Joint Design of Asynchronous Sleep-wake Scheduling and Opportunistic Routing in Wireless Sensor Networks

Ming-Shing Kuo, Shi-Chen Wang, Chih-Cheng Hsu, and Cheng-Fu Chou

Abstract—Designing a lifetime-maximization routing in wireless sensor networks poses a great challenge mainly due to unreliable wireless links and limited power supply. Recently, two natural advantages of opportunistic routing, i.e., path diversity and the improvement of transmission reliability, are exploited to develop a lifetime-extended opportunistic routing for wireless sensor networks. Besides, asynchronous sleep-wake scheduling is an effective mechanism to reduce energy consumption by appropriately arranging sensor nodes to sleep. Hence, in this paper, we propose a joint design of Asynchronous Sleep-wake Schedules and Opportunistic Routing, called ASSORT, to maximize the network lifetime. Simulation results show that ASSORT effectively achieves network lifetime extension compared with other routing schemes.

Index Terms—Wireless communication, sensor networks, opportunistic routing, asynchronous sleep-wake scheduling.

1 INTRODUCTION

Wireless sensors, which combine the capability of sensing and wireless communication, are suitable for lots of applications such as military surveillance, temperature monitoring, wildlife tracking, and disaster warning system. For instance, [1] deployed sensors at Reventador, an active volcano in Ecuador, to detect volcanic activity; and [2] monitors the life of a 70-meter tall redwood tree by deploying sensors in Grove of the Old Trees in Sonoma, California. Since it is difficult to recharge batteries of a large number of sensors in remote or hostile environments, energy is regarded as the most critical resource for wireless sensor networks. Due to unreliable wireless links and limited power supply, the routing must be carefully designed to conserve energy for extending the network lifetime, which is defined as the time until the first node depletes its energy [3], [4].

Due to channel fading and wireless contention, data transmission over wireless links is prone to failure. To provide a reliable data delivery, the retransmission scheme is widely used in multi-hop wireless networks; however, certain poor links would spend much energy and dominate the end-to-end reliability. Recently, a new routing paradigm, i.e., opportunistic routing, is proposed to cope with the unreliable wireless transmissions by exploiting the broadcast nature and spatial diversity of the wireless medium. According to the characteristics of the wireless medium, when a packet is sent to a specific node, those nodes within the transmission range of the sender would successfully receive this packet as well. By involving multiple forwarders, called forwarding set, to overhear packets, opportunistic routing is able to reduce the energy consumption for retransmissions based on actual reception conditions at forwarding nodes. In contrast to deterministic routing, which constructs a specific route for each source-destination pair, opportunistic routing can instantly create route diversity to balance load by adaptively selecting forwarders at each intermediate hop. Therefore, such advantages demonstrate that opportunistic routing can be applied to wireless sensor networks for reducing energy consumption caused by retransmissions and dynamically detouring critical sensor nodes with less energy.

Another aspect for energy conservation is sleep-wake scheduling, which appropriately arrange sensor nodes to sleep when data transmission/reception is not needed. In a sleep mode, the communication module of a sensor node is turned off; thus, the energy consumption is fairly low, e.g., $3\mu W$ and $60\mu W$ for the sleep mode and the idle mode of a MicaZ node [5], respectively. Moreover, to avoid the energy consumption of clock synchronization, asynchronous sleep-wake scheduling makes sensor nodes wake up independently with the given rate. Asynchronous sleep-wake scheduling is suitable for a wireless sensor network, because it is easily implemented and can be done locally without additional communication overhead. In addition,

Based on the above discussions, both opportunistic routing and asynchronous sleep-wake scheduling can effectively reduce energy consumption for a wireless sensor network. Hence, an interesting question arises: “can these two mechanisms be combined to get better performance?” We observe that, asynchronous sleep-wake scheduling could result in few forwarders being woken up simultaneously, which might degrade the performance of opportunistic routing. On the other hand, different from the current opportunistic routing protocol, the forwarding set selection has to take

- M.-S. Kuo, S.-C. Wang, C.-C. Hsu, and C.-F. Chou are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10617 Taiwan(R.O.C). E-mail: {kuokuo321, wangsichen, kenneth, ccf}@cmlab.csie.ntu.edu.tw
the wake-up probability into account. Therefore, the goal of this paper is to develop a joint design of Asynchronous Sleep-wake Scheduling and Opportunistic Routing Technology, called ASSORT, in order to prolong the network lifetime of a wireless sensor network. More specifically, ASSORT allows sensor nodes to switch into sleep modes for power saving, and to forward data by jointly considering energy capability, link reliability, and sleep-wake schedules. The contributions of this paper are as follows.

- In section 3.2, we study the energy consumption of asynchronous sleep-wake scheduling and propose a methodology that determines the wake-up rate $\lambda$ and the awake period $\Delta_{wake}$ to minimize the energy consumption and to mitigate the negative impact on opportunistic routing as much as possible.

- In section 3.3, by jointly considering the residual energy and the sleep-wake schedule of each sensor node, we design the metric, called Opportunistic Energy Cost with Sleep-wake schedules (OECS). According to the OECS metric, each intermediate node can choose its forwarding set and the relay sequence to forward data for prolonging the network lifetime.

- In section 4, a comprehensive study shows that our proposed joint method ASSORT is able to achieve a balanced energy consumption among sensor nodes and substantially outperform (a) anycasting routing with sleep-wake scheduling and (b) lifetime-extended opportunistic routing without sleep-wake scheduling, in term of the network lifetime.

Section 2 presents related work, and section 5 gives the future works of this paper.

2 Related Work

In traditional deterministic routing schemes over wireless networks, each node selects a specific node to relay data based on a given metric. Energy-aware routing protocols [6], [7] focus on minimizing end-to-end hop counts. Hence, these works are effective in reducing end-to-end cost, i.e., total energy consumption. For another approach of deterministic routing, lifetime maximization routing schemes [3], [4], [8], [9] consider residual energy of nodes as selecting routing paths. They aim to sustain availability of sensor nodes with less energy by distributing the traffic load over those with higher residual energy.

In contrast to deterministic routing, opportunistic routing [10], [11], [12], [13] is proposed to exploit broadcast nature to involve multiple neighboring nodes as forwarders for dealing with unreliable wireless links. Each forwarder can overhear data and cooperate with each other to relay data according to the actual reception conditions. More recently, some existing research works [14], [15], [16] applied the idea of opportunistic routing to wireless sensor networks for efficient energy consumption. In [14] and [15], authors used geographic information of the nodes, i.e., Euclidean distance to the destination, to assign forwarder priority. In [16], Hung et al. proposed EFFORT which considered the residual energy as the metric for prioritization and the main idea of EFFORT is to prevent some critical node from draining out energy quickly. In contrast to [14] and [15], EFFORT is able to effectively improve the network lifetime with balanced energy consumption. However, the main problem of EFFORT is that nodes still consumes energy as it is idle. This is the reason why our work combines the idea of sleep-wake scheduling, with which nodes sleep for most of time and wake for transmission, into the design of opportunistic routing.

On the other hand, sleep-wake scheduling is an effective method to save energy consumption of sensor nodes while they are in idle states. Prior literature works focus on the synchronized scheduling in the wireless sensor network [17], [18]. These protocols have to periodically exchange the global information to achieve synchronization, which incurs additional communication overhead, i.e., energy consumption and wireless bandwidth. In our work, we apply asynchronous sleep-wake scheduling, as proposed in [19], [20], which do not need to exchange information and is easy to implement in practical.

3 Design of ASSORT

3.1 System Overview

We consider a wireless sensor network composed of sensor nodes and multiple sinks, in which each sensor node is in charge of both detecting events and relaying packets. In addition, sinks are connected to each other by wired links and can be deemed as equal nodes. While a sensor node detects an event, it encapsulates data into packets and transmits these packets to any of multiple sinks by multiple-hop forwarding. Moreover, we assume that sink nodes are rechargeable; so, the sink nodes can be regarded as having infinite energy. TABLE 1 summarizes the notations used in this paper.

In order to efficiently utilize sensor energy, we study a joint design of Asynchronous Sleep-wake Scheduling and Opportunistic Routing, named ASSORT. Sleep-wake scheduling can reduce the energy consumption because sensor nodes can enter sleep mode while there is no packet to send or relay. Furthermore, an opportunistic routing mechanism not only can reduce the number of retransmissions under unreliable wireless links but also can disperse the consumed energy to a group of forwarders. Hence, it is able to (a) avoid some critical node quickly draining out its energy, and (b) extend the network lifetime.

According to sleep-wake schedules, each sensor node switches between active and sleep modes. See Fig. 1 as an example, when sensor node $u$ has data to transmit, the sleep-wake schedule is suspended and node $u$ immediately wakes up for a period of length $T_b + T_s$, called probing period. At the beginning
a probing, node $u$ broadcasts beacons, i.e., $T_b$ of Fig. 1, to notify nodes in its forwarding set that there exists data to be forwarded, and then waits for responses, i.e., $T_s$ of Fig. 1. Node $u$ can start to do data transmission if it successfully receives an ACK response from any node in its forwarding set. If no other node wakes up for relaying data, node $u$ returns to sleep mode and then re-broadcasts beacons until at least one forwarder wakes up. On the other hand, if a sensor node has no data to transmit, it sleeps for at least one forwarder wakes up. On the other hand, if a sensor node has no data to transmit, it sleeps for a random length of intervals $T_{sleep}$ that are independent and exponentially distributed with a given average rate $\lambda$. Note that, the interval between wake periods is defined as the period from the time when the $k$-th wake period ends to the time when $(k+1)$-th wake period begins. When a sensor node $v$ wakes up, it senses the channel for a period of length $\Delta_{wake}$, called wake period. If node $v$ sense the probing beacon from node $u$ and node $v$ is in the forwarding set of node $u$, node $v$ sends an ACK, i.e., $T_u$ of Fig. 1, to node $u$, and keeps awake to receive the packets sent from node $v$; otherwise, node $v$ returns to the sleep mode for energy saving. Moreover, to avoid that a receiver node wakes up between two beacon broadcasting but does not sense any beacon, the interval between two beacon broadcasting is set to be $\Delta_{wake}$. Now, an interesting question arises: what are proper values of $\Delta_{wake}$ and $\lambda$ for sleep-wake scheduling? Therefore, we propose an evaluation model and a decision algorithm to cope with this problem in section 3.2.

Given sleep-wake schedules, in section 3.3, an opportunistic routing method is proposed to cooperate with schedules and to forward data under unreliable wireless links for maximizing network lifetime. Here, network lifetime is defined as the time until the first sensor node runs out of energy.

### 3.2 Asynchronous Sleep-wake Scheduling

In this paper, each sensor node with sleep-wake scheduling wakes up for a wake period of length $\Delta_{wake}$ and enter the sleep mode of length $T_{sleep}$. Moreover, the scheduling of each sensor node is independent. This assumption is reasonable and easy to implement in a wireless sensor network, because it does not need to synchronize clocks of sensor nodes and exchange additional information, which incurs more communication overhead and consumes more energy. Similar to the assumption in [20], we let a wake-up event follows a Poisson process with rate $\lambda$. According to the definition of a Poisson process, the value of $T_{sleep}$ is an exponentially random variable with mean $1/\lambda$. The advantage of Poisson scheduling is its memoryless property, which makes computations tractable. Besides, we assume that the total data generation rate $\gamma$, which is the total number of data packets generated (excluding the relayed data packets) per time unit, is given.

The wake rate $\lambda$, which affects the amount of energy saved by sleeping and the bandwidth supported by a sensor node, is the frequency of switching into wake mode. As for the length of wake period $\Delta_{wake}$, it consumes lots of energy if $\Delta_{wake}$ is large; otherwise, if $\Delta_{wake}$ is small, there might be not enough forwarders to wake up to receive probing beacon for forwarding data; and, this would result in the negative impact on opportunistic routing. We define scheduling overhead to be the energy consumption except for data transmission and reception. Therefore, the problem is to find the values of $\Delta_{wake}$ and $\lambda$ to minimize the scheduling overhead, such that (i) the bandwidth of each sensor node is larger than the given data generation rate $\gamma$; and, (ii) on average, there are $\beta\%$ of nodes in the forwarding set wake up to relay data.

#### 3.2.1 Formulation of Scheduling Overhead

There are two major factors in scheduling overhead: (a) $E_{prob}$, the energy consumption for probing periods before sending data, and (b) $E_{wake}$, the energy consumption for wake periods. Suppose there are $N$ sensor nodes in a network, and the average hop count to the sink is $H$. That is, a data packet is transmitted by $H$ times in a network. Therefore, the expected number of sensor nodes transmitting data per time unit is $\gamma \cdot H$. Hence, the objective is to

\[
\text{Minimize} \quad \gamma \cdot H \cdot E_{prob} + (N - \gamma \cdot H) \cdot E_{wake}
\]

Furthermore, Eq. (2) shows the calculation of $E_{prob}$, where $e_{tx}$ and $e_{rx}$ and $e_{idle}$ represent the energy consumption for transmission, reception, and idle state per time unit. Moreover, $E[n_{prob}]$ indicates the expected number of probing periods before data transmission. That is, in each probing period, the sender node consumes $T_b$ and $T_s$ time units to transmit a beacon and to wait for ACK return, respectively. Moreover, a sender node in the last probing period consumes at most $T_s$ time units to receive ACK.

\[
E_{prob} = E[n_{prob}] \cdot (T_b \cdot e_{tx} + T_s \cdot e_{idle}) + T_s \cdot e_{rx} \tag{2}
\]

Suppose a sender node broadcasts a beacon at time $t$. Since the length of a wake period is $\Delta_{wake}$, the receivers wake up between $t - \Delta_{wake}$ and $t$ can receive a probing beacon. Moreover, because wake-up events
follow Poisson process with rate $\lambda$, the probability that at least one sensor node in a forwarding set wake up between $t - \Delta_{wake}$ and $t$ is 
$q = 1 - e^{-\lambda \cdot |F| \cdot \Delta_{wake}}$, where $|F|$ is the average size of a forwarding set. Thus, the value of $E[p_{prob}]$ is evaluated by Eq. (3).

$$E[p_{prob}] = \sum_{i=1}^{\infty} i \cdot (1 - q)^{i-1} = \frac{1}{1 - e^{-\lambda \cdot |F| \cdot \Delta_{wake}}}$$  

On the other hand, the value of $E_{wake}$, i.e., the energy consumption for wake periods, is expressed by Eq. (4), that includes the energy cost for wake periods and for beacon reception as well as ACK transmission. As the expected length of a sleep mode is $1/\lambda$ in the long term, the rate of the accumulated time in wake mode is $\frac{\Delta_{wake}}{1/\lambda + \Delta_{wake}}$. Besides, whenever a receiver node wakes up and receives a probing beacon, i.e., $T_b \cdot e_{rx}$ of Eq. (4), it has to return an ACK and spend $T_s$ time units to wait for data transmission.

$$E_{wake} = E[p_{prob}] \cdot \Delta_{wake} \cdot \frac{\Delta_{wake} \cdot e_{idle}}{\lambda} + \gamma \cdot (T_b \cdot e_{rx} + T_s \cdot e_{tx} + T_s \cdot e_{idle})$$

### 3.2.2 Formulation of Constraints

Let us turn our attention to the constraint (i): the bandwidth of each sensor node is larger than the given data generation rate $\gamma$. We denote the size of data per transmission by $\delta$. A sender node transmits data after $E[p_{prob}]$ probing periods, and then all data packets of its buffer are transmitted. So, the constraint (i) can be formulated by Eq. (5), where $c$ is the channel capacity for a sensor node, e.g., 250 kbps for MicaZ nodes. Note that, $\alpha \cdot \gamma$ determines the bandwidth requirement of a sensor node to support data generation rate $\gamma$. In this paper, we conservatively set the value of $\alpha$ to be the number of sinks in a network.

$$E[p_{prob}] \cdot \Delta_{wake} + \frac{\delta}{\gamma} \geq \alpha \cdot \gamma \cdot \delta$$  

For constraint (ii): on average, there are $\beta\%$ of nodes in the forwarding set wake up to relay data. We let $\tilde{q}$ to be probability that a receiver node receives a beacon in a period of lengths $\Delta_{wake}$, i.e., $\tilde{q} = 1 - e^{-\lambda \cdot \Delta_{wake}}$. Thus, Eq. (6) shows the constraint (ii).

$$\sum_{i=1}^{\infty} \left(1 - q\right)^{i-1} \cdot \tilde{q} \cdot \left(1 - \tilde{q}\right) \cdot \gamma \cdot \left(1 - \tilde{q}\right) \cdot \sum_{i=1}^{\infty} \left(1 - q\right)^{i-1} \cdot \beta \cdot |F| \geq 1$$

### 3.2.3 Determination of $\Delta_{wake}$ and $\lambda$

According to the proposed model, we are able to find the values of $\Delta_{wake}$ and $\lambda$ to minimize the scheduling overhead (Eq. (1)), such that both constraints (Eqs. (5) and (6)) are met. That is, to determine $\Delta_{wake}$ and $\lambda$, we exploit a simple search algorithm. Due to space limitation, the pseudo code can be found in the technical report [21]. At first, the lower bound and the upper bound of $\lambda$ are found out as follows. Low wake-up rate could result in sender’s buffer overflow and unacceptable end-to-end latency, because a sender needs to wait longer before transmitting data. Thus, to avoid such situation, we assume that, on average, a receiver node wakes up at least once between two consecutive data events, as shown by Eq. (7), which is used to get the lower bound of $\lambda$.

$$\frac{1}{\gamma} \cdot \delta \cdot \lambda \geq 1$$

On the other hand, excessively frequent wake-up events could cause ineffective power consumption; and hence, we let the expected value of $T_{sleep}$ must be larger than the time of a beacon interval, i.e., $1/\lambda \geq T_b$, which is used to obtain the upper bound of $\lambda$. Therefore, we can vary the values of $\lambda$ between its upper bound and lower bound to find the appropriate values of $\Delta_{wake}$ such that both Eqs. (5) and (6) are met. In other words, as the value of $\lambda$ is given, the upper bound and lower bound of $\Delta_{wake}$ are derived from Eqs. (5) and (6) respectively. Thus, we can examine every valid $\lambda \cdot \Delta_{wake}$ pair to find the one such that the scheduling overhead is minimal. In order to find a feasible solution in reasonable time, we set the search granularity of $\lambda$ or $\Delta_{wake}$ to be their lower bound.

### 3.3 Opportunistic Routing Scheme

In this section, we present the proposed OECS metric (Opportunistic Energy Cost with Sleep-wake schedules) and describe the designed routing scheme.

#### 3.3.1 OECS Metric Formulation

Note that the same amount of energy consumption has different degree of impact on sensor nodes with various residual energy. More specifically, a sensor

| Table 1 |
|-------------------------|-------------------------|-------------------------|-------------------------|
| $T_p$ | time to send/receive a beacon | $T_a$ | time to send/receive an ACK |
| $p_{ij}$ | reliability of link $i \rightarrow j$ | $T_w$ | time to wait for ACK sent from awake receivers |
| $N$ | total number of sensor nodes | $\lambda$ | rate of wake-up events with Poisson processes |
| $\Delta_{wake}$ | length of a wake period | $H$ | average hop count from a node to the sink |
| $\gamma$ | total data generation rate in a network | $E[p_{prob}]$ | expected number of probing period before sending data |
| $\delta$ | size of data per transmission | $c$ | channel capacity of a sensor node |
| $e_{rx}/e_{tx}/e_{idle}$ | the energy consumption for transmission/reception/idle state per time unit | $\beta$ | the probability that at least one sensor node in a forwarding set wakes up in $\Delta_{wake}$ |
| $\tilde{q}$ | the probability that a receiver node receives a probing beacon in $\Delta_{wake}$ | $\gamma$ | time to send/receive a beacon |
| $\gamma$ | the probability that at least one sensor node in a forwarding set wake up in $\Delta_{wake}$ | $\Delta_{wake}$ | time to send/receive a beacon |
| $\delta$ | size of data per transmission | $e_{rx}$ | data generation rate in a network |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
node with less residual energy should more conservatively manipulate its energy. For example, the residual energy of two sensor nodes \( u \) and \( v \) are 10 units and 2 units, respectively. In contrast with node \( u \), consuming one unit of energy is significant for node \( v \) since its energy is going to be drained out. Therefore, we define energy cost as the ratio of energy consumption to a sensor node's residual energy for OECs metric; the goal is to minimize energy cost caused by each data transmission. The rationale is to balance energy usage, so the network lifetime can be prolonged.

In order to compute the end-to-end energy cost from a sensor node to the sink, every sensor node has to take the OECs value of all its forwarder into consideration when computing its own OECs value. With this recursive method, every sensor node computes the expected energy cost of data forwarding from a sender to the sink. More specifically, the OECs value of sensor node \( u \) is the expected opportunistic energy cost from node \( u \) to the sink, which is the summation of (1) the energy cost of probing its forwarders, (2) the energy cost in a period from the time when forwarders received probing beacon to the time when forwarders start receiving data, (3) the energy cost of transmitting data from node \( u \), (4) the energy cost of receiving data by all awake forwarders, (5) the expected OECs value of node \( u \)'s forwarders to the sinks, and (6) the energy cost of retransmission. Therefore, the definition of OECs metric for node \( u \) is represented by Eq. (8), where \( F_u \) and \( P_u \) indicate the forwarding set of node \( u \) and priority assignment for all nodes of \( F_u \). Moreover, \( P_{FS} \) is the probability that at least one forwarder successfully receives data.

\[
OECs_u(F_u, P_u) = \frac{C_{prob} + C_{wake} + C_{tx} + C_{rx} + C_{fwd\rightarrow sink}}{P_{FS}} \tag{8}
\]

Each term in Eq. (8) is described in detail as follows, where \( r_u \) is the residual energy of sensor node \( u \).

- \( C_{prob} \) indicates the energy cost of a sender node broadcasting beacon to its forwarders. Hence, \( C_{prob} \) is similar to that of \( E_{prob} \) in Eq. (2).

\[
C_{prob} = \frac{E[n_{prob}] \cdot (T_b \cdot e_{tx} + T_s \cdot e_{idle}) + T_s \cdot e_{rx}}{r_u} \tag{9}
\]

- \( C_{wake} \) denotes the energy cost of the awake forwarders for receiving probing beacon, returning an ACK, and waiting for \( T_s \) time unit before they receive data, as shown in Fig. 1, so \( C_{wait} \) is similar to the second term of \( E_{wake} \) in Eq. (4).

\[
C_{wake} = \frac{T_b \cdot e_{rx} + T_a \cdot e_{tx} + T_s \cdot e_{idle}}{r_u} \tag{10}
\]

- \( C_{tx} \) represents the energy cost of broadcasting data of size \( \delta \) to forwarders. Recall that \( c \) is the channel capacity for a sensor node.

\[
C_{tx} = \frac{\delta \cdot e_{tx}}{c \cdot r_u} \tag{11}
\]

- \( C_{rx} \) is the energy consumption of receiving data sent from the sender. Only awake forwarders can receive data, so we also consider the probability that a sensor node is awake, i.e.,

\[
C_{rx} = \sum_{\forall i \in F_u} \frac{\delta \cdot \tilde{q} \cdot e_{rx}}{c \cdot r_i} \tag{12}
\]

- \( C_{fwd\rightarrow sink} \) denotes the expected energy cost of forwarders, which relay the received data after they received data sent from the sender. For those nodes assigned lower priority, they will not relay data unless the nodes, with higher priority in the forwarding set, fail to relay data. In other words, forwarder \( i \) only has to relay data if forwarder \( i \) receives the packet correctly and all nodes, having higher priority in the forwarding set, fail to receive data. Hence, the probability that node \( i \) relays data is shown by Eq. (13), where \( p_{u,i} \) is the reliability of link \((u, i)\) and \( \rho_i \) is the assigned priority of node \( i \).

\[
P[fwd = i] = \tilde{q} \cdot p_{u,i} \prod_{\rho_i > \rho_j, j \in F_u} (1 - \tilde{q} \cdot p_{u,j}) \tag{13}
\]

Thus, Eq. (14) shows \( C_{fwd\rightarrow sink} \), where \( F_i^* \) and \( P_i^* \) indicate node \( i \)'s forwarding set and relay priority.

\[
C_{fwd\rightarrow sink} = \sum_{\forall i \in F_u} OECs_{i}(F_i^*, P_i^*) \cdot P[fwd = i] \tag{14}
\]

- If all forwarders in a forwarding set fail to relay data, sender node \( u \) needs to transmit data again. Hence, the probability that at least one forwarder successfully receives data is represented by Eq. (15).

\[
P_{FS} = 1 - \prod_{\forall i \in F_u} (1 - \tilde{q} \cdot p_{u,i}) \tag{15}
\]

Since we consider the energy cost of both a sender and a receiver, the value of OECs varies by selecting different nodes to a forwarding set during the metric computation procedure. Besides, priorities determine the order for forwarders to relay data, thus, the priority assignment affects the value of \( C_{fwd\rightarrow sink} \) of OECs. Hence, in order to extend lifetime, each sensor node can minimize the opportunistic energy cost from itself to the sink by determining the proper forwarding set and the priority to minimize the value of OECs.

### 3.3.2 Opportunistic Routing Framework

Based on the definition of OECs metrics, an opportunistic routing protocol is proposed to select forwarders, to assign priorities, and to disseminate data. When a wireless sensor network is initiated, the initial computation of the OECs metric starts from sinks. We let the OECs of a sink is zero, and the initial metric of each node is \( \infty \). The computation procedure is similar to distance-vector routing algorithm. Once a sensor node receives the metric of its neighbor node, it computes the OECs metric and broadcasts its own metric to the one-hop neighboring set. Through the propagation of metric computation, each sensor nodes in a network can obtain the OECs metric. The procedure of initial OECs metric computation will go on until each node’s metric converges.
While a sensor node determines its forwarding set, the OECS metrics of its neighbor nodes ar given and a greedy selection method is applied. Initially, sensor node $u$ set its forwarding set $F_u$ to be empty. Then, node $u$ adds the neighbor node with smallest OECS metric into $F_u$, and relay order $P_u$, the priority of forwarders to relay data, is assigned according to a forwarder’s OECS metric in an ascending order. Similar to existing OR schemes [13, 14], the nodes in a forwarding set are able to hear each other to guarantee that these nodes can follow the given relay priority to forward data. Besides, because a sensor node’s OECS metric depends on its forwarding set’s OECS metrics, the value of metric $OECS_u(F_u, P_u)$ is recomputed after adding one neighbor to $F_u$. This procedure repeats until no neighbor node can be added to $F_u$ to improve the value of $OECS_u(F_u, P_u)$.

When node $u$ needs to send data, it has to broadcast a probing beacon to realize which forwarders are awake. After probing, node $u$ broadcasts data to awake forwarders. If those forwarders receive data correctly, they sequentially relay data according to the relay priority $P_u$. Once node $u$ assures that data has been relayed by its forwarders, the sleep-wake schedule of node $u$ is restored.

Note that a sensor node’s OECS metric depends on its residual energy and its forwarding set’s OECS metrics, so each node’s OECS metric varies with time and interacts with each other. In order to effectively re-compute OECS, each sensor node encapsulates its up-to-date OECS into an ACK packet, and thus a sender node can update its neighbor node’s OECS and re-compute its own OECS locally when receiving the ACK packet. In addition, a sensor node re-determines its forwarding set and relay priority; and then, such up-to-date information is encapsulated into the beacon packets to notify active receivers. Hence, each active receiver can get its up-to-date role of each data transmission. By using control packets to update information, each sensor node can locally re-compute its routing structure when it receive a ACK or beacon packet with new information, i.e., any OECS value of its neighboring nodes changes.

4 Performance Evaluation

To verify whether the proposed routing scheme ASSORT, which jointly considers sleep-wake scheduling and opportunistic routing, can effectively improve the sensor network lifetime, we compare ASSORT with the following routing schemes via NS2 simulator. (1) ANYCAST with sleep-wake scheduling, i.e. Kim et al. [20], only employs sleep-wake schedule to maximize the network lifetime and to ensure that the actual end-to-end delay does not exceed a pre-required delay bound; (2) EFFORT [16] is an opportunistic routing scheme without sleep-wake scheduling for network lifetime extension; and (3) D-SW is a deterministic routing (based on OECS metric, i.e., a single forwarder is selected) with sleep-wake scheduling. We evaluate the performance of four schemes in terms of the amount of data received by sinks within network lifetime, which is defined as the time period until the first node runs out of its energy. Besides, to do a fair comparison, we consider the same pre-required delay bound in [20] for ASSORT, ANYCAST, and EFFORT protocols. That is, the data is regarded as valid if the data is successfully received at any sink and its end-to-end delay is smaller than the pre-required delay bound.

In our experiment, the sensor hardware parameters are set to those of MICAz mote specification [5]; the parameter setting of wireless channel is the same as that in [22], such as the computation of SNR. In each experiment, we randomly select nodes to detect events, the amount of data generated for each event is uniformly distributed from 1 to 200 packets, and the data event is followed by Poisson distribution with rates $0.01 - 0.2$ events per second (default value is $0.1^3$). That is, the average data generation rate $\gamma$ is $1 - 20$ packets per second. All our experiments are under a multiple-sink environment, i.e., 4 is default. All sensor nodes will forward packets toward the sinks, and the transmission is successful if a packet is delivered to any one of the multiple sinks. That is, a sensor node is capable of forwarding data to any sink; therefore, all sinks are regarded as a virtual sink, so a sensor node maintains one routing table for this virtual sink. For each experiment, we randomly generate 10 topologies and report their average as the result (confidence intervals shown at 95% confidence level). Other environment settings used in our simulation are as followed: 1) area field is $500m \times 500m$; 2) transmission range of a sensor is 80m; 3) the data packet size is 46Bytes, in which 25Bytes is for payload; 4) the beacon (or ACK) packet size is 23Bytes (or 4Bytes), i.e., $T_b$ (or $T_a$) is 23Bytes (or 4Bytes)/250kbps; 5) $T_s = 100 \times T_a$; 6) the initial energy of a sensor is 1.5 Joule; and 7) the total number of sensors is 175 – 300 (default value is 200) so the average hop count $H$ from a node to the sink is $3.2 - 2.7$.

Fig. 2 gives the network lifetime comparison of 4 different routing schemes under various event rates. First, we observe that the lifetime of ASSORT, D-SW, or EFFORT is longer than that of ANYCAST, e.g., ASSORT increases the amount of data received during lifetime by 125%-191% compared to ANYCAST. This is because all of them consider the residual energy of each sensor when choosing relay nodes; hence, such routing detour could avoid selecting some lower-energy sensor nodes for the routing path to the sink. In addition, ASSORT and EFFORT apply the idea of opportunistic routing to reduce the energy con-

3. Many applications in sensor networks are event driven, such as, an environment monitoring system can detect activity, e.g., vibrate signal, every 10 seconds on average.
sumption of retransmissions by adaptively involving multiple forwarders for each hop transmission. Note that ASSORT outperforms EFFORT (by 17%-82%) because ASSORT exploits the advantages of sleep-wake scheduling to reduce the energy consumption of idle states. Furthermore, because OR can deal with unreliable wireless transmission, ASSORT outperforms D-SW, especially when the data rate becomes higher, i.e., more collisions occur. This is also the reason why EFFORT can outperform D-SW as the event rate is higher than 0.1.

Fig. 3 depicts the lifetime comparison under different sensor densities. It is not surprising to see that ASSORT still performs best. This is because, with opportunistic routing, ASSORT and EFFORT could get more improvement as the density of sensors becomes larger. That is, as the density increases, the more sensors can be substituted for the critical sensors to avoid consuming residual energy. Also, ASSORT can collect more nodes as forwarders under the case of the same wake probability, which could further improve the performance. In addition, D-SW is able to dynamically select a single forwarder with highest OECS metric to detour low-energy nodes. When the density is higher, each sensor node has more forwarder candidates, thus D-SW can perform more balanced energy consumption. With the help of sleep-wake scheduling and OECS metric, the performance of D-SW is comparable with that of EFFORT, even though D-SW does not involve multiple forwarders to deal with unreliable wireless links. On the contrary, the density has minor impact on the performance of ANYCAST, since the ANYCAST only concerns with which nodes are awake, those nodes are selected as relay nodes with higher probability and they would run out of energy quickly.

In a sink-based scenario, the sensor nodes around sinks would become hotspot nodes with high probability. Hence, we study the impact of sink number, as shown in Fig. 4. The figure demonstrates that for any given approach, the lifetime increases as the number of sinks increase provided nodes are evenly distributed and within reach of these sinks either through single or multi-hops. This is because, more sinks can disperse the traffic load around sinks.

Let us turn our attention to $\beta$ which is a parameter of ASSORT and is given when a network is initialized. $\beta$ means the proportion of how many forwarders, which ASSORT is expected to choose as relay nodes. To help system designers make better decisions for the setting of $\beta$, Fig. 5 presents the impact of different $\beta$ values on ASSORT lifetime under different data generated rates. Thus, the higher $\beta$ value results in the longer wake period and the larger wake up rate; so, by doing that in ASSORT, nodes in ASSORT might consume more energy to collect more nodes as relay nodes. Hence, if the data generated rate is low, ASSORT should choose lower $\beta$ to get longer lifetime. On the other hand, if the data rate increases like 0.2 case, ASSORT should choose larger $\beta$ value, e.g., 0.6, to get better performance. That is, it is worth spending little more time to collect more nodes as relay nodes for ASSORT and this could enhance the successful data delivery probability (from the advantage of opportunistic routing) without involving extra re-transmissions. A dynamic adjustment of $\beta$ can improve performance, but it is out of the scope of this paper.

In Fig. 6, we show the energy consumption of all compared routing protocols. For each compared method, all nodes’ energy consumption are depicted.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Effect of event rate}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{Effect of network density}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4.png}
\caption{Effect of number of sinks}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5.png}
\caption{Effect of $\beta$}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{Energy consumption of sensor nodes}
\end{figure}
in an ascendant order; therefore, a point \((x, y)\) on the curve means that the node, which consumes the \(x\)-th least energy, has already spent energy by \(y\) mJoul.

As illustrated in Fig. 6, we observe that the distribution of energy consumption among the sensors is more balanced in ASSORT, EFFORT, and D-SW than in ANYCAST. This is because they all consider residual energy of sensor nodes, and the nodes remaining more energy are selected as forwarders. We can see that the total energy consumption in EFFORT are higher than those of ASSORT, ANYCAST, and D-SW by 71%, 91%, and 55%, because sensor nodes in EFFORT never go to sleep mode for energy conservation when they are idle. Moreover, we observe that the total energy consumption in ASSORT is slightly higher than that of ANYCAST by 11%. This can be explained as follows. ANYCAST does not need to let multiple forwarders to wake up simultaneously for data forwarding, which allows sensor nodes to sleep longer. However, the links with poor reliability would spend more energy for retransmissions, which incurs unbalanced energy consumption. Besides, because D-SW do not exploit opportunistic routing to reduce the energy consumption caused by unreliable wireless links, the total energy consumption of D-SW is higher than that of ASSORT.

5 Conclusion and Future Works

In this work, we have proposed ASSORT, which is a joint design of synchronous sleep-wake scheduling and opportunistic routing protocol, to enhance network-lifetime for wireless sensor networks. The operation of ASSORT is based on (a) determining the wake-up rate \(\lambda\) and the awake period \(\Delta_{\text{wake}}\) to minimize the additional energy consumption, and (b) an opportunistic metric called OCES which considers the residual energy, link reliability, and sleep-wake schedules. Simulation results show that ASSORT achieves network-lifetime enhancement compared with other schemes.

Our current performance studies are done through NS2 simulator, so one direction of future works is to implement ASSORT on real sensor devices, such as, MicaZ or TelosB. Although ASSORT employs asynchronous sleep-wake scheduling, which is done locally without additional message exchanges, there are several potential challenges that need to be resolved. For instance, we have to add probing/ACK mechanism to the 2.5 layer over the existing CSMA-based MAC protocol, e.g., IEEE 802.15.4, on sensor hardware, such that it works with sleep-wake scheduling. The control packets, e.g., beacon and ACK, play key roles for routing protocols; thus, protecting these packets from corruption is an important issue. Moreover, the wireless channel condition varies over time; so, adaptively monitoring the qualities of wireless links is another practical issue that needs of consideration.

References