we calculate the ESL according to the above metric for different routing policies. According to ETX, node A will select node  $B_1$  (with the best link quality) to forward, and then sleep latency is 100. For DESS algorithm, node A intends to select node  $B_3$  which has minimum waiting time among  $B_1$ ,  $B_2$ ,  $B_3$ ; the delay value of first transmission is 25. Once failed (with a high probability of 60%), the delivery delay could be longer since node A needs to wait for a whole period ( $T = 100$ ) to obtain the chance of retransmission. Suppose that infinite retransmissions could be performed; the ESL values of successful delivery through three candidates  $B_1$ ,  $B_2$ , and  $B_3$  are 100, 75, and 175, respectively. In order to minimize delivery delay, data forwarding via neighbor  $B_2$  might be the best choice in this example.

**4.3. Delay-Driven Routing Algorithms.** Based on the metric, we propose (i) a local-optimized routing algorithm by which each node select next hop to reduce one-hop sleep latency

```

Input: Forwarding Set,  $FS(i)$ 
Output: the optimal forwarder,  $OptFwder$ 
(1)  $\min ESL \leftarrow \infty$ 
(2)  $OptFwder \leftarrow -1$ 
(3) for  $j \leftarrow 1$  to  $n$  do
(4)   calculate  $ESL$  from node  $i$  to candidate
       $B_j$ , ( $B_j \in FS(i)$ ) using (5);
(5)   if  $ESL(i, j) < \min ESL$  then
(6)      $\min ESL \leftarrow ESL(i, j)$ 
(7)      $OptFwder \leftarrow B_j$ 
(8)   end if
(9) end for
(10) return  $OptFwder$ 

```

ALGORITHM 1: ESL algorithm at node  $i$ .

(referred to as ESL); (ii) a global-coordinated routing algorithm which can minimize the ESL from current sender to the sink node (referred to as MSL).

In ESL, we utilize the proposed metric in a way as the example explained in Section 4.2. As shown in Algorithm 1, the sender firstly evaluates the ESL value for each candidate according to Equation (5) and then selects neighbors with minimized ESL as the optimal forwarder [line 3–9]. Instead of building whole optimal forwarding path, the ESL algorithm is completely localized and distributed.

In MSL, we propose routing algorithm to find optimal path with minimized end-to-end delay from each node to the sink. The expected E2E delay from node  $i$  at level  $h$  to the sink, denoted by  $EED_i(h)$ , is the ESL for the packets sent by node  $i$  and received by the sink node over multihop path. Formally, we can evaluate the minimized EED in a recursive way:

$$EED_i(h) = \min_{j \in FS(i)} (EED_j(h-1) + ESL(i, j)). \quad (8)$$

Note that the above calculation of E2E delay can be started from the sink and then propagated towards the rest of network. For sink node, its ESL is zero since no further data forwarding is necessary once the packet arrives at the sink node. The routing algorithm for each node except the sink node is shown in Algorithm 2. Similar to Dijkstra's algorithm (but use sink as destination), the algorithm begins with the broadcasting of EED value by sink node. At first, each node sets its minimized EED to  $\infty$  [line 1]. After receiving the broadcasted values from neighbors, each node starts to compute minimized EEDs based on Equation (8). With multiple iterations, the process converges at a node until it receives no more information from all neighbors [see line 2–12]. Finally, the sender selects potential candidate with minimized EED as its next hop.

In fact, we can build a spanning tree with minimized E2E sleep latency, with which each source can forward data packets via its parent to minimize delivery delay.

## 5. Extensive Simulation

In this section, we evaluate and compare performance of proposed routing algorithms with extensive simulations.

**5.1. Baseline of Routing Algorithms.** To verify the effectiveness of our design, we compare the proposed algorithms with the state-of-the-art derived in wireless networks.

- (i) Geographic-based routing algorithm [12] (termed PRRxD): the product of the packet reception ratio (PRR) and the distance traversed towards destination (D) is selected as the optimal forwarding metric in order to improve energy efficiency.
- (ii) ETX-based routing algorithm [13] (termed ETX): the ETX can be represented as the reciprocal of link quality. Note that multiple neighbors may have the same link quality in dense networks, and the farthest one among them is selected.
- (iii) DESS-based algorithm [22] (termed DESS): original DESS is proposed based on perfect channel model. To deliver packets successfully, we assume that retransmissions can be performed via the wake-up earliest candidates. If one or more neighbors wake up at the same time, the one with better link quality is chosen.

**5.2. Simulation Setup and Performance Metrics.** Without specification, 400 nodes are randomly deployed in a 200 m × 200 m network, where the source and sink node are located in the opposite of sensor area. The link qualities between node pairs are set according to channel model proposed in [27], where radio parameters strictly follow Chipcon CC2420 hardware specification [28]. Each experiment is repeated 10 times with different random seeds, node deployments, and working schedules. Simulation results are obtained with the average value of 100 source-to-sink data transmissions.

In the simulation, we measure *delivery delay* as period of time from a message being sent out by the source until it reaches the sink node. Since energy consumed by data reception for most wireless transmitter is close to that of idle state [28], *energy cost* is evaluated by the number of transmissions. For all experiments, the expected delivery ratio of one-hop ( $\delta$ ) is set to 0.8.

### 5.3. Performance Comparison

**5.3.1. Different Network Sizes.** We evaluate the performance of all algorithms in different network sizes. For side length

```

Input: Forwarding Set,  $FS(i)$ 
Input: Minimized EED from candidate  $B_j$  at level  $h - 1$  to
the sink,  $EED_j(h - 1)$ 
Output: the optimal forwarder,  $OptFwder$ 
(1)  $\min EED_i(h) \leftarrow \infty$ 
(2)  $OptFwder \leftarrow -1$ 
(3) for  $j \leftarrow 1$  to  $n$  do
(4) calculate ESL from node  $i$  to candidate
 $B_j (B_j \in FS)$  using (5);
(5) if  $EED_j(h - 1) \neq \infty$  then
(6)  $d \leftarrow (EED_j(h - 1) + ESL(i, j))$ 
(7) end if
(8) if  $d < \min EED_i(h)$  then
(9)  $\min EED_i(h) \leftarrow d$ 
(10)  $OptFwder \leftarrow B_j$ 
(11) end if
(12) end for
(13) Broadcast the minimized value,  $\min EED_i(h)$ ;
(14) return  $OptFwder$ 

```

ALGORITHM 2: MSL algorithm at node  $i$ .

from 100 to 300, the number of nodes is set from 100 to 900 in order to keep the same network density.

Figure 5(a) shows that the E2E delay increases as the size length of network. In particular, MSL reduces delay by a factor of 6 times compared with ETX. Even for DESS, our algorithm can obtain a factor of 2 times delay reduction. Note that ESL is suboptimal in delivery delay but still can obtain a large delay gain compared with other algorithms. For example, the delivery delay of ESL is 50% less than that of DESS when the size length is 150. Energy cost, quantified by number of transmissions, is plotted in Figure 5(b). Since PRRxD is proposed to optimize energy efficiency, it has the lowest energy consumption among all algorithms. Based on the global optimization, MSL is also energy efficient compared with ETX and DESS. On the contrary, ESL consume a little more energy because the average length of routing path is longer compared with other algorithms.

**5.3.2. Different Network Densities.** To evaluate the performance in different network densities, 200~800 nodes are randomly deployed on a  $200\text{ m} \times 200\text{ m}$  field. Figure 6(a) shows that the EEDs in all algorithms decrease as network density. Again, both MSL and ESL can reduce delivery delay compared with other algorithms. Figure 6(b) shows that DESS and ESL consume more energy and the change of network densities has little impact on them. The energy efficiency of ETX is monotonously increasing since the average link quality can be improved with more neighbors. For MSL and PRRxD, there is a turning point near 600 nodes, from which the energy dissipation keeps unchanged. In other words, more neighbors has no influence on the energy consumption.

**5.3.3. Different Duty Cycles.** We change working period of nodes ( $T$ ) from 100 to 500 and obtain the corresponding duty-cycle from 1% to 0.2%. Figure 7(a) demonstrates that

delivery delay is monotonously decreasing with duty cycles for all algorithms. For example, the E2E delay for ESL decreases from 890 to 128, while the duty cycle increases from 0.2% to 1%. On the other hand, the number of transmissions keeps stable for all algorithms as shown in Figure 7(b). Note that duty-cycled operation saves energy consumed by idle listening instead of data transmission. The number of transmissions for PRRxD and MSL is smaller than that of other algorithms. According to our investigation, both of them prefer to send packets along reliable routing path.

**5.3.4. Impact of Link Quality.** In this section, we investigate the impact of link quality on the network performance. To set the given link qualities, we first used CC2420 radio specification and delay model derived in [27] to build the neighbor table for each sensor node and then generated the pairwise link quality according to the simulation settings.

Figure 8 shows that both E2E delay and energy consumption decreases as the average link quality. It is because the waiting time of retransmissions is reduced when the average link quality increases. In particular, the average delivery delay for DESS, ESL, and MSL is converging to the same point when the link quality approximates to perfection. In this case, all of them tend to select the earliest wake-up neighbors as the optimal forwarder. On the other hand, MSL instead of PRRxD has the lowest energy cost since the prior tends to select short path with better link quality. The energy consumption for ETX, DESS, and ESL is very similar if we take the path length and link quality into consideration. At last, Figure 8(c) shows that the delivery ratio from source nodes to the sink node for all schemes increases when the average link quality grows. Also, MSL has the best delivery ratio among them.

**5.4. Investigations.** To investigate the rationale behind our design, we recorded average number of hops and delivery

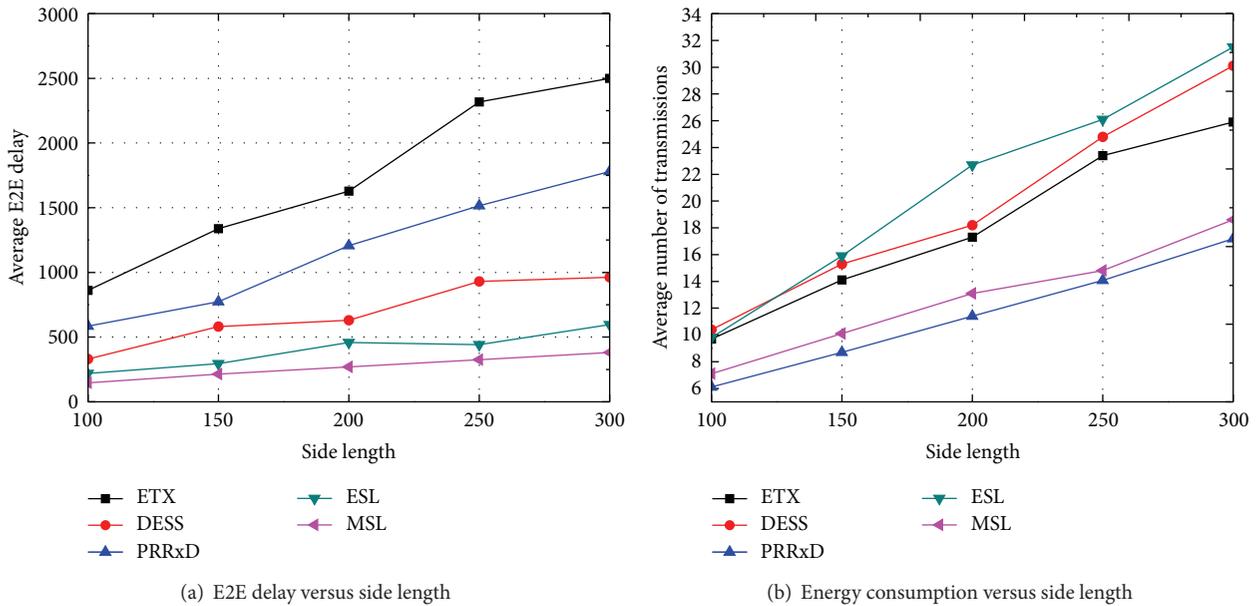


FIGURE 5: Network length.

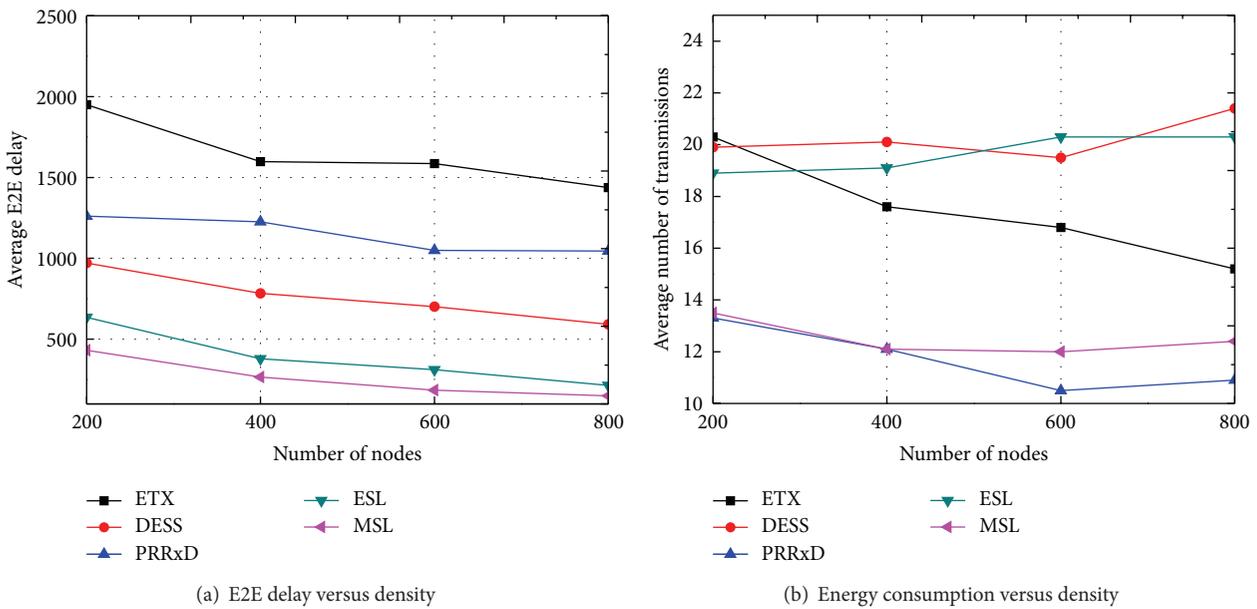


FIGURE 6: Network density.

delay per hop taken along the routing path in Tables 1 and 2, respectively.

Table 1 shows that PRRxD and MSL have shorter routing path than the other algorithms. For example, the average path length for PRRxD and MSL is 5.8 and 6.7, while the side length of network is 100. On the other hand, it takes more than 9 hops for other algorithms to deliver the packets. In particular, ESL has the longest forwarding path and consumes more energy.

As shown in Table 2, both MSL and ESL can minimize the average sleep latency per hop. For example, the sleep latencies per hop for them are both 10 when the working

period is 100, while the value is as large as 55 for PRRxD. Consequently, ESL and MSL outperform other algorithms in delivery delay. Taking both the length and single-hop delay into consideration, MSL can provide the best delivery performance for low-duty-cycle sensor networks.

## 6. Testbed Implementation

In this section, we implement ETX, DESS, and ESL on the TinyOS/Mote platform with micaZ nodes. The modules of algorithm include initialization, link quality measurement, working schedule sharing, metric evaluation, and data forwarding module.

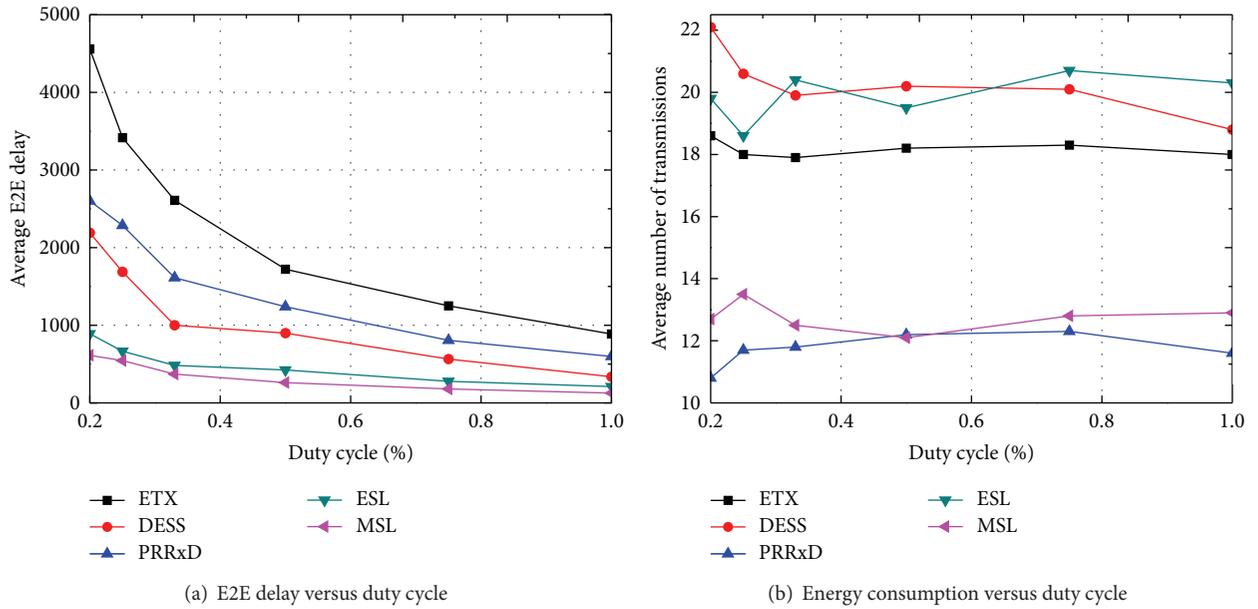


FIGURE 7: Duty cycles.

TABLE 1: Investigation of path length.

Side length	Average path length				
	PRRxD	ESL	ETX	DESS	MSL
100	5.8	9.7	9.7	9.3	6.7
150	8.2	15.8	14.1	13	9.6
200	10.8	22.5	17.3	16.1	12.7
250	13.4	26	23.4	21.6	14.6
300	16.22	31.3	25.9	27.2	18.2

TABLE 2: Investigation of delay per hop.

Working period ( $T$ )	Average delay per hop				
	PRRxD	ESL	ETX	DESS	MSL
100	55	10	48	19	10
200	108	22	94	54	22
300	142	23	145	56	30
400	208	35	189	97	42
500	255	45	245	116	50

## 6.1. Implementation

**6.1.1. Initialization.** At the boot-up phase, each node keeps awake and begins to exchange control messages in order to discover its neighbors. Next, initiated by the sink node which broadcasts beacon packets with hops-count equaling zero, nodes which receive such packets select the minimum hop count and add one as their own hop count and then broadcast packets to neighbors except the sender. This phase works similarly to breadth-first search algorithm until convergence is achieved.

**6.1.2. Link Quality Measurement.** Each node needs to know link quality with its neighboring nodes in order to calculate

metrics. To measure link quality, each node sends a number of packets to each of its neighbors and utilizes acknowledge from MAC layer to evaluate the pairwise link quality. To reduce communication overhead, the four-bit link estimator can be used for such low-power sensor networks [13].

**6.1.3. Working Schedule Sharing.** Each node sets up its working schedule individually and then share with its neighbors. In our implementation, we randomly generate working schedules on a base station and then broadcast them by sink node.

**6.1.4. Metric Evaluation.** Evaluation component is the kernel of our design, where each individual node executes the given algorithm with different metric. For MSL algorithm, Algorithm 2 is performed, while the node receives updated information from its neighboring nodes. For other local algorithms, such as ESL, only neighbor information is needed for the evaluation of routing metric. After this process, each node can find the optimal forwarder via which the packet can be delivered to the sink.

**6.1.5. Data Forwarding.** Data forwarding component can be used by all routing protocols. When the source node has packets ready to send, it can forward them via the selected candidate. If failure happens, the node attempts to transmit again until the delivery succeeds or the packet is discarded when the maximum allowable number of retransmissions has been performed.

**6.2. Performance Validation.** In the experiments, 30 micaZ nodes are uniformly deployed along with the walkway in our office building. We adjust the transmission power of these nodes so that they form a maximum 7-hop network. At the startup, each node performs initialization, measures link

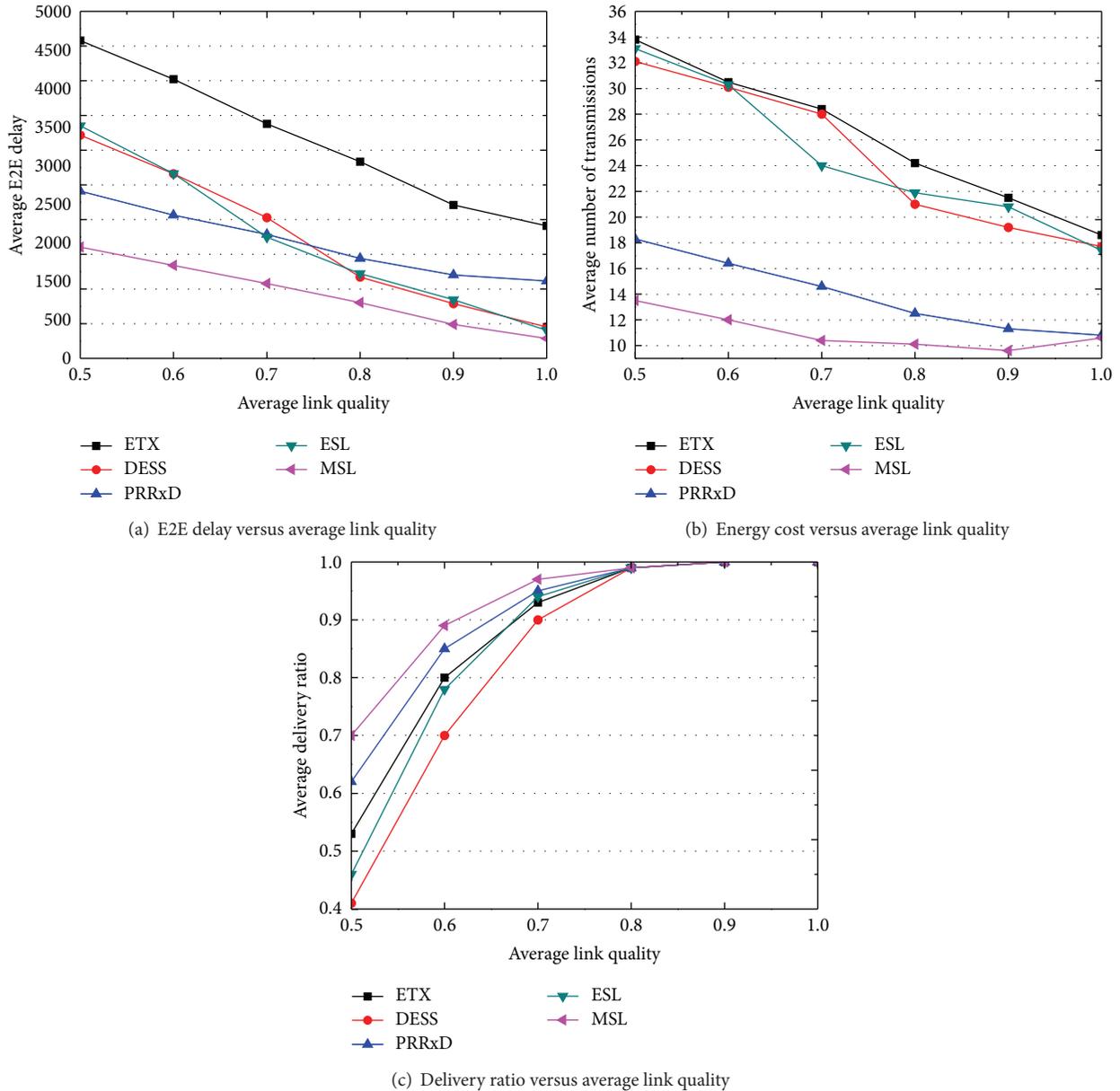


FIGURE 8: Impact of link quality.

quality, and receives working schedule. Next, they find the optimal forwarder with given routing algorithm. To minimize the effect of temporal link quality, the farthest nodes (7-hop away from the sink node) send 100 packets to the sink with different algorithms in turn during the data forwarding phase.

Figure 9(a) reports the E2E delay for all three algorithms. In the experiment, the working period of nodes is set to  $100\tau$ ; that is, the node operates at 1% duty cycle. Each time slot is set to 20 ms, and then the whole working period takes up 2 s. From the figure, we can see that more than 65% of packets can be delivered within three working periods for ESL, while the ratios for ETX and DESS are 52% and 49%, respectively.

This is because ESL algorithm selects the optimal forwarder to reduce the ESL.

Figure 9(b) plots cumulative distribution of the number of transmissions from the source nodes to the sink, which shows that our algorithm takes a little more energy cost than ETX but less than DESS. For example, with 9 transmissions, our algorithm can deliver 93% of packets, while ETX can achieve 95% of packet transmissions. As we discussed, both ESL and DESS send packets via certain bad links. However, we found that the length of longest path for DESS is 14, while the length for ESL is 11, which verifies that ESL can make a better tradeoff between link quality and delay gain.

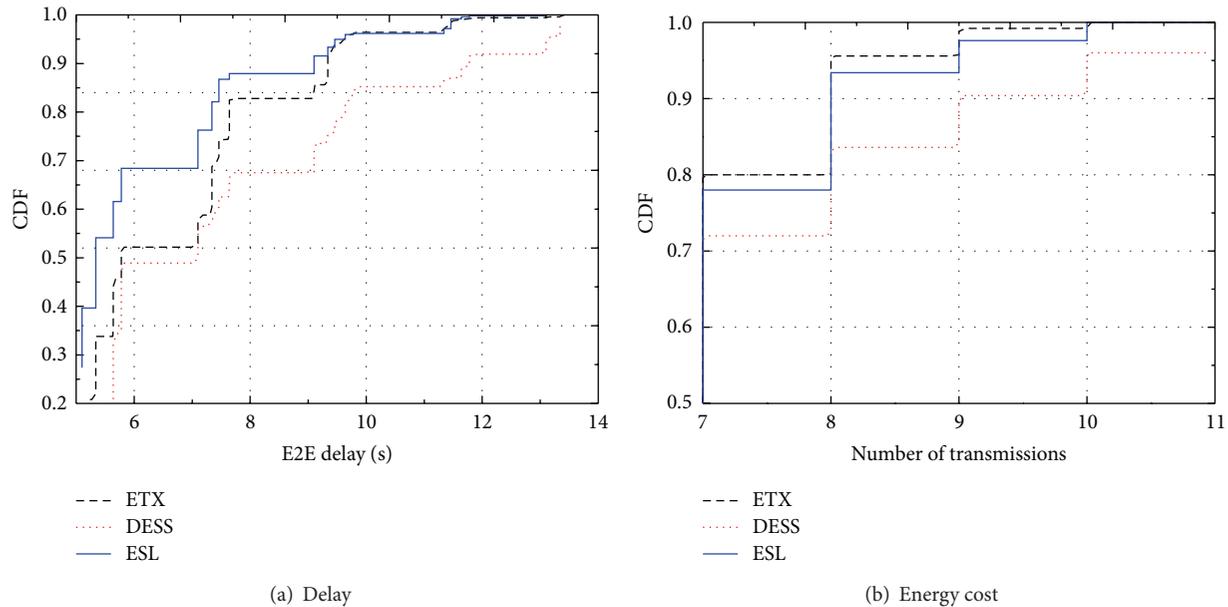


FIGURE 9: Performance comparison.

## 7. Conclusion and Future Work

This paper studies the routing algorithms for the optimization of delivery delay in low-duty-cycle sensor networks. We present a new routing metric, which takes both link quality and working schedule of sensor nodes into consideration. Based on the metric, delay-driven routing algorithms are proposed to reduce source-to-sink delivery delay. Both testbed experiments and large-scale simulations show that our algorithm can optimize sleep latency compared with other routing algorithms.

In future work, we will focus on the following aspects. First, we are aware that more experimental results from extensive testbed systems deployed in different scenarios are needed to further verify our design. Second, instead of single-parent forwarding delivery, we plan to compare our design with multiparent forwarding scheme proposed in [17]. Third, we also want to explore the performance of delay-driven routing algorithms in energy-harvesting sensor network [29], in which the energy supply is highly dynamic and environment-dependent. At last, we plan to study the impact of transmission power on the sleep latency in low-duty-cycle sensor networks.

## Acknowledgments

The author would like to thank the anonymous reviewers for their helpful suggestions that improved this work. This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant no. 21610318 and in part by NSFC under Grant no. 61070165.

## References

- [1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, supplement, pp. 20–41, 2001.
- [2] P. Vicaire, T. He, Q. Cao et al., "Achieving long-term surveillance in VigilNet," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, article 9, 39 pages, 2009.
- [3] L. Mo, Y. He, Y. Liu et al., "Canopy closure estimates with GreenOrbs: sustainable sensing in the forest," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 99–112, November 2009.
- [4] V. Dyo, S. A. Ellwood, D. W. MacDonald et al., "Evolution and sustainability of a wildlife monitoring sensor network," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 127–140, November 2010.
- [5] G. Werner-Allen, K. Lorincz, M. Welsh et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [6] N. Ramanathan, T. Schoellhammer, E. Kohler, K. Whitehouse, T. Harmon, and D. Estrin, "Suelo: human-assisted sensing for exploratory soil monitoring studies," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 197–210, November 2009.
- [7] S. He, J. Chen, and Y. Sun, "Coverage and connectivity in duty-cycled wireless sensor network for event monitoring," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 475–482, 2012.
- [8] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 20–27, April 2005.
- [9] C. F. Hsin and M. Liu, "Randomly duty-cycled wireless sensor networks: dynamics of coverage," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3182–3192, 2006.
- [10] V. Paruchuri, S. Basavaraju, A. Durrresi, R. Kannan, and S. S. Iyengar, "Random asynchronous wakeup protocol for sensor networks," in *Proceedings of the 1st International Conference on Broadband Networks (BroadNets '04)*, pp. 710–717, October 2004.

- [11] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, ACM, August 2000.
- [12] M. Z. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmy, "Efficient geographic routing over lossy links in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, article 12, 2008.
- [13] D. S. J. de Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
- [14] Y. Gu and T. He, "Bounding communication delay in energy harvesting sensor networks," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 837–847, Genoa, Italy, June 2010.
- [15] Y. Gu, T. He, M. Lin, and J. Xu, "Spatiotemporal delay control for low-duty-cycle sensor networks," in *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS '09)*, pp. 127–137, Washington, DC, USA, December 2009.
- [16] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 322–333, May 2006.
- [17] Y. Gu and T. He, "Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1741–1754, 2011.
- [18] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP '08)*, pp. 278–287, Orlando, Fla, USA, October 2008.
- [19] S. Lai and B. Ravindran, "On distributed time-dependent shortest paths over duty-cycled wireless sensor networks," in *Proceedings of the IEEE INFOCOM 2010*, pp. 1–9, San Diego, Calif, USA, March 2010.
- [20] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling," in *Proceedings of the 11th ACM/IEEE Conference on Information Processing in Sensing Networks (IPSN '12)*, pp. 185–196, April 2012.
- [21] E. Ghadimi, O. Landsiedel, P. Soldati, and M. Johansson, "A metric for opportunistic routing in duty cycled wireless sensor networks," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '12)*, pp. 335–343, Seoul, Republic of Korea, June 2012.
- [22] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proceedings of the IEEE INFOCOM 2005*, vol. 4, pp. 2470–2481, March 2005.
- [23] X. Yang and N. H. Vaidya, "A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay," in *Proceedings of the RTAS 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '04)*, pp. 19–26, May 2004.
- [24] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 244–255, March 2004.
- [25] B. Sundaraman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [26] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 14–27, November 2003.
- [27] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 517–526, October 2004.
- [28] Chipcon, CC2420 Data Sheet, <http://www.ti.com/>.
- [29] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transaction on Embedded Computing Systems*, vol. 6, no. 4, article 32, 2007.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

