Exploiting Concurrency for Opportunistic Forwarding in Duty-Cycled IoT Networks

Article *in* ACM Transactions on Sensor Networks · May 2019

DOI: 10.1145/3322496

CITATIONS
6

READS
37

6 authors, including:

Cao Zhichao
Tianjin University
35 PUBLICATIONS **454** CITATIONS
SEE PROFILE

Yuan He
Tsinghua University
127 PUBLICATIONS **2,588** CITATIONS
SEE PROFILE

Xiaoyu Ji
The Hong Kong University of Science and Technology
10 PUBLICATIONS **104** CITATIONS
SEE PROFILE

Exploiting Concurrency for Opportunistic Forwarding in Duty-Cycled IoT Networks

DAIBO LIU, Hunan University
ZHICHAO CAO and YUAN HE, Tsinghua University
XIAOYU JI, Zhejiang University
MENGSHU HOU, University of Electronic Science and Technology of China
HONGBO JIANG, Hunan University

Due to limited energy supply of Internet of Things (Zhao et al. 2018) (IoT) devices, asynchronous duty cycle radio management is widely adopted to save energy. Since the sleep schedules of nodes are unsynchronized, a sender has to repeatedly send frames to coordinate with its receiver or keep sleeping until the receiver's wake-up time will come according to receiver's sleep-wake schedule. In such contexts, opportunistic forwarding, which takes the earliest forwarding opportunity instead of a deterministic forwarder, shows great advantage in utilizing channel resource for duty-cycled IoT networks. The multiple forwarding choices with temporal and spatial diversity increase the chance of collision tolerance in opportunistic forwarding, potentially enhancing the overall performance of duty-cycled multi-hop networks. However, since the current channel contention mechanisms mainly focus on collision avoidance, it is too conservative to exploit concurrency.

To address this problem, in this article, we propose COF to fully exploit the potential Concurrency for Opportunistic Forwarding in duty-cycled IoT networks. COF achieves concurrent transmission by: (i) measuring conditional link quality under the interference of on-going transmissions, and then (ii) further modeling the benefit of potential concurrency opportunities. According to the expected benefit of concurrency, COF decides whether or not to transmit in concurrent way. COF also adopts concurrency flag and signal features to avoid data collision caused by disordered concurrent transmissions and enhance the accuracy of conditional link quality estimation. COF can be easily integrated into the conventional unsynchronized and duty-cycled protocols. We have implemented COF and evaluated its performance on a 40-node testbed. The results show that COF can effectively exploit potential concurrency in opportunistic forwarding and COF outperforms the state-of-art protocols under diverse traffic load and network density.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

This work was partially supported by the National Key Research and Development Program of China (2018YFC0831800), NSFC 61772184, 61732017, 61502162, 61572219, 61702175, 61702451, ZJNSF Grant LGG19F020020, and the Fundamental Research Funds for the Central Universities.

Authors' addresses: D. B. Liu and H. B. Jiang, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, P.R. China; email: dbliu@hnu.edu.cn; Z. C. Cao (corresponding author) and Y. He, School of Software, TNLIST, Tsinghua University, Beijing 100084, P.R. China; emails: {caozc, heyuan}@tsinghua.edu.cn; X. Y. Ji, Electrical Engineering and Computer Science, Zhejiang University, Hangzhou 310058, P.R. China; email: xji@zju.edu.cn; M. S. Hou, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. China; email: mshou@uestc.edu.cn.

1550-4859/2019/05-ART31 $15.00
https://doi.org/10.1145/3322496

Additional Key Words and Phrases: Internet of thing (IoT), opportunistic forwarding, concurrent transmission, duty-cycled node, industrial sensor networks

ACM Reference format:
Daibo Liu, Zhichao Cao, Yuan He, Xiaoyu Ji, Mengshu Hou, and Hongbo Jiang. 2019. Exploiting Concurrency for Opportunistic Forwarding in Duty-Cycled IoT Networks. *ACM Trans. Sen. Netw.* 15, 3, Article 31 (May 2019), 33 pages.
https://doi.org/10.1145/3322496

1 INTRODUCTION

Energy efficiency is a fundamental issue in the design of forwarding protocols for Internet of Things (IoT) applications. Due to limited energy supply of IoT devices, asynchronous duty cycle radio management is widely adopted to save energy. In duty-cycled mode (Polastre et al. 2004; Liu et al. 2015), a node periodically switches its radio state between on (awake state) and off (sleeping state). A widely adopted protocol of duty-cycled media access control (MAC) is low power listening (LPL) (Polastre et al. 2004). Take BoX-MAC (Moss and Levis 2008), an advanced LPL-based protocol that has been widely applied by duty-cycled IoT networks, as an example. As shown in Figure 1, a node periodically turns its radio on to detect on-going traffic by checking channel signal strength. If channel is clear, it turns off the radio. Because the sleep schedules of different nodes are unsynchronized, transmitter (TX) has to wait until the receiver (RX) turns on its radio and acknowledges the receipt of a data packet. During the waiting period, TX has to continuously transmit the same data packet, referred to as *frame*, until the receiver's acknowledgment is received, or by adopting a phase-lock mechanism, TX keeps sleeping and starts its repeated frame transmissions just before the receiver will supposedly wake up.

Blind waiting in duty-cycled networks is generally energy-inefficient and brings about extra delivery latency. Especially when unexpected surge of traffic occurs, it can further lead to insecure data delivery performance. To shorten the blind waiting time, a practical approach is opportunistic routing (Biswas and Morris 2004), which takes the earliest forwarding opportunity instead of waiting for the wake-up of a deterministic forwarder. The forwarding opportunities include all the neighbors that are awake and offer sufficient routing progress toward sink node. The state-of-the-art opportunistic routing protocols, such as ORW (Ghadimi et al. 2014) and DOF (Liu et al. 2016), have shown promising improvement in terms of delivery latency, energy efficiency, and network reliability in duty-cycled networks with different traffic load and network density.

To what extent can we seize the forwarding opportunities is the key of opportunistic routing. Despite the fact that the waiting period is shortened, the practical performance of conventional opportunistic routing is still far from satisfactory. Our key observation is that existing collision-avoidance-based MACs are too conservative for duty-cycled opportunistic routing. Specifically, the multiple forwarding choices with temporal and spatial diversity increase the chance to tolerate collision in opportunistic forwarding. The interference from a specific neighbor is likely to have different influences on different candidate forwarders. If at least one of the potential forwarders can successfully receive a sender's frame under interference conditions or free from interference, opportunistic forwarding should be promoted rather than arbitrarily suppressed. We use the term *opportunistic exposed terminal* to denote such a phenomenon. Moreover, by hearing a frame of the on-going transmission, it is easy to get sufficient information for determining the potential benefit of concurrent transmissions, hereinafter referred to as CT. Hence, in duty-cycled networks, it is feasible and profitable to exploit potential concurrency for low power opportunistic forwarding. A lot of researches, such as capture effect (Ji et al. 2017; Lu and Whitehouse 2009; Roberts 1975; Whitehouse et al. 2005), conflict graph (Gupta and Kumar 2000; Rhee et al. 2006; Zhou et al. 2013),

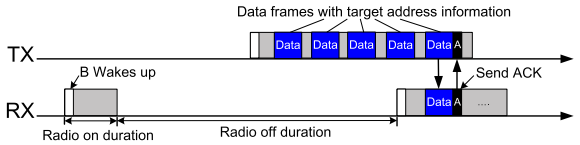


Fig. 1. An example of enhanced low power listening mechanism, namely BoX-MAC.

and physical interference model (Sha et al. 2009; Liu et al. 2010), have been proposed to improve channel utilization in the scenario of deterministic forwarding. However, concurrency for low-power opportunistic forwarding has not yet been well studied so far.

To address this problem, in this article, we propose COF, a practical approach to fully exploit potential *C*oncurrencies for *O*pportunistic *F*orwarding in duty-cycled IoT networks. COF addresses this problem by modeling conditional link quality under interference of on-going transmissions, and further modeling the benefit of potential concurrency opportunities based on estimated conditional link quality by considering the overall performance gain of all active nodes in local network. According to the expected benefit, COF can decide whether or not to transmit in concurrent way appropriately. Besides, COF also adopts concurrency flag to avoid data collision caused by disordered concurrent transmissions, and exploits signal features to monitor the completion of concurrent sender's transmission in real time, and therefore to enhance the accuracy of conditional link quality estimation. COF can be easily integrated into the conventional unsynchronized and duty-cycled protocols. The contributions of this work are summarized as follows:

—Based on the observation of *opportunistic exposed terminal* phenomenon, we propose COF to fully exploit potential concurrency for opportunistic forwarding in duty-cycled networks. This is the first work to achieve concurrent transmission in opportunistic routing.
—We adopt a distributed scheme to estimate conditional link quality under interference of other on-going transmissions, and model the benefit of potential concurrency opportunities.
—We adopt a concurrency flag and signal features to avoid data collision caused by disordered concurrent transmissions and enhance the accuracy of conditional link quality estimation.
—We have implemented COF in TinyOS-2.1.1 (Levis and Gay 2009) and evaluated its performance on a 40-node indoor testbed. Experimental results demonstrate that COF outperforms the state of art in terms of energy efficiency, delivery latency, and network reliability.

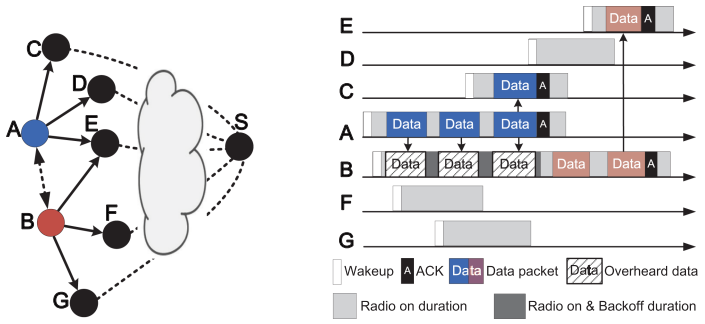
The rest of the article is organized as follows: We first explain the motivation of this article and introduce the preliminary knowledge of low-power opportunistic forwarding in the next section. Section 3 introduces the design overview of COF and the basis data structure used for conditional link quality estimation. Section 4 gives the detailed design of COF. We discuss the implement issues of COF in Section 5 and evaluate its performance in Sections 6. By introducing the related works and discussing the limitations of COF in Section 7 and Section 8 respectively, we finally conclude this paper in Section 9.

2 MOTIVATION AND PRELIMINARY

In this section, we first explain the requirements of low-power opportunistic forwarding in practical applications, and further introduce the preliminary knowledge and potential of CT in low-power opportunistic forwarding.

2.1 Application Requirements

A typical application of IoT is to monitor an environment for events that are of interest to the users. Usually, the events are rare. Yet when an event occurs, a large burst of packets are often generated that needs to be transported reliably and in real-time to an appointed base station.



(a) Topology for opportunistic forwarding (b) Process of low power opportunistic forwarding

Fig. 2. Example of traditional low-power opportunistic forwarding. (a) is network topology, where *A* (with candidate receivers *C*, *D*, and *E*) and *B* (with candidate receivers *E*, *F*, and *G*) are two senders that can interfere with each other, and (b) is a general process of low-power opportunistic forwarding.

Exemplary event-driven applications include volcano monitoring (Werner-Allen et al. 2006), structural monitoring (Xu et al. 2004), underground monitoring (Li and Liu 2007), and the like. Besides, as sensor nodes are normally battery powered, limited battery capacity requires sensor nodes to work in duty-cycled mode. Hence, a data-forwarding protocol that not only quickly transports the large amounts of data in heavy traffic-load scenario, but also saves energy if there is no data for transmitting is essential for these applications. Since low-power opportunistic forwarding has these characteristics inherently, it is regarded as the best candidate for this kind of application scenarios. However, opportunistic forwarding protocols are basically built upon collision avoidance mechanism. To avoid data collision, existing approaches significantly shrank spatial reuse, as described below. To further improve data-forwarding performance under network scenarios with unexpected surge of traffic, we should try to explore the improvement of spatial reuse in low-power opportunistic forwarding.

2.2 Preliminary Knowledge

2.2.1 Low-Power Opportunistic Forwarding. LPL has been widely applied for duty-cycled IoT networks. Low-power opportunistic forwarding is also built upon LPL. In low-power opportunistic forwarding, a frame may be heard and acknowledged by an earlier wake-up neighbor which can provide sufficient routing progress. This neighboring node is called a *forwarder*. By receiving an ACK that was replied by any one of the candidate forwarders, the period of repeating frames is therefore shortened. As shown in Figure 2(a), node *A* maintains a set of candidate forwarders which are denoted as $\mathbf{F}_A$ ($\mathbf{F}_A$={C, D, E}). The set of candidate forwarders of *B* is $\mathbf{F}_B$ ($\mathbf{F}_B$={E, F, G}). To send a packet to the appointed destination *S*, as shown in Figure 2(b), node *A* sends frames until *C* wakes up and acknowledges the reception of its frame.

Now we consider a general scenario. When *A* is transmitting, *B* is also holding a packet to transmit. In the current low-power opportunistic forwarding mechanisms, such as ORW (Ghadimi et al. 2014) and DOF (Liu et al. 2016), *A* or *B* can only exclusively access wireless channel. Thus, in this situation, *B* keeps its radio on and continuously conducts backoff, so as to wait for the channel to be free. In fact, during the waiting period, *B* misses the early opportunities to send its packet to *F* or *G*, which are free from the interference of *A*. Finally, *B* forwards its data packet to *E* after a long waiting period, resulting in relatively high energy consumption and delivery latency.

2.2.2 Opportunistic Exposed Terminal. The current designs of duty-cycled IoT networks generally adopt energy-efficient MAC protocols which are based on contention avoidance mechanism.

The low-power opportunistic forwarding protocols are directly built upon such MAC (e.g., LPL), ignoring a fundamental characteristic of opportunistic forwarding: *In opportunistic forwarding, each node maintains a set of candidate forwarders with temporal and spatial diversity, which means the impact of an interferer on different candidate forwarders (e.g., the impact of* A *on* E, F, *and* G *in Figure 2(a)) is likely to be different.* When two neighboring senders transmit in a concurrent way, even though in most of the candidate forwarders the received frames are corrupted, there may exist some forwarders that can successfully decode the frames. We use the term *Opportunistic Exposed Terminal* to denote such a phenomenon. By tolerating data collision in opportunistic exposed terminals, it is likely that opportunistic forwarding collaborating with concurrent transmission can significantly enhance network performance. However, existing opportunistic forwarding protocols strictly observe collision-avoidance mechanism, which hinders the harnessing of potential forwarding opportunities.

2.3 Potential of Concurrency

Here, we conduct experiments to qualitatively show the potential benefits of CT in low-power opportunistic forwarding and the possible harm if concurrency opportunities are not properly exploited. The experiments are conducted on an indoor testbed as shown in Figure 7. Similarly, nodes run ORW to forwarding data packets. The average number of forwarders of each node is 4.3, while the maximum and the minimum number are 7 and 2, respectively. The network diameter is 4 hops.

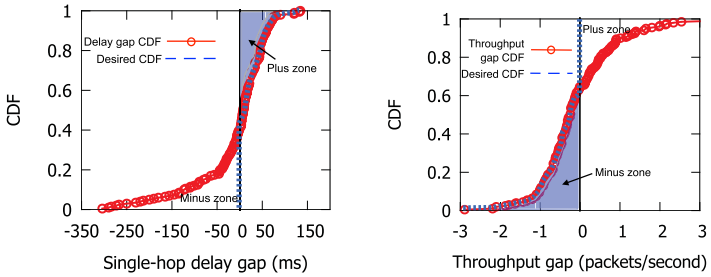
In the experiments, we select two neighboring nodes as appointed senders like node A and B in Figure 2(a), which are within the carrier sense range of each other and continuously generate data packets. The other network nodes generate data packets at an inter-packet interval (IPI) of 5 minutes. We repeat the experiments by respectively enabling and disabling the CSMA mechanism at the two senders. The CSMA mechanism of all the other network nodes are enabled all the time. The experiments are conducted for more than 100 times by selecting different neighboring senders. We record the single-hop delivery time of each packet, the total number of received packets at sink node, and the duration of each experiment. We compute the average single-hop transmission delay of the two senders with enabled CSMA (referred to as T_{csma}) and disabled CSMA (referred to as T_{nocsma}), and the average network throughput (received packets per second) of them with enabled CSMA (referred to as TP_{csma}) and disabled CSMA (referred to as TP_{nocsma}). Then we compute the single-hop delay gap (referred to as T_{gap}) and throughput gap (referred to as TP_{gap}) according to

$$T_{gap} = T_{csma} - T_{nocsma},$$
$$TP_{gap} = TP_{csma} - TP_{nocsma},$$

and plot the CDF of those 100+ experimental results in Figure 3(a) and Figure 3(b), respectively.

As shown in Figure 3, when CSMA is disabled in low-power opportunistic forwarding, a large part of transmissions can achieve shorter single-hop delay (the plus zone of Figure 3(a)) and finally achieve higher throughput (the minus zone of Figure 3(b)) than the CSMA-enabled cases. The plus zone of Figure 3(a) denotes that the enabled CSMA mechanism brings about a larger single-hop delay compared with disabled CSMA for a subset of neighboring senders, and the minus zone of Figure 3(b) denotes the enabled CSMA mechanism can bring about shrunk throughput for the corresponding part of neighboring senders. This observation indicates that there are many data transmissions (about 63.7% in our experiments) suffering from *opportunistic exposed terminal* problem. Exploiting the potential concurrency opportunities can alleviate this problem.

On the other hand, when all candidate forwarders are likely to be seriously interfered by concurrent senders, low-power opportunistic forwarding with disabled CSMA may induce serious data collisions and longer transmission delay (minus zone of Figure 3(a)) when CT is out of



(a) Single-hop delay gap (b) Throughput gap

Fig. 3. The CDF of the gaps of (a) single-hop delay, and (b) average radio duty cycle between the experimental results with CSMA disabled and enabled, respectively.

control. In this case, the network throughput will be sharply degraded (plus zone of Figure 3(b)). As illustrated in Figure 3(a), the minus zone denotes the disabled CSMA could instead bring about increased transmission delay because immoderate concurrent transmission could result in serious transmission collisions and retransmissions in the experiment, and the plus zone of Figure 3(b) denotes the corresponding shrunk throughput of disabled CSMA.

The experimental results imply that: we should allow neighboring senders to transmit in a concurrent way in the presence of *opportunistic exposed terminal*, and suppress CT when it is likely to hurt network performance. By doing this, we could achieve the desired single hop delay gap and throughput gap as shown by the bold blue dotted lines plotted in Figure 3. Visually, the dash areas of both Figure 3(a) and 3(b) are the potential improvement space for low-power opportunistic forwarding.

The empirical studies shed light on the potential of exploiting concurrency for low-power opportunistic forwarding. To achieve this goal, network nodes need to model the conditional link quality under inference of on-going transmissions and further model the benefit of concurrent transmission. Motivated by the above-mentioned results, we give the design of COF in the following sections.

3 COF OVERVIEW AND DATA TRANSMISSION STATUS

In this section, we first present the design overview of COF, then we introduce the basis data structure for recording data transmission status in COF, which plays a key role in measuring conditional link quality and modeling the benefit of CT mode.

3.1 Overview

Take Figure 4 as an example. A and B are with the case of *opportunistic exposed terminal*. As a neighbor of A, C is beyond the carrier sense range of B and F; F is a neighbor of B, and it cannot hear both A and C. As shown, A is transmitting now and B has a data packet to be sent. Because the assessed channel is busy (A is transmitting), B first defers its frame transmission. By hearing a frame transmitted by A, B knows the on-going sender. Then, according to the expected benefit of CT (see Section 4.2), COF authorizes B to transmit its frame concurrently with A. For each CT frame, the sender must add a flag to indicate which neighbor is the concurrent sender. For example, B adds a concurrency flag in its frame to record the identification (ID) of A. If there is no concurrent sender, the flag will be erased, such as the first frame transmitted by A.

Because A doesn't know B is in concurrent mode, by capturing a busy channel, A will defer its frame transmission until a frame is heard. According to the heard frame, A learns that B is

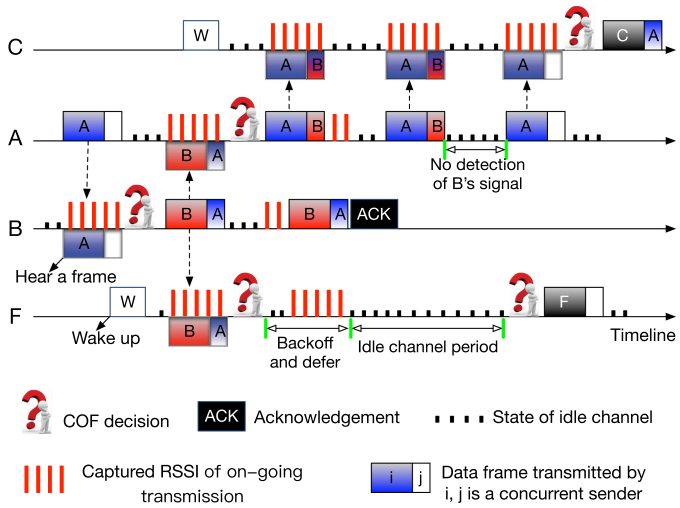


Fig. 4. Overview of COF for exploiting concurrency opportunity.

transmitting concurrently with itself. Hence, based on the expected benefit of CT, *A* will continue frame transmission by setting the concurrency flag to *B*. At the same time, *F* has also heard the frame transmitted by *B*. By checking the concurrency flag, *F* knows *B* is transmitting concurrently with another node. To avoid a data collision caused by disordered concurrent transmissions, *F* defers its frame transmission and conducts random backoff until the shared channel is free again or *B* is no longer in CT mode, just as *C* will do.

Besides, in CT mode, all concurrent senders periodically sample the channel signal strength indicator (RSSI) during the period of inter-frame interval and the period of deferment and backoff. Because of the special PHY modulation technique (DSSS), ZigBee (IEEE Computer Society 2003) has little fluctuation of signal power. By comparing the sampled RSSI with the average signal power of concurrent sender, each node can credibly determine whether/when its concurrent sender has completed frame transmission. As node *A* does in the figure, once the sampled RSSIs indicate *B* has completed frame transmission, *A* clears the concurrency flag in subsequent frames, and then *C* will join in CT mode.

The criteria of COF decision is based on the expected benefit of concurrent transmission defined in Section 4.2. The expected benefit of concurrent transmission is computed by using conditional link quality under influence of neighboring sender. In the following subsection, we explain how to record data transmission status for the computation of conditional link quality in Section 4.

3.2 Data Transmission Status

To compute the conditional link quality, several statuses of data transmission are indispensable. For each data transmission, the sender needs to know which forwarder successfully received the frame. Meanwhile, sender needs to know what caused a failed transmission: i) data collision at forwarders' receiving end; or ii) ACK collision at its own transmitting end. In this section, we introduce a data structure for recording data transmission statuses that could reveal the truth concerned above.

3.2.1 Bitmap for Status Recording. In low power opportunistic forwarding, each node acts as both a sender and a forwarder. As a sender, it transmits data packets cached in sending buffer and records basic information for each transmission. Note that each data transmission is assigned to a

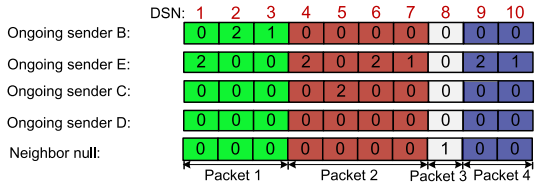
unique data sequence number (DSN), and an unacknowledged transmission lasts for a sleep-wake cycle (e.g., 512ms in TinyOS (Levis and Gay 2009)) to guarantee each forwarder has at least one opportunity to receive the packet. During this transmission period, the sender periodically transmits frames by carrying the same packet as in Figure 1. These periodically transmitted frames share the same DSN. In the following sections, the sleep-wake cycle transmission is expressed in terms of *data transmission*. The data transmission is organized by network layer protocol, while the transmission of periodic frames is organized by MAC layer protocol. As a forwarder, it acknowledges received frames that were transmitted by its children nodes and records the number of received frames assigned with the same DSN.

3.2.2 Transmitting Statuses. We first consider the case that a node acts as a sender. In this case, each node maintains a set of bitmaps for all neighboring nodes to record the transmission results of data transmissions. Hereinafter, we uniformly call them *sender bitmaps*. Each neighboring node is assigned a unique bitmap. These bitmaps are allocated with the same size (10 bytes) and equally divided into a fixed number of units. Each unit of the bitmaps is used to record the status (see below) of a data transmission result. Each bitmap is organized circularly and orderly according to DSN.

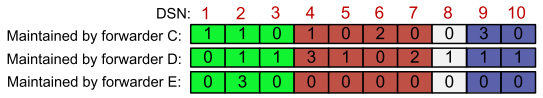
In the design of COF, each unit of a bitmap should distinguish: whether the corresponding data transmission was acknowledged or not, and whether there was a neighboring node that transmitted concurrently with the data transmission. Hence, we use four statuses to record and distinguish these information, and record the status of each data transmission in a unit. Necessarily, the unit size is two bits. Status 1 denotes that an ACK is received for a data transmission, status 2 denotes the data transmission is not acknowledged, and status 0 denotes a neighboring node corresponding to the bitmap didn't transmit during the sender's data transmission period. We also assign a bitmap to record the resulting statuses of data transmissions when no neighboring node transmits concurrently with the sender (marked as neighbor null). The statuses of all units are initialized to 0.

By completing a data transmission assigned with DSN n, the units of all bitmaps corresponding to DSN n are consistently updated. For simplicity, these corresponding units are denoted as μ_n. If the data transmission is transmitted concurrently with a neighboring node, it first sets the unit μ_n of the corresponding bitmap to a non-zero status, and simultaneously sets the μ_ns of the other bitmaps to 0. Specifically, the non-zero status should be set to 2 if the data transmission is not acknowledged, and 1 if an ACK is received. According to the maintained bitmaps and recorded statuses, we can know exactly which neighboring node transmitted concurrently with the sender and the transmission result. Taking node A in Figure 2(a) as an example, it maintains five bitmaps for neighbors B, C, D, E, and neighbor null as shown in Figure 5(a). Node A successfully transmits 4 packets over total 10 data transmissions, separately assigned with DSN from 1 to 10. The same color square boxes correspond to the same packet in Figure 5. Sender A transmitted the first data transmission (DSN 1) concurrently with E, but the data transmission was not acknowledged. Then, A retransmitted the data transmission (DSN 2 and 3) concurrently with B and the last retransmission was acknowledged.

3.2.3 Receiving Statuses. As a forwarder, it also maintains a set of bitmaps for its potential senders. In the following sections, we called them *forwarder bitmaps*. Each bitmap unit is used to record the number of received frames which are transmitted by a potential sender and are assigned with the same DSN. In opportunistic forwarding, once a forwarder receives a frame transmitted by a children node, it immediately replies with an ACK. Hence, the received multiple (k) frames assigned with the same DSN indicate that at least the previous $k-1$ ACKs were collided and lost at the sender. Furthermore, if the next data transmission is a retransmission with increased DSN, we can conclude that the previous k ACKs were all lost. Note that as a forwarder, it cannot know whether a data transmission is a retransmission only according to the receiving statuses



(a) Bitmaps recorded by sender A



(b) Bitmaps recorded by A's candidate forwarders

Fig. 5. (a) Bitmaps recording the state of each transmission at sender A, and (b) Bitmap recording the received copies of each transmission at A's candidate forwarders. Each color represents the complete transmissions of a packet.

recorded in forwarder bitmaps. But by feeding back this information to the corresponding sender, combined with the above-mentioned transmitting status maintained by the sender, it can know the exact number of lost ACKs. Hence, the computation of conditional link quality is conducted at each node acting as a sender. For example, in Figure 5(b), E consecutively received three frames assigned with DSN 2 transmitted by A, however, its acknowledgments all collide at A's transceiver because A retransmitted the same packet by assigning with DSN 3 (see the transmitting statuses in Figure 5(a)). We also use two bits to record the number of received frames of each data transmission. In duty-cycled IoT networks, the duration for keeping in the active state after each wake-up is very short, and receiving a duplicate frame will not trigger an extended wake-up time. Hence, two bits are enough to record the number of received frames for the vast majority of cases. By explaining how to get the status of each data transmission, we next introduce the detail design of COF.

4 PROTOCOL DESIGN

In this section, we present the detail design of COF for exploiting concurrency opportunities in low-power opportunistic forwarding. We first describe the symbols involved in this article in Table 1. Then, we introduce the computation of conditional link quality (see Section 4.1) and model the expected benefit of CT in Section 4.2. Beyond that, we explain how COF exploits concurrency flag and RSSI characteristics to avoid disordered CT mode and enhance the accuracy of conditional link quality (Section 4.3). Finally, we introduce the COF decision in Section 4.4 and the initialization of COF in Section 4.5.

4.1 Conditional Link Quality

We use conditional packet delivery ratio (*cpdr*) to denote the conditional link quality under interference of the on-going neighboring sender. We define *cpdr* as the probability, $P_{i,j}^{N}$, that during a data transmission period, forwarder j can receive at least one frame transmitted by sender i when neighbor N transmits concurrently. By collecting the recorded receiving statuses from each candidate forwarder (see Section 4.6) and using the transmitting statuses maintained by itself, for each data transmission under the interference of a specific neighboring node, a sender can know the number of frames that each forwarder received, whether the data transmission was acknowledged or not, and the number of ACKs that it lost. Furthermore, it can figure out the cause of a failed data transmission: data collision at the forwarders or ACK collision at its own transceiver front-end.

Table 1. Symbols Description

Symbols	Description of Symbol
$P_{i,j}^{N}$	The probability that forwarder j will receive the packet sent by i when neighbor N is concurrently transmitting.
$cpdr$	A link's conditional packet delivery ratio under the influence of a neighbor's transmission.
$epdr(A\|B)$	The expected packet delivery ratio of A's transmission under the influence of B.
$epdr(A\|\emptyset)$	The expected packet delivery ratio of A's transmission when there is no other on-going transmitter.
$EGain(A\|B)$	The expected benefit of CT than the individual transmission of B.
BTable	A table maintaining the expected benefit of CT.
CPDR	A table maintaining links' $cpdr$s.
$\mathbf{S}_i^N$	The set of data transmissions transmitted by i and influenced by N.
$ACK_m(j)$	The number of ACKs replied by j for data transmission m.
π_m	A correction parameter for accurately computing $cpdr$.
$\delta_m(j)$	Indicating whether forwarder j received the mth data transmission.
θ, α	Parameters of moving average for updating link's $cpdr$.

Hence, the sender can compute the links' bidirectional $cpdr$s between it and each of its candidate forwarders.

To compute the bidirectional $cpdr$s between a sender i and each of its candidate forwarders, sender i should first collect the forwarder bitmaps maintained by its candidate forwarders. Then, for a specific neighboring node N, i could compute the up-to-date $cpdr$s of all links between it and each of the candidate forwarders under influence of the data transmission of N. To achieve that, sender i should assemble all the data transmissions (DSNs) whose resulting statuses are non-zero in the sender bitmap assigned to N, and also assemble the receiving statuses corresponding to these DSNs in the forwarder bitmap maintained and fed back by N. The non-zero statuses denote the sender i and neighbor N transmitted concurrently. We use S_i^N to express the set of data transmissions.

First, the sender i can compute the up-to-date $cpdr$ of the link from i to a specific forwarder j under the influence of an on-going data transmission of N, denoted as $P_{i,j}^{N}$, according to

$$P_{i,j}^{N} = \frac{\sum_{m \in \mathbf{S}_i^N} \delta_m(j)}{|\mathbf{S}_i^N| - \sum_{m \in \mathbf{S}_i^N} (A_m - \pi_m)}, \tag{1}$$

where j is a candidate forwarder of the sender i. $\mathbf{S}_i^N$ denotes the set of data transmissions of sender i interfered by the neighboring node N. Taking Figure 5 as an example, there are two data transmissions in $\mathbf{S}_A^B$. The m denotes the mth data transmission in $\mathbf{S}_i^N$ influenced by N, and $\delta_m(j)$ denotes whether the forwarder j received at least one frame of the mth data transmission. COF refers to the receiving status at the forwarder bitmap maintained by j to determine the value of δ_m by

$$\delta_m(j) = \begin{cases} 1 & \textit{if j received at least one frame;} \\ 0 & \textit{if j didn't receive any frame of the m.} \end{cases}$$

Hence, the numerator of Equation (1) denotes the number of data transmissions that the forwarder j has successfully received under the influence of N.

On the other hand, the denominator of Equation (1) is the number of total data transmissions transmitted by i under the influence of neighboring node N. Specifically, $|\mathrm{S}_i^N|$ denotes the number of data transmissions in the set. A_m denotes whether the data transmission m was acknowledged. If an ACK is received for the data transmission m, A_m is 1. Otherwise, A_m is 0. Before considering the accuracy of Equation (1), we should first note that the acknowledged data transmission can only indicate some forwarder has successfully received the transmitted packet and then replied an ACK. However, for the other forwarders which did not receive the packet, we cannot infer whether the frames of the data transmission were lost due to the influence of the neighboring node N or the frames were missed due to forwarders' sleeping in an asynchronous low-power opportunistic forwarding. Hence, we add a correction parameter π_k in Equation (1). If the forwarder j replies an ACK and sender i happens to receive an ACK for a data transmission, the value of π_k is 1, no matter whether the received ACK does come from the forwarder or not (because it is unable to accurately know it). Otherwise, π_k is 0. This assignment makes sense, because if a forwarder stays in the sleeping state and has no opportunity to acknowledge the sender's data transmission but the sender receives an ACK replied by another forwarder, the last data transmission will not be used to compute the unidirectional *cpdr* of the link between the sender and the sleeping forwarder. Considering that a sender could have multiple candidate forwarders, there may be several sleeping forwarders when a data transmission is acknowledged.

Equation (1) computes the probability that a data transmission could successfully carry a packet to a specific forwarder j under the influence of a neighboring node N. In the same way, under the influence of N, the *cpdr* of an ACK transmitted from j to sender i can also be computed according to

$$P_{j,i}^N = \frac{\sum_{m \in \mathrm{S}_i^N} \pi_m}{\sum_{m \in \mathrm{S}_i^N} ACK_m(j)}, \tag{2}$$

where $ACK_m(j)$ is the number of ACKs replied by j for acknowledging the received frames of the data transmission m, which is transmitted by i and is influenced by N. π_m is the same as that of Equation (1). If j replies an ACK for the data transmission m and sender i does receive an ACK, π_m is 1, otherwise, π_m is 0. Hence, $\sum_{m \in \mathrm{S}_i^N} \pi_m$ is the number of ACKs that were replied by j and (probably) received by i, and the denominator of Equation (2) is the number of all ACKs replied by j. Hence, $P_{j,i}^N$ denotes the probability that an ACK can be successfully delivered from j to i under the influenced of N.

Note that the up-to-date *cpdr*s is a partial view of the overall *cpdr*s. To fully show the *cpdr*s considering both accuracy and network dynamics, we use a moving average to update both $P_{i,j}^N$ and $P_{j,i}^N$ by

$$P_{i,j}^N = (1-\theta) \times P_{i,j}^{N,old} + \theta \times P_{i,j}^{N,new}, \tag{3}$$

$$P_{j,i}^N = (1-\alpha) \times P_{j,i}^{N,old} + \alpha \times P_{j,i}^{N,new}. \tag{4}$$

Both θ and α will be discussed in the implementation of COF in detail. By computing and updating link *cpdr*s, we further compute the expected benefit of CT in the next section.

4.2 Expected Benefit of CT

We define the expected benefit of CT as the expected gain (*EGain*) of the number delivered packets for the period of a data transmission. By computing and updating *cpdr*s, each node i constructs a **CPDR** table to maintain *cpdr*s. The table consists of multiple entries. Each entry corresponds to a neighboring node N which could concurrently transmit with itself. The entry forms as (N, $< P_{i,F_1}^N, P_{F_1,i}^N >, \ldots, < P_{i,F_n}^N, P_{F_n,i}^N >, epdr(i|N)$), where $\{F_1, \ldots F_n\}$ is the forwarder set of node i,

and $epdr(i|N)$ is the expected packet delivery ratio (*epdr*) of a data transmission transmitted by i and influenced by neighboring node N. According to the computed *cpdr*s listed in each entry, COF computes $epdr(i|N)$ by

$$epdr(i|N) = 1 - \prod_{j \in \mathbf{F}_i} \left(1 - P^N_{i,j} \times P^N_{j,i}\right), \tag{5}$$

where $P^N_{i,j} \times P^N_{j,i}$ denotes the probability that both a data transmission from i to j and the replied ACK from j to i succeed under the influence of N, and $\prod_{j \in \mathbf{F}_i}(1 - P^N_{i,j} \times P^N_{j,i})$ denotes i cannot receive an ACK from any candidate forwarder after a data transmission. Note that $epdr(i|N)$ only indicates the impact of N on the data transmission of sender i.

By computing the *epdr* and recording it in each entry of **CPDR** table, each node should notify all neighboring nodes by broadcasting the ID of a neighboring node and the corresponding *epdr*, such as $< N, epdr(i|N) >$ by taking node i and neighboring node N as an example. The broadcast strategy is introduced in Section 4.6. Once overhearing the notified information, COF extracts the items involving itself and records (or updates) them in the benefit table which is marked as **BTable**. For example, if node i overhears the broadcasted *epdr* information from the neighboring node N, it only extracts the items: $< i, epdr(N|i) >$ and $< i, epdr(N|\text{ø}) >$, where $epdr(N|\text{ø})$ denotes the expected packet delivery ratio of N's data transmission when N is not interfered with by other nodes. Then i maintains these items in an entry to the **BTable** assigned to N. The entry to the **BTable** is written as $< N, epdr(i|N), epdr(N|i), epdr(N|\text{ø})$, *permission or denial of concurrency*$>$, where $epdr(i|N)$ is computed by node i in accordance with Equation (5).

Based on the maintained/updated entries to the **BTable** table, each node could compute the expected benefit of CT (marked as *EGain*) then transmit in sequence. If N is the on-going transmitter and i intends to transmit at this time, i should check the overall gain of CT, namely as $T(i|N)$. $T(i|N)$ is computed by

$$T(i|N) = epdr(i|N) + epdr(N|i). \tag{6}$$

Note that $epdr(N|\text{ø})$ is also recorded in each entry to **BTable** table. Hence, the overall benefit ($EGain(i|N)$) of CT can be computed by

$$EGain(i|N) = T(i|N) - epdr(N|\text{ø}). \tag{7}$$

If $EGain(i|N)$ satisfies the condition

$$EGain(i|N) > \omega, \tag{8}$$

we consider that it is more beneficial to transmit in the way of concurrency. ω is a compensation value for the extra consumption (e.g., energy) of CT and will be discussed in Section 5. To ensure the consistency of decisions made by both i and N, i also checks N's $EGain(N|i)$ by

$$EGain(N|i) = T(N|i) - epdr(i|\text{ø}) > \omega. \tag{9}$$

According to the double check of both Equation (8) and Equation (9), COF can guarantee the consistency of decisions made by concurrent senders.

If both Equations (8) and (9) are satisfied, COF will permit the transmission of i when N is transmitting, and adds/updates the permission marker (*yes*) in the last column (permission or denial of concurrent) of the **BTable** table. Otherwise, it adds/updates the denied marker (*no*) in the table.

4.3 Concurrency Control and Sender Monitoring

To guarantee the accuracy of conditional link quality, COF should avoid concurrency collision caused by disorder CTs and capture the end of concurrent sender in time. COF adopts concurrency flag to control concurrent transmissions between neighboring senders and exploits signal features to monitor the completion of concurrent sender's transmission in real time.

Table 2. Characteristics of Common 2.4GHz Technologies

Wireless technology	On-air time	PAPR
ZigBee	$[576, 4256]\mu s$	$\leqslant 1.3$
WiFi	$[192, 542]\mu s$	$\geqslant 1.9$
Bluetooth	$366\mu s$	$\leqslant 1.3$
MWO	10ms	$\geqslant 2.9$

4.3.1 Concurrency Control. COF supports concurrent transmission for neighboring nodes. It controls concurrency mode according to the following policies.

No Concurrency. No concurrency contains two situations. For the first situation, no neighboring node is transmitting and the sampled channel state indicates a clear channel. In this situation, a sender having a new packet can transmit immediately. Another situation is when only one neighboring node is transmitting with unassigned concurrency flag. According to the expected benefit of CT mode, a new sender decides whether or not it should join in CT mode, as illustrated by the first heard frame at node *B* in Figure 4.

CT Mode. CT mode denotes when a node transmits frames concurrently with an on-going neighboring sender. In CT mode, the concurrency flag of transmitted frames is set to the ID of neighboring sender, and only the node indicated by the concurrency flag is permitted to transmit in this mode.

Exclusive Mode. As a sender works in the CT mode, it occupies the shared wireless channel in the exclusive mode for all neighboring nodes, except the on-going sender indicated by its concurrency flag. The other neighboring nodes having data packets to transmit have to be deferred until CT mode in heard frames is canceled or channel becomes idle. Besides, if the channel is busy and no frame was heard, frame transmission is not allowed.

4.3.2 Sender Monitoring. We introduce the features that can be used to distinguish Zigbee from other coexistent interference, and further adopt the features of short-term RSSI sequence proposed by Meter (Liu et al. 2017) and Zisense (Zheng et al. 2017) to determine whether or not the concurrent sender has finished its transmission, and when. Based on this information, the accuracy of conditional link quality is enhanced. The pseudocode of monitoring strategy is given by Algorithm 1.

Signal Features. One of the significant features of the ZigBee signal is the stability of the signal strength. Peak to Average Power Ratio (PAPR) is a common measure of the fluctuation of signal power. The different modulation techniques lead different *PAPR*. As shown by previous studies (Schurgers 2001), 802.11g/n has a large *PAPR* ($\geqslant$1.9). This is due to Orthogonal Frequency Division Multiplexing (OFDM) (Le Floch et al. 1995), the multiple sub-carriers modulation technique adopted by WiFi (IEEE Computer Society 2012). In OFDM, each subcarrier has a certain level variation of signal strength. The received signal is a sum of the signals on all the orthogonal sub-carriers. Thus, the variation of the sum will be larger than that of a single carrier. ZigBee adopts Direct Sequence Spread Spectrum (DSSS) which utilizes the entire frequency range to transmit data so that its *PAPR* is relatively stable. The *PAPR*s of common 2.4GHz technologies are listed in Table 2.

Another feature is *on-air time*, which indicates the transmission period of an individual frame. Due to the different data rate and maximum frame size of different techniques, their *on-air time* is usually different. The on-air time of a normal ZigBee frame of CC2420 radio is between [576,

ALGORITHM 1: Monitoring the State of Concurrent Transmission in CT Mode

```
Input: CT_RSSI, CT_RSSI;
Output: State of concurrent transmission.
if Signal was detected before the scheduled time of the next frame transmission then
    Sample entire signal or no less than 4 RSSIs;
    Record RSSI sequences in TEMP_RSSI;
    Calculate TEMP_RSSI;
    Identify Zigbee frame;
    if Identified as Zigbee frame then
        if |TEMP_RSSI − CT_RSSI)| ⩽ 1dBm then
            Return Stay in CT mode;
        else
            Defer to next frame transmission;
            Keep in listening state;
            Return Stay in CT mode;
        end
    else
        Conduct backoff;
        Return Stay in CT mode;
    end
else if Hear a frame in listening state then
    if Concurrency flag is set to this node OR Concurrency flag is not set then
        Update CT_RSSI and CT_RSSI;
        Update my concurrency flag;
        if Transmitter is different to my previous concurrency flag then
            Feed back the completion of previous concurrent sender's transmission;
        Return Stay in CT mode;
    else
        Cancel concurrency flag;
        Clear CT_RSSI;
        Return Exit CT mode;
    end
```

4256]μs. The valid packet lengths and data rates specified by the underlying IEEE standard 802.11 (IEEE Computer Society 2012) limits the on-air time of a WiFi packet in [192,542]μs. Bluetooth (IEEE Computer Society 2005) adopts a frequency hopping technique. The standard hopping rate is 1600 hop/s, i.e., 625μs residence time in one channel. The standard also specifies that the transmission time in one channel is 366μs. As shown in Table 2, unlike ZigBee, WiFi, and Bluetooth have a shorter on-air time, while microwave ovens (MWO) have a longer on-air time.

Monitoring Strategy. To join in CT mode, both the new sender (like node B in Figure 4) and the on-going sender (like A in Figure 4) necessarily go through two stages, respectively: (i) collision detection and deferring frame transmission and (ii) hearing a frame of neighboring sender for the concurrent transmission decision. During the first stage, each node periodically samples the signal strength of the shared channel. If the channel is busy according to the Clear Channel Assessment (CCA) threshold, it records the sampled RSSI sequences in CT monitoring set: **CT_RSSI**=$\{R_0, R_1, \ldots, R_n\}$, where R_i denotes the ith sampled RSSI value. At the second stage, if a frame is received and the CT mode is permitted by COF (see the next section), the recorded

RSSI sequences become effective; otherwise, COF clears recorded RSSI information. The averaged RSSI of the CT monitoring set can be calculated by $\overline{CT\_RSSI} = \frac{\sum_0^n R_i}{n}$.

In the CT mode, to monitor the completion of another sender's transmission in real time, each concurrent sender also periodically samples the signal strength of the shared channel during the period of inter-frame interval. We label the consecutive sequence of RSSIs above the CCA threshold as the temporary monitoring set: **TEMP_RSSI**=$\{TR_0, TR_1, \ldots, TR_k\}$. Note that the radio chip, such as CC2420 (Texas Instruments 2006), usually has a built-in RSSI module recording the instantaneous received signal strength. The RSSI value is always averaged over eight symbol periods ($128\mu s$). Hence, the sample interval T_i is set to $128\mu s$. By getting no less than four samples in **TEMP_RSSI**, COF exploits signal features to quickly identify wireless technology of the signal corresponding to **TEMP_RSSI** by adopting Zigbee frame identification algorithm proposed in Liu et al. (2017) (see Line 5 in Alg. 1). Note that the *PAPR* of **TEMP_RSSI** can be calculated according to

$$PAPR(TEMP\_RSSI) = \frac{max\{TR_i^2 | 0 \leqslant i \leqslant k\}}{\overline{TEMP\_RSSI^2}}, \tag{10}$$

where $\overline{TEMP\_RSSI^2}$ denotes the average of the squared values of the elements in CT monitoring set **TEMP_RSSI**. $\overline{TEMP\_RSSI^2}$ can be calculated by

$$\overline{TEMP\_RSSI^2} = \frac{\sum_0^k R_i^2}{k}. \tag{11}$$

If the identified signal is not a Zigbee transmission, COF defers its frame transmission (Line 14-15). Otherwise, it continues to compare the averaged RSSI value of **TEMP_RSSI** with $\overline{CT\_RSSI}$ (Line 7). According to Liu et al. (2017), if

$$|\overline{TEMP\_RSSI} - \overline{CT\_RSSI})| \leqslant 1dBm, \tag{12}$$

COF treats the on-going sender as the concurrent sender marked in its concurrency flag (Line 8). The only exception is that the sender recording **TEMP_RSSI** is hidden to the on-going sender other than the one marked by concurrency flag, while the sender can sense the on-going sender and the signal strength is very close to that of the concurrent sender marked by concurrency flag. However, the probability of this case is extremely low. In order to avoid the influence of hidden terminal problem, if Equation (12) is not satisfied, COF defers its data transmission until it hears another frame (Line 16). Then, it decides whether or not to join in the CT mode according to the strategy mentioned in the above section: if the heard frame's concurrency flag is set to it or not set (Line 17), COF just records the new **CT_RSSI** and $\overline{CT\_RSSI}$, and updates the corresponding concurrency flag. Otherwise, COF cancels its concurrency flag (Line 24), clears **CT_RSSI**, and exits the CT mode (Line 26).

Different from above-mentioned cases, if the concurrent sender was undetected in two consecutive inter-frame intervals, COF thinks the concurrent sender's transmission has finished. Hence, it cancels CT mode by clearing concurrency flag. Then other neighboring senders have opportunity to join in CT mode.

4.4 Transmission Decision

COF is a built upon conventional duty-cycled MAC protocol. It persistently monitors the current transmissions of neighboring nodes by sampling shared wireless channel and hearing potential frames during non-transmission period. Then, it accurately assesses the feasibility of transmission in the CT mode in real-time.

To transmit a data packet, the MAC layer not only uses carrier sense to determine whether a channel is busy, but also passes a transmission notification event to COF. COF first confirms whether a frame was heard during the last several milliseconds (ms). If nothing was received, COF returns a value denoting *no recommendation* to the MAC protocol. Then MAC makes the transmission decision according to the principle of exclusive use of shared channel. Otherwise, if a frame sent by a neighboring sender was heard, COF first queries the **BTable** table to verify whether CT is permitted according to Equations (8) and (9). Then it returns a value denoting the *permission of CT* (*yes*) or *denial of CT* (*no*) to the MAC protocol.

If the returned value from COF is *permission of CT*, MAC immediately transmits the pending data packet by disabling the carrier sensing for one data transmission period. However, if the returned value is *denial of CT*, no matter whether the MAC layer is transmitting or not, it pauses the MAC layer's transmission and conducts backoff. After each backoff, if the node wants to transmit again, the MAC protocol will also conduct carrier sense and send a transmission notification event to COF as mentioned above. If COF returns *no recommendation*, the MAC protocol decides whether or not to transmit by only referring to the carrier sense result. During the CT mode, COF also monitors the transmission of the concurrent sender by exploiting signal features to enhance the accuracy of the link quality estimation. Once detecting the completion of the previous concurrent sender's frame transmission, it immediately updates the corresponding sender bitmap.

4.5 Initialization of COF

Initially, the **CPDR** table and the **BTable** table are empty, and there is no *permission* or *deny* rule supporting the COF decision. In order to fast construct and optimize the **CPDR** table, COF initially sets a link's *cpdr* to its routing link quality, aggressively allowing nodes to concurrently transmit.

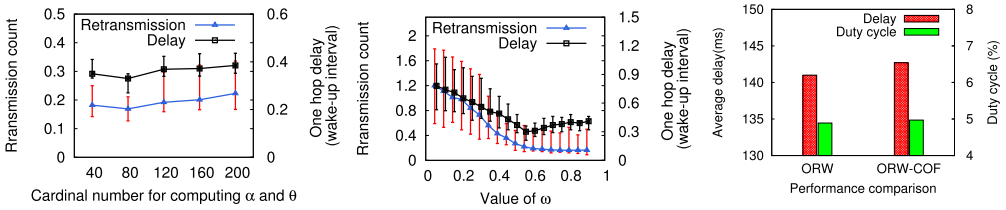
Note that the excessive indulgence of CT in the initial stage may result in consecutive transmission failures. To address this problem, COF uses the enforcement of denial of concurrent: Once COF becomes aware of consecutive failures in routing layer (exceeding six retransmissions in our implementation), it actively issues a *denial of concurrent* event to the *transmission decision module* of the MAC protocol to enable carrier sense for the next transmission.

4.6 Information Collection

We adopt two ways to feed back the maintained forwarder bitmaps to all candidate children nodes by exploiting network probe and data packet footer, respectively. The footer is defined as the extra space of the difference between the maximum payload size and the actual payload size.

Without changing the original mechanism of the probe transmission, COF only broadcasts a probe carrying the forwarder bitmaps and recorded information in the **BTable** table with a long time interval. In COF, the interval is adaptively set to 5 to 10 minutes according to traffic load. A COF probe will not be concurrently transmitted with another on-going sender by setting a specific concurrency flag, and any data transmission is banned to concurrently transmit with a broadcast probe. In addition, we also fully utilize the free space of system network probes by adding the most frequently updated bitmaps into probe footer.

Additionally, COF also exploits the possible opportunity of small data packets, which have free space to carry at least one forwarder bitmap. By attaching the most frequently updated bitmaps into the appointed packet footer, each node could quickly disseminate the frequently updated bitmaps. Note that the exploitation of the free space of data packet is independent of probe transmission. COF doesn't guarantee that the attached bitmaps in data packets can be heard by all children nodes. But as time goes on, each node could collect sufficient information to calculate and update *cpdr*. By exchanging recorded information in the **BTable** table between each pair of



(a) Computation of α and θ (b) Value of ω (c) Overhead of COF

Fig. 6. Discussion about the implement issues of COF. (a) Effect of changing the cardinal number for computing α and θ on performance (average retransmission count and average one-hop delay); (b) Effect of ω on performance; and (c) Effect of the extra overhead of COF on performance.

neighboring nodes, they will immediately update the expected benefit of CT, and keep the consistency of permission of concurrency between them.

5 IMPLEMENTATION ISSUES

We have implemented COF in TinyOS 2.1.1 (Levis and Gay 2009). The RAM and ROM consumptions of COF are 947 bytes and 3186 bytes, respectively. As an opportunistic forwarding protocol, COF uses the recently proposed routing metric EDC (Ghadimi et al. 2012) to construct network topology. In this section, we give in-depth discussions on several implementation issues.

5.0.1 Link cpdr Update. In Equations (3) and (4), the selected values of θ and α should consider both the accuracy and the adaptation of *cpdr*. Because the number of consumed DSNs of each update over an individual link may be different, the corresponding change rate of *cpdr* should also be different for each update. When updating the *cpdr* of a link, we mark the number of DSNs related to the link as N_i which is the denominator of Equations (1) or (2). Then we set a cardinal number CN to update θ and α, where $\theta = \frac{N_i}{CN}$ (or $\alpha = \frac{N_i}{CN}$). Since the maximum number of N_i is 40 in our implementation (2 bits denote a DSN and a 10Bytes bitmap can accommodate 40 DSNs), we set CN to different values (ranging from 40 to 200) to test the effect of θ and α on the average retransmission count and a single-hop delay. It is reasonable to suppose that the optimal θ and α can result in good performance. In each experiment, we set each node's IPI to 4 seconds, and each experiment lasts 2 hours in an indoor testbed. We plot the average retransmission count and the average one-hop delay by changing the value of CN in Figure 6(a). The delay is transformed from the time cost to wake-up interval by $\frac{time}{wake-up\ interval}$, and the wake-up interval is set to 512 milliseconds. From the experiment results shown in the figure, setting CN to 80 can achieve a good performance. Although it is difficult to justify the optimality of CN here, the value of CN is reasonable. We will give an in-depth discussion on the optimization of parameter setting in future work.

5.0.2 Compensation Value ω. In Equations (8) and (9), the weight ω is a compensation value for the expected benefit of CT. Generally, a large ω could reduce the opportunity for CT, but it also reduces retransmission rate. On the other hand, assigning a very low value to ω could increase retransmission rate caused by data collision and result in a high transmission delay. Thus, assigning an appropriate value to ω is important for achieving high network performance, such as the one-hop delay and transmission efficiency. We conduct experiments in the indoor testbed by calibrating ω. We plot the average retransmission count and the average one-hop delay by changing the setting of ω in Figure 6(b). From the experimental results plotted in the figure, 0.55 is a reasonable value.

5.0.3 Network Overhead. COF adopts both network probe and data packet footer to share the recorded forwarder bitmaps. The overhead is very limited. Here, we conduct two experiments in

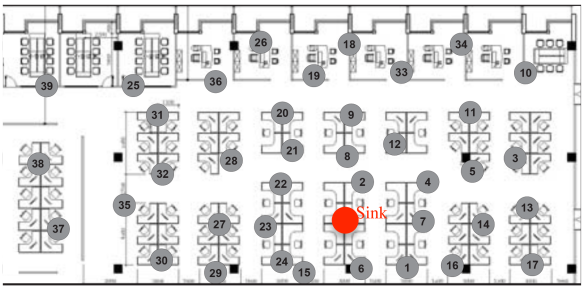


Fig. 7. Indoor testbed with 40 Telosb nodes deployed on our $40\times70m^2$ office. The red node denotes sink node.

the indoor testbed to test the extra overhead introduced by COF: the first one runs the original version of ORW by setting the nodes' IPI to 4 minutes; and the second one runs COF also setting IPI to 4 minutes, but COF always returns a *no recommendation* to disable CT. We compute the average single-hop delay and average radio duty cycle of all nodes in Figure 6(c). As shown in the figure, the extra overhead of COF brings 0.9% extra delay and 0.75% extra energy consumption. Considering the benefit of exploiting concurrency opportunity, we are confident that the performance of COF is superior to that of ORW, as demonstrated by the following evaluation results.

6 EVALUATION RESULTS

In this section, we conduct extensive experiments to test the performance of COF. We first introduce the experimental testbed and performance indicators. Then, we conduct experiments to assess the performance of concurrency control and sender monitoring, and further conduct specific experiments to quantify the expected benefit of CT in the presence of an *opportunistic exposed terminal*. After that, we respectively evaluate the performance of COF from different aspects and compare it with the state-of-the-art protocols.

6.1 Experimental Testbed and Performance Indicator

Our experiments are conducted in indoor testbeds with 40 Telosb nodes which are deployed on our $40\times70m^2$ office as shown in Figure 7. By setting different transmission power levels (RF output power) to testbed networks, nodes automatically form multi-hop networks with different densities. All experiments are conducted in the 19th Zigbee wireless channel which is overlapped with part of WiFi operating frequency used by the office APs. All senders transmit 80-byte data packets in the experiments. The wake-up interval is set to 512ms. When a node wakes up, it has to perform a Clear Channel Assessment (CCA) to assess channel condition. This period is a constant about 11ms. During this period, if the sensor node detects a busy channel condition, it extends its radio-on period to receive potential incoming packets. The extended active period in TinyOS is defaulted to 30ms. Except for 512ms, we have also evaluated the effect of different wake-up intervals on performance. The experimental results show the same conclusion as 512ms and the performance of COF increases with the decrease of wakeup rates which was demonstrated by Ghadimi et al. (2014). All network nodes work in duty-cycled mode except sink node. When a node wakes up, if the shared channel is clear and the node has no data packet to transmit, it will keep in active state for 11ms, and then it turns off the radio and returns to the sleeping state. However, if it has a data packet to transmit, it will stay in the listening state to occupy the shared channel. After the completion of data transmission, it returns to the sleeping state by turning off radio. Note that external interference could cause node's active state to be extended even if the disturbed node has no data packet to transmit.

In the following sections, we use packet delivery ratio as the indicator of network reliability. The energy consumption is measured by duty cycle, the portion of radio-on time, as a platform-independent metric for energy efficiency. This metric is a good proxy for power, because typical sensor platforms have their power profile dominated by the radio chip and transmitting and listening operations commonly have a similar current draw. Besides, we use single-hop delay to approximate delivery latency, and use the single-hop transmission count to indicate data collision caused by concurrent transmission, because aggressively exploiting concurrency opportunities could bring about more serious network interference. The delivery latency is defined as the time duration from the time when a packet is put into the sender's transmission buffer to the time when the sender receives an ACK.

6.2 Concurrency Control and Monitoring

In order to assess the performance of concurrency control and sender monitoring introduced in Section 4.3, we use COF to construct networks with different traffic load in the indoor testbed. Because the disorder CT mode could directly lead to higher data collision and appropriate concurrency decision can decrease the single-hop transmission time, we use average number of retries of each transmission (referred to as RTX) as the indicator of data collision and use one-hop delay (referred to as Delay) to indicate the transmission efficiency. For comparison, we also evaluate the performance of the state-of-the-art opportunistic forwarding protocol, ORW (Ghadimi et al. 2014), using the same network configuration. ORW is a traditional opportunistic forwarding protocol that can fully exploit candidate forwarders to forward data packets rather than exploit concurrency opportunities.

We test the performance of COF and ORW by setting four different traffic loads in testbed networks. The four traffic loads correspond to different IPIs: (i) low traffic load by setting IPI to 60 seconds; (ii) moderate traffic load by setting IPI to 30 seconds; (iii) high traffic load by setting IPI to 10 seconds; and (iv) bursty traffic in part of the network. For each network configuration, we repeat experiment for at least 5 times and each lasts for 2 hours:

Low Traffic Load (LTL). Each node generates a data packet every 1 minute. This allows us to evaluate RXT and Delay under low traffic load that has small numbers of concurrency opportunities. In this case, COF cannot give full play to the advantage of exploiting potential concurrency opportunities.

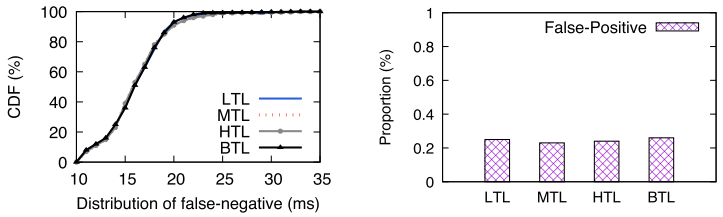
Moderate Traffic Load (MTL). The IPI of network nodes is set to 30 seconds. Under this configuration, COF can fully exploit concurrency opportunities, while ORW uses the shared channel to transmit data packet in exclusive mode. In theory, COF can significantly decrease Delay compared with ORW.

High Traffic Load (HTL). This configuration lets network nodes produce data packets at intervals of 10 seconds. In this condition, COF activates concurrency control mechanism to avoid data collision caused by disordered CT mode. If the concurrency control mechanism works well, the RXT will not be appreciably increased compared with the configuration of moderate traffic load, while the Delay could rise.

Bursty Traffic Load. In this configuration, we select 10 neighboring senders in the testbed network to continuously produce data packets lasting for 1 minute. Once a data packet has been successfully delivered, a new data packet will be produced after 20ms. This generates a sudden surge of local traffic, which accords well with actual network traffic characteristics. In this case, we only evaluate the RXT and Delay of selected neighboring senders.

Table 3. Performance of Concurrency Control in Testbed Networks with Different Traffic Load

Protocols	Low Traffic Load		Moderate Traffic Load		High Traffic Load		Bursty Traffic Load	
	RXT	Delay (ms)	RXT	Delay (ms)	RXT	Delay (ms)	RXT	Delay (ms)
ORW	0.07	97	0.15	125	0.21	152	0.21	245
COF	0.08	93	0.18	97	0.26	101	0.24	125

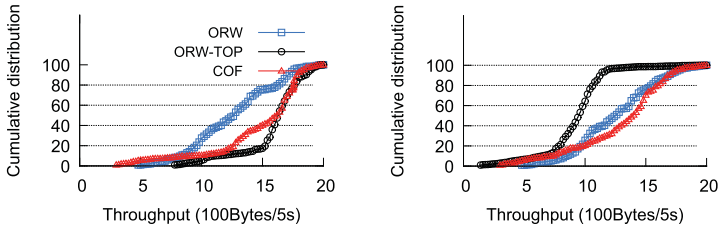


(a) False-negative distribution (b) False-positive proportion

Fig. 8. Distribution of (a) false-negative and (b) false-positive proportion.

6.2.1 Effectiveness of Concurrency Control. Throughout the study, we compare the average retries of each transmission and single-hop delay between COF and ORW. In our experiment, each network node records the retransmission count and transmission delay of all data packets, and reports back to central computer through wire cables. According to the information feedback from the network nodes, we compute the average retries of each transmission (RXT) and single-hop transmission time (Delay). Table 3 summarizes the results of the experiments with different configurations. The results show that, compared with ORW, the average retries of each data transmission in COF networks has not significantly increased with increasing traffic load. Under the high traffic load configuration, the RXT increases from ORW's 0.21 to COF's 0.26. Because ORW exploits candidate forwarders to forward data packets rather than exploits concurrency opportunities, the retransmissions of COF are mainly caused by the lossy link, the failure of the collision avoidance mechanism, and a hidden terminal problem. All of them may bring about data loss. Note that in distributed IoT networks, this proportion of retransmissions is inevitable. On this basis, by further exploiting concurrency opportunities, due to the traffic surging in COF network, the failure of the collision avoidance mechanism and the hidden terminal problem may bring about a more serious data collision. Even so, the increased RXT is very limited compared with ORW. The results show that the concurrency control mechanism of COF works well. The avoidance of the disorder CT mode guarantees the benefit of harnessing concurrency opportunities. Furthermore, with the exploitation of concurrency opportunities, COF significantly reduces the single-hop transmission delay in networks with a different traffic load compared with ORW. This conclusion also applies to networks with a bursty traffic load.

6.2.2 Accuracy of Sender Monitoring. In above-mentioned experiments, to test the accuracy of sender monitoring, all network nodes are synchronized. Besides, for each data transmission, the sender records the starting and ending time t_{ending}. If in the CT mode, the sender also records the corresponding concurrent sender and the measured finish time t_{finish}. All network nodes report this information back to the central computer. By analyzing the difference between t_{ending} and t_{finish} in the CT mode, we classify all recorded t_{finish}s into two sets: false-negative and false-positive. The false-negative indicates that the actual ending time t_{ending} of the concurrent sender is earlier than the measured finish time t_{finish} by COF. On the contrary, false-positive indicates that the



(a) Opportunistic exposed terminals (b) Within range

Fig. 9. COF exploits concurrency opportunity to achieve a 1.64× gain shown in (a) and COF defers some harmful CT to improve performance shown in (b).

measured finish time is earlier than actual ending time. False-negative is normal, for only when the concurrent sender's transmission signal cannot be detected for two consecutive inter-frame intervals, can COF think the concurrent sender's transmission has finished. The emphasis here is that too large time interval between t_{ending} and t_{finish} in false-negative could diminish neighboring senders' opportunity to join in CT mode. False-positive denotes a wrong measurement result which could solicit more neighboring senders to transmit concurrently and result in a data collision.

We plot the distribution of false-negative and the proportion of false-positive on testbed networks with different configurations in Figure 8(a) and 8(b), respectively. As shown, almost 90% of the false-negatives are less than 20ms and the distributions of false-negative in testbed networks with different configurations are relatively similar. Moreover, the proportion of false-positive keeps steady at 0.2% in different configurations. This is mainly caused by the overlapping of frame transmissions, where one sender's frame transmission completely overlaps the other neighboring sender's frame transmission so that one of them cannot detect the other one in consecutive two inter-frame intervals. The experimental results demonstrate the accuracy and validity of sender monitoring. COF can detect the neighboring sender's end of transmission within the average of 18ms and the extremely low false-positive guarantees the high accuracy of sender monitoring.

From the analysis results, we can observe that the false-negative and the false-positive are really not changed with traffic loads. It benefits from the concurrency control manner of COF, which allows only two neighboring senders to transmit in concurrent way. Hence, no matter low traffic load or high/bursty traffic load, a sender's data transmissions usually can be overlapped with the data transmission of another neighboring sender. In the calculation of false-negative and false-positive, the data transmissions that do not set a concurrency flag are filtered out. We just use the data packets transmitted in concurrent way to calculate both false-negative and false-positive. The concurrency control mode makes them remain steady.

The effective concurrency control mechanism and accurate sender monitoring strategy will eventually enhance the measurement of the conditional link quality and the modeling of the benefit of CT mode. Both of them play decisive roles in exploiting concurrency opportunities in duty-cycled opportunistic forwarding.

6.3 Exploitation of Concurrency Opportunity

In this section, we conduct experiments to quantify the expected gain of the number of delivered packets (referred to as throughput for simplicity) in part of the network by using COF in presence of *opportunistic exposed terminal*. This experiment involves with all neighboring sender pairs that satisfy the condition "permission of CT" in the indoor testbed. Each neighboring sender pair has similar characteristics: (1) two neighboring senders are within the interference range of each other and (2) each sender has a set of candidate forwarders and at least one of them is out of the two

senders' interference region. These neighboring sender pairs are determined by COF according to the estimated conditional link quality and the expected benefit of CT recorded in the **BTable**. Testbed nodes are all controlled by a central computer through wire cables. By collecting all network nodes' **BTable**s, the central computer periodically selects a neighboring pair and orders them to produce a large number of data packets at intervals of 512ms. The other network nodes without being selected produce data packets at intervals of 1 minute.

We respectively use ORW, COF, and a variation of ORW without using of carrier sense and without requesting ACK (referred to as ORW-TOP, because it can achieve almost theoretically optimal performance) to construct networks and deliver data packets. For each pair of neighboring senders, we order them to do the same operation in the three networks by respectively using the three protocols. Note that ORW, ORW-TOP, and COF adopt the same routing metric EDC (Ghadimi et al. 2012) to construct network topology. Although it's not exactly the same, we think the overall network structures of them are similar. To minimize errors due to network structure difference, we repeat each experiment no less than 10 times. In addition, we should also note that ORW-TOP allows a pair of neighboring senders to transmit concurrently without requesting the forwarders' acknowledgments. Hence, it can maximally exploit the spatial diversity of forwarders without considering the influence of ACK collision at transmitting end. Overall, in theory, ORW-TOP has the optimal performance in presence of *opportunistic exposed terminal*.

We select more than 90 neighboring sender pairs from the testbed meeting the above-mentioned conditions. By selecting the same neighboring sender pair under networks respectively using ORW, ORW-TOP, and COF, we quantify the corresponding throughputs by counting the total packets received by their candidate forwarders during a period of 5-second windows. We have eliminated all duplicate packets before computing throughput. The theoretical maximum throughput in this set of experiments is 20 packets per 5-second window unit. Each evaluation is run for 10 minutes and repeated no less than 10 times. We plot the cumulative distribution of actual throughputs in Figure 9(a). As shown, the throughput of ORW is far less than that of ORW-TOP, because, in this case, ORW suppresses the opportunities of CT, while ORW-TOP can fully exploit concurrency opportunities without requesting forwarders' acknowledgments. This No-ACK mechanism can maximally exploit the spatial diversity of forwarders without considering the influence of the ACK collision at the transmitting end. The candidate forwarders, which are located outside the interference range of the neighboring senders, can successfully receive data packets transmitted concurrently. Although ORW-TOP can theoretically attain the largest throughput, we must note that it is not suitable for practical applications due to insecure data delivery performance. Compared with ORW, COF improves the expected throughput by 64%. The reason is that COF can quickly confirm the feasibility of exploiting the concurrency opportunity by considering the packet receipt rate at both the forwarder's ending and transmitting end.

6.4 Avoidance of the Overshooting Concurrency

Aside from the *opportunistic exposed terminal*, we also evaluate the performance of COF on special topologies that a pair of neighboring senders and all their candidate forwarders are within the carrier sense range of each other (referred to as *within range*). We deliberately choose 50 applicative topologies from the testbed. Except for the topology, the experiment setup is the same as the above-mentioned experiments. For each topology, we evaluate the performance of ORW, ORW-TOP, and COF, respectively, by lasting for 10 minutes and repeating 10 times.

Figure 9(b) presents the distribution of throughput. Since severe interference can bring about packet collisions at all candidate forwarders, ORW-TOP attains very low throughput, with the average of 9.4 packets per 5-second window, far less than the attained throughput under the *opportunistic exposed terminal*, whereas ORW attains almost the same throughput as that in the case

Table 4. Concurrency and Potential Forwarders Utilized by COF and the State-of-the-Arts

Protocols	Concurrent forwarding	Opportunistic forwarding
ORW (Ghadimi et al. 2014)	×	√
BoX-MAC (Moss and Levis 2008)	×	×
COF	√	√
CMAP (Vutukuru et al. 2008)	√	×
NoPSM (Chen et al. 2017)	√	×

of the *opportunistic exposed terminal*, because both the cases of *opportunistic exposed terminal* and within range suppress the opportunities for CT. Compared with ORW, COF improves the mean throughput by about 10.3% because COF can utilize potential concurrencies to the maximum. In this experiment, more than 20% of the topologies support CT even though neighboring senders and all candidate forwarders are within the same interference range.

6.5 Data Forwarding Performance

In this section, we conduct experiments to evaluate the data forwarding performance of COF in terms of delivery ratio, network retransmission count, latency, and energy efficiency in testbed networks with a different traffic load and a different network density. Besides, we also compare COF with ORW (Ghadimi et al. 2014), BoX-MAC-2 (Moss and Levis 2008) (referred to as BoX-MAC), CMAP (Vutukuru et al. 2008), and NoPSM (Chen et al. 2017) with the same network configuration.

BoX-MAC is an enhanced CSMA-based protocol for duty-cycled networks, which means nodes in the interference range of current transmitters defer to send packets. Hence, performance of BoX-MAC is regarded as a baseline to see how much improvement can be taken from concurrent transmission or/and opportunistic forwarding. As previously mentioned, ORW (Ghadimi et al. 2014) is a traditional opportunistic forwarding protocol by exploiting candidate forwarders to forward data packets. The difference between ORW and COF is that ORW is built upon BoX-MAC and it ignores all potential concurrency opportunities. CMAP (Vutukuru et al. 2008) was designed for wireless *ad hoc* networks to improve channel usage through addressing the exposed terminal problem. It makes decision for concurrent transmission according to reactively constructed interference relationships. Except for concurrency opportunities, the difference between CMAP and COF is that CMAP makes a binary transmission decision based on the defer patterns inferred from the interference list, while COF makes the positive decision of transmission concurrency only when it is beneficial for throughput gain of all active nodes in local network. Based on CMAP, NoPSM (Chen et al. 2017) further uses time synchronization to improve the estimation of packet overlapping so that the interference relationships can be measured more accurately. Because COF is first proposed for exploiting concurrency opportunities in opportunistic forwarding, ORW, CMAP, and NoPSM are the most related work with ours. The properties of the related protocols and COF are listed in Table 4. A tick denotes a characteristic that a specified protocol has, and a cross denotes a characteristic that it doesn't have.

6.5.1 Diverse Network Load. To evaluate the effects of network load on performance, we construct testbed networks with COF and the four other related protocols by setting different traffic patterns. For each protocol, we respectively set all network nodes' IPI to 16 seconds (s), 32s, 64s, 128s, and 256s. Besides, we also construct bursty traffic load by selecting 10 neighboring senders in the testbed network to continuously produce data packets lasting for 5 minutes. In this case, the IPI of selected 10 nodes is respectively set to 1s, 2s, 4s, and 8s. When a new data packet is generated and the previous data packet hasn't been successfully delivered, the new packet is put

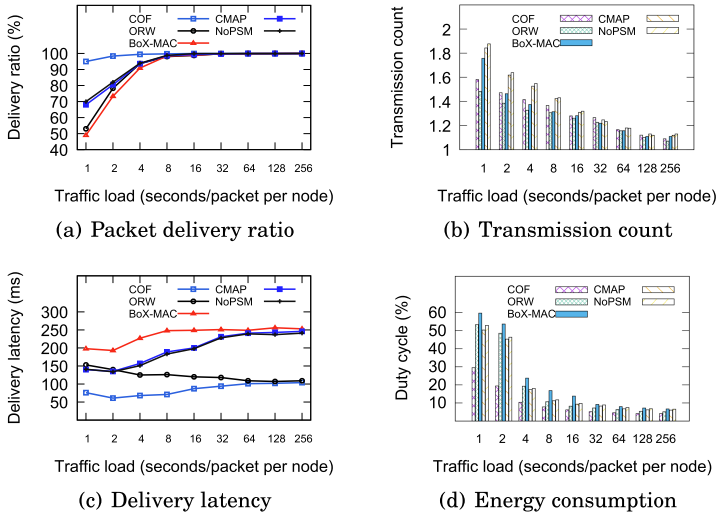


Fig. 10. Data forwarding performance of ORW, BoX-MAC, CMAP, NoPSM, and ORW in tested networks with different traffic loads. Each node has about six neighbors on the average.

into the sending buffer. In this experiment, the nodes' wake-up interval is set to 512ms and the RF output power is set to level 5 in CC2420 (Texas Instruments 2006). Then, the average number of neighbors of each network node is about 6. For each traffic load, we respectively run experiments with the five protocols, and compute the average of 10 times of running as the result.

Delivery Ratio. Figure 10(a) compares the delivery ratio of the five protocols with different traffic loads. Note that the values 1, 2, 4, and 8 on horizontal axis denotes the testbed networks with bursty traffic pattern. We can see that in all scenarios with bursty traffic load, the delivery ratio of COF significantly outperforms BoX-MAC, CMAP, NoPSM, and ORW. By increasing the traffic load (the IPI of selected neighboring nodes decreases from 8s to 1s), the delivery ratio of BoX-MAC decreases from 98% to 49%, CMAP from 98.6% to 68%, NoPSM from 98.8% to 70%, and ORW from 98.08% to 43%, respectively. In contrast, the delivery ratio of COF has always been kept over 95%. The outperformed delivery ratio of COF is mainly attributed to the utilization of both concurrency opportunities and forwarder opportunities. The exploiting of concurrency opportunities can reduce the blind waiting time when neighboring nodes are transmitting, and the exploiting of potential forwarders can further reduce the delivery latency in asynchronous duty-cycled networks because the wake-up phases of potential forwarders are randomly distributed in a sleep-wake cycle. In addition, the performance improvement between CMAP/NoPSM and BoX-MAC also demonstrates the benefit of exploiting concurrency opportunities, and the delivery ratio gap between COF and CMAP/NoPSM indicates the benefit of exploiting potential forwarders on data delivery performance. However, with the increasing of traffic load, the delivery ratio of ORW is as low as BoX-MAC even though ORW has exploited potential forwarder to deliver data packets. This is because the duplicate transmissions consume a great portion of channel resource under high traffic load. The significant decreasing of delivery ratio of both ORW and BoX-MAC in high traffic load is due to the inefficient channel utilization. Compared with ORW, the advantage of using concurrency opportunities in COF makes up for the disadvantage of exploiting potential forwarders, because it can further improve channel spatial reuse and luckily reduce the probability that multiple forwarders simultaneously receive the same packet.

Delivery Latency and Transmission Count. By exploiting concurrency opportunities, although delivery ratio performance can be improved, it brings about more serious intra-network interference and directly results in more data collisions. Figure 10(b) compares the average one-hop transmission count of each packet. We can see that in all scenarios with different traffic loads, COF has a transmission count that is 6.5–11 percent higher than ORW, and CMAP/NoPSM has a transmission count that is 4.7–13.5 percent higher than BoX-MAC. Compared with CMAP/NoPSM, COF having smaller transmission count is mainly achieved from the opportunistic forwarding mechanism, because the multiple forwarding choices with temporal and spatial diversity increase the chance to tolerate collision in duty-cycled opportunistic forwarding. In spite of more one-hop transmission counts, the utilization of concurrency opportunities can significantly reduce data delivery latency in the scenarios of both exposed terminals and opportunistic exposed terminals. Figure 10(c) shows the average delivery latency of the five protocols under different traffic loads. From this figure, we can see that BoX-MAC, CMAP, and NoPSM have almost the same latency when the traffic load is low (IPI larger than 64s), and they all have decreasing latency as the traffic load become high because it may increase the probability that a sender's receiver could be awake during the sender's transmission phase. Compared with ORW, COF also significantly reduces the delivery latency with the increasing of traffic load. Unlike BoX-MAC, CMAP, and NoPSM, with the increasing of the traffic load, the delivery latency of ORW is slightly increased because of the explosive increase of duplicate transmissions and the lack of effective duplication suppression mechanism (Liu et al. 2016) in ORW. COF utilizes concurrency opportunities to improve spatial reuse, and then the increased intra-network interference decreases the probability that multiple forwarders receive the same data packet simultaneously. The advantage of exploiting concurrency opportunities makes up for the deficiency of exploiting multiple potential forwarders under high traffic load.

Energy Consumption. Figure 10(d) gives the duty cycle of BoX-MAC, CMAP, NoPSM, ORW, and COF with different traffic loads. We can see that COF significantly outperforms the other four protocols in energy efficiency when an unexpected surge of traffic occurs (i.e., IPI of selected nodes is less than 16s). This mainly results from fully used concurrency opportunities and potential forwarders. Both of them can significantly reduce data delivery latency and then reduce the radio on time. COF also outperforms the other four protocols in energy efficiency with the decrease of traffic load (the IPI of network nodes increases from 16s to 256s). This is mainly attributed to the utilization of potential forwarders for data forwarding which shortens the frame transmission time as shown in Figure 2. In Figure 10(d), ORW ranks behind only COF in energy efficiency when IPI is no less than 16s. By using concurrency opportunities, CMAP and NoPSM outperform BoX-MAC in the scenarios with bursty traffic load, and the duty cycle is close to BoX-MAC with the decrease of traffic load, because of the low utilization of concurrency opportunities in testbed network with traffic load.

In short, COF has an advantage over the state of the art in networks with different traffic load. With increase of traffic load, COF significantly outperforms all of them by utilizing both concurrency opportunities and potential forwarders.

6.5.2 Diverse Network Densities. To compare performance of COF with BoX-MAC, CMAP, NoPSM, and ORW in testbed networks with different density, we set different RF output powers to nodes in testbed networks. In our experiments, the RF output power level (Texas Instruments 2006) is respectively set to level 3 (-25dBm), level 5, level 7 (-15dBm), and level 9. For each RF output power, the traffic pattern is generated by setting all network nodes' IPI to 16s. We test the performance of the five protocols and compute the average 10 times of running as the result.

Delivery Ratio. Figure 11(a) compares the delivery ratio of the five protocols with different network densities. Note that the horizontal axis shows the average number of neighbors by setting

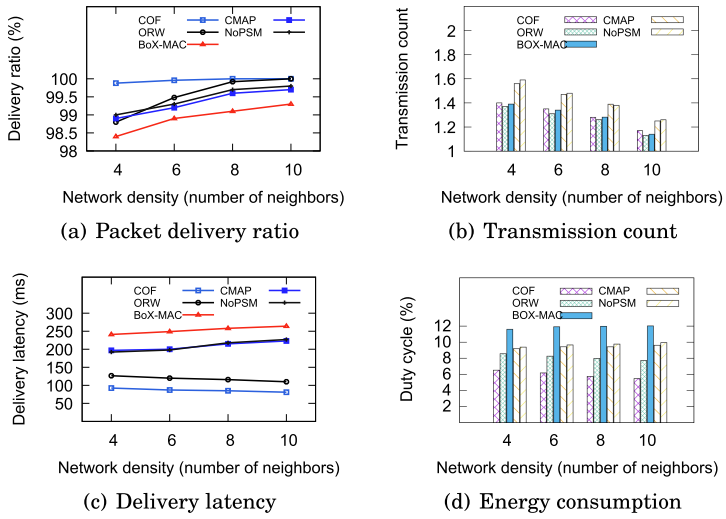


Fig. 11. Data forwarding performance of COF, BoX-MAC, CMAP, and ORW in testbed networks with different network densities. All network nodes' IPIs are set to 16 seconds. The horizontal axis shows the average number of neighbors by setting different RF output powers.

different RF output powers. It is referred to as the network density factor. From this figure, we can see that denser deployment of nodes can improve delivery ratio for all the evaluated MAC protocols, because the impact of the hidden terminal on data delivery becomes weaker with the increase of network density. Especially for COF, on any particular network density, COF always outperforms ORW, BoX-MAC, CMAP, and NoPSM. The outperformed delivery ratio of COF is mainly attributed to the utilization of both concurrency opportunities and potential forwarders. In addition, the spatial diversity of multiple forwarders in opportunistic forwarding has mitigated the influence of hidden terminal problem. In low-density testbed networks (i.e., average number of neighbors is 4), the delivery ratio of ORW is higher than BoX-MAC, but slightly lower than CMAP and NoPSM. This is because the networks have relatively more concurrency opportunities compared with potential forwarders opportunities by setting low density and high traffic load (IPI is 16s). Even so, anycast forwarding allows opportunistic forwarding protocol to transmit a packet faster than traditional unicast forwarding (see Figure 11(c)). Overall, opportunistic forwarding protocols work best at high network densities, as this gives the most choices for forwarding. As a result, ORW shows the better delivery ratio for dense topologies, i.e., in testbed networks with network density factor 6, 8, and 10.

Delivery Latency and Transmission Count. As mentioned above, the impact of the hidden terminal on data delivery becomes weaker with increase of network density. This can be demonstrated by Figure 11(b). The average transmission count of the five protocols decreases with the increase of network density. Moreover, network density directly affects delivery latency, because more neighboring nodes share the same channel resource with the increase of network density. Figure 11(c) compares the delivery latency of COF, ORW, BoX-MAC, CMAP, and NoPSM with different network density. We can see that, compared with BoX-MAC, by either exploiting concurrency opportunities or exploiting potential forwarders to transmit data packets, delivery latency can be significantly reduced. Compared with the other four protocols, COF has the smallest delivery latency. For BoX-MAC, CMAP, and NoPSM, the delivery latency declines with the increase of network density, because more neighboring nodes share the same channel to deliver data. However, opportunistic

forwarding protocols work better at high network densities, as this gives the most choices for forwarding. As a result, ORW and COF show slightly reduced delivery latency with the increase of network density.

Energy Consumption. Figure 11(d) shows the average duty cycle of these MAC protocols with different network densities. We can see that COF can outperform all the other four protocols in energy efficiency whether in low or high network density. In short, COF outperforms the state of the art in all aspects of network performance with diverse network density.

7 RELATED WORKS

In this section, we survey the related work of low-power opportunistic forwarding schemes and the techniques used for concurrent transmission.

7.1 Low-Power Opportunistic Forwarding

GeRaF (Zorzi and Rao 2003) pioneered the concept of anycast routing in duty-cycled wireless sensor networks (WSNs). It utilizes geographic routing to determine routing progress of its neighboring nodes and a busy tone protocol to ensure a unique forwarder. The structure of opportunistic routing concept was first shaped in ExOR (Biswas and Morris 2005) in 2005. The focus of this protocol is to improve the performance of traditional routing schemes in wireless networks by exploiting multiple transmission opportunities created by the broadcast nature of the wireless medium. CMAC (Sha Liu et al. 2009) combines the concepts of GeRaF and ExOR by using prioritized forwarders and slotted acknowledgments, and overhearing of acknowledgments to determine a unique forwarder as in ExOR. However, relying solely on geographic routing, CMAC does not address the key challenges for opportunistic routing in duty-cycled networks such as anycast routing metrics and wireless link dynamics.

On that basis, Ghadimi et al. proposed ORW (Ghadimi et al. 2014) in 2014 to achieve opportunistic forwarding for duty-cycled WSNs. ORW uses Expected Duty Cycle (EDC) (Ghadimi et al. 2012) as the routing metric, which represents the number of MAC wakeup periods required to reach the sink and is the equivalent of the ETX metric in asynchronous and duty-cycled networks. ORW showed that opportunistic routing is also beneficial in duty-cycled data-collection networks. Instead of waiting for a specific neighbor to wake up, nodes anycast the packet until any valid forwarder receives and acknowledges it. This increases robustness and shortens the wakeup phase of low-power listening. By introducing the opportunistic routing metric EDC and the forwarding mechanism of ORW, Duquennoy et al. further proposed a scalable opportunistic routing protocol ORPL (Duquennoy et al. 2013) to support any-to-any data forwarding. Although opportunistic forwarding significantly improves data forwarding performance compared with traditional routing schemes, the explosive increase of duplicate transmissions come with an unexpected surge of traffic in duty-cycled WSNs. Liu et al. proposed DOF (Liu et al. 2016) to suppress duplicate by quickly distinguishing potential forwarders before data transmission, and then considering link quality to arrange forwarding schedule.

The application of opportunistic routing in duty-cycled networks also received great attention from a more theoretical perspective (Dubois-Ferrière et al. 2011; Kim et al. 2009; Unterschütz et al. 2012). While they omit the harnessing of potential concurrency opportunities for opportunistic forwarding in duty-cycled networks that this article addresses, their results strongly motivated our work.

7.2 Concurrency for Data Forwarding

Concurrent transmission is a well-known concept employed in wireless communications to enhance channel utilization by improving spatial reuse. It is crucial to performance of data

forwarding protocols, concerning both traditional unicast protocols or opportunistic protocols. Researchers have proposed a lot of methods and mechanisms to exploit concurrency in wireless networks. Here we have a brief discussion on the existing methods in three aspects, respectively.

Collision Tolerance. In the wireless communication community, capture effect has been a well-known phenomenon (Roberts 1975; Whitehouse et al. 2005) for collision tolerance and various capture models have been proposed. Unlike collision avoidance, the idea of capture effect allows collisions. Flash flooding (Lu and Whitehouse 2009), Chorus (Zhang and Shin 2010), Glossy (Ferrari et al. 2011), Splash (Doddavenkatappa et al. 2013), Chaos (Landsiedel et al. 2013), LWB (Ferrari et al. 2012), and Pando (Du et al. 2015) were subsequently proposed for efficient data transmission exploiting capture effect in flooding scenarios. They detect and recover packets from collisions taking advantage of capture effect, whereby a packet with the stronger signal strength can be received in spite of a collision. However, the common limitation of these techniques trying to use collision tolerance is that they can be only applied in flooding or broadcasting scenarios, where transmitted packets must carry the same data. This requirement greatly limits their application scope (Tan et al. 2010), especially in large-scaled data collection networks. Moreover, the techniques using capture effect are heavily dependent on highly precise time synchronization. In resource-restricted duty-cycled networks, considering network dynamics, it is not cost effective to achieve this level time synchronization by paying considerable energy consumption.

Some other related works, such as partial packet recovery (Jamieson and Balakrishnan 2007) and interference cancellations (Gollakota and Katabi 2008), have also been proposed to achieve concurrent transmission. However, these techniques are heavily dependent on highly precise time synchronization (microsecond level) and extensive computation for real-time processing enormous amount of matrix computation. In resource-restricted duty-cycled networks, considering network dynamics, it is difficult to fully meet these conditions. Hence, these techniques are difficult to be directly applicable to duty-cycled networks.

PRR-SINR Model. A physical interference model is another effective way for improving channel utilization in wireless networks. Son et al. (2006) and Sha et al. (2009) studied the PRR-SINR model in sensor networks and showed the modeling accuracies and impacts on link scheduling performance. In particular, it is shown that adopting the PRR-SINR model can lead to significant link throughput improvement. Reis et al. (2006) presented interference and packet delivery models that can be instantiated by packet transmission traces. Qiu et al. (2007) proposed a general interference model to characterize the interference among arbitrary number of 802.11 senders and predict the resultant throughput. In (Kashyap et al. 2007), a measurement-based approach is proposed to model the interference and link capacity in 802.11 networks. Aguayo et al. (2004) experimentally studied the effect of SINR on the causes of packet loss in a 802.11 mesh network (Roofnet).

In the domain of duty-cycled sensor networks, a part of concurrent MAC protocols, such as C-MAC (Sha et al. 2009) and PIM (Liu et al. 2010), are based on proactively constructed PRR-SINR model. These concurrent MACs were designed to exploit transmission concurrency to improve throughput of data intensive WSNs, but they require a network downtime periodically to make RSSI measurement at whole network scale and construct SINR-PRR model for each node in advance, which incur high overheads.

Conflict Relationship. The conflict graph has been used to model wireless interference between neighboring nodes. The conflict graph provides a simplified description of the interference status, which greatly eases the design of channel assignment/spectrum allocation algorithms, and consequently gives birth to a series of highly efficient wireless network optimization algorithms (Joo et al. 2016; Cheng et al. 2012; Subramanian et al. 2008).

Existing works can be divided into two categories based on the type of conflict graphs they use. The first category uses per-link signal measurements to capture interference conditions among individual links, using either active measurements (Sha Liu et al. 2009, 2009b; Vutukuru et al. 2008) or passive measurements (Shrivastava et al. 2011; Vutukuru et al. 2008; Chen et al. 2017). These link-based conflict graphs are for immobile networks where transmission links are known *a priori*. The second category of works builds coverage-based conflict graphs based on propagation models (Gupta and Kumar 2000; Rhee et al. 2006; Zhou et al. 2013). Zhou et al. (2013) used real-world measurements to evaluate the conflict graph accuracy of coverage-based conflict graph.

In wireless sensor networks, CMAP (Vutukuru et al. 2008) was designed to improve channel usage through addressing the exposed terminal problem. It exploits potential opportunity for the exposed nodes to transmit data concurrently. CMAP makes decision of transmission concurrency with reactively constructed interference relationships rather than the proactively constructed SINR-PRR model. Based on CMAP, NoPSM (Chen et al. 2017) further uses time synchronization to improve the estimation of packet overlapping so that the interference relationships can be measured more accurately. Although the link-based conflict graphs are the most related work to COF, however, existing work doesn't apply to opportunistic forwarding scheme. Our work is the first exploiting concurrency for low-power opportunistic forwarding by using per-link conflict relationship.

8 DISCUSSIONS

8.1 Network Dynamics

The main limitation of COF is that our link model cannot be timely updated when the network links are highly dynamic. With inaccurate link estimation, the efficiency of the concurrent transmission decision may be degraded. However, the opportunistic forwarder selection and moving average link quality update can improve fault-tolerance capability and guarantee the routing efficiency to some degree. In opportunistic forwarding, the multiple forwarding choices with temporal and spatial diversity increase the chance to tolerate collision in opportunistic forwarding, because interference from a specific neighboring sender is likely to have different influence on these candidate forwarders. After failure of some links, COF still has great chances to deliver its data packet to one of its candidate forwarders. Furthermore, COF adopts moving average to update each link's overall *cpdr*, even suffering from serious network dynamics, with time COF can measure the accurate conditional link quality and make the concurrent transmission decision more accurate.

8.2 Overhead and Energy Consumption

The computation and update of COF depend on the receiving of forwarder's feedback of forwarder bitmaps. The higher the network traffic, the more the feedback of receiving statuses from candidate forwarders. Hence, the cost of computation and update increases with the rising network traffic. But overall, the time overhead and energy cost is similar to the cost of the built-in link estimator, because their working processes are almost the same. The most frequent operations of COF are the setting or resetting of some bits in sender bitmaps. But the computation overhead and energy consumption is negligible.

Besides, the communication overhead of COF is also lightweight. COF exploits both the network probe and data packet footer to share forwarder bitmaps. COF only broadcasts a probe carrying the forwarder bitmaps and recorded information in the **BTable** table with a long time interval. In addition, COF also fully utilizes the free space of system network probes and small data packets to share the most frequently updated bitmaps.

8.3 Limitation of Experimental Scenarios

Due to the limited experimental environment and resource constraints, we only conducted indoor experiments to quantitatively and qualitatively evaluate the performance of COF under different experimental scenarios. We also think that the indoor testbed networks cannot fully represent complex and diverse outdoor applications. Even so, the evaluation results can demonstrate the feasibility and effectiveness of exploiting concurrency for low-power opportunistic forwarding by COF to a great degree. In this article, we have actually tried our best to increase network diversity by constructing multi-hop networks with different system setting, topologies, and external interference.

9 CONCLUSION

In this article, we propose COF to exploit potential concurrency opportunities for low-power opportunistic forwarding. COF achieves concurrent transmission by measuring conditional link quality under the interference of on-going transmissions in distributed way, and then modeling the benefit of potential concurrency opportunities. According to the expected benefit of concurrency, COF decides whether or not to transmit in concurrent way. COF also adopts concurrency flag and signal features to avoid data collision caused by disordered concurrent transmissions and enhance the accuracy of conditional link quality estimation. COF can be easily integrated into the conventional unsynchronized and duty-cycled protocols. We implement COF and evaluate its performance on an indoor testbed. The results show that COF outperforms the state of art protocols.

ACKNOWLEDGMENTS

We sincerely thank the editor and anonymous reviewers for insightful comments to improve our work.

REFERENCES

Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. 2004. Link-level measurements from an 802.11B mesh network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'04)*. ACM, New York, 121–132. DOI: https://doi.org/10.1145/1015467.1015482

Sanjit Biswas and Robert Morris. 2004. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'05)* ACM, New York, 133–144. DOI: https://10.1145/1080091.1080108

Haiming Chen, Zhaoliang Zhang, Li Cui, and Changcheng Huang. 2017. NoPSM: A concurrent MAC protocol over low-data-rate low-power wireless channel without PRR-SINR model. *IEEE Transactions on Mobile Computing* 16, 2 (March 2017), 435–452. DOI: https://doi.org/10.1109/TMC.2016.2547867

Yu Cheng, Hongkun Li, and Peng-Jun Wan. 2012. A theoretical framework for optimal cooperative networking in multiradio multichannel wireless networks. *IEEE Wireless Communications* 19, 2 (April 2012), 66–73. DOI: https://doi.org/10.1109/MWC.2012.6189415

Manjunath Doddavenkatappa, Mun-Choon Chan, and Ben Leong. 2013. Splash: Fast data dissemination with constructive interference in wireless sensor networks. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI'13)*. USENIX Association, Berkeley, CA, 269–282.

Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li. 2015. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*. ACM, New York, 365–378. DOI: https://doi.org/10.1145/2809695.2809721

Henri Dubois-Ferrière, Matthias Grossglauser, and Martin Vetterli. 2011. Valuable detours: Least-cost anypath routing. *IEEE/ACM Trans. Netw.* 19, 2 (April 2011), 333–346. DOI: https://doi.org/10.1109/TNET.2010.2070844

Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. 2013. Let the tree Bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys'13)*. ACM, New York, Article 2, 14 pages. DOI: https://doi.org/10.1145/2517351.2517369

Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. 2012. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys'12)*. ACM, New York, 1–14. DOI: https://doi.org/10.1145/2426656.2426658

Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. 2011. Efficient network flooding and time synchronization with Glossy. In *IPSN*, Xenofon D. Koutsoukos, Koen Langendoen, Gregory J. Pottie, and Vijay Raghunathan (Eds.). IEEE, 73–84. http://dblp.uni-trier.de/db/conf/ipsn/ipsn2011.html#FerrariZTS11.

E. Ghadimi, O. Landsiedel, and P. Soldati. 2012. A metric for opportunistic routing in duty cycled wireless sensor networks. In *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'12)*. 335–343. DOI: https://doi.org/10.1109/SECON.2012.6275795

Euhanna Ghadimi, Olaf Landsiedel, Pablo Soldati, Simon Duquennoy, and Mikael Johansson. 2014. Opportunistic routing in low duty-cycle wireless sensor networks. *ACM Trans. Sen. Netw.* 10, 4, Article 67 (June 2014), 39 pages. DOI: https://doi.org/10.1145/2533686

Shyamnath Gollakota and Dina Katabi. 2008. Zigzag decoding: Combating hidden terminals in wireless networks. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication (SIGCOMM'08)*. 159–170. DOI: https://doi.org/10.1145/1402958.1402977

Piyush Gupta and P. R. Kumar. 2000. The capacity of wireless networks. *IEEE Transactions on Mobile Computing* 46, 2 (March 2000), 388–404. DOI: https://doi.org/10.1109/18.825799

IEEE Computer Society. 2003. IEEE Standard 802.15.4. https://standards.ieee.org/findstds/standard/802.15.4-2015.html.

IEEE Computer Society. 2005. IEEE Standard 802.15.1. https://standards.ieee.org/findstds/standard/802.15.1-2005.html.

IEEE Computer Society. 2012. IEEE standard 802.11. https://standards.ieee.org/findstds/standard/802.11-2016.html.

Kyle Jamieson and Hari Balakrishnan. 2007. PPR: Partial packet recovery for wireless networks. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'07)*. 409–420. DOI: https://doi.org/10.1145/1282380.1282426

Xiaoyu Ji, Yuan He, Jiliang Wang, Kaishun Wu, Daibo Liu, Ke Yi, and Yunhao Liu. 2017. On improving wireless channel utilization: A collision tolerance-based approach. *IEEE Transactions on Mobile Computing* 16, 3 (March 2017), 787–800. DOI: https://doi.org/10.1109/TMC.2016.2567380

Changhee Joo, Xiaojun Lin, Jiho Ryu, and Ness B. Shroff. 2016. Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels. *IEEE/ACM Trans. Netw.* 24, 3 (June 2016), 1476–1488. DOI: https://doi.org/10.1109/TNET.2015.2417861

Anand Kashyap, Samrat Ganguly, and Samir R. Das. 2007. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'07)*. ACM, New York, NY, USA, 242–253. DOI: https://doi.org/10.1145/1287853.1287883

Joohwan Kim, Xiaojun Lin, and Ness Shroff. 2009. Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks. In *Proceedings of INFOCOM (INFOCOM'09)*. 612–620. DOI: https://doi.org/10.1109/INFCOM.2009.5061968

Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. 2013. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys'13)*. ACM, New York, Article 1, 14 pages. DOI: https://doi.org/10.1145/2517351.2517358

Bernard Le Floch, Michel Alard, and Claude Berrou. 1995. Coded orthogonal frequency division multiplex. *Proceedings of IEEE* 83, 6 (June 1995), 982–996. DOI: https://doi.org/10.1109/5.387096

Philip Levis and David Gay. 2009. *TinyOS Programming* (1st ed.). Cambridge University Press, New York.

Mo Li and Yunhao Liu. 2007. Underground structure monitoring with wireless sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*. ACM, New York, NY, USA, 69–78. DOI: https://doi.org/10.1145/1236360.1236370

Daibo Liu, Zhichao Cao, Yi Zhang, and Mengshu Hou. 2017. Achieving accurate and real-time link estimation for low power wireless sensor networks. *IEEE/ACM Trans. Netw.* 25, 4 (Aug. 2017), 2096–2109. DOI: https://doi.org/10.1109/TNET.2017.2682276

Daibo Liu, Mengshu Hou, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. 2016. Duplicate detectable opportunistic forwarding in duty-cycled wireless sensor networks. *IEEE/ACM Trans. Netw.* 24, 2 (April 2016), 662–673. DOI: https://doi.org/10.1109/TNET.2014.2387440

Daibo Liu, Xiaopei Wu, Zhichao Cao, Mingyan Liu, Yujun Li, and Mengshu Hou. 2015. CD-MAC: A contention detectable MAC for low duty-cycled wireless sensor networks. In *Proceedings of the 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'15)*. IEEE. DOI: https://doi.org/10.1109/SAHCN.2015.7338289

Sha Liu, Kai-Wei Fan, and Prasun Sinha. 2009. CMAC: An energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Trans. Sen. Netw.* 5, 4, Article 29 (Nov. 2009), 34 pages. DOI: https://doi.org/10.1145/1614379.1614381

Shucheng Liu, Guoliang Xing, Hongwei Zhang, Jianping Wang, Jun Huang, Mo Sha, and Liusheng Huang. 2010. Passive interference measurement in wireless sensor networks. In *Proceedings of the 18th IEEE International Conference on Network Protocols*. 52–61. DOI: https://doi.org/10.1109/ICNP.2010.5762754

Xi Liu, Anmol Sheth, Michael Kaminsky, Konstantina Papagiannaki, Srinivasan Seshan, and Peter Steenkiste. 2009. DIRC: Increasing indoor wireless capacity using directional antennas. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication (SIGCOMM'09)*. ACM, New York, 171–182. DOI: https://doi.org/10.1145/1592568.1592589

Jiakang Lu and Kamin Whitehouse. 2009. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *Proceedings IEEE INFOCOM*. 2491–2499.

David Moss and Philip Levis. 2008. *BoX-MACs: Exploiting Physical and Link Layer Boundaries in LowPower Networking*. Technical Report.

Joseph Polastre, Jason Hill, and David Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM, New York, 95–107. DOI: https://doi.org/10.1145/1031495.1031508

Lili Qiu, Yin Zhang, Feng Wang, Mi Kyung Han, and Ratul Mahajan. 2007. A general model of wireless interference. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'07)*. ACM, New York, 171–182. DOI: https://doi.org/10.1145/1287853.1287874

Charles Reis, Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. 2006. Measurement-based models of delivery and interference in static wireless networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'06)*. ACM, New York, 51–62. DOI: https://doi.org/10.1145/1159913.1159921

Injong Rhee, Ajit Warrier, Jeongki Min, and Lisong Xu. 2006. DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*. ACM, New York, 190–201. DOI: https://doi.org/10.1145/1132905.1132927

Lawrence Roberts. 1975. ALOHA packet system with and without slots and capture. *SIGCOMM Comput. Commun. Rev.* 5, 2 (April 1975), 28–42. DOI: https://doi.org/10.1145/1024916.1024920

Curt Schurgers. 2001. Systematic approach to peak-to-average power ratio in OFDM. In *Advanced Signal Processing Algorithms, Architectures, and Implementations XI*, F. T. Luk (Ed.), Vol. 4474. 454–464. DOI: https://doi.org/10.1117/12.448680

Mo Sha, Guoliang Xing, Gang Zhou, and S. Liu. 2009. C-MAC: Model-driven concurrent medium access control for wireless sensor networks. In *Proceedings of the IEEE INFOCOM*. DOI: https://doi.org/10.1109/INFCOM.2009.5062105

Vivek Shrivastava, Shravan Rayanchu, Suman S., and Konstantina Papagiannaki. 2011. PIE in the sky: Online passive interference estimation for enterprise WLANs. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. USENIX Association, Berkeley, CA, 337–350. http://dl.acm.org/citation.cfm?id=1972457.1972492

Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. 2006. Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM, New York, 237–250. DOI: https://doi.org/10.1145/1182807.1182831

Anand Prabhu Subramanian, Himanshu Gupta, Samir R. Das, and Jing Cao. 2008. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing* 7, 12 (May 2008), 1459–1473. DOI: https://doi.org/10.1109/TMC.2008.70

Rui Tan, Guoliang Xing, Jianping Wang, and Hing Cheung So. 2010. Exploiting reactive mobility for collaborative target detection in wireless sensor networks. *IEEE Transactions on Mobile Computing* 9, 3 (July 2010), 317–332. DOI: https://doi.org/10.1109/TMC.2009.125

Texas Instruments. 2006. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. https://www.ti.com/lit/ds/symlink/cc2420.pdf.

Stefan Unterschütz, Christian Renner, and Volker Turau. 2012. Opportunistic, receiver-initiated data-collection protocol. In *Proceedings of European Conference on Wireless Sensor Networks (EWSN'12)*. 1–16. DOI: https://doi.org/10.1007/978-3-642-28169-3_1

Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan. 2008. Harnessing exposed terminals in wireless networks. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI'08)*. USENIX Association, Berkeley, CA, 59–72. http://dl.acm.org/citation.cfm?id=1387589.1387594

Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. 2006. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI'06)*. USENIX Association, Berkeley, CA, 381–396. http://dl.acm.org/citation.cfm?id=1298455.1298491

Kamin Whitehouse, Alec Woo, Fred Jiang, Joseph Polastre, and David Culler. 2005. Exploiting the capture effect for collision detection and recovery. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNets'05)*. IEEE Computer Society, Washington, DC, 45–52. http://dl.acm.org/citation.cfm?id=1251990.1253398

Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. 2004. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM, New York, 13–24. DOI: https://doi.org/10.1145/1031495.1031498

Xinyu Zhang and Kang Shin. 2010. Chorus: Collision resolution for efficient wireless broadcast. In *Proceedings of INFOCOM (INFOCOM'10)*. 1–9. DOI: https://doi.org/10.1109/INFCOM.2010.5461994

Zhiwei Zhao, Geyong Min, Weifeng Gao, Yulei Wu, Hancong Duan, and Qiang Ni. 2018. Deploying edge computing nodes for large-scale IoT: A diversity aware approach. *IEEE Internet of Things Journal* 5, 5 (2018), 3606–3614.

Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He, and Yunhao Liu. 2017. Interference resilient duty cycling for sensor networks under co-existing environments. *IEEE Transactions on Communications* 65, 7 (April 2017), 2971–2984. DOI: https://doi.org/10.1109/TCOMM.2017.2692758

Xia Zhou, Zengbin Zhang, Gang Wang, Xiaoxiao Yu, Ben Y. Zhao, and Haitao Zheng. 2013. Practical conflict graphs for dynamic spectrum distribution. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'13)*. ACM, New York, 5–16. DOI: https://doi.org/10.1145/2465529.2465545

Michele Zorzi and Ramesh R. Rao. 2003. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. *IEEE Transactions on Mobile Computing* 2, 4 (Oct. 2003), 337–348. DOI: https://doi.org/10.1109/TMC.2003.1255648

Received April 2018; revised March 2019; accepted March 2019