DOF: Duplicate Detectable Opportunistic Forwarding in Duty-Cycled Wireless Sensor Networks

Daibo Liu[1], Zhichao Cao[2], Jiliang Wang[2], Yuan He[2], Mengshu Hou[1], Yunhao Liu[2]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[2]School of Software, TNLIST, Tsinghua University
{dbliu, mshou}@uestc.edu.cn, {caozc, jiliang, he, yunhao}@greenorbs.com

***Abstract*—Opportunistic routing, offering relatively efficient and adaptive forwarding in low-duty-cycled sensor networks, generally allows multiple nodes to forward the same packet simultaneously, especially in networks with intensive traffic. Uncoordinated transmissions often incur a number of duplicate packets, which are further forwarded in the network, occupy the limited network resource, and hinder the packet delivery performance. Existing solutions to this issue, e.g. overhearing or coordination based approaches, either cannot scale up with the system size, or suffers high control overhead. We present Duplicate-Detectable Opportunistic Forwarding (DOF), a duplicate free opportunistic forwarding protocol for low-duty-cycled wireless sensor networks. DOF enables senders to obtain the information of all potential forwarders via a slotted acknowledgement scheme, so the data packets can be sent to the deterministic next-hop forwarder. Based on light-weight coordination, DOF explores the opportunities as many as possible and removes duplicate packets from the forwarding process. We implement DOF and evaluate its performance on an indoor test-bed with 20 TelosB nodes. The experimental results show that DOF reduces the average duplicate ratio by 90%, compared to state-of-the-art opportunistic protocols, and achieves 61.5% enhancement in network yield and 51.4% saving in energy consumption.**

I. Introduction

Wireless sensor networks [1] [2] [3] [4] are usually duty cycled to prolong the network lifetime. A widely adopted low-duty-cycled media access mechanism is low power listening (LPL) [5]. Taking X-MAC [6] as a typical example of LPL, each node periodically wakes up and checks the received signal strength to detect the potential traffic. If the channel is clear, it turns off the radio to sleep for a certain period. Note that the sleep schedule of different nodes is generally unsynchronized. A sender probably has to spend much time waiting for its corresponding forwarder to wake up. During the waiting time, the sender continuously transmits the same data packet (called preamble) till the preset timer expires or an acknowledgement is received. As a result, if the forwarder is deterministic, the end-to-end delay is likely high. Obviously, sender energy is wasted on waiting for the forwarder. The duty-cycled communication nature makes the deterministic forwarding schemes inefficient.

To shorten the waiting time, an intuitive idea is to take the earliest forwarding opportunity instead of waiting for the deterministic forwarder, like opportunistic routing [7]. Temporally available links may be exploited to reduce the transmission cost in wireless mesh networks. Landsiedel et al. propose ORW [8], an opportunistic forwarding protocol for low-duty-cycled unsynchronized sensor networks. In ORW, any forwarder with certain routing progress can acknowledge the preamble transmission in LPL. The first wake-up neighbor that successfully receives the packet is selected as the next-hop forwarder. Nevertheless, ORW cannot support high traffic load applications due to channel capacity degradation incurred by the inherent duplicate problem.

978-1-4799-1270-4/13/$31.00 ©2013 IEEE

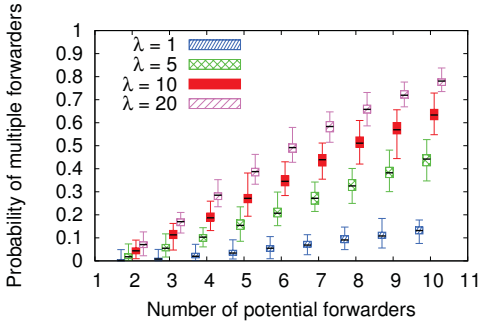
Most duplicate packets are generated when several forwarders keep awake and receive the same data packet during the same period. In low-duty-cycled sensor networks, the high traffic load will significantly increase the risk of producing duplicates. Although several duplicate suppression mechanisms are proposed [7] [8] [9], the overhearing based approaches are not well adapted to the bursty traffic, especially in the large-scale networks with dynamic links. Moreover, according to MORE [10], the long coordination process diminishes the benefits brought by opportunistic routing. The amount of duplicate packets might increase exponentially along the multi-hop relay such that the network throughput is significantly degraded.

In order to address the above issues, we propose Duplicate Detectable Opportunistic Forwarding (DOF). Instead of direct data transmission in LPL, a sender sends a probe and asks the potential forwarders to acknowledge the probe respectively in different time slots. By utilizing the temporal diversity of multiple acknowledgements, the sender detects the quantity and differentiates the priority of all potential forwarders. The sender then forwards its data in the deterministic way to avoid multiple forwarders hearing the same packets. We develop methods to resolve possible collisions among multiple acknowledgments and exploit temporal long good links for opportunistic forwarding. With the light-weight mechanism to suppress duplicates, DOF can adapt to various traffic loads in duty-cycled sensor networks and enhances the system performance with respect to both network yield and energy efficiency.

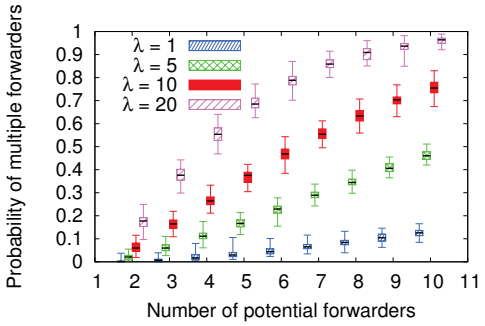
The contributions of this work are as follows:

• Under the context of duty-cycled sensor networks, we extend the opportunistic routing to fit the needs of various traffic loads. This work presents a more comprehensive solution to low-duty-cycled opportunistic forwarding.

• We propose DOF, a practical duplicate-free opportunistic forwarding protocol by exploiting the temporal diversity of the acknowledgements. DOF minimizes the control overhead and improves the reliability of duplicate suppression. It can be easily extended to opportunistic routing in other networks.



(a) Poisson traffic model



(b) Uniform traffic model

Fig. 1. The probability of multiple waking forwarders at one moment with different number of potential forwarders. λ indicates the average traffic load of each forwarder (packets per 10 seconds).

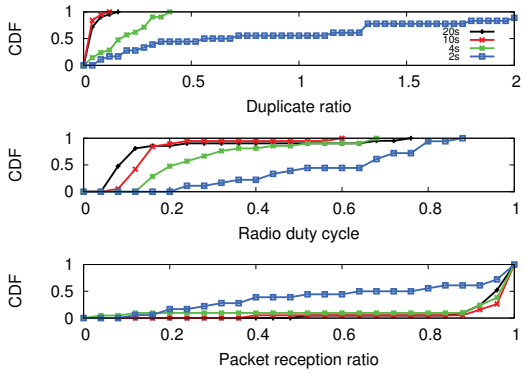


Fig. 2. The CDF of the duplicate ratio, radio duty cycle, and packet reception ratio with different traffic loads on test-bed experiment, when the sleep interval is 512ms.

TABLE I. THE DISTRIBUTION OF THE TRAFFIC LOAD CARRIED BY EACH NODE.

Inter Packet Interval (IPI) (s)	Average Traffic Load (packets/10s)		
	Minimum	Median	Maximum
20	0.55	1.26	7.2
10	1	1.80	8.14
4	2.41	5.05	17.78
2	5.02	11.03	28.81

• We implement DOF and evaluate it on a testbed with 20 TelosB nodes. In high traffic load settings, the evaluation results show that DOF reduces the average duplicate ratio by 90%, compared to state-of-the-art protocols. Meanwhile, DOF achieves 61.5% enhancement in network yield and 51.4% saving in energy consumption.

The rest of the paper is organized as follows. Section II presents the motivation of this work. Section III introduces the system design and analysis, followed by its implementation and evaluation in Sections IV and V, respectively. Section VI discusses the related work. Section VII concludes this paper.

II. MOTIVATION

In this section, we examine the performance degradation brought by inherent duplicates of state-of-the-art opportunistic routing in duty-cycled sensor networks. First, under different probabilistic models of traffic loads, e.g. poisson and uniform distribution, we analyze the probability for multiple forwarders to wake up simultaneously under different network densities and protocol settings. Then, through testbed experiments of ORW, we show the relationship between the duplicates and system performance. Finally, we explain why the current duplicate suppression mechanisms are inefficient in duty-cycled sensor networks.

A. Protocol Analysis

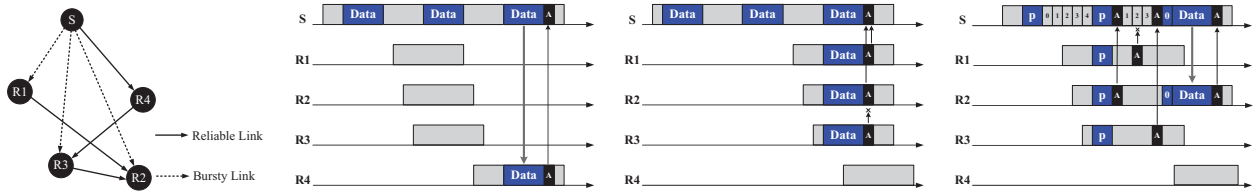
The forwarders that simultaneously keep awake may receive the same data packet. The number of duplicates goes up as the probability of multiple simultaneously waking forwarders increases. Due to the long preamble transmission of LPL, data forwarding with LPL significantly prolongs the waking time of the forwarders. Thus, the number of potential forwarders and the traffic load can influence the probability of multiple forwarders being awake at one moment.

In the analysis, we assume each forwarder periodically wakes up every 512ms. The forwarder stays awake for 20ms after it wakes up. The traffic model of a forwarder follows either poisson or uniform distribution. λ indicates the average number of data packets passing each forwarder in 10s. The time of the preamble transmission of individual data forwarding is calculated according to the traffic model. We simulate the data forwarding process to calculate the probability of multiple waking forwarders at one moment.

Figure 1 shows the probability distribution under poisson and uniform traffic model, respectively. In both scenarios, the probability goes up with the increasing traffic load. For the same traffic load, the probability increases as the number of potential forwarders increases. Thus, the duplicates tend to appear in the areas with bursty traffic or high node density. Specifically, in both Figures 1(a) and 1(b), when the average traffic load is 1 packet/second and there are 6 potential forwarders, the probability of multiple simultaneously waking forwarders is about 30% - 50%. This is much higher than that in the low traffic load setting [8].

B. System Measurement

Based on the implementation of ORW in TinyOS, we further evaluate the influence of duplicates on a testbed with 25 TelosB nodes. We set the radio power at 1 in TinyOS and the sleep interval is 512 ms. On the testbed, the minimum, median and maximum number of available next-hop forwarders of different nodes are 1, 5 and 11, respectively. The average



(a) Topology (b) Deterministic forwarding (c) ORW (d) DOF

Fig. 3. Different from deterministic forwarding and ORW, in DOF the sender distinguishes the multiple waking forwarders by the temporal diversity of ACKs. Then it sends the data packet to an exclusive forwarder by adding in the ACK slot information.

length of routing paths is 2.08 hops. The maximum length is 7 hops. Each node generates data periodically. We select four different traffic loads with the inter packet interval (IPI) to be 2s, 4s, 10s and 20s, respectively. The actual distribution of the traffic load is shown in Table I.

According to the sequence number of the data packets received by sink, we take the duplicate ratio, i.e., the number of duplicates to the number of different packets received, as the metric of duplicates. According to Figure 2, the duplicate ratio is low when the traffic load is low. It increases quickly with the increase of traffic load and the maximum duplicate ratio reaches 200%. When the IPI is 2s, the duplicate ratio of over 50% of nodes is higher than 100%.

We then take the radio duty cycle as the energy consumption indicator for a node. We can see a significant increase of the radio duty cycle when the duplicate ratio increases. When the IPI is 2s, the radio duty cycle of over 40% of nodes is higher than 60%. About half of the energy is wasted on the transmission of duplicates. According to packet reception ratio (PRR) of Figure 2, we can see the PRR stays stable when the packet interval is 4s, but it decreases quickly when the traffic load gets higher. The main cause of packet drops is forwarding queue overflow, where the queue size is 10.

The experiments show that many duplicates indeed exist with the state-of-the-art low-duty-cycled opportunistic routing protocols, especially when the traffic load is high. Moreover, the duplicates significantly degrade the system performance and should be avoided.

C. Duplicate Suppression Mechanism

Most of existing duplicate suppression mechanisms are based on overhearing. When a forwarder overhears a packet, which is identical to a pending packet in the forwarding queue, it deletes the packet from the queue. However, in current sensor operating systems like TinyOS, the non-preemptive task abstraction does not allow a node to interrupt on ongoing transmission tasks. Moreover, the bursty traffic, especially in large-scale networks with dynamic links, further makes a forwarder hard to exactly overhear every packet relayed by others.

To further reduce duplicates in the bursty traffic, packet transmissions are coordinated among different nodes in the network. For example, ExOR [7] arranges the forwarding order according to the routing progress and the quantity of received data packets. According to More [10], however, the coordination process introduces extra overhead. Moreover, the coordination restricts concurrent transmissions and hence reduces the network yield.

We want to develop a forwarding approach, which can detect the simultaneous waking-up forwarders and inherently avoid the duplicates. Meanwhile, the forwarding approach keeps the spatial diversity of the opportunistic routing as much as possible.

III. System Design

DOF targets on developing a practical opportunistic forwarding scheme for various duty-cycled sensor network applications. In this section, we discuss several issues: (1) the overview of how DOF detects the potential forwarders by slotted acknowledgement (ACK), (2) the algorithm of ACK slot assignment and forwarding strategy, (3) the adaptive routing metric. For simplicity we here illustrate the basic design of DOF using X-MAC, a well adopted unsynchronized LPL MAC as we mentioned above.

A. Overview of DOF

As Figure 3(a) shows, *S* sends packets to the intended destination *R2*. There are three potential relay nodes *R1*, *R3* and *R4*. The links are either reliable or bursty indicated as the solid or dashed lines, respectively.

As Figure 3(b) shows, in traditional deterministic forwarding, *S* continuously sends the data to the predetermined relay node *R4* till it wakes up. As Figure 3(c) shows, ORW takes the early wake-up nodes (*R1*, *R2* or *R3*) which receives the data and provides routing progress as the next-hop forwarder. However, as Figure 3(c) shows, *R1, R2* and *R3* may receive the data simultaneously. The duplicates then significantly degrade the system performance as introduced above.

DOF detects potential duplicates by using adaptive slotted ACK when multiple forwarders are awake simultaneously. As Figure 3(d) shows, instead of directly sending data, a sequence of probes are first broadcast by *S*. The interval of two adjacent probes is divided into multiple time slots. Each slot is long enough to receive an ACK. When *R1, R2*, and *R3* receive one probe and any of them offers routing progress, each of them independently selects a slot (2, 0, and 4) to send the ACK back. According to the slot information of the received ACKs (0 and 4), *S* sends the data packet to a forwarder (*R2*) by adding in the slot information (0).

To minimize the duplicates and keep the benefit of opportunistic routing, the design of DOF faces several challenges: (1) Different forwarders should acknowledge the probe at

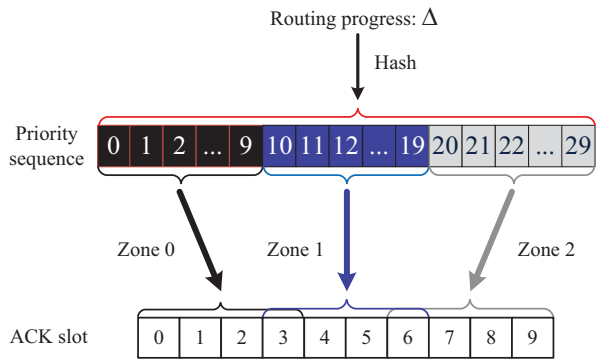


Fig. 4. DOF splits all the ACK slots into 3 slightly overlapped priority zones. According to the routing progress, DOF randomly maps each forwarder into a slot in different priority zones.

different slots. In addition, the routing progress of different forwarders should be distinguished because the forwarder with more routing progress should be used with a higher priority. (2) Although the communication overhead caused by probe transmissions for each data packet is little, it should be avoided when the traffic load is high. (3) DOF may explore temporally available links to forward data. However, the data ACK loss over these links may lead undesirable retransmissions due to the bursty loss. So the short-term link performance should be considered.

B. ACK Slot Assignment

As we mentioned in the previous subsection, the two requirements of ACK slot assignment are that multiple forwarders should be distributed into different slots and the sender should infer the routing progress of different forwarders by ACK slot distribution. As Figure 4 shows, the basic strategy is as follows: first, according to a hash function, the forwarder matches its routing progress Δ to a location H_{sf} on the priority sequence. The priority sequence is like a ruler to measure the routing progress; then, we split all the ACK slots into multiple slightly overlapped zones, which are matched to different segments of the priority sequence (e.g. zone 0 $\rightarrow \{0-9\}$); last, according to H_{sf}, we randomly assign one slot in the selected zone.

There are six parameters in the calculation procedure, as shown in Table II. When forwarder f receives the probe sent by s, the routing progress is calculated by

$$\Delta_{sf} = W_s - W_f \tag{1}$$

W_s is carried in the probe and W_f is local routing information. If Δ_{sf} is larger than Δ_{max}, we set it as Δ_{max}. By Eq. (2), the routing progress Δ_{sf} is mapped to a location H_{sf} in the priority sequence. A forwarder with a larger routing progress is mapped into the head area of the sequence.

$$H_{sf} = \lfloor (1 - \frac{\Delta_{sf}}{\Delta_{max}}) * N \rfloor \tag{2}$$

f calculates in which ACK zone ($zone_f$) it should acknowledge the probe and the offset (δ_f) in the segment of priority sequence corresponding to $zone_f$.

$$zone_f = \lfloor \frac{H_{sf} \cdot L}{N} \rfloor \tag{3}$$

TABLE II. THE DESCRIPTION OF THE SYMBOLS IN THE ACK ASSIGNMENT ALGORITHM.

Symbol	Description
$\mathbf{W_s}$	the routing metric value of the source node
$\mathbf{\Delta_{max}}$	the maximum routing progress over one hop
N	the total length of the priority sequence
M	The total number of ACK slots
L	the number of priority zone of ACK slots
R	the number of ACK slots in each priority zone

$$\delta_f = H_{sf} - \lfloor zone_f \cdot \frac{N}{L} \rfloor \tag{4}$$

f randomly maps H_{sf} into the final ACK slot, $slot_f$, as Eq. (5) shows.

$$slot_f = zone_f \cdot \lfloor \frac{M}{L} \rfloor + \lfloor \frac{\delta_f \cdot L \cdot R}{N} \rfloor + rand() \tag{5}$$

where $rand()$ is a random number between 0 and R. If $slot_f$ is larger than M, we make $slot_f$ equal to M.

Let's take Figure 4 as an example. We assume L, Δ_{max}, N, M and R are 3, 5, 30, 10 and 4, respectively. If the forwarder provides routing progress Δ_{sf} as 2.8, the location of priority sequence, H_{sf}, equals to 13. Then, $zone_f$ and δ_f are 1 and 3. If we assume the random number $rand()$ is 3, the $slot_f$ will be the 7^{th} slot.

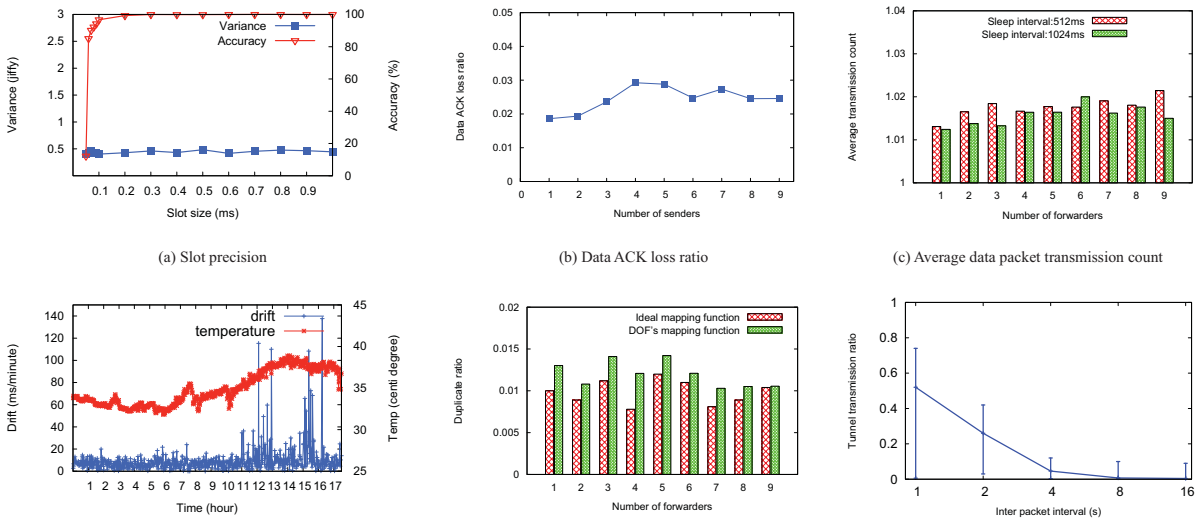
Rather than assigning each forwarder a fixed ACK slot, our method is more flexible to utilize all temporarily available links. Moreover, the parameters of our method are predetermined based on the local routing information so that there is no extra communication overhead. The computation complexity of the algorithm is low. However, this algorithm does not guarantee that multiple forwarders do not choose the same ACK slot. We show in practice this situation rarely happens in Section IV.

C. Forwarding Management

Note that a forwarder may serve multiple senders during a short period. Each forwarder maintains a sender table, which records the ACK slot information to trace the potential senders. Each entry of the sender table includes: the sender's address, expected data sequence number (DSN), and the selected ACK slot.

When a probe is received, the forwarder first checks the attached routing metric W_s of the sender s. If the forwarder can provide routing progress ($\Delta_{sf} > 0$), it selects an ACK slot $slot_f$ to acknowledge the sender. Then, if there is a record of the same sender, the forwarder updates the corresponding record in the sender table. Otherwise, the forwarder adds a new entry into the table. Note that the DSN attaching in the received probe copies that of the sender's pending data packet. Upon the acknowledged probe, the sender attaches the DSN and the selected ACK slot number as the virtual intended forwarder address. When the forwarder receives a data packet, it queries the sender table. If there is no matched entry, the forwarder drops the packet and does nothing. Otherwise, it will take the responsibility to forward the data packet.

Moreover, although the forwarder acknowledges the received probe, it still receives the duplicate of the same probe.



(a) Slot precision

(b) Data ACK loss ratio

(c) Average data packet transmission count

(d) Clock drift

(e) Comparison between two slot assignment algorithms

(f) Probability of tunnel transmission

Fig. 5. The detail verification of DOF implementation. (a) relationship between the ACK slot time span and the prediction accuracy; (b) the situation of data ACK loss with multiple senders; (c) the average transmission count of the successfully delivered packet; (d) the clock drift as temperature changes; (e) duplicate ratio of DOF slot assignment algorithm compared to the ideal assignment; (f) the ratio of tunnel transmissions over the total transmissions under different traffic loads.

The probe duplicate indicates the sender has not received the ACK for the previous probe due to the asymmetric link or link dynamics. If the forwarder receives the duplicate probe, it goes back to sleep to save energy.

On the other hand, the sender may receive multiple ACKs distributed in different slots after sending a probe. According to our ACK slot assignment algorithm, the forwarder corresponding to the earlier coming ACK provides relatively high routing progress. Thus, the sender inserts the DSN and the minimum slot number of ACK received to the pending data packet and sends it.

When the sender prepares to send a batch of packets, the intended forwarder will keep awake during the batched sending. Besides the probes of the first packet, the probes of the rest packets are not needed. Thus, to save the extra overhead of the probe transmission, the sender directly sends the rest packets with the connection (called Tunnel) found by the probes of the first packet till either the loss of data ACK or there is no pending data packets.

When the pending data packet is acknowledged, the sender finishes this transmission. However, because of the lossy link or misalignment of the probe ACK slots, the sender may not receive the data ACK from the intended forwarder. Hence, with a larger retransmission limit is inadvisable. Whether we should keep retransmitting the data packet or send a probe again to detect new forwarders is an important problem for the agility and efficiency of the protocol. According to [11] [12], the packet loss tends to be bursty over temporally available links. We propose the Limited Retransmission Strategy (LRS) to address the data ACK loss. The basic idea is to estimate the available period of those links and then adaptively bound the number of retransmissions. The specific setting of LRS is shown in Section IV.

D. Low-Duty-Cycled Opportunistic Routing

In DOF, a packet is sent to one of the waking neighbors, which provides certain routing progress. As a result, the routing topology towards the sink is not fixed. A packet may be forwarded to the sink along different paths. Moreover, considering the unsynchronized sleep schedule in LPL, DOF drives two requirements on routing. First, the routing metric should reflect the waiting time of the link layer transmissions. Second, each node should adaptively choose a set of forwarders from all neighbors to determine the local routing metric.

Considering the two requirements above, EDC (expected duty cycle), which is introduced by ORW [8], acts well on the whole. Hence, we adopt the concept of EDC as the routing metric. Our method of duplicate detection can be easily built on other routing metrics as well, such as end-to-end delay or ETX[13].

IV. Implementation

We implement DOF on TelosB nodes in TinyOS 2.1.1. The RAM and ROM consumption of the program are 6268 bytes and 39714 bytes, respectively. Next, several implementation issues are carefully discussed.

A. ACK Slot Settings

For the forwarder, when the radio (CC2420) has received a packet, it will generate an interrupt (FIFOP) to trigger the handler function. Meanwhile, for the sender, it will also generate an interrupt (falling-edge SFD) when the transmission has finished. We neglect the propagation delay so that the FIFOP interrupts of the probe on different forwarders happen simultaneously. Thus we take the simultaneous interrupts as the beginning of ACK response for the sender and forwarders.

When the FIFOP interrupt is generated, the forwarder calculates its ACK slot K_f and sends the ACK after $K_f T_{slot}$

time, where T_{slot} is the slot time span. Upon receiving the ACKs from different forwarders, the sender needs to determine the slot number for different forwarders. Since the ACK slot calculation and ACK reception take certain time, there is a shift between the time the probe is sent and the time the first ACK is received. We denote the shift between the earliest received ACK (i.e. the ACK sent in the first slot) and the falling-edge SFD interrupt as T_{base}. T_{base} is close to a constant time. In our implementation, the measured T_{base} is about 2.3ms. Thus, according the interval T_r from the falling-edge SFD interrupt to the ACK received, the sender calculates the slot number K_s as:

$$K_s = \lfloor (T_r - T_{base})/T_{slot} \rfloor \quad (6)$$

where K_s should equal to K_f. If the maximum ACK waiting duration is T_{max}, the maximum number $slot_{max}$ of the ACK slots is calculated as:

$$slot_{max} = \lfloor (T_{max} - T_{base})/T_{slot} \rfloor \quad (7)$$

In practice, the clock drift and the variance of the software execution time will incur the mismatch between K_f and K_s. Assume the clock drift between the forwarder and sender is σ, a mismatch only occurs when

$$\sigma T_r \geq T_{slot} \quad (8)$$

Thus, increasing the slot time span reduces the probability of mismatch, while it limits the number of available ACK slots. In practice the less available ACK slots might increase the probability that multiple receivers choose the same ACK slot.

We conduct experiments to show the prediction accuracy and the average time variance with different slot time spans. In the experiments, two senders transmit packets to the same receiver in the office environment. The results are shown in Figure 5(a). We can see the predication accuracy is close to 100%, when the slot time span is larger than 0.1ms. The average variance is relatively stable, about 0.5 jiffy (1 jiffy = 1/32 ms). Considering the more complicate environment in practice, the slot time span is conservatively set as 0.2ms and the total number of ACK slots is 10.

Moreover, we measure the clock drift under different temperature between a pair of nodes. As Figure 5(d) shows, the clock drift goes up when temperature rises from hour 11 to hour 13. The clock drift is less stable when the temperature is higher than 35 degree centigrade. The maximum clock drift is about 140ms per minute (about 0.0023ms per 1ms). Under the above settings, the maximum ACK waiting duration is about 4.3ms. Thus, the maximum variance incurred by the clock drift is about 0.009ms, which is far less than 0.2ms.

B. Data ACK Loss and Retransmission

As Section III-C mentioned, the data ACK may be lost due to the mismatch of the probe ACK slot between sender and forwarder or link dynamics. Figure 5(b) shows the data ACK loss rate when multiple senders send packets periodically to the same receiver. The data ACK loss ratio is no more than 3% with different number of senders. In DOF, the sender opportunistically utilizes the temporally available links, for which the probe ACKs have been successfully received. Thus, the data ACK loss is rare in the experiments. The probe ACK loss will reduce the available opportunities, but not degrade the reliability when the firm links exist.

Due to the possible bursty loss, we propose the limited retransmission strategy (LRS) to bound the number of data retransmissions. To determine the maximum retransmission count in practical networks, we make multiple senders send packets to the same receiver. For each packet, the maximum number of retransmission is initially set to 10. The packet will be dropped when the retransmission is larger than 10. We measure the average transmission count of the successful data transmissions of all senders. As shown in Figure 5(c), the average transmission count of a successful transmission is smaller than 2. Thus, we set the transmission threshold as 2. When the sender does not hear the ACK, it will retransmit the data packet once. If the retransmission also fails, the sender will broadcast the probe to find the available forwarder again.

TABLE III. SYSTEM PARAMETERS.

Parameters	Value
M	10
L	3
R	4
N	30
Slot time span	0.2ms
LRS threshold	2
Δ_{max}	3

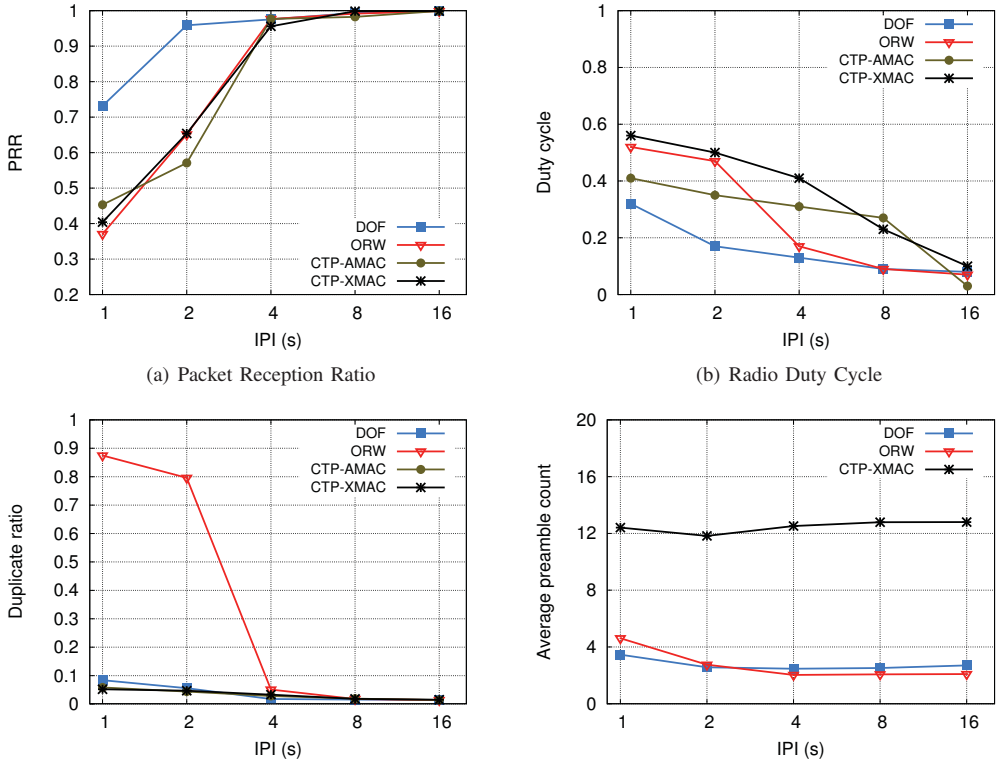
C. Tunnel Transmission

The probe is utilized to detect the potential receivers. As Section III-C mentioned, when there are several packets in forwarding queue, the sender will take the tunnel transmission to save the energy consumption on the probe transmission. Figure 5(f) shows the ratio between the number of tunnel transmissions and total data transmissions for different traffic loads on testbed experiment with 20 TelosB nodes. We could see for high traffic loads the portion of tunnel transmission ratio is high. Especially, over 50% of transmissions are tunnel transmissions when the inter packet interval is 1s.

D. Slot Assignment

As mentioned in Section III-B , the slot assignment algorithm of DOF may map different forwarders into the same ACK slot. In this situation, it is possible that multiple forwarders receive the same packet so that duplications occur. We compare our algorithm with the ideal slot assignment, in which we manually assign an unique ACK slot for each of the forwarders. Figure 5(e) shows that, in practice, the duplicate ratio of our slot assignment algorithm is just a little higher than the ideal method. The duplicate in the ideal assignment algorithm is induced by the data ACK loss. In such a case, the sender will transmit the data packet again to a new forwarder, while the previous forwarder actually has received the packet.

In summary, we show the details of the implementation settings in Table III.



(a) Packet Reception Ratio

(b) Radio Duty Cycle

(c) Duplicate Ratio

(d) Average Preamble Count

Fig. 6. The overall comparison of different system performance metrics between DOF and the other forwarding protocols.

V. Evaluation

In this section, we evaluate DOF through various testbed experiments. We compare the network yield, energy consumption and duplicate ratio of DOF with two unsynchronized low-duty-cycled forwarding protocols such as ORW and CTP-XMAC. We also compare the performance of DOF with CTP-AMAC [14], which is the-state-of-art synchronized receiver-initiated low-duty-cycled protocol. In addition, considering the system stability under network churns, we compare DOF with L^2 [15], which is proposed to optimize the energy efficiency by incorporating with the synchronized rendezvous, link burstiness and dynamic forwarding.

We use the packet reception ratio (PRR) as the indicator of the network yield. It also indicates the network throughput combining with the inter packet interval. In our implementation, the on-board sensor and flash memory is rarely used and thus the radio consumes most of the energy [16]. The energy consumption is measured by the radio duty cycle. Moreover, we use the average preamble count to approximate the delay. In CTP-XMAC and ORW, it is the average number of the data transmissions. In DOF, it is the sum of the probes and data transmissions on average. Normally, the smaller the average preamble count is, the less the delay is.

A. Evaluation Setup

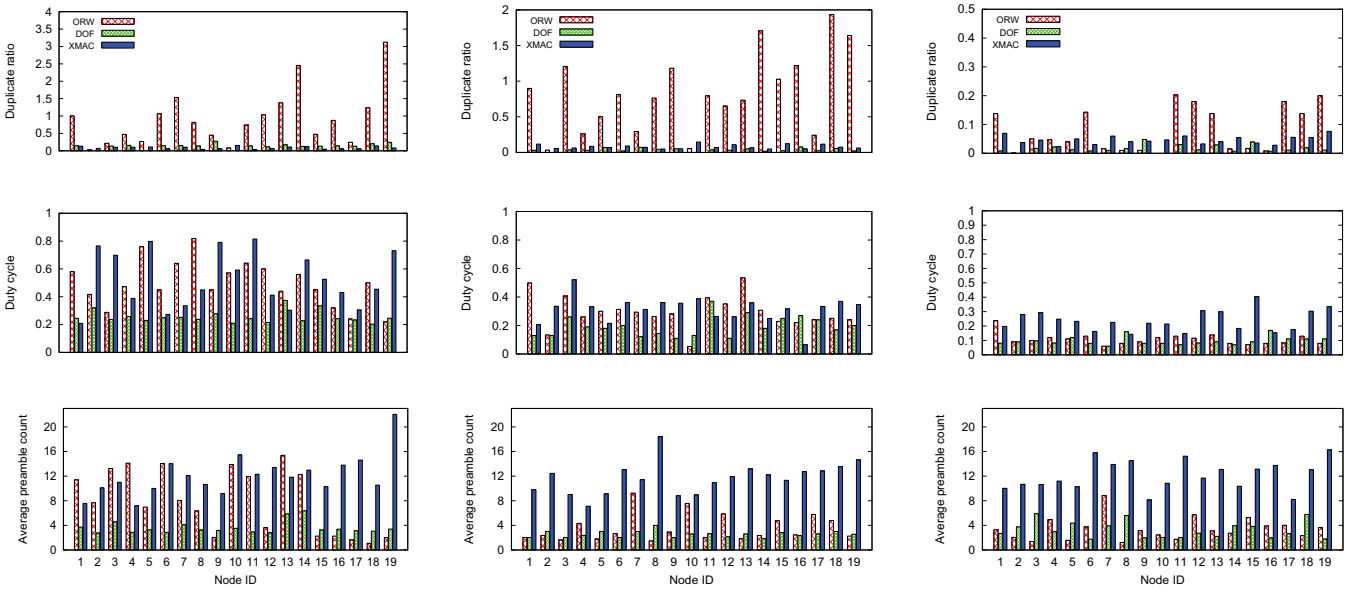
We evaluate the performance of different forwarding protocols on an indoor testbed with 20 TelosB sensor nodes. We set the transmission power of CC2420 as 1 to ensure multi-hop communication (maximum hop is 3).

Each node generates packets with a fixed IPI. We vary the traffic load by setting different IPIs, such as 1s, 2s, 4s, 8s, and 16s. The packet length is 80 bytes. Thus, the sleep interval of other protocols is set to 512ms, except AMAC which is set to the default setting 128ms. For each traffic load, the experiments last for at least 30 minutes and are repeated three times. The experiments are often conducted during the night to mitigate the influence of the human behavior.

B. Network Yield

Figure 6(a) shows the experiment results of the average PRR for different forwarding protocols with different traffic loads. We can see that when the IPI is no less than 8s, the packet loss of all forwarding protocols is small. However, with the decreasing of IPI, the PRR of ORW, CTP-AMAC, and CTP-XMAC sharply decreases from 95% to less than 50%. In contrast, the PRR of DOF is still higher than 90% and 70% when the IPI is 2s and 1s, respectively. When the IPI is 1s and 2s, the network yield of DOF is about 46.5% and 61.5% higher than the best of ORW, CTP-AMAC, and CTP-XMAC.

The significant decreasing of PRR in a high traffic load is due to the inefficient channel utilization. In CTP-XMAC, each sender will occupy the channel for a long time till the intended receiver wakes up. As shown in Figure 6(d), the average preamble count of CTP-XMAC is much larger than both ORW and DOF. Although ORW has the smallest average



(a) IPI=1s (b) IPI=2s (c) IPI=4s

Fig. 7. Per node comparison of different system performance metrics between DOF and the other forwarding protocols in high traffic loads.

preamble count, the duplicate ratio is much higher than others as shown in Figure 6(c). The duplicate degrades the channel utilization, as explained in Section II. In AMAC, instead of the continuous data packet transmission, the sender waits for the probe from the receiver when it wakes up. Upon receiving a probe, the sender sends the pending packet to the receiver immediately. When multiple senders have packets for the same receiver, packet collisions occur and data retransmissions will significantly reduce the channel utilization. Hence, DOF is more adaptive for various traffic loads and thus the network yield of DOF is better than others.

C. Energy Consumption

As Figure 6(b) shows, the average duty cycle of DOF is the smallest among various traffic loads, except for the case of CTP-AMAC when the traffic load is low (IPI = 16s). AMAC is high energy efficient in low traffic load because it uses one-hop synchronization. The advantages of transmission solicitation and synchronization in AMAC transform to the weakness as the traffic load increases. This is because the solicitation from the receiver to effectively synchronize packet transmissions results in concentration in packet transmission, and thus contention and collision. DOF saves at least 21.4% and 51.4% of energy compared with others when IPI is 1s and 2s, respectively. The CTP-XMAC has the lowest energy efficiency. ORW performs badly when IPI is less than 4s. The energy consumption of CTP-AMAC increases slowly when IPI is less than 8s.

In a high traffic load, the sharp increasing of energy consumption of CTP-XMAC and ORW is due to the degradation of the channel utilization mentioned above. However, the energy consumption in CTP-AMAC increases slowly. We guess the reason is the synchronized sleep schedule of AMAC.

The histograms in the middle of Figure 7 clearly show the duty cycle of each node for different protocols with different traffic loads. When the IPI is 1s and 2s, on most of nodes, the duty cycle of DOF is much better than ORW and CTP-XMAC. ORW consumes almost the same energy as CTP-XMAC. When IPI is 4s, on most of nodes, the duty cycle of ORW is close to that of DOF, and both of them are better than CTP-XMAC. The results verify that although ORW performs well in a low traffic load network, DOF can keep the energy consumption low for various traffic loads.

DOF utilizes the probe to detect potential forwarders so that it induces a low communication overhead. The experiments show that in the high traffic load, the energy efficiency brought by the probe is much larger than the overhead. In a low traffic load, the overhead of the probe transmission is also limited.

D. Duplicate Ratio

In Figure 6(c), compared with ORW, DOF significantly reduces duplicate ratio when IPI is less than $4s$. The average duplicate ratio of ORW is about 85% when IPI is 1s. This is about 10 times larger than DOF. The duplicate ratio of DOF is comparable with the deterministic routing under various traffic loads. As the histograms on the top of Figure 7 show, on most of the nodes, the duplicate ratio of DOF and CTP-XMAC are much less than ORW. When IPI is 1s, the highest duplicate ratio of ORW exceeds 300% (e.g., node 19). Compared to ORW's high duplicate ratio, DOF always keeps the duplicate ratio low, which is close to the duplicate ratio of CTP-XMAC in different traffic loads. The results of duplicate ratio verify the efficiency of the ACK slot assignment algorithm.

E. Delay

As Figure 6(d) shows, the average preamble count of CTP-XMAC do not change significantly with the increasing of

network traffic load. The average preamble count of DOF, which is comparable with ORW, is 3 times less than CTP-XMAC. The average preamble count increases when the traffic load increases. The histograms on the bottom of Figure 7 show the average preamble count of each node in different traffic loads. We can see CTP-XMAC always has the largest delay on every node. Due to channel degradation, the maximum average preamble count reaches 8 and 12 when IPI is 2s and 1s, respectively. DOF has less delay than ORW in the scenarios when IPI is no more than 2s, because the waking forwarders simultaneously send ACKs resulting in ACK collisions in ORW protocol. Then, the sender will broadcast the data packet again. In a low traffic load, the overhead brought by probe transmission makes the average preamble count of DOF a little greater than that of ORW in single-hop propagation. Considering the preferential use of the links with high routing progress, we believe that DOF can reduce hop count compared to ORW.

F. Impact of Network Churn

The node reboot, node replacement or link dynamic will incur network churn, which may further lead to the degradation of system performance. We evaluate the impact of network churn on DOF and L^2. L^2 optimizes the energy utility by incorporating the synchronized rendezvous, link burstiness and dynamic forwarding. In the implementation of L^2, each node will take 10 minutes to synchronize the rendezvous with neighbors by routing beacons before it begins to generate data. When the synchronization is stable, the frequency of the routing beacon is reduced to one per several minutes.

We set the IPI as 4s and do the experiments for one hour in the daytime with the influence of the human behavior and WiFi. Moreover, for DOF, we randomly remove and add nodes. For L^2, we randomly reboot nodes. The top figure of Figure 8 shows the number of nodes sending packets.

The middle figure of Figure 8 illustrates the variance of the PRR. We can see that there is almost no influence on DOF since there is no need of any extra control message in DOF. However, we can see the PRR of L^2 decreases even without churn. The reason is that the a high traffic load will lead to time error accumulation in TinyOS system so that the synchronization accuracy will decrease quickly. Without enough routing beacons to re-synchronization, routing loops or data retransmissions will significantly degrade the PRR. When there are network churns, the PRR of L^2 tends to be more dynamic.

The bottom figure of Figure 8 illustrates the variance of the radio duty cycle along the time. We can see that the energy consumption of DOF without churn is relatively stable. When there are network churns, since the reduced data amount brings low channel contention, the energy consumption could be reduced as shown around the 15^{th} time unit. However, if nodes around the sink are removed, the path length will increase. Thus the churn might also increase the energy consumption as shown around the 22^{th} time unit. The window-based transmission of L^2 makes the energy consumption of a single transmission is bounded, even with packet loss. We can also see the energy consumption of L^2 last increasing slowly.

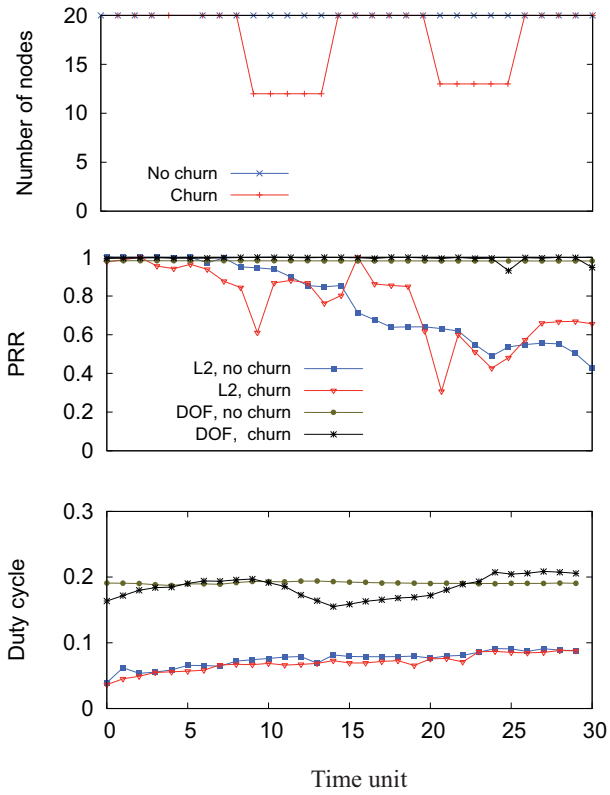


Fig. 8. Impact of churn on DOF and L^2 in the network with IPI=4s and removing or adding several nodes. The x-axis indicates the time units. Each unit is corresponding to 2 minutes.

Although L^2 has less energy consumption, it needs extra control overhead, i.e. routing beacon, to deal with network churns and thus the energy consumption increases. In contrast, DOF explores the temporally available links by probe, but does not need up-to-date link state maintenance. So DOF is more adaptive to practical large scale network deployments.

G. Discussion and Limitations

DOF uses software acknowledgement in the implementation. However, in current CC2420 radio stack in TinyOS system, we found the software ACK is vulnerable when the traffic load is high. The limitation is due to the slow buffer swapping between MCU and CC2420. One consequence is that a new packet may arrive when the sender is waiting for the ACK. The processing time of the received packet might affect the accuracy of the ACK slot calculation. The other consequence is increasing the collision probability between ACK and data packets.

For DOF in a high traffic load, the transmission of probe, data packet and ACK are mixed, which might lead to ACK loss or ACK slot prediction error. This explains why the PRR in Figure 5(a) is lower than 80% when IPI is 1s. The average preamble count when IPI is 1s is higher than that when IPI is 2s. We believe that DOF could work better with more fine-grained timing control in the radio stack.

VI. Related Work

In this section, we discuss related work on opportunistic and dynamic forwarding mechanisms. Moreover, we illustrate the advantage of our adaptive duplicate suppression schemes in unsynchronized duty-cycled WSN.

ExOR [7] develops a complete opportunistic routing for wireless network. ExOR assigns each receiver to further transmit in distinct time slot, the receiver overhears others' transmissions to avoid the duplicates. MORE [10] targets on the inefficient coordination process of ExOR and proposes a coding approach to eliminate the overhead. Rather than network coding, DOF takes a light weight method to mitigate the overhead for WSN.

BRE [12] develops the overhearing scheme on CTP [17] to capture the temporally good links. The sender changes the next-hop receiver when the opportunity appears to reduce the transmission count. However, BRE does not address the duty cycle issue, in which the waiting time dominates the energy efficiency.

In DSF [18], each node knows when the schedule of neighbor nodes by synchronization. DSF dynamically selects multiple next-hop forwarders based on the sleep schedules and routing metrics of the neighbors. L^2 [15] further notices the link bursty to optimize the energy consumption on each packet and improve the network yield. However, DSF and L^2 need extra control overhead to stabilize the forwarding schedule, which is vulnerable to dynamic links and network churn.

ORW [8] implements the opportunistic routing for unsynchronized low-duty-cycled WSN, but shows the limited performance for high traffic load applications. DOF extends this work to more general purpose WSN applications. CMAC [9] includes the slotted acknowledgements, but CMAC still determines the unique forwarder by overhearing other's acknowledgments. In DOF, the sender distinguishes the forwarders, and then considers the link quality to arrange the forwarding schedule.

There are also some theoretical works focusing on opportunistic routing [19] [20] and dynamic forwarding [21] [22] for wireless sensor networks. Although the models and simulation show the efficiency of the opportunistic routing, they neglect the practical issues addressed by DOF.

VII. Conclusion

Developing an adaptive and efficient forwarding protocol is urgent for duty-cycled wireless sensor network. In this paper, we propose DOF, a duplicate-detectable unsynchronized low-power opportunistic forwarding which is adaptive to various traffic loads. Based on the slotted acknowledgement, DOF mainly solves the channel degradation problem incurred by the large amount of duplicates in traditional opportunistic forwarding and retains the benefits of the opportunistic routing as much as possible. The testbed experiments show DOF is more efficient and reliable than state-of-the-art low-duty-cycled forwarding protocols.

Acknowledgment

This study is supported in part by NSFC under Grant 61073177, National Basic Research Program (973) under Grant No. 2014CB347800, NSFC under Grant 61202359, NSFC Major Program 61190110, NSFC under Grant 61170213, and the Fundamental Research Funds for the Central Universities under grant ZYGX2011J062.

References

[1] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "Citysee: Urban CO_2 monitoring with sensors," in *Infocom*. IEEE, 2012, pp. 1611–1619.

[2] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai, "Canopy closure estimates with greenorbs: sustainable sensing in the forest," in *Sensys*. ACM, 2009, pp. 99–112.

[3] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy *et al.*, "Is there light at the ends of the tunnel? wireless sensor networks for adaptive lighting in road tunnels," in *IPSN*. IEEE, 2011, pp. 187–198.

[4] X. Wu, M. Liu, and Y. Wu, "In-situ soil moisture sensing: Optimal sensor placement and field estimation," *TOSN*, vol. 8, no. 4, p. 33, 2012.

[5] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Sensys*. ACM, 2004, pp. 95–107.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Sensys*. ACM, 2006, pp. 307–320.

[7] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 133–144.

[8] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling," in *IPSN*. ACM, 2012, pp. 185–196.

[9] S. Liu, K.-W. Fan, and P. Sinha, "Cmac: an energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks," *TOSN*, vol. 5, no. 4, p. 29, 2009.

[10] C. Szymon, J. Michael, K. Sachin, and K. Dina, "More: network coding approach to opportunistic routing," *MIT-CSAIL-TR-2006-049*, 2006.

[11] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *Sensys*. ACM, 2008, pp. 29–42.

[12] M. H. Alizai, O. Landsiedel, J. Á. B. Link, S. Götz, and K. Wehrle, "Bursty traffic over bursty links," in *Sensys*. ACM, 2009, pp. 71–84.

[13] D. S. 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] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Sensys*. ACM, 2010, pp. 1–14.

[15] Z. Cao, Y. He, and Y. Liu, "L^2: Lazy forwarding in low duty cycle wireless sensor networks," in *Infocom*. IEEE, 2012, pp. 1323–1331.

[16] R. Fonseca, P. Dutta, P. Levis, and I. Stoica, "Quanto: Tracking energy in networked embedded systems," in *OSDI*. USENIX Association, 2008, pp. 323–338.

[17] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Sensys*. ACM, 2009, pp. 1–14.

[18] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *Sensys*. ACM, 2007, pp. 321–334.

[19] X. Mao, X.-Y. Li, W.-Z. Song, P. Xu, and K. Moaveni-Nejad, "Energy efficient opportunistic routing in wireless networks," in *MSWiM*. ACM, 2009, pp. 253–260.

[20] G. Schaefer, F. Ingelrest, and M. Vetterli, "Potentials of opportunistic routing in energy-constrained wireless sensor networks," in *Wireless Sensor Networks*. Springer, 2009, pp. 118–133.

[21] J. Kim, X. Lin, and N. Shroff, "Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks," in *Infocom*, 2009, pp. 612–620.

[22] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, "On maximizing the lifetime of delay-sensitive wireless sensor networks with anycast," in *Infocom*. IEEE, 2008, pp. 807–815.